

Bid Delivery Address

Department of Administration, Purchasing Division
2019 Washington Street East
Charleston, WV 25305-0130

VENDOR NAME: Voatz, Inc.

BUYER: Secretary of State

SOLICITATION NO.: CRFP-1600-SOS2400000001

BID OPENING DATE: 08-Nov-2023

BID OPENING TIME: 01:30 pm

FAX NUMBER: n/a

Part-I: Technical Proposal

RECEIVED

2023 NOV -7 PM 1:28

WV PURCHASING
DIVISION



Response To

Request for Proposal

WVSOS Election Division E-Ballot Delivery Technology

Part I, Technical Response

State of West Virginia
CRFP-1600-SOS2400000001

Submission Date:
2023-11-07

Vendor: Voatz, Inc.
50 Milk St. Floor 16
Boston, MA 02109
Phone: 617-669-6366
Fax: N/A
Contact: Nimit Sawhney
Email: ns@voatz.com


Signature 

Table of Contents

| | |
|---|------------|
| Cover Letter | 1 |
| RFP required certification | 2 |
| Technical Proposal - Voatz's proposed approach | 3 |
| 4.1 Project Goals and Mandatory Requirements | 3 |
| 4.1.1 Goals and Objectives | 3 |
| 4.1.2 Mandatory Project Requirements | 4 |
| 4.1.3 Establish Cyber Security Systems and Controls | 17 |
| 4.2 Qualifications and Experience | 18 |
| 4.2.1 Qualification and Experience Information | 18 |
| 4.2.2 Mandatory Qualification/Experience Requirements | 19 |
| Addendum Acknowledgement Form - Addendum No 1 | 23 |
| Addendum Acknowledgement Form - Addendum No 2 | 24 |
| 1. Attachment B: OWASP Application Level Security Verification Levels 1 – 3: Documentation | 25 |
| Attachment B: Supporting Documentation | 47 |
| 2. Attachment C: OWASP Mobile Application Level Security Verification: Table | 133 |
| Attachment C: Supporting Documentation | 137 |
| 3. Attachment D: Security Requirements for Databases: Table | 155 |
| Attachment D: Supporting Documentation | 157 |
| 4. Attachment E: Select Controls from the StateRAMP Moderate Baseline: Table | 172 |
| Attachment E: Supporting Documentation | 176 |
| 5. Attachment F: POA&M Tracker | 202 |
| 6. Attachment G: Voatz, Inc. Accessibility Conformance Report | 203 |
| 7. Attachment H: Full ProV&V Report | 204 |
| 8. Attachment I: Synack Assessment Report | 206 |
| 9. Attachment J: Proposed team and Resumes of Key Personnel | 207 |
| 10. Attachment K: Sample Project Plan | 209 |
| Phase I - Solution Implementation | 210 |
| Project Kick-off | 210 |
| Platform Integration and Customization Work | 211 |
| Phase II - Deployment in Election | 213 |
| Pre-Election Election Staging and Support Work | 213 |
| Voting Window Work | 214 |
| Post-Election Work | 215 |
| 11. Attachment L: Detailed Voting Processes and Voter Guides | 216 |
| Attachment L-1: Detailed Voting processes | 216 |

| | |
|--|------------|
| Attachment L-1-1: End to End Voting Process | 216 |
| Attachment L-1-2: Election Administration Process | 217 |
| Attachment L-2: Voter Guides | 218 |
| Attachment L-2-1: VWA (Voatz Web App) with Electronic Return Voter Guide | 218 |
| Attachment L-2-2: VMA (Voatz Mobile App) with Electronic Return Voter Guide | 219 |
| Attachment L-2-3: VWA (Voatz Web App) with Mark, Print and Mail Ballot Voter Guide | 220 |
| 12. Attachment M: About Voatz | 221 |
| 13. Attachment N: Additional Supporting Documentation | 224 |

Cover Letter

Voatz, Inc. (“Voatz”) is pleased to present this submission to the West Virginia Secretary of State (hereinafter referred to as “WV SOS”, the “State” or the “Client”), in response to the bid request for “WVSOS Election Division E-Ballot Delivery Technology” for the 2024 primary and general elections (solicitation CRFP-1600-SOS2400000001).

With this bid submission, Voatz is pleased to describe our Remote Accessible Ballot Delivery, Marking and Return system, which makes the voting experience more secure, private, accessible, and user-friendly for remote voters. Voatz offers a complete solution that will meet and exceed West Virginia’s expectations, including:

- ✓ Local presence in the United States and similar successful references in several US states.
- ✓ Unsurpassed elections and cybersecurity team experience.
- ✓ Multiple positive project references for electronic ballot delivery and ballot return.
- ✓ Unique and differentiated security, implementing advanced cryptographic protocols to provide a secure and reliable election.
- ✓ Robust denial of service (DoS) and distributed denial of service (DDoS) protection.
- ✓ Securely and redundantly hosted infrastructure in the United States.
- ✓ Customizable solution.
- ✓ Accessible voting using native iOS and Android mobile voting apps and also accessible voting using a browser-based voting app with ability to return electronically or print/mail the ballots.
- ✓ Different voter authentication options, including maximum security option requiring the provision of a government issued credential and liveness test.
- ✓ Ability to require voters to sign an affidavit as part of ballot submission.
- ✓ Ability for jurisdictions to seamlessly print a tabulatable version (machine-readable) of electronically returned ballots.
- ✓ Ballots securely stored on an immutable blockchain ledger.
- ✓ Unique audit portal to enable voters to verify their ballot submissions.
- ✓ Online help and live support for voters and election administrators.
- ✓ Voting solution successfully battle tested and audited multiple times.

Voatz looks forward to working again with West Virginia to meet the goals of future elections within the State. We are happy to answer any additional questions the State may have.

Sincerely,

Nimit Sawhney
Co-Founder/CEO

RFP required certification

By signing below, I certify that I have reviewed this Request for Proposal in its entirety; understand the requirements, terms and conditions, and other information contained herein; that I am submitting this proposal for review and consideration; that I am authorized by the bidder to execute this bid or any documents related thereto on bidder's behalf; that I am authorized to bind the bidder in a contractual relationship; and that, to the best of my knowledge, the bidder has properly registered with any State agency that may require registration.

Voatz, Inc.

(Company)



Nimit Sawhney, Co-Founder/CEO

(Representative Name, Title)

617-669-6366 (Fax N/A)

(Contact Phone/Fax Number)

11/07/2023

(Date)

Technical Proposal - Voatz's proposed approach

Note: Section numbers in the Technical Proposal align with those in the RFP.

4.1 Project Goals and Mandatory Requirements

4.1.1 Goals and Objectives

4.1.1.1 The Vendor provides an electronic ballot delivery and marking tool to all 55 West Virginia Counties in the State. The tool shall be ready for go-live use by no later than the statutory absentee ballot mailing deadline on March 29, 2024. All development, proofing, training, and other necessary actions shall be complete prior to that date.

✓ Voatz fully complies with this requirement and the associated project timelines.

Voatz's award winning system permits seamless implementation and administration by authorized county or state election officials. Voatz's team is composed of highly experienced project management staff, technical support staff and training resources to deliver the project in compliance with the stated requirements.

The Voatz platform is complete and ready to meet the needs of West Virginia's voters. Our support center is fully operational and staffed and ready to support voters beginning on March 29, 2024.

A Sample Project Plan has also been included in **Attachment K** of this proposal for reference.

4.1.1.2 The tool satisfies all West Virginia and federal requirements for electronic absentee voting, including but not limited to W. Va. Code § 3-3-1 et seq., the Uniformed and Overseas Absentee Voting Act, the Military and Overseas Voter Empowerment Act, and the Americans with Disabilities Act.

✓ Voatz fully complies with this requirement.

Voatz is the first remote ballot delivery system to undergo comprehensive VSTL testing to evaluate compliance for the applicable subset of sections under the VVSG 1.1 standards for Usability, Accessibility, Functionality, Security and Accuracy. The VSTL testing summary and report are included in **Attachment H**.

4.1.1.3 The tool's functionality allows convenient confirmation of voter eligibility, voter identity, and accessibility.

✓ Voatz fully complies with this requirement.

The Voatz system is conveniently accessible to voters via native iOS and Android mobile applications as well as a voter web portal.

The Voatz system confirms voter eligibility using the Jurisdiction's voter registration databases. The Voatz system integrates with voter data in multiple ways:

- **Batch file transfer** of the most current voter database. This is handled typically via Secure File Transfer Protocols (SFTP or SCP). Voatz also supports frequent or on-demand re-syncing of the batch files to accommodate any changes to the voter data during the voting window.
- **Web services API integration** (typically via an **encrypted read-only API** into the state voter registration database allowing real-time syncing or a replica system with equivalent accuracy).

The import and management of voter data can be fully automated in the background or initiated manually by the election officials via the Admin portal. All imported voter data is visible to the designed officials via the Administrative Interface.

The Voatz system supports multiple methods of confirming voter identity such as the use of Govt-issued photo-id documents, last 4 of SSN, biometrics, etc. The Voatz system also supports a strict identity verification process based on NIST 800-63 guidelines to verify the identity of a voter.

Only once the initial verification process is completed and the data is matched to the state/county voter files, the voter is authorized to receive the appropriate ballot during the election windows.

The guides included in **Attachment L** describe the detailed processes of confirming voter eligibility and identity.

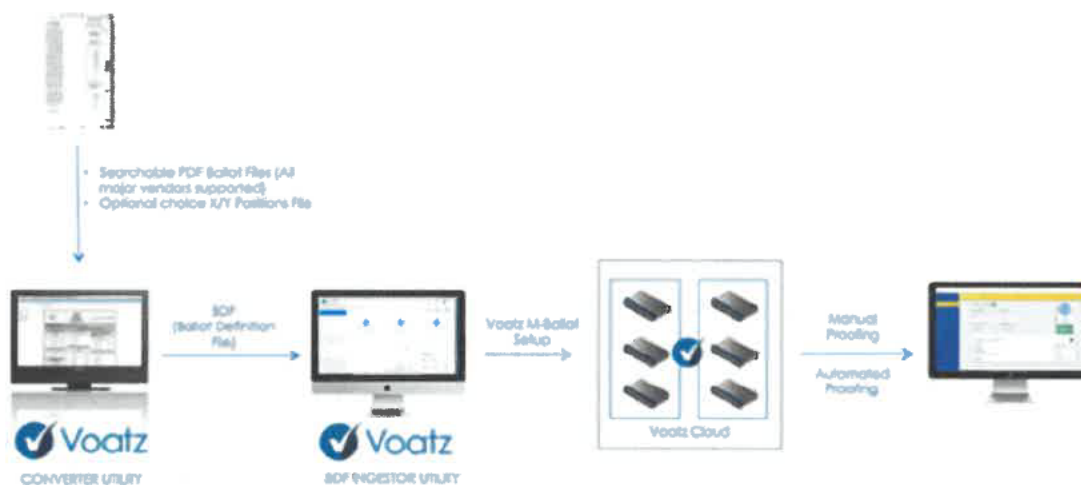
Accessibility is an inherent feature of the Voatz platform. The Voatz native mobile apps and web browser applications provide advanced accessibility features to ensure voters with disabilities can access the system in an easy and convenient manner. Our Accessibility Compliance (VPAT) statements are included in **Attachment G**, which also includes Voatz's general Accessibility Statement.

4.1.2 Mandatory Project Requirements

4.1.2.1 The tool is capable of recognizing and reading each ballot style based on the "Ballot Design" files in the format provided by the Agency, a county, or a county's ballot programming vendor.

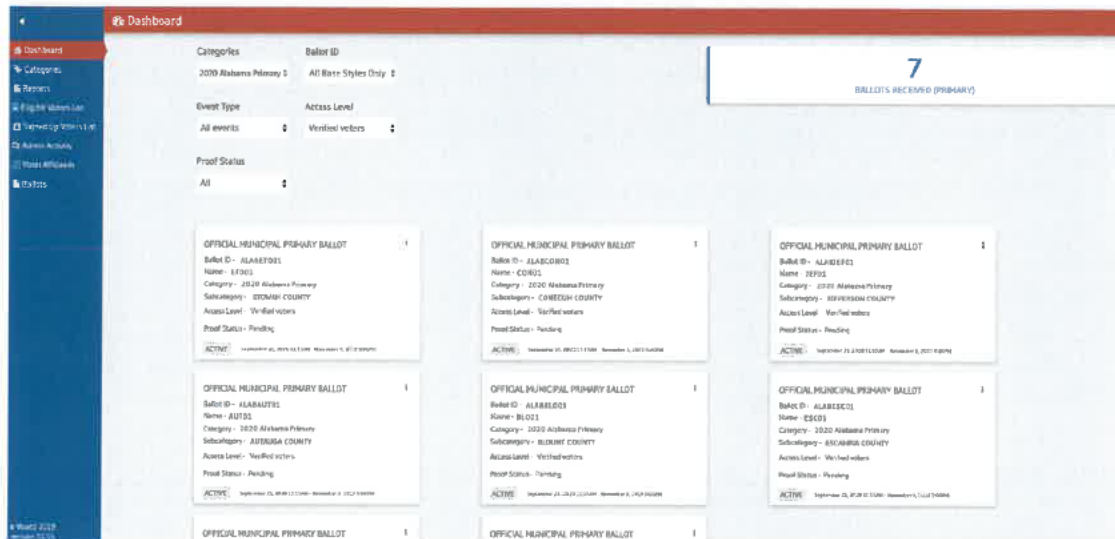
✓ Voatz fully complies with this requirement.

Voatz coordinates with the primary election system vendor that produces the paper ballot designs. We have a set of automated utilities that read the Ballot Definition File ("BDF") from the paper ballot design exports, PDF files (or XML files), and any available XY coordinate descriptions of the paper ballot design in order to produce the corresponding digital/accessible ballot style files for the Voatz system.

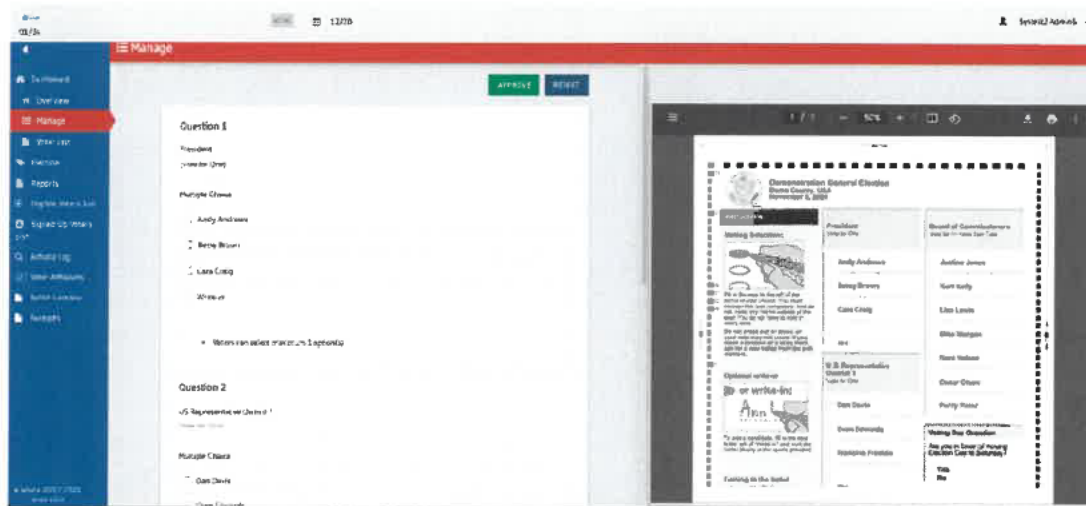


Ballot Import Process


Voatz has a multi-pronged internal proofing process to ensure the accuracy of all electronic ballot styles. In addition, Voatz also provides authorized election officials with a secure Admin Portal to conduct proofing of electronic ballot styles.



Sample Screenshots of the Admin Portal for Ballot Proofing (above & below)



4.1.2.2 The tool includes a cloud server or equivalent backend which securely processes each electronic absentee ballot submission into a cast vote record (CVR) format, stores the records in a tamper-resistant manner, and enables all participating counties to access the CVRs as required by the election schedule and process for in-county tallying.

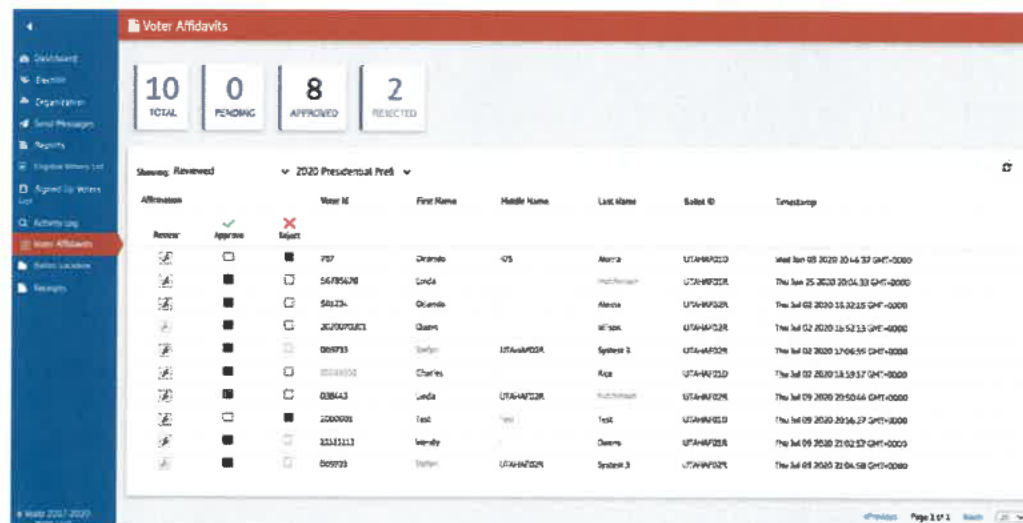
 Voatz fully complies with this requirement.

The Voatz system includes a highly resilient cloud server backend (built using distributed ledger technology) that can securely process each electronic ballot submission and store all the records in a highly tamper resistant manner.

Using the Admin Portal provided as part of the Voatz system, election officials in participating counties can access CVRs, approve/reject voter affidavits, print machine tabulatable ballot PDFs and (if used) voter ballot receipts, etc. as part of the overall in-county tallying process.



Sample Screenshots of CVR Download



Sample Screenshots of Voter Affidavit Approval, CVR Download

Access to the Admin Portal is role-based and two-factor authentication. Each Admin Portal user is assigned a *user role*. The user role determines what the user can see and do in the Admin Portal. A user's role aligns with their duties as an election official and reflects two things:

- (1) Their election responsibilities and (2) their precinct assignment(s), if any

For example, if you are a poll worker for Kanawha County, you will be assigned the role Organization User access to data from Kanawha County. Depending on your role, certain Admin Portal screens and

activities may not be visible or enabled for you. Admin user roles and associated permissions are described in greater detail in the table below:


| Role | Typically Assigned To | Description |
|-----------------------------------|---|--|
| Organization User | Precinct poll worker | An Organization User is generally akin to a poll worker for a given precinct. Organization Users generally have read-only access to data, and may be limited to data for their precinct only. Organization Users do not have access to view voter affidavits, voter ballots, or ballot receipts. |
| Organization Admin - Affiliated | Precinct-level election administrator such as a town or county clerk | An Affiliated Organization Admin is an election administrator for one or more specific precincts. Affiliated Organization Admins can work with ballot styles, voter lists, voter affidavits, voter ballots, and ballot receipts, but are generally limited to data for their assigned precincts. |
| Organization Admin - Unaffiliated | Jurisdiction-level election administrator such as a supervisor of elections | An Unaffiliated Organization Admin has administrative access to functions and data for all precincts within the jurisdiction. |

Ballot Tallying

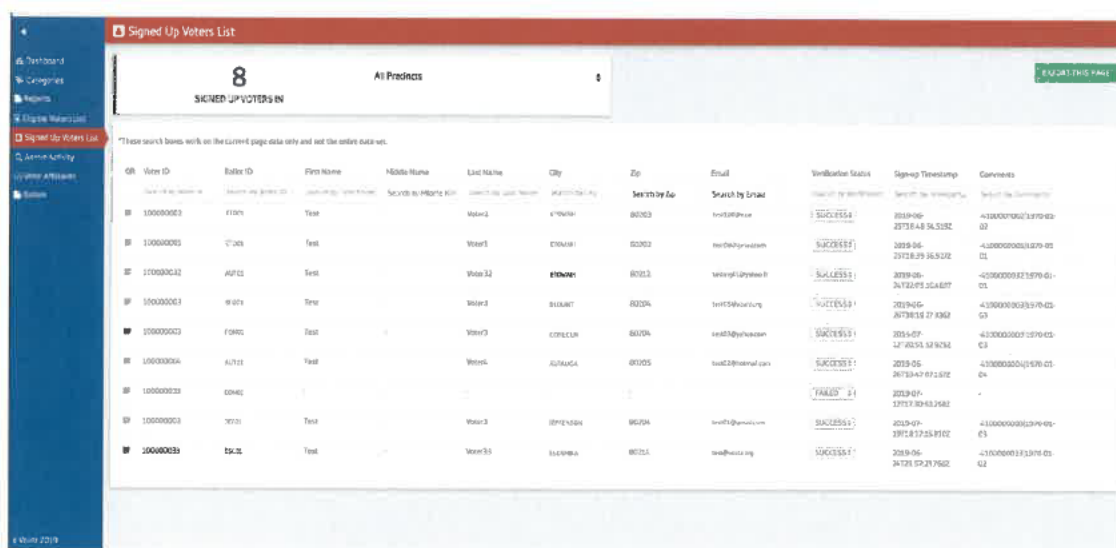
Voter ballot selections are written in aggregate to the Cast Vote Record, a secure, encrypted data file from which election results are tabulated. This feature also supports a fully digital tallying and results preparation within the system in addition to the seamless generation of machine tabulatable paper ballots.

Each individual voter's selections can be rendered as a PDF of a fully-marked, machine-tabulatable paper ballot. These PDF ballots are stored securely in a digital lockbox. These PDFs can be printed and scanned by the jurisdiction's primary tabulation system. They are also available for post-election auditing purposes.

4.1.2.3 The tool includes a web-based or equivalent administration console for reporting and tracking voter participation.

 Voatz fully complies with this requirement.

The Voatz system includes a web based Admin Portal that election officials in participating counties can use for various purposes including reporting, management of voter lists for eligible, signed up voters, approval/rejection of affidavits, oaths, signature verification, etc. thereby providing a comprehensive mechanism for tracking voter participation easily.



Signed Up Voters List

8 All Previews

These search boxes work on the current page data only and not the entire data set.

| ID | Voter ID | Ballot ID | First Name | Middle Name | Last Name | City | Zip | Email | Verification Status | Sign-up Timestamp | Comments |
|------------|------------|------------|------------|-------------|-----------|------------|-------|----------------------|---------------------|--------------------------|----------------------|
| 1000000001 | 1000000001 | 1000000001 | John | | Smith | Washington | 20000 | john.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000010000000000 |
| 1000000002 | 1000000002 | 1000000002 | Jane | | Smith | Washington | 20000 | jane.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000020000000000 |
| 1000000003 | 1000000003 | 1000000003 | John | | Smith | Washington | 20000 | john.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000030000000000 |
| 1000000004 | 1000000004 | 1000000004 | Jane | | Smith | Washington | 20000 | jane.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000040000000000 |
| 1000000005 | 1000000005 | 1000000005 | John | | Smith | Washington | 20000 | john.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000050000000000 |
| 1000000006 | 1000000006 | 1000000006 | Jane | | Smith | Washington | 20000 | jane.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000060000000000 |
| 1000000007 | 1000000007 | 1000000007 | John | | Smith | Washington | 20000 | john.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000070000000000 |
| 1000000008 | 1000000008 | 1000000008 | Jane | | Smith | Washington | 20000 | jane.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000080000000000 |
| 1000000009 | 1000000009 | 1000000009 | John | | Smith | Washington | 20000 | john.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000090000000000 |
| 1000000010 | 1000000010 | 1000000010 | Jane | | Smith | Washington | 20000 | jane.smith@voatz.com | SUCCESS | 2019-06-20T15:48:34.532Z | 41000000100000000000 |

Sample Screenshot of Voter Management, Approval

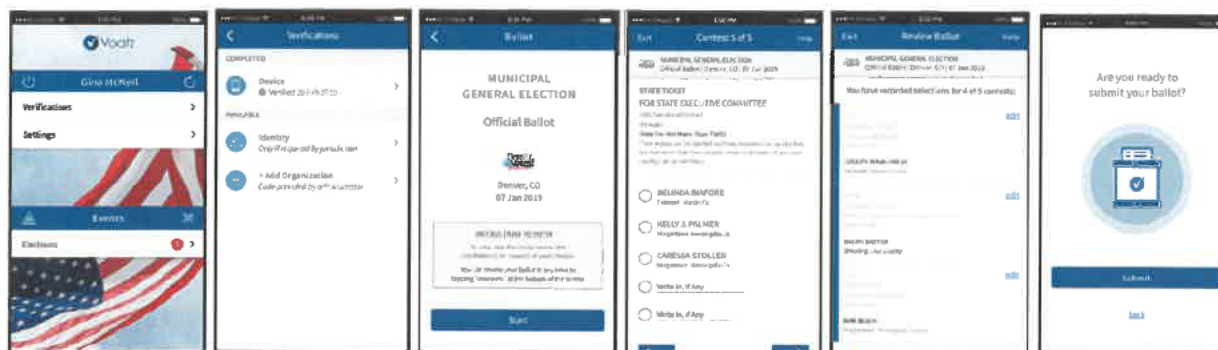
Prior-to, during, and following the election, election officials can use the Voatz Admin Portal to review summary reports and statistics such as:

- Eligible voters
- Signed up voters
- Ballots received
- Ballots rejected
- Affidavits received
- Affidavits approved
- Security monitoring reports
- Votes by device type
- Votes by jurisdiction
- Audit logs/reports

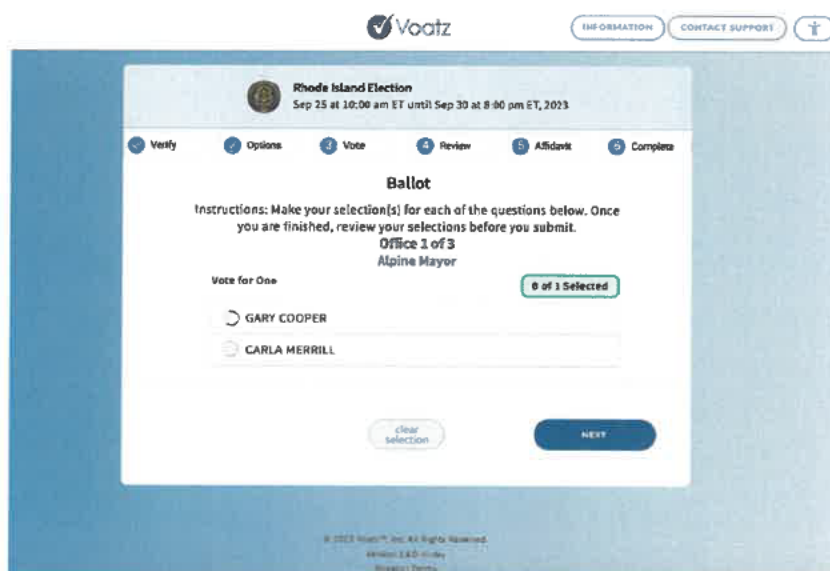
4.1.2.4 The tool permits a voter to mark a ballot independently and without assistance.

 Voatz fully complies with this requirement.

Voatz designs and builds its product with usability and accessibility as priorities. The look, feel and continued evolution of the technology is informed by a design process that leverages in-depth user research, interviews, and prototyping with user personas ranging from 18-year-old to 80-year-old voters. Voatz prioritizes simplicity and ease in order to ensure a user-friendly application navigable by all without assistance.



Example Screens Viewed by Voters Using The Mobile App



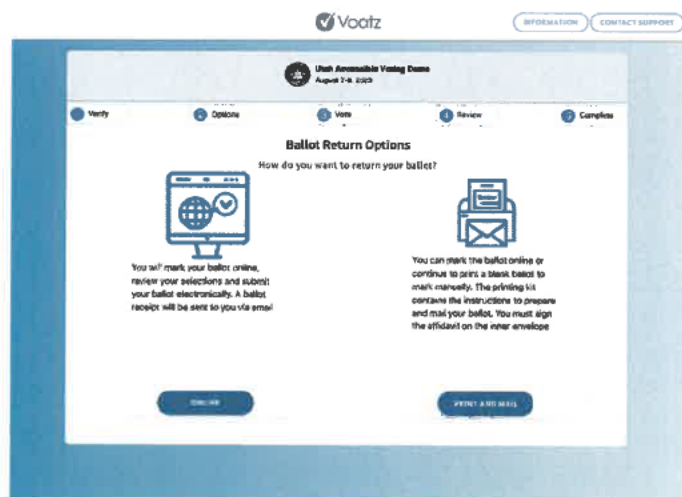
Example Screens Viewed by Voters Using The Web App

Further details of the independent ballot marking process are available in the included Voter guides in **Attachment L**.

4.1.2.5 The tool provides the Agency the option to permit a voter to transmit a marked ballot, along with a return packet that includes the requisite forms and disclosures, to the county clerk electronically, or alternatively to print a voted ballot with the aforementioned return packet for return via other approved means to the county clerk.

 Voatz fully complies with this requirement.

The Voatz system permits the voter to return the marked ballot along with all the requisite forms and disclosures electronically via the native iOS/Android mobile apps or the Voatz web application.



Sample VWA screen to select Ballot Return option

Alternatively, the voter has the option to print the marked ballot via the Voatz web application. The voter is presented the option to print the ballot, either unmarked or after making and verifying their selections. This is available to any voters who either can not or do not want to submit electronically.

See **Attachment L-2-3** (VWA - ID Verification, Mark, Print and Mail Ballot) for additional details and screenshots # 9, 13, 14, 15 and 16.

4.1.2.5.1 The option for a voter to return a ballot shall be an optional functionality available to the Agency at no cost, and the Vendor shall not be compensated in any manner in the event the Agency opts to permit voters to return a ballot electronically.

✔ Voatz fully complies with this requirement.

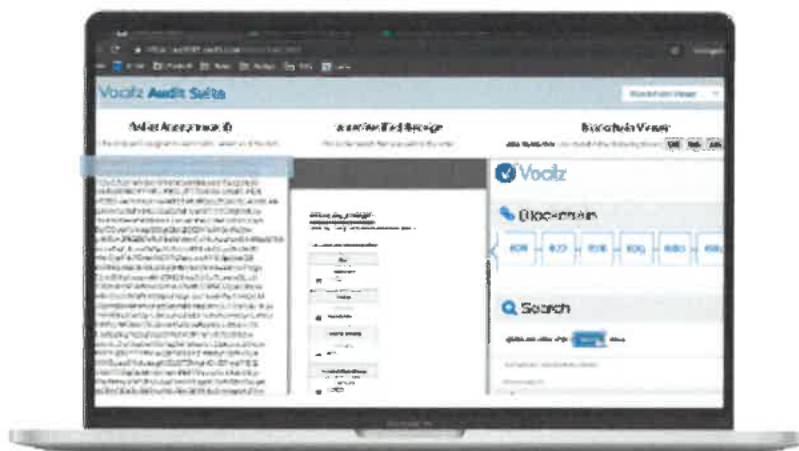
Electronic ballot return is available as an optional feature at no cost to the State.

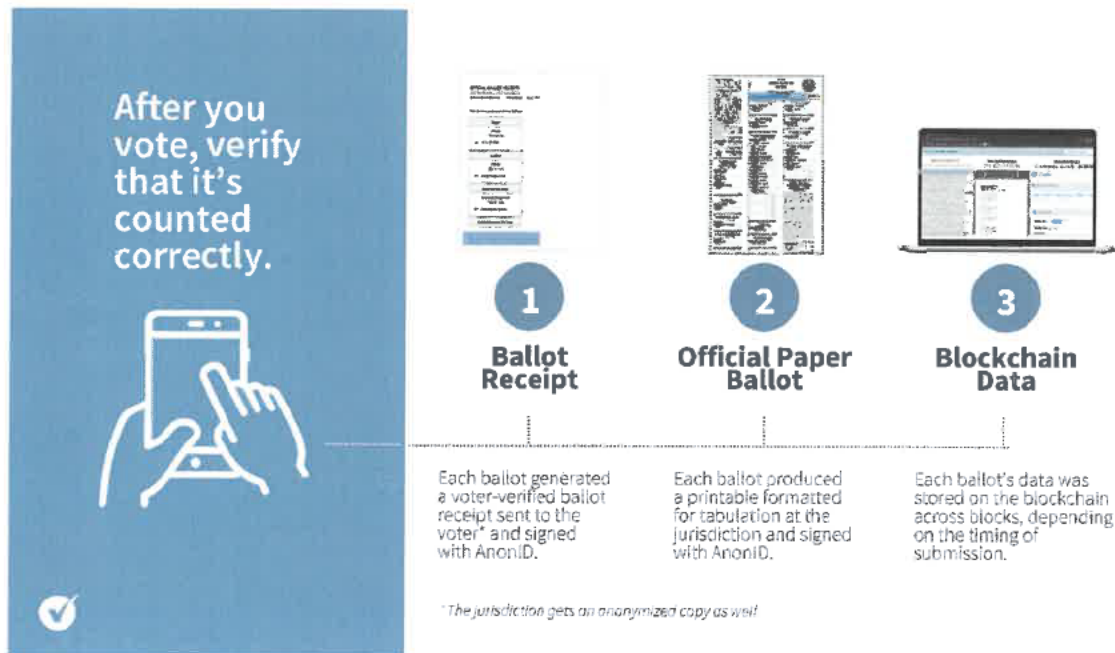
4.1.2.6 The tool includes a verification portal that permits a voter to review their marked, submitted ballot, in a secure and anonymous manner, and in a read-only format, affording the voter the ability to confirm the ballot cast is the ballot received by the county.

✔ Voatz fully complies with this requirement.


After ballot submission, voters automatically receive a ballot receipt containing an anonymous ballot ID. Once a ballot is accepted by a county via the Admin Portal, the county also receives a copy of the ballot receipt with the anonymous ballot ID, separated from the voter's profile. The same anonymous ballot ID is also recorded as part of the blockchain data. When accepted ballot PDFs are printed for tabulation, the printed ballots also contain this anonymous ballot ID (optional). Only the voter knows their own anonymous ballot ID. It is not tied to the voter in any part of the platform once the ballot packet has been accepted by the county.

The Voatz Audit Portal allows for a full end-to-end audit of the election data by tracing the anonymous ballot ID across all channels. If a voter participates in an audit of the election, they can look up their own anonymous ballot ID in the bulletin board and compare their copy of their ballot receipt to the copy that the county received, as well as comparing it to the blockchain data to ensure that the ballot cast was the one received by the county.





4.1.2.7 The Vendor provides training and support to the Agency and counties during the duration of the contract.

 Voatz fully complies with this requirement.

Voatz provides comprehensive training services (both in-person and online) covering the various aspects of the system including voting and the administrative functions. Voatz also provides mock community elections (prior to the actual election) wherein both citizens and officials can participate, provide feedback, conduct logic/accuracy tests, etc.

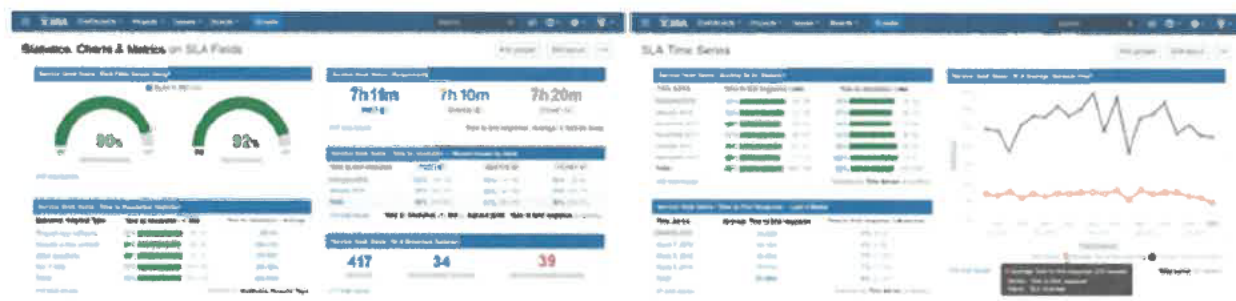
Pursuant to state and local health guidelines concerning travel and COVID-19: In the event that in-person training is not permitted due to state or local health guidelines, online virtual training will be provided in its place.

Furthermore, Voatz provides comprehensive documentation and user guides for the various activities pertaining to the system.

Voatz will provide comprehensive on-site support to the county elections personnel on the desired dates as needed. In the event that on-site support is not permitted due to state or local health guidelines, virtual support will be provided in its place.

Attachment J of this document includes further details of the Voatz team designated to work on this project along with their project responsibilities, including also their Curricula vitae.

Voatz provides comprehensive help desk services for voters and election administration personnel via phone, email, text chat and support web portals, and will maintain Help Desk statistics on help request volume, resolution, and response time, and provide reports to the Secretary of State upon request via our robust support reporting tools.



Sample Screenshot of Help Desk Statistics Dashboard

4.1.2.8 Section 508 Compliance

Section 508 of the Rehabilitation Act, as amended by the Workforce Investment Act of 1998 (P.L. 105-220) requires that when Federal agencies develop, procure, maintain, or use Information and Communication Technology (ICT), it shall be accessible to people living with disabilities. Federal employees and members of the public who have disabilities must have access to, and use of, information and data that is comparable to people without disabilities.

Products, platforms and services delivered as part of this work statement that are ICT, or contain ICT, must conform to the Revised 508 Standards, which are located at 36 C.F.R. § 1194.1 & Apps. A, C & D, and available at <https://www.access-board.gov/guidelines-andstandards/communications-and-it/about-the-ict-refresWfinal-rule/text-of-the-standardsand-guidelines>

☒ Voatz fully complies with this requirement.

The Voatz™ mobile and web app elections platform provides robust support for all voters, including those with visual, cognitive, mobility, and dexterity disabilities. (Hearing is not required for voting unless using a screen reader.) Voters of all abilities can vote privately, independently and securely without the need for paper ballots or postal mail, using their own COTS device such as an iOS or Android mobile phone, or desktop/laptop computer.

The Voatz Mobile App and Voatz Web App support various accessibility features, including those listed here.

- VoiceOver, TalkBack screen readers (on mobile apps), NVDA, VoiceOver, TalkBack and JAWS screen readers (on the web browser app)
- Predictable navigation
- Configurable font size
- Speech-to-Text (for write-ins)
- Voice Control (on iOS and MacOS)
- Flexible session timeouts
- Support for Bluetooth assistive devices


The Voatz mobile and web apps are designed to support these capabilities and meet WCAG 2.1 as described further in our Accessibility Compliance Statements (aka. VPATs) which accompany this proposal (**Attachment G**). The aforementioned attachment also includes Voatz's general Accessibility Statement.

4.1.2.8.1 Provide a list of item(s) that contains ICT. For each item, the following requirements apply:

Voter-facing software:

- Voatz Mobile App (VMA) for iOS smartphones
- Voatz Mobile App (VMA) for Android smartphones
- Voatz Web App (VWA) for compatible browsers

4.1.2.8.2 All functional performance criteria apply when using an alternative design or technology that achieves substantially equivalent or greater accessibility and usability by individuals with disabilities, than would be provided by conformance to one or more of the requirements in Chapters 4-6 of the Revised 508 Standards, or when Chapters 4-6 do not address one or more functions of ICT.

 Voatz fully complies with this requirement.

4.1.2.8.2 Software features and components: All WCAG Level AA Success Criteria, 502 Interoperability with Assistive Technology, 503 Application.

 Voatz fully complies with this requirement.

4.1.2.8.3 Hardware features and components: All requirements apply.


 Voatz fully complies with this requirement.

Voatz does not provide hardware. However, Voatz mobile apps and the web app utilize the accessibility features native to iOS/Android smartphones as well as on modern web browsers.

4.1.2.8.4 Applicable support services and documentation: All requirements apply.


 Voatz fully complies with this requirement.

4.1.2.8.2 Provide an Accessibility Conformance Report (ACR) for each commercially available ICT item offered through this contract. Create the ACR using the Voluntary Product Accessibility Template Version 2.1 or later, located at <https://www.itic.org/policy/accessibility/vpat>. Complete each ACR in accordance with the instructions provided in the VPAT template. Each ACR must address the applicable Section 508 requirements referenced in the Work Statement. Each ACR shall state exactly how the ICT meets the applicable standards in the remarks/explanations column, or through additional narrative. All "Not Applicable" (N/A) responses must be explained in the remarks/explanations column or through additional narrative. Address each standard individually and with specificity, and clarify whether conformance is achieved throughout the entire ICT Item (for example - user functionality, administrator functionality, and reporting), or only in limited areas of the ICT Item.

 Voatz fully complies with this requirement.

Please see **Attachment G** - Voatz, Inc. Accessibility Conformance Report.

4.1.2.8.3 Provide a description of the evaluation methods used to support Section 508 conformance claims. The Agency reserves the right, prior to making an award decision, to perform testing on some or all of the Vendor's proposed ICT items to validate Section 508 conformance claims made in the ACR.

 Voatz incorporates accessibility and usability design principles throughout its development process, and incorporates a continuous practice of both real-world evaluation and design-to-standards in engineering.

- Voatz partnered with the National Center for Accessible Media (NCAM) for recommendations on usability and accessibility (further details are included at the end of Attachment G).
- Voatz software is evaluated in-house by professional testers with previous WCAG 2.1 and certified voting systems experience. The two Accessibility Compliance (VPAT) statements are included in **Attachment G**, which also includes Voatz's general Accessibility Statement.
- Voatz engaged Pro V&V, an independent EAC-approved voting system testing laboratory (VSTL) certified in assessing accessibility compliance with the U.S. Election Assistance Commission's Voluntary Voting System Guidelines (VVSG 1.1). View Pro V&V's Accessibility and Usability Test Report included in **Attachment H**.
- In addition to rigorous internal testing and remediation, Voatz continuously incorporates feedback from accessibility experts and voters with disabilities who have participated in elections using Voatz.

4.1.2.8.4 Describe your approach to incorporating universal design principles to ensure ICT products or services are designed to support disabled users.

✓ Voatz incorporates accessibility and usability design throughout its development process:

- Voatz partnered with the National Center for Accessible Media (NCAM) for recommendations on usability, accessibility and to help test the Voatz applications for accessibility compliance (further details are included at the end of Attachment G).
- Voatz also incorporates design recommendations made by the Center for Civic Design.
- Voatz software is evaluated in-house by professional testers with previous WCAG 2.1 and certified voting systems experience. Read the VPAT statements included in **Attachment G**.
- Voatz engaged Pro V&V, an independent EAC-approved voting system testing laboratory (VSTL) certified in assessing accessibility compliance with the U.S. Election Assistance Commission's Voluntary Voting System Guidelines (VVSG 1.1). View Pro V&V's Accessibility and Usability Test Report included in **Attachment H**.
- In addition to rigorous internal testing and remediation, Voatz continuously incorporates feedback from accessibility experts and voters with disabilities who have participated in elections using Voatz.

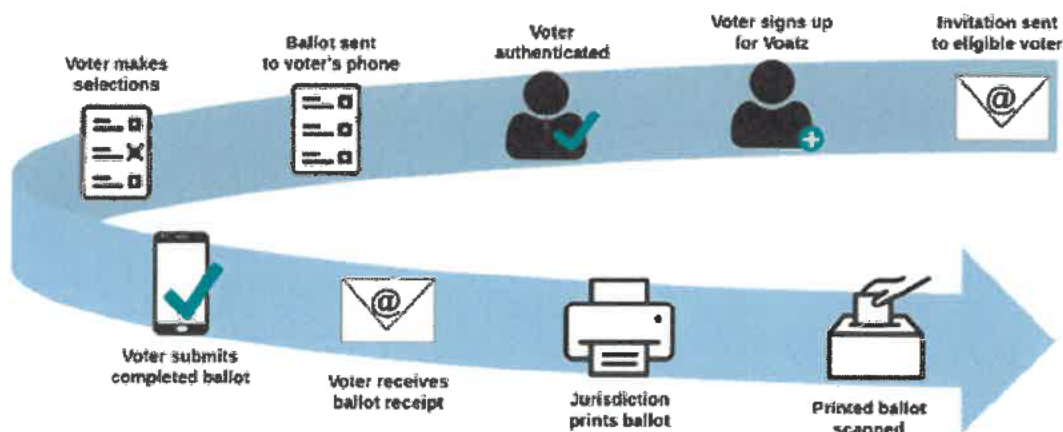
4.1.2.8.5 Describe plans for features that do not fully conform to the Section 508 Standards.

✓ Neither accessibility nor security are concepts that are ever "done" 100%. Voatz is committed to continually improving both, and expects that both standards and user expectations will continue to drive those improvements. As a partner with many national and local accessibility user groups and standards bodies, Voatz intends to not only maintain compliance with applicable standards, but to lead the way in ensuring that voters with disabilities are able to vote privately and independently, regardless of standards or requirements.

For jurisdictions requiring use of government ID verification, a sighted individual can assist a visually-impaired voter by scanning a photo ID and following video 'selfie' instructions. This can be performed prior to voting without compromising the privacy of the voter's ballot marking process.

4.1.2.8.6 Describe "typical" user scenarios and tasks, including individuals with disabilities, to ensure fair and accurate accessibility testing of the ICT product or service being offered.

✓ Typical user tasks are described in the following process chart:



Voatz RABDMR Process Flow

Each step in this process is routinely tested by user groups with disabilities to identify weaknesses in the platform.

The Voatz system is composed of a mobile application (iOS, Android), an administrative portal, blockchain vote storage, and a public bulletin for vote verification and auditing. A traditional browser-based voter portal is also offered for voters who may not have a compatible device, while encouraging most voters to use the most secure, accessible, and convenient mobile option.

A voter is able to download the mobile application and sign up with an active mobile number and email. Once signed up in the Voatz app, voters are required to authenticate themselves before gaining access to eligible elections. Authentication is a three-step process that uses the smartphone's camera and its biometric feature (i.e., fingerprint recognition or facial recognition): (1) the voter scans their state driver's license or passport, (2) takes a live facial snapshot (a video "selfie"), and (3) touches the fingerprint reader on the smartphone or activates the phone facial lock, which ties the voter's device to the voter. Once the voter is authenticated, the app matches the voter's "selfie" to the facial picture on their passport or driver's license and confirms the voter's eligibility to vote against the state's voter registration database.

Once the voter is verified, election jurisdictions initiate the voting process by sending a qualified voter a mobile ballot. Contained in the mobile ballot are "tokens" — think of them as potential votes — which are cryptographically tied to a candidate or ballot measure question. The number of tokens a given voter receives is the same as the number of ovals he or she would have received on a paper ballot handed out at the voter's precinct or sent through the mail.

A voter must then provide their PIN or biometric information to access the mobile application and again to submit any ballots. Voters will see eligible elections and ballots within the Voatz app during the corresponding voting period, and the voter will then make their desired selections and submit their ballot. These selections alter the tokens (like filling in a ballot oval). Overvotes are prevented, as each

voter only receives a total number of tokens as they have potential votes. Once submitted, the votes for choices on the ballot are verified by multiple distributed servers called “verifiers”, or validating nodes. Upon verification, the token is debited (i.e. subtracted) from the voter’s ledger and credited (i.e. added) to the candidate’s ledger. The blockchain on every verifier is automatically updated and the process repeats as additional voters submit their selections. Voters receive a digitally signed and encrypted pdf of their choices (optional, if desired by the jurisdiction) in which they may verify their submitted ballot.

The following are the steps of the simple process that voters experience using the Voatz Web App.

1. Authorized voters access the Voatz Web App via a URL provided to them for the election.
2. A Login screen prompts them to enter a mix of personal identification data as required by the jurisdiction, for example, name, date of birth, and Voter ID. This information is validated against voter data in the Voatz database. See **Attachment L-2-1 image # 2**.
3. Voters arrive at the first screen for their event. This can be a simple welcome screen that provides information about the election, or a declaration that the voter must agree with before proceeding with the ballot.
4. Voters are taken to the ballot. A contest is displayed where voters can select multiple candidates. To make their choices, voters select the candidates they wish to vote for by clicking on their name or anywhere in their designated box on the screen. If desired, voters can select ‘more information’ to view additional information and resources for each candidate. The information found under the ‘more information’ tab can be customized by the jurisdiction and can include embedded media and links to external resources. If a voter has under-voted, they will receive a warning before the system allows them to move on to the next question. Overvoting is not allowed by the system. See **Attachment L-2-1 images #9 & 10**.
5. After completing the ballot, voters are shown a review screen where they can see an overview of their selections. Voters can navigate back to any contest and change their selections by clicking the ‘change’ button located next to each contest. See **Attachment L-2-1 image #12**.
6. Voters are prompted to confirm their intent to submit their ballot by clicking ‘submit’ when the popup appears as shown above. Until they click ‘submit,’ voters still have the option of navigating back to their ballot and changing their selections, by clicking ‘back.’ Additional submission requirements, such as an e-signature for affidavits (captured via touch screen or mouse), can be included in the submission process. See **Appendix Attachment L-2-1 images #13, 14 and 15** (the images show a sample affidavit signature screen in VWA.).
7. After submitting their ballot, voters are taken to a confirmation screen where they are provided with a confirmation number which they can print or email for their records. Or, optionally, an emailed vote receipt (with visual confirmation of a voter’s individual oval choices) can be delivered to the voter upon submission (in addition to the confirmation code provided on screen, or in place of the confirmation code). See **Attachment L-2-1 image #16**.

4.1.2.9 *The tool meets all mandatory security or control requirements as indicated in Attachments D and E.*

 Voatz fully complies with this requirement. See further details in **Attachment D and E**.

4.1.2.10 Prior to acceptance, the Agency reserves the right to perform testing on required ICT items to Validate the Vendor's Section 508 conformance claims. If the Agency determines that Section 508 conformance claims provided by the Vendor represent a higher level of conformance than what is actually provided to the agency, the government shall, at its option, require the Vendor to remediate the item to align with the Vendor's original Section 508 conformance claims prior to acceptance.

☒ Voatz will fully comply with this requirement, and collaborate with the Agency as required.

4.1.3 Establish Cyber Security Systems and Controls

The tables in **Attachments B-E** contain lists the requirements for Sections 4.1.3.1.1-4, along with an indication of Yes/No/Partial to note whether or not we meet the given requirement. Those items that are not applicable are indicated with a "Yes" in the table, but have an "N/A" designation in the subsequent description in the attachment, with relevant comments.

4.1.3.1 Cybersecurity systems and controls are essential to distinguish, counteract, or decrease security risks. These measures are required to manage threats targeting computer systems and networks. These measures must be adaptive and robust. To determine whether your cyber security systems and controls meet our desired standards:

See below.

4.1.3.1.1 Please complete Attachment B — OWASP Application Level Security Verification Levels 1 —3. For all listed requirements please provide 2 pieces of supporting documentation. For any requirements that are lacking supporting documentation, please add to Attachment F — POA&M Tracker.

☒ Voatz complies with this requirement: Voatz complies with **241 of 241** requirements in Attachment B: OWASP Application Level Security Verification Levels 1 - 3. Eight of these requirements are in an improvement development phase for the Voatz Admin Portal, with analogous requirements currently not applicable in case of the mobile/web apps.

See **Attachment B** for the documentation for each requirement, along with comments.

4.1.3.1.2 Please complete Attachment C — OWASP Mobile Application Level Security Verification if applicable. For all listed requirements please provide 2 pieces of supporting documentation. For any requirements that are lacking supporting documentation, please add to Attachment F — POA&M Tracker. If not applicable, please put N/A by all requirements.

☒ Voatz fully complies with **69 of 69** requirements in Attachment C: OWASP Mobile Application Level Security Verification.

See **Attachment C** for the documentation for each requirement, along with comments.

4.1.3.1.3 Please complete Attachment D — Security Requirements for Databases. For all listed requirements please provide 2 pieces of supporting documentation. For any requirements that are lacking supporting documentation, please add to Attachment F — POA&M Tracker. All requirements flagged as "Mandatory" must be met for award eligibility.

☒ Voatz fully complies with **43 of 43** requirements in Attachment D: Security Requirements for Databases.

See **Attachment D** for the documentation for each requirement, along with comments.

4.1.3.1.4 Please complete Attachment E — Select Controls from NIST SP 800-171. For all listed requirements please provide 2 pieces of supporting documentation. For any requirements that are lacking supporting documentation, please add to Attachment F — POA&M Tracker. All requirements flagged as “Mandatory” must be met for award eligibility.

☒ Voatz fully complies with **80 of 80** requirements in Attachment E: Select Controls from the StateRAMP Moderate Baseline. Voatz has also recently completed a StateRAMP Security Snapshot.

See **Attachment E** for the documentation for each requirement, along with comments.

4.1.3.1.5 Please complete Attachment F — POA&M Tracker in the provided format as indicated above for any requirements that are lacking supporting documentation. Select Controls from NIST SP 800-171. For all listed requirements please provide 2 pieces of supporting documentation. For any requirements that are lacking supporting documentation, please add to Attachment F — POA&M Tracker. All requirements flagged as “Mandatory” must be met for award eligibility.

☒ See Attachment F as required. Please note that there are no requirements that are lacking supporting documentation.

4.1.3.2 Supporting Documentation

☒ Compliance reports, security and risk assessment documentation is included in **Attachments H and I**. Additional supporting documentation is included in **Attachment N**.

4.1.3.3 Sensitive, confidential, proprietary or critical elements of technology.

☒ Included in **Attachment N**.

4.2 Qualifications and Experience

Vendor should provide information and documentation regarding its qualifications and experience in providing services or solving problems similar in size, scope and complexity to those requested in this RFP. Information and documentation should include, but are not limited to, proposed staffing plans, descriptions of past projects completed (descriptions should include the location of the project, project manager name and contact information, type of project, and what the project goals and objectives were and how they were met.), references for prior projects including the value and period of performance of past projects, and any other information that Vendor deems relevant to the items identified as desirable or mandatory below.

Please see Attachment M, which provides an overview of Voatz’s experience. Attachment J lists the proposed staffing plan as well.

4.2.1 Qualification and Experience Information

Vendor should describe in its proposal how it meets the desirable qualification and experience requirements listed below.

4.2.1.1 Vendor's tool has been reviewed by at least one (1) independent, nationally recognized organization supporting the Disability Community for its user acceptance and Section 508 conformity for

voters living with disabilities. Copies of any reports or public statements by the organization(s) should be provided to the Agency for Confidential review.

☒ Yes, Voatz complies with this requirement.




Please see the 'Voatz Mobile App Accessibility Conformance Report,' **Attachment G** for further details. Also, please see **Attachment H** for the review conducted by Pro V&V.



4.2.2 Mandatory Qualification/Experience Requirements

The following mandatory qualification/experience requirements must be met by the Vendor as a part of its submitted proposal. Vendor should describe how it meets the mandatory requirements and include any areas where it exceeds the mandatory requirements. Failure to comply with mandatory requirements will lead to disqualification, but areas where the mandatory requirements are exceeded will be included in technical scores where appropriate. The mandatory qualifications/experience requirements are listed below.

4.2.2.1 Implemented tool in at least two (2) previous federal elections. A list of all previous federal elections, including the jurisdiction, shall be provided to the Agency.

☒ Yes, Voatz complies with this requirement.

| Jurisdiction Name /Project Owner | Contract Description / Work Performed by Bidder | Contact Person: Name Title Contact Info |
|--|---|---|
| Utah County, Utah  | Voatz provided accessible ballot delivery, marking and electronic return services for UOCAVA voters and voters with disabilities and their caretakers in Utah County, Utah. Voatz also provided an iPad tablet-based solution for accessible in-home or curbside voters. <ul style="list-style-type: none"> ● 2020 and 2022 Primary & General Elections ● 2019 and 2021 Municipal & Special Elections | Rozan Mitchell County Elections Manager at Utah County, Utah (385) 321-2935 /rozanmit@utahcounty.gov |
| Jackson County, Oregon  | Voatz provided accessible ballot delivery, marking and electronic return services for deployed military and overseas voters. <ul style="list-style-type: none"> ● 2020 General Elections ● 2019 and 2021 Municipal & Special Elections ● 2022 Primary & Mid-Term Federal Elections ● 2023 Special Elections | Christine Walker County Clerk at Jackson County, Oregon (541) 774-6125 /walkercd@jacksoncounty.org |
| State of West Virginia  | Voatz provided remote ballot delivery, marking and returns services for deployed military and overseas voters from the state of West Virginia during the 2018 Primary Elections and Midterm Election. The use of native mobile applications and a blockchain based infrastructure was a pioneering and historical event in US Election History. | Donald Kearsey Deputy Secretary of State at the State of West Virginia (866) 767-8683 /dkersey@wvsos.gov |

| | | |
|---|--|--|
| | <p>Voatz provided the following services: Project Management, Product Customization, Communications and Media Preparation, Staff Training, Digital Ballot Design & Proofing, Voter Data List Integration, Help Desk Support to the County staff and to Voters, Online Ballot Marking, Production of Paper Ballot Audit Trail, Post Election Reports and Independent Audit, Cybersecurity Monitoring</p> <ul style="list-style-type: none"> • 2018 Primary & Mid-Term Federal Elections | |
| <p>Washington County, Oregon</p>  | <p>Voatz provided accessible ballot delivery, marking and electronic return services for deployed military and overseas voters.</p> <ul style="list-style-type: none"> • 2022 Special and General Elections • 2023 Special Elections | <p>Dan Forester Elections Director at Washington County, Oregon (801)234-0676 /Dan_Forester@washingtoncountyor.gov</p> |
| <p>Summit County, Utah</p>  | <p>Voatz provided accessible ballot delivery, marking and electronic return services for UOCAVA voters and voters with disabilities and their caretakers.</p> <ul style="list-style-type: none"> • 2021 Municipal & Special Elections • 2022 Primary & General Elections | <p>Eve Furse County Clerk at Summit County, Utah (435) 336-3203 /efurse@summitcounty.org</p> |

4.2.2.2 Vendor's applicable network and systems or tool have been assessed for security vulnerabilities by at least two (2) independent, federally recognized, certified, or industry specific equivalent technology or cybersecurity auditors. Copies of all assessments or equivalent reports shall be provided to the Agency.

☒ Yes, Voatz complies with this requirement.

Please see the list of independent security reviews below:

| Year | Category | Scope | Conducted by |
|------|------------------------------------|---|-----------------------------|
| 2016 | Whitebox Testing | Mobile Applications | Independent Security Vendor |
| 2018 | Whitebox Testing, Blackbox Testing | Mobile Applications, Core Servers, DLT Infrastructure | Independent Security Vendor |

| | | | |
|------|--|---|--|
| 2019 | Citizens Audit #1 | Post-election Audit (Denver County, CO) | Pool of volunteer citizens, election officials, experts |
| 2019 | Citizens Audit #2 | Post-election Audit (Utah County, UT) | Pool of volunteer citizens, election officials, experts |
| 2019 | Hunt Assessment | Full Infrastructure Audit | CISA (DHS) Hunt and Incident Response Team (HIRT) |
| 2020 | Whitebox Analysis, Threat Modeling | Mobile Applications, Core Servers, Blockchain Infrastructure | Independent Security Vendor |
| 2020 | Citizens Audit #3 | Post-Election Audit, Utah GOP State Party | Pool of volunteer citizens, election officials, experts |
| 2020 | VVSG 1.1 Compliance Testing (including Usability and Accessibility), Whitebox Testing | Mobile Applications, Core Servers | Pro V&V - EAC/Federally Certified Voting Systems Test Laboratory (VSTL) |
| 2020 | Whitebox Analysis, Blackbox Testing, Threat Modeling | Mobile Applications, Core Servers, Blockchain Infrastructure | Public Election Security Lab |
| 2021 | Red Team Testing | Mobile Applications, Core Servers | Independent Security Vendor |

| | | | |
|------|---|---|-----------------------------|
| 2022 | Red Team Testing | Mobile Applications, Core Servers | Independent Security Vendor |
| 2022 | Application Penetration Testing | Web Applications, Mobile Applications, Core Servers | Independent Security Vendor |
| 2023 | StateRAMP Snapshot | Web Applications, Mobile Applications, Core Servers | StateRAMP PMO |
| 2023 | Cloud Security Alliance STAR Compliance | Cloud Infrastructure | (in progress) |

Please refer to Attachments H, I and O for additional information.

Addendum Acknowledgement Form - Addendum No 1

Please see the following pages with the signed Form.

ADDENDUM ACKNOWLEDGEMENT FORM
SOLICITATION NO.: CRFP SOS24*001

Instructions: Please acknowledge receipt of all addenda issued with this solicitation by completing this addendum acknowledgment form. Check the box next to each addendum received and sign below. Failure to acknowledge addenda may result in bid disqualification.

Acknowledgment: I hereby acknowledge receipt of the following addenda and have made the necessary revisions to my proposal, plans and/or specification, etc.

Addendum Numbers Received:

(Check the box next to each addendum received)

| | |
|--|--|
| <input checked="" type="checkbox"/> Addendum No. 1 | <input type="checkbox"/> Addendum No. 6 |
| <input type="checkbox"/> Addendum No. 2 | <input type="checkbox"/> Addendum No. 7 |
| <input type="checkbox"/> Addendum No. 3 | <input type="checkbox"/> Addendum No. 8 |
| <input type="checkbox"/> Addendum No. 4 | <input type="checkbox"/> Addendum No. 9 |
| <input type="checkbox"/> Addendum No. 5 | <input type="checkbox"/> Addendum No. 10 |

I understand that failure to confirm the receipt of addenda may be cause for rejection of this bid. I further understand that any verbal representation made or assumed to be made during any oral discussion held between Vendor's representatives and any state personnel is not binding. Only the information issued in writing and added to the specifications by an official addendum is binding.

Voatz, Inc.

Company



Authorized Signature

11/07/2023

Date

NOTE: This addendum acknowledgement should be submitted with the bid to expedite document processing.
Revised 6/8/2012

Addendum Acknowledgement Form - Addendum No 2

Please see the following pages with the signed Form.

ADDENDUM ACKNOWLEDGEMENT FORM
SOLICITATION NO.: CRFP SOS24*001

Instructions: Please acknowledge receipt of all addenda issued with this solicitation by completing this addendum acknowledgment form. Check the box next to each addendum received and sign below. Failure to acknowledge addenda may result in bid disqualification.

Acknowledgment: I hereby acknowledge receipt of the following addenda and have made the necessary revisions to my proposal, plans and/or specification, etc.

Addendum Numbers Received:

(Check the box next to each addendum received)

| | |
|--|--|
| <input type="checkbox"/> Addendum No. 1 | <input type="checkbox"/> Addendum No. 6 |
| <input checked="" type="checkbox"/> Addendum No. 2 | <input type="checkbox"/> Addendum No. 7 |
| <input type="checkbox"/> Addendum No. 3 | <input type="checkbox"/> Addendum No. 8 |
| <input type="checkbox"/> Addendum No. 4 | <input type="checkbox"/> Addendum No. 9 |
| <input type="checkbox"/> Addendum No. 5 | <input type="checkbox"/> Addendum No. 10 |

I understand that failure to confirm the receipt of addenda may be cause for rejection of this bid. I further understand that any verbal representation made or assumed to be made during any oral discussion held between Vendor's representatives and any state personnel is not binding. Only the information issued in writing and added to the specifications by an official addendum is binding.

Voatz, Inc.

Company



Authorized Signature

11/07/2023

Date

NOTE: This addendum acknowledgment should be submitted with the bid to expedite document processing.

Revised 6/8/2012

1. Attachment B: OWASP Application Level Security Verification Levels 1 – 3: Documentation

| Attachment B: OWASP Application Level Security Verification Levels 1 - 3 | | Yes/No/ Partial |
|--|--|--------------------|
| Open Web Application Security Project - Application Security Verification Standard 4.0.2 Level 1 | | |
| Password Security Requirements | | |
| Verify that user set passwords are at least 8 characters in length (after multiple spaces are combined). | | Yes |
| Verify that passwords 64 characters are longer are permitted but may be no longer than 128 characters. | | Yes |
| Verify that password truncation is not performed. However, consecutive multiple spaces may be replaced by a single space. | | Yes |
| Verify that any printable Unicode character, including language neutral characters such as spaces and Emojis are permitted in passwords. | | Yes |
| Verify users can change their password. | | Yes |
| Verify that password change functionality requires the user's current and new password. | | Yes |
| Verify that passwords submitted during account registration, login, and password change are checked against a set of breached passwords either locally (such as the top 1,000 or 10,000 most common passwords which match the system's password policy) or using an external API. If using an API a zero knowledge proof or other mechanism should be used to ensure that the plain text password is not sent or used in verifying the breach status of the password. If the password is breached, the application must require the user to set a new non-breached password. | | Yes |
| Verify that a password strength meter is provided to help users set a stronger password. | | Yes |
| Verify that there are no password composition rules limiting the type of characters permitted. There should be no requirement for upper or lower case or numbers or special characters. | | Yes |
| Verify that there are no periodic credential rotation or password history requirements. | | Yes |

| | |
|---|-----|
| Verify that "paste" functionality, browser password helpers, and external password managers are permitted. | Yes |
| Verify that the user can choose to either temporarily view the entire masked password, or temporarily view the last typed character of the password on platforms that do not have this as a built-in functionality. | Yes |
| General Authenticator Requirements | |
| Verify that anti-automation controls are effective at mitigating breached credential testing, brute force, and account lockout attacks. Such controls include blocking the most common breached passwords, soft lockouts, rate limiting, CAPTCHA, ever increasing delays between attempts, IP address restrictions, or risk-based restrictions such as location, first login on a device, recent attempts to unlock the account, or similar. Verify that no more than 100 failed attempts per hour is possible on a single account. | Yes |
| Verify that the use of weak authenticators (such as SMS and email) is limited to secondary verification and transaction approval and not as a replacement for more secure authentication methods. Verify that stronger methods are offered before weak methods, users are aware of the risks, or that proper measures are in place to limit the risks of account compromise. | Yes |
| Verify that secure notifications are sent to users after updates to authentication details, such as credential resets, email or address changes, logging in from unknown or risky locations. The use of push notifications - rather than SMS or email - is preferred, but in the absence of push notifications, SMS or email is acceptable as long as no sensitive information is disclosed in the notification. | Yes |
| Authenticator Lifecycle Requirements | |
| Verify system generated initial passwords or activation codes SHOULD be securely randomly generated, SHOULD be at least 6 characters long, and MAY contain letters and numbers, and expire after a short period of time. These initial secrets must not be permitted to become the long term password. | Yes |
| Credential Recovery Requirements | |
| Verify that a system generated initial activation or recovery secret is not sent in clear text to the user. | Yes |
| Verify password hints or knowledge-based authentication (so-called "secret questions") are not present. | Yes |
| Verify password credential recovery does not reveal the current password in any way. | Yes |
| Verify shared or default accounts are not present (e.g. "root", "admin", or "sa"). | Yes |

| | |
|---|-----|
| Verify that if an authentication factor is changed or replaced, that the user is notified of this event. | Yes |
| Verify forgotten password, and other recovery paths use a secure recovery mechanism, such as time-based OTP (TOTP) or other soft token, mobile push, or another offline recovery mechanism. (C6) | Yes |
| Out of Band Verifier Requirements | |
| Verify that clear text out of band (NIST "restricted") authenticators, such as SMS or PSTN, are not offered by default, and stronger alternatives such as push notifications are offered first. | Yes |
| Verify that the out of band verifier expires out of band authentication requests, codes, or tokens after 10 minutes. | Yes |
| Verify that the out of band verifier authentication requests, codes, or tokens are only usable once, and only for the original authentication request. | Yes |
| Verify that the out of band authenticator and verifier communicates over a secure independent channel. | Yes |
| Single or Multi-factor One Time Verifier Requirements | |
| Verify that time-based OTPs have a defined lifetime before expiring | Yes |
| Fundamental Session Management Requirements | |
| Verify the application never reveals session tokens in URL parameters | Yes |
| Session Binding Requirements | |
| Verify the application generates a new session token on user authentication. | Yes |
| Verify that session tokens possess at least 64 bits of entropy. | Yes |
| Verify the application only stores session tokens in the browser using secure methods such as appropriately secured cookies (see section 3.4) or HTML 5 session storage. | Yes |
| Session Logout and Timeout Requirements | |
| Verify that logout and expiration invalidate the session token, such that the back button or a downstream relying party does not resume an authenticated session, including across relying parties. | Yes |
| Cookie Based Session Management | Yes |

| | |
|--|-----|
| Verify that cookie-based session tokens have the 'Secure' attribute set. | Yes |
| Verify that cookie-based session tokens have the 'HttpOnly' attribute set. | Yes |
| Verify that cookie-based session tokens utilize the 'SameSite' attribute to limit exposure to cross-site request forgery attacks. | Yes |
| Verify that cookie-based session tokens use "__Host-" prefix (see references) to provide session cookie confidentiality. | Yes |
| Verify that if the application is published under a domain name with other applications that set or use session cookies that might override or disclose the session cookies, set the path attribute in cookie-based session tokens using the most precise path possible. | Yes |
| General Access Control Design | |
| Verify that the application enforces access control rules on a trusted service layer, especially if client-side access control is present and could be bypassed. | Yes |
| Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized. | Yes |
| Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege. | Yes |
| Verify that the principle of deny by default exists whereby new users/roles start with minimal or no permissions and users/roles do not receive access to new features until access is explicitly assigned. | Yes |
| Verify that access controls fail securely including when an exception occurs. | Yes |
| Operation Level Access Control | |
| Verify that the sensitive data and APIs are protected against Insecure Direct Object Reference (IDOR) attacks targeting creation, reading, updating and deletion of records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records. | Yes |
| Verify that the application or framework enforces a strong anti-CSRF mechanism to protect authenticated functionality, and effective anti-automation or anti-CSRF protects unauthenticated functionality. | Yes |
| Other Access Control Considerations | |
| Verify administrative interfaces use appropriate multi-factor authentication to prevent unauthorized use. | Yes |

| | |
|---|-----|
| Verify that directory browsing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of files or directory metadata such as Thumbs.db, .DS_Store, .git or .svn folders. | Yes |
| Input Validation Requirements | |
| Verify that the application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, or environment variables). | Yes |
| Verify that frameworks protect against mass parameter assignment attacks, or that the application has countermeasures to protect against unsafe parameter assignment, such as marking fields private or similar. | Yes |
| Verify that all input (HTML form fields, REST requests, URL parameters, HTTP headers, cookies, batch files, RSS feeds, etc) is validated using positive validation (allow lists). | Yes |
| Verify that structured data is strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers or telephone, or validating that two related fields are reasonable, such as checking that suburb and zip/postcode match). | Yes |
| Verify that URL redirects and forwards only allow destinations which appear on an allow list, or show a warning when redirecting to potentially untrusted content. | Yes |
| Sanitization and Sandboxing Requirements | |
| Verify that all untrusted HTML input from WYSIWYG editors or similar is properly sanitized with an HTML sanitizer library or framework feature. | Yes |
| Verify that unstructured data is sanitized to enforce safety measures such as allowed characters and length. | Yes |
| Verify that the application sanitizes user input before passing to mail systems to protect against SMTP or IMAP injection. | Yes |
| Verify that the application avoids the use of eval() or other dynamic code execution features. Where there is no alternative, any user input being included must be sanitized or sandboxed before being executed. | Yes |
| Verify that the application protects against template injection attacks by ensuring that any user input being included is sanitized or sandboxed. | Yes |
| Verify that the application protects against SSRF attacks, by validating or sanitizing untrusted data or HTTP file metadata, such as filenames and URL input fields, use whitelisting of protocols, domains, paths and ports. | Yes |

| | |
|---|-----|
| Verify that the application sanitizes, disables, or sandboxes user-supplied SVG scriptable content, especially as they relate to XSS resulting from inline scripts, and foreignObject. | Yes |
| Verify that the application sanitizes, disables, or sandboxes user-supplied scriptable or expression template language content, such as Markdown, CSS or XSL stylesheets, BBCode, or similar. | Yes |
| Output Encoding and Injection Prevention Requirements | |
| Verify that output encoding is relevant for the interpreter and context required. For example, use encoders specifically for HTML values, HTML attributes, JavaScript, URL parameters, HTTP headers, SMTP, and others as the context requires, especially from untrusted inputs (e.g. names with Unicode or apostrophes, such as ねこ or O'Hara). | Yes |
| Verify that output encoding preserves the user's chosen character set and locale, such that any Unicode character point is valid and safely handled. | Yes |
| Verify that context-aware, preferably automated - or at worst, manual - output escaping protects against reflected, stored, and DOM based XSS. | Yes |
| Verify that data selection or database queries (e.g. SQL, HQL, ORM, NoSQL) use parameterized queries, ORMs, entity frameworks, or are otherwise protected from database injection attacks. | Yes |
| Verify that where parameterized or safer mechanisms are not present, context-specific output encoding is used to protect against injection attacks, such as the use of SQL escaping to protect against SQL injection. | Yes |
| Verify that the application projects against JavaScript or JSON injection attacks, including for eval attacks, remote JavaScript includes, CSP bypasses, DOM XSS, and JavaScript expression evaluation. | Yes |
| Verify that the application protects against LDAP Injection vulnerabilities, or that specific security controls to prevent LDAP Injection have been implemented. | Yes |
| Verify that the application protects against OS command injection and that operating system calls use parameterized OS queries or use contextual command line output encoding. | Yes |
| Verify that the application protects against Local File Inclusion (LFI) or Remote File Inclusion (RFI) attacks. | Yes |
| Verify that the application protects against XPath injection or XML injection attacks. | Yes |
| Deserialization Prevention Requirements | |
| Verify that serialized objects use integrity checks or are encrypted to prevent hostile object creation or data tampering. | Yes |

| | |
|---|-----|
| Verify that the application correctly restricts XML parsers to only use the most restrictive configuration possible and to ensure that unsafe features such as resolving external entities are disabled to prevent XML eXternal Entity (XXE) attacks. | Yes |
| Verify that desterialization of untrusted data is avoided or protected in both custom code and third-party libraries (such as JSON, XML and YAML parsers). | Yes |
| Verify that when parsing JSON in browsers or JavaScript-based backends, JSON.parse is used to parse the JSON document. Do not use eval() to parse JSON. | Yes |
| Algorithms | |
| Verify that all cryptographic modules fail securely, and errors are handled in a way that does not enable Padding Oracle attacks. | Yes |
| Log Content Requirements | |
| Verify that the application does not log credentials or payment details. Session tokens should only be stored in logs in an irreversible, hashed form. | Yes |
| Verify that the application does not log other sensitive data as defined under local privacy laws or relevant security policy. | Yes |
| Error Handling | |
| Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate. | Yes |
| Client-side Data Protection | |
| Verify the application sets sufficient anti-caching headers so that sensitive data is not cached in modern browsers. | Yes |
| Verify that data stored in client side storage (such as HTML5 local storage, session storage, IndexedDB, regular cookies or Flash cookies) does not contain sensitive data or PII. | Yes |
| Verify that authenticated data is cleared from client storage, such as the browser DOM, after the client or session is terminated. | Yes |
| Sensitive Private Data | |
| Verify that sensitive data is sent to the server in the HTTP message body or headers, and that query string parameters from any HTTP verb do not contain sensitive data. | Yes |
| Verify that users have a method to remove or export their data on demand. | Yes |

| | |
|---|-----|
| Verify that users are provided clear language regarding collection and use of supplied personal information and that users have provided opt-in consent for the use of that data before it is used in any way. | Yes |
| Verify that all sensitive data created and processed by the application has been identified, and ensure that a policy is in place on how to deal with sensitive data. | Yes |
| Deployed Application Integrity Controls | |
| Verify that if the application has a client or server auto-update feature, updates should be obtained over secure channels and digitally signed. The update code must validate the digital signature of the update before installing or executing the update. | Yes |
| Verify that the application employs integrity protections, such as code signing or sub-resource integrity. The application must not load or execute code from untrusted sources, such as loading includes, modules, plugins, code, or libraries from untrusted sources or the Internet. | Yes |
| Verify that the application has protection from sub-domain takeovers if the application relies upon DNS entries or DNS sub-domains, such as expired domain names, out of date DNS pointers or CNAMEs, expired projects at public source code repos, or transient cloud APIs, serverless functions, or storage buckets (<i>autogen-bucket-id.cloud.example.com</i>) or similar. Protections can include ensuring that DNS names used by applications are regularly checked for expiry or change. | Yes |
| Business Logic Security Requirements | |
| Verify the application will only process business logic flows for the same user in sequential step order and without skipping steps. | Yes |
| Verify the application will only process business logic flows with all steps being processed in realistic human time, i.e. transactions are not submitted too quickly. | Yes |
| Verify the application has appropriate limits for specific business actions or transactions which are correctly enforced on a per user basis. | Yes |
| Verify the application has sufficient anti-automation controls to detect and protect against data exfiltration, excessive business logic requests, excessive file uploads or denial of service attacks. | Yes |
| Verify the application has business logic limits or validation to protect against likely business risks or threats, identified using threat modeling or similar methodologies. | Yes |
| File Upload Requirements | |
| Verify that the application will not accept large files that could fill up storage or cause a denial of service. | Yes |

| File Execution Requirements | |
|---|-----|
| Verify that user-submitted filename metadata is not used directly with system or framework file and URL API to protect against path traversal. | Yes |
| Verify that user-submitted filename metadata is validated or ignored to prevent the disclosure, creation, updating or removal of local files (LFI). | Yes |
| Verify that user-submitted filename metadata is validated or ignored to prevent the disclosure or execution of remote files (RFI), which may also lead to SSRF. | Yes |
| Verify that the application protects against reflective file download (RFD) by validating or ignoring user-submitted filenames in a JSON, JSONP, or URL parameter, the response Content-Type header should be set to text/plain, and the Content-Disposition header should have a fixed filename. | Yes |
| Verify that untrusted file metadata is not used directly with system API or libraries, to protect against OS command injection. | Yes |
| File Storage Requirements | |
| Verify that files obtained from untrusted sources are stored outside the web root, with limited permissions, preferably with strong validation. | Yes |
| Verify that files obtained from untrusted sources are scanned by antivirus scanners to prevent upload of known malicious content. | Yes |
| File Download Requirements | |
| Verify that the web tier is configured to serve only files with specific file extensions to prevent unintentional information and source code leakage. For example, backup files (e.g. .bak), temporary working files (e.g. .swp), compressed files (.zip, .tar.gz, etc) and other extensions commonly used by editors should be blocked unless required. | Yes |
| Verify that direct requests to uploaded files will never be executed as HTML/JavaScript content. | Yes |
| SSRF Protection Requirements | |
| Verify that the web or application server is configured with an allow list of resources or systems to which the server can send requests or load data/files from. | Yes |
| Generic Web Service Security Verification Requirements | |
| Verify that all application components use the same encodings and parsers to avoid parsing attacks that exploit different URI or file parsing behavior that could be used in SSRF and RFI attacks. | Yes |

| | |
|--|-----|
| Verify that access to administration and management functions is limited to authorized administrators. | Yes |
| Verify API URLs do not expose sensitive information, such as the API key, session tokens etc. | Yes |
| RESTful Web Service Verification Requirements | |
| Verify that enabled RESTful HTTP methods are a valid choice for the user or action, such as preventing normal users using DELETE or PUT on protected API or resources. | Yes |
| Verify that JSON schema validation is in place and verified before accepting input. | Yes |
| Verify that RESTful web services that utilize cookies are protected from Cross-Site Request Forgery via the use of at least one or more of the following: triple or double submit cookie pattern (see references), CSRF nonces, or Origin request header checks. | Yes |
| SOAP Web Service Verification Requirements | |
| Verify that XSD schema validation takes place to ensure a properly formed XML document, followed by validation of each input field before any processing of that data takes place. | Yes |
| Dependency | |
| Verify that all components are up to date, preferably using a dependency checker during build or compile time. | Yes |
| Verify that all unneeded features, documentation, samples, configurations are removed, such as sample applications, platform documentation, and default or example users. | Yes |
| Verify that if application assets, such as JavaScript libraries, CSS stylesheets or web fonts, are hosted externally on a content delivery network (CDN) or external provider, Subresource Integrity (SRI) is used to validate the integrity of the asset. | Yes |
| Unintended Security Disclosure Requirements | |
| Verify that web or application server and framework error messages are configured to deliver user actionable, customized responses to eliminate any unintended security disclosures. | Yes |
| Verify that web or application server and application framework debug modes are disabled in production to eliminate debug features, developer consoles, and unintended security disclosures. | Yes |
| Verify that the HTTP headers or any part of the HTTP response do not expose detailed version information of system components. | Yes |
| HTTP Security Headers Requirements | |

| | |
|---|-----|
| Verify that every HTTP response contains a Content-Type header. text/*, /+xml and application/xml content types should also specify a safe character set (e.g., UTF-8, ISO-8859-1). | Yes |
| Verify that all API responses contain Content-Disposition: attachment; filename="api.json" header (or other appropriate filename for the content type). | Yes |
| Verify that a Content Security Policy (CSP) response header is in place that helps mitigate impact for XSS attacks like HTML, DOM, JSON, and JavaScript injection vulnerabilities. | Yes |
| Verify that all responses contain an X-Content-Type-Options: nosniff header. | Yes |
| Verify that a Strict-Transport-Security header is included on all responses and for all subdomains, such as Strict-Transport-Security: max-age=15724800; includeSubdomains. | Yes |
| Verify that a suitable "Referrer-Policy" header is included, such as "no-referrer" or "same-origin". | Yes |
| Verify that the content of a web application cannot be embedded in a third party site by default and that embedding of the exact resources is only allowed where necessary by using suitable Content-Security-Policy: frame-ancestors and X-Frame-Options response headers. | Yes |
| Validate HTTP Request Header Requirements | |
| Verify that the application server only accepts the HTTP methods in use by the application/API, including pre-flight OPTIONS and logs/alerts on any requests that are not valid for the application context. | Yes |
| Verify that the supplied Origin header is not used for authentication or access control decisions, as the Origin header can easily be changed by an attacker. | Yes |
| Verify that the Cross Origin Resource Sharing (CORS) Access-Control-Allow-Origin header uses a strict allow list of trusted domains and subdomains to match against and does not support the "null" origin. | Yes |
| Select Controls from Open Web Application Security Project - Application Security Verification Standard 4.0.2 Level 2 | |
| Secure Software Development Lifecycle | |
| Verify the use of a secure software development lifecycle that addresses security in all stages of development. | Yes |
| Verify documentation and justification of all the application's trust boundaries, components, and significant data flows. | Yes |

| | |
|---|-----|
| Verify implementation of centralized, simple (economy of design), vetted, secure, and reusable security controls to avoid duplicate, missing, ineffective, or insecure controls. | Yes |
| Verify availability of a secure coding checklist, security requirements, guideline, or policy to all developers and testers. | Yes |
| Authentication Architectural Requirements | |
| Verify that communications between application components, including APIs, middleware and data layers, are authenticated. Components should have the least necessary privileges needed. | Yes |
| Verify that the application uses a single vetted authentication mechanism that is known to be secure, can be extended to include strong authentication, and has sufficient logging and monitoring to detect abuse or breaches. | Yes |
| Verify that all authentication pathways and identity management APIs implement consistent authentication security control strength, such that there are no weaker alternatives per the risk of the application. | Yes |
| Access Control Architectural Requirements | |
| Verify enforcement of the principle of least privilege in functions, data files, URLs, controllers, services, and other resources. This implies protection against spoofing and elevation of privilege. | Yes |
| Verify the application uses a single and well-vetted access control mechanism for accessing protected data and resources. All requests must pass through this single mechanism to avoid copy and paste or insecure alternative paths. | Yes |
| Verify that attribute or feature-based access control is used whereby the code checks the user's authorization for a feature/data item rather than just their role. Permissions should still be allocated using roles. | Yes |
| Input and Output Architectural Requirements | |
| Verify that input and output requirements clearly define how to handle and process data based on type, content, and applicable laws, regulations, and other policy compliance. | Yes |
| Verify that serialization is not used when communicating with untrusted clients. If this is not possible, ensure that adequate integrity controls (and possibly encryption if sensitive data is sent) are enforced to prevent deserialization attacks including object injection. | Yes |
| Cryptographic Architectural Requirements | |
| Verify that there is an explicit policy for management of cryptographic keys and that a cryptographic key lifecycle follows a key management standard such as NIST SP 800-57. | Yes |

| | |
|--|-----|
| Verify that consumers of cryptographic services protect key material and other secrets by using key vaults or API based alternatives. | Yes |
| Verify that all keys and passwords are replaceable and are part of a well-defined process to re-encrypt sensitive data. | Yes |
| Verify that the architecture treats client-side secrets – such as symmetric keys, passwords, or API tokens – as insecure and never uses them to protect or access sensitive data. | Yes |
| Errors, Logging and Auditing Architectural Requirements | |
| Verify that a common logging format and approach is used across the system. | Yes |
| Verify that logs are securely transmitted to a preferably remote system for analysis, detection, alerting, and escalation. | Yes |
| Data Protection and Privacy Architectural Requirements | |
| Verify that all sensitive data is identified and classified into protection levels. | Yes |
| Verify that all protection levels have an associated set of protection requirements, such as encryption requirements, integrity requirements, retention, privacy and other confidentiality requirements, and that these are applied in the architecture. | Yes |
| Communications Architecture Requirements | |
| Verify the application encrypts communications between components, particularly when these components are in different containers, systems, sites, or cloud providers. | Yes |
| Verify that application components verify the authenticity of each side in a communication link to prevent person-in-the-middle attacks. For example, application components should validate TLS certificates and chains. | Yes |
| Malicious Software Architectural Requirements | |
| Verify that a source code control system is in use, with procedures to ensure that check-ins are accompanied by issues or change tickets. The source code control system should have access control and identifiable users to allow traceability of any changes. | Yes |
| Secure File Upload Architectural Requirements | |
| Verify that user-uploaded files are stored outside of the web root. | Yes |
| Verify that user-uploaded files - if required to be displayed or downloaded from the application - are served by either octet stream downloads, or from an unrelated domain, such as a cloud | Yes |

| | |
|--|-----|
| file storage bucket. Implement a suitable content security policy to reduce the risk from XSS vectors or other attacks from the uploaded file. | |
| Configuration Architectural Requirements | |
| Verify the segregation of components of differing trust levels through well-defined security controls, firewall rules, API gateways, reverse proxies, cloud-based security groups, or similar mechanisms. | Yes |
| Verify that the build pipeline warns of out-of-date or insecure components and takes appropriate actions. | Yes |
| Verify that the build pipeline contains a build step to automatically build and verify the secure deployment of the application, particularly if the application infrastructure is software defined, such as cloud environment build scripts. | Yes |
| Verify that application deployments adequately sandbox, containerize and/or isolate at the network level to delay and deter attackers from attacking other applications, especially when they are performing sensitive or dangerous actions such as deserialization. | Yes |
| Verify the application does not use unsupported, insecure, or deprecated client-side technologies such as NSAPI plugins, Flash, Shockwave, ActiveX, Silverlight, NACL, or client-side Java applets. | Yes |
| Credential Storage Requirements | |
| Verify that passwords are stored in a form that is resistant to offline attacks. Passwords SHALL be salted and hashed using an approved one-way key derivation or password hashing function. Key derivation and password hashing functions take a password, a salt, and a cost factor as inputs when generating a password hash. | Yes |
| Verify that the salt is at least 32 bits in length and be chosen arbitrarily to minimize salt value collisions among stored hashes. For each credential, a unique salt value and the resulting hash SHALL be stored. | Yes |
| Verify that if PBKDF2 is used, the iteration count SHOULD be as large as verification server performance will allow, typically at least 100,000 iterations. | Yes |
| Credential Recovery Requirements | |
| Verify that if OTP or multi-factor authentication factors are lost, that evidence of identity proofing is performed at the same level as during enrollment. | Yes |
| Look-up Secret Verifier Requirements | |
| Verify that lookup secrets can be used only once. | Yes |

| | |
|---|-----|
| Verify that lookup secrets have sufficient randomness (112 bits of entropy), or if less than 112 bits of entropy, salted with a unique and random 32-bit salt and hashed with an approved one-way hash. | Yes |
| Verify that lookup secrets are resistant to offline attacks, such as predictable values. | Yes |
| Out of Band Verifier Requirements | |
| Verify that the out of band verifier retains only a hashed version of the authentication code. | Yes |
| Verify that the initial authentication code is generated by a secure random number generator, containing at least 20 bits of entropy (typically a six digit random number is sufficient). | Yes |
| Single or Multi-factor One Time Verifier Requirements | |
| Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage. | Yes |
| Verify that approved cryptographic algorithms are used in the generation, seeding, and verification of OTPs. | Yes |
| Verify that time-based OTP can be used only once within the validity period. | Yes |
| Verify that if a time-based multi factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device. | Yes |
| Cryptographic Software and Devices Verifier Requirements | |
| Verify that cryptographic keys used in verification are stored securely and protected against disclosure, such as using a Trusted Platform Module (TPM) or Hardware Security Module (HSM), or an OS service that can use this secure storage. | Yes |
| Verify that the challenge nonce is at least 64 bits in length, and statistically unique or unique over the lifetime of the cryptographic device. | Yes |
| Verify that approved cryptographic algorithms are used in the generation, seeding, and verification. | Yes |
| Service Authentication Requirements | |
| Verify that if passwords are required for service authentication, the service account used is not a default credential (e.g. root/root or admin/admin are default in some services during installation). | Yes |
| Verify that passwords are stored with sufficient protection to prevent offline recovery attacks, including local system access. | Yes |

| | |
|--|-----|
| Session Binding Requirements | |
| Verify that session tokens are generated using approved cryptographic algorithms. | Yes |
| Session Logout and Timeout Requirements | |
| Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties. | Yes |
| Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices. | Yes |
| Token Based Session Management | |
| Verify the application uses session tokens rather than static API secrets and keys, except with legacy implementations. | Yes |
| Verify that stateless session tokens use digital signatures, encryption, and other countermeasures to protect against tampering, enveloping, replay, null cipher, and key substitution attacks. | Yes |
| Other Access Control Considerations | |
| Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems and/or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud. | Yes |
| Data Classification | |
| Verify that regulated private data is stored encrypted while at rest, such as Personally Identifiable Information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR. | Yes |
| Algorithms | |
| Verify that industry proven or government approved cryptographic algorithms, modes, and libraries are used, instead of custom coded cryptography. | Yes |
| Verify that encryption initialization vector, cipher configuration, and block modes are configured securely using the latest advice. | Yes |
| Verify that random number, encryption or hashing algorithms, key lengths, rounds, ciphers or modes, can be reconfigured, upgraded, or swapped at any time, to protect against cryptographic breaks. | Yes |

| | |
|---|-----|
| Verify that known insecure block modes (i.e. ECB, etc.), padding modes (i.e. PKCS#1 v1.5, etc.), ciphers with small block sizes (i.e. Triple-DES, Blowfish, etc.), and weak hashing algorithms (i.e. MD5, SHA1, etc.) are not used unless required for backwards compatibility. | Yes |
| Verify that nonces, initialization vectors, and other single use numbers must not be used more than once with a given encryption key. The method of generation must be appropriate for the algorithm being used. | Yes |
| Random Values | |
| Verify that all random numbers, random file names, random GUIDs, and random strings are generated using the cryptographic module's approved cryptographically secure random number generator when these random values are intended to be not guessable by an attacker. | Yes |
| Verify that random GUIDs are created using the GUID v4 algorithm, and a Cryptographically-secure Pseudo-random Number Generator (CSPRNG). GUIDs created using other pseudo-random number generators may be predictable. | Yes |
| Secret Management | |
| Verify that a secrets management solution such as a key vault is used to securely create, store, control access to and destroy secrets. | Yes |
| Verify that key material is not exposed to the application but instead uses an isolated security module like a vault for cryptographic operations. | Yes |
| Log Content Requirements | |
| Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures. | Yes |
| Verify that each log event includes necessary information that would allow for a detailed investigation of the timeline when an event happens. | Yes |
| Log Processing Requirements | |
| Verify that all authentication decisions are logged, without storing sensitive session identifiers or passwords. This should include requests with relevant metadata needed for security investigations. | Yes |
| Verify that all access control decisions can be logged and all failed decisions are logged. This should include requests with relevant metadata needed for security investigations. | Yes |
| Log Protection Requirements | |

| | |
|---|-----|
| Verify that the application appropriately encodes user-supplied data to prevent log injection. | Yes |
| Verify that all events are protected from injection when viewed in log viewing software. | Yes |
| Verify that security logs are protected from unauthorized access and modification. | Yes |
| Verify that time sources are synchronized to the correct time and time zone. Strongly consider logging only in UTC if systems are global to assist with post-incident forensic analysis. | Yes |
| Error Handling | |
| Verify that a "last resort" error handler is defined which will catch all unhandled exceptions. | Yes |
| General Data Protection | |
| Verify the application protects sensitive data from being cached in server components such as load balancers and application caches. | Yes |
| Verify that all cached or temporary copies of sensitive data stored on the server are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data. | Yes |
| Verify the application minimizes the number of parameters in a request, such as hidden fields, Ajax variables, cookies and header values. | Yes |
| Verify the application can detect and alert on abnormal numbers of requests, such as by IP, user, total per hour or day, or whatever makes sense for the application. | Yes |
| Sensitive Private Data | |
| Verify accessing sensitive data is audited (without logging the sensitive data itself), if the data is collected under relevant data protection directives or where logging of access is required. | Yes |
| Verify that sensitive information contained in memory is overwritten as soon as it is no longer required to mitigate memory dumping attacks, using zeroes or random data. | Yes |
| Verify that sensitive or private information that is required to be encrypted, is encrypted using approved algorithms that provide both confidentiality and integrity. | Yes |
| Verify that sensitive personal information is subject to data retention classification, such that old or out of date data is deleted automatically, on a schedule, or as the situation requires. | Yes |
| Client Communications Security Requirements | |
| Verify that secured TLS is used for all client connectivity, and does not fall back to insecure or unencrypted protocols. | Yes |

| | |
|--|-----|
| Verify using online or up to date TLS testing tools that only strong algorithms, ciphers, and protocols are enabled, with the strongest algorithms and ciphers set as preferred. | Yes |
| Verify that old versions of SSL and TLS protocols, algorithms, ciphers, and configuration are disabled, such as SSLv2, SSLv3, or TLS 1.0 and TLS 1.1. The latest version of TLS should be the preferred cipher suite. | Yes |
| Server Communications Security Requirements | |
| Verify that connections to and from the server use trusted TLS certificates. Where internally generated or self-signed certificates are used, the server must be configured to only trust specific internal CAs and specific self-signed certificates. All others should be rejected. | Yes |
| Verify that encrypted communications such as TLS is used for all inbound and outbound connections, including for management ports, monitoring, authentication, API, or web service calls, database, cloud, serverless, mainframe, external, and partner connections. The server must not fall back to insecure or unencrypted protocols. | Yes |
| Verify that all encrypted connections to external systems that involve sensitive information or functions are authenticated. | Yes |
| Verify that proper certification revocation, such as Online Certificate Status Protocol (OCSP) Stapling, is enabled and configured. | Yes |
| Malicious Code Search | |
| Verify that the application source code and third party libraries do not contain unauthorized phone home or data collection capabilities. Where such functionality exists, obtain the user's permission for it to operate before collecting any data. | Yes |
| Verify that the application does not ask for unnecessary or excessive permissions to privacy related features or sensors, such as contacts, cameras, microphones, or location. | Yes |
| Business Logic Security Requirements | |
| Verify the application has configurable alerting when automated attacks or unusual activity is detected. | Yes |
| File Integrity Requirements | |
| Verify that the files obtained from untrusted sources are validated to be of expected type based on the file's content. | Yes |
| File Execution Requirements | |

| | |
|---|-----|
| Verify that the application does not include and execute functionality from untrusted sources, such as unverified content distribution networks, JavaScript libraries, node npm libraries, or server-side DLLs. | Yes |
| RESTful Web Service Verification Requirements | |
| Verify that the message headers and payload are trustworthy and not modified in transit. Requiring strong encryption for transport (TLS only) may be sufficient in many cases as it provides both confidentiality and integrity protection. Per-message digital signatures can provide additional assurance on top of the transport protections for high-security applications but bring with them additional complexity and risks to weigh against the benefits. | Yes |
| SOAP Web Service Verification Requirements | |
| Verify that the message payload is signed using WS-Security to ensure reliable transport between client and service. | Yes |
| GraphQL and other Web Service Data Layer Security Requirements | |
| Verify that a query allow list or a combination of depth limiting and amount limiting should be used to prevent GraphQL or data layer expression Denial of Service (DoS) as a result of expensive, nested queries. For more advanced scenarios, query cost analysis should be used. | Yes |
| Build | |
| Verify that the application build and deployment processes are performed in a secure and repeatable way, such as CI / CD automation, automated configuration management, and automated deployment scripts. | Yes |
| Verify that compiler flags are configured to enable all available buffer overflow protections and warnings, including stack randomization, data execution prevention, and to break the build if an unsafe pointer, memory, format string, integer, or string operations are found. | Yes |
| Verify that server configuration is hardened as per the recommendations of the application server and frameworks in use. | Yes |
| Verify that the application, configuration, and all dependencies can be re-deployed using automated deployment scripts, built from a documented and tested runbook in a reasonable time, or restored from backups in a timely fashion. | Yes |
| Dependency | |
| Verify that third party components come from pre-defined, trusted and continually maintained repositories. | Yes |

| | |
|--|-----|
| Verify that the attack surface is reduced by sandboxing or encapsulating third party libraries to expose only the required behavior into the application. | Yes |
| Validate HTTP Request Header Requirements | |
| Verify that HTTP headers added by a trusted proxy or SSO devices, such as bearer token, are authenticated by the application. | Yes |
| Select Controls from Open Web Application Security Project - Application Security Verification Standard 4.0.2 Level 3 | |
| General Authenticator Requirements | |
| Verify impersonation resistance against phishing, such as the use of multi-factor authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side certificates. | Yes |
| Verify that where a credential service provider (CSP) and the application verifying authentication are separated, mutually authenticated TLS is in place between the two endpoints. | Yes |
| Session Logout and Timeout Requirements | |
| If authenticators permit users to remain logged in, verify that re-authentication occurs periodically with 2FA both when actively used after 12 hours or after an idle period of 15 minutes. | Yes |
| Algorithms | |
| Verify that encrypted data is authenticated via signatures, authenticated cipher modes, or HMAC to ensure that ciphertext is not altered by an unauthorized party | Yes |
| General Data Protection | |
| Verify that regular backups of important data are performed and that test restoration of data is performed. | Yes |
| Verify that backups are stored securely to prevent data from being stolen or corrupted. | Yes |
| Code Integrity Controls | |
| Verify that a code analysis tool is in use that can detect potentially malicious code, such as time functions, unsafe file operations and network connections. | Yes |
| Malicious Code Search | |

| | |
|---|-----|
| Verify that the application source code and third party libraries do not contain back doors, such as hard-coded or additional undocumented accounts or keys, code obfuscation, undocumented binary blobs, rootkits, or anti-debugging, insecure debugging features, or otherwise out of date, insecure, or hidden functionality that could be used maliciously if discovered. | Yes |
| Verify that the application source code and third party libraries do not contain time bombs by searching for date and time related functions. | Yes |
| Verify that the application source code and third party libraries do not contain malicious code, such as salami attacks, logic bypasses, or logic bombs. | Yes |
| Verify that the application source code and third party libraries do not contain Easter eggs or any other potentially unwanted functionality. | Yes |

Attachment B: Supporting Documentation

Open Web Application Security Project - Application Security Verification Standard 4.0.2 Level 1

Password Security Requirements

Verify that user set passwords are at least 8 characters in length (after multiple spaces are combined).

1. Onboarding and Identity Verification with Voatz: See video at time 0:46.
<https://vimeo.com/533255584/1baa3910a6>
2. Blockchain Technology Pilot Using Mobile Voting, City of Chandler, AZ: See time 0:46.
<https://www.chandleraz.gov/government/elections-and-voting/blockchain-mobile-voting>

Comments: Voatz mobile apps require a minimum of 8 characters (iOS devices require an 8-digit PIN and Android devices require a 12-digit PIN). Web apps require a complex password of 10 characters or more.

Verify that passwords 64 characters or longer are permitted but may be no longer than 128 characters.

1. The Voatz Administration Portal allows users to select passwords between 64 and 128 characters.
2. This is not applicable in VMA, since devices that support biometric identification have stronger security than a 64-character password, so biometric identification in VMA aligns with the spirit of this requirement. Currently, VMA requires a minimum of 8 characters (iOS devices require an 8-digit PIN and Android devices require a 12-digit PIN), in alignment with NIST guidelines from October 2020. Furthermore, internal work tickets VMA-1895: [iOS] and VMA-1896: [Android] “Transition from numeric PIN to complex password” are being considered to implement such password functionality in VMA by February 2024. However, these may not be implemented for accessibility reasons.

Verify that password truncation is not performed. However, consecutive multiple spaces may be replaced by a single space.

1. The Voatz Mobile App (VMA) does not perform password truncation, in compliance with updated NIST guidance below from October 2020. Notable aspects of the NIST recommendations include the following.

NIST 800-63B, Section 5.1.1.2: Memorized Secret Verifiers:
Truncation of the secret SHALL NOT be performed.
2. The Voatz Web App (VWA) does not perform password truncation, in compliance with updated NIST guidance below from October 2020. Notable aspects of the NIST recommendations include the following.

NIST 800-63B, Section 5.1.1.2: Memorized Secret Verifiers:
Truncation of the secret SHALL NOT be performed.

Verify that any printable Unicode character, including language neutral characters such as spaces and emojis are permitted in passwords.

1. The Voatz Administration Portal allows any UTF-8 encoded character combination to be set as the password. Internal work ticket VAP-343 will add support for emojis and is in active development following the updated NIST guidelines from October 2020. Notable aspects of the NIST recommendations include the following.
 - No restrictions on character combinations (e.g., requiring uppercase, lowercase, number and special character)
 - No restrictions on any printable Unicode character, including emojis.
2. This is not applicable in VMA for devices that support biometric identification, which is stronger than password-based authentication, so VMA aligns with the spirit of this requirement. Furthermore, internal work tickets VMA-1895: [iOS] and VMA-1896: [Android] “Transition from numeric PIN to complex password” are being considered to implement such password functionality in VMA by February 2024. However, these features may not be implemented for accessibility reasons.

Verify users can change their password.

1. The Voatz Administration Portal allows users to change their password via a two step process involving two-factor authentication. The first step requires an active session with a valid existing password. This sends a TOTP to the user’s device which is then used in the second step along with the new password to change the user’s current password.
2. This is not applicable in VMA for devices that support biometric identification, which is stronger than password-based authentication, so VMA aligns with the spirit of this requirement. To prevent misuse, the Voatz platform currently requires the user to go through a comprehensive re-verification and account reset workflow to change the password. As part of internal work tickets VMA-1895: [iOS] and VMA-1896: [Android], this feature is being considered for implementation in VMA by February 2024. However, this feature may not be implemented for accessibility reasons.

Verify that password change functionality requires the user’s current and new password.

1. The Voatz Administration Portal allows users to change their password via a two step process involving two-factor authentication. The first step requires an active session with a valid existing password. This sends a TOTP to the user’s device which is then used in the second step along with the new password to change the user’s current password.
2. This is not applicable in VMA for devices that support biometric identification, which is stronger than password-based authentication, so VMA aligns with the spirit of this requirement. To prevent misuse, the Voatz platform currently requires the voter to go through a comprehensive re-verification and account reset workflow to change the password. As part of internal work

tickets VMA-1895: [iOS] and VMA-1896: [Android], this feature is being considered for implementation in VMA by February 2024. However, this feature may not be implemented for accessibility reasons.

Verify that passwords submitted during account registration, login, and password change are checked against a set of breached passwords either locally (such as the top 1,000 or 10,000 most common passwords which match the system's password policy) or using an external API. If using an API a zero knowledge proof or other mechanism should be used to ensure that the plain text password is not sent or used in verifying the breach status of the password. If the password is breached, the application must require the user to set a new non-breached password.

1. Internal work ticket VAP-343, which will improve the login process of the Voatz Administration Portal, is in active improvement development and will implement such functionality in alignment with the updated NIST guidelines from October 2020. Notable aspects of the NIST recommendations include the following.
 - Restrict passwords appearing in a blacklist dictionary.
2. This is not applicable in VMA for devices that support biometric identification, which is stronger than password-based authentication, so VMA aligns with the spirit of this requirement. Furthermore, as part of internal work tickets VMA-1895: [iOS] and VMA-1896: [Android], this feature is being considered for implementation in VMA by February 2024. However, this feature may not be implemented for accessibility reasons.

Verify that a password strength meter is provided to help users set a stronger password.

1. Internal work ticket VAP-343, which will improve the login process of the Voatz Administration Portal, is in active improvement development and will implement such functionality in alignment with the updated NIST guidelines from October 2020. Notable aspects of the NIST recommendations include the following.
 - Provide a strength indicator to users.
2. This is not applicable in VMA for devices that support biometric identification, which is stronger than password-based authentication, so VMA aligns with the spirit of this requirement. However, the following code is from the VMA Android app. The app does not allow a PIN with 3 or more consecutive or repeating digits, and the PIN must be at least 12 digits. The app alerts the voter to a weak PIN and helps the voter set a stronger password.

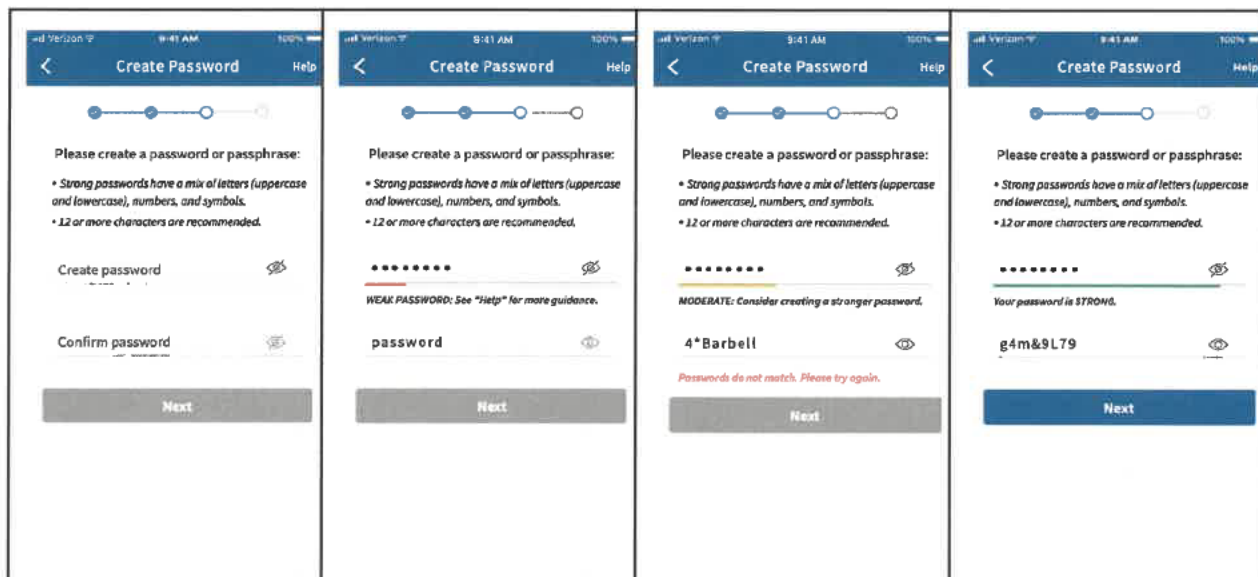
```
private fun isPinValid(pin: String): Boolean {
    val regex = "(?=.{0,12}$)(([0-9])\\2{2}(?!\\2))+\\$"
    return if (pin.length < 12) {
        errorString = resources.getString(R.string.error)
        return false
    } else if (!pin.matches(regex.toRegex()) && checkConsecutive(pin)) {
        errorString = resources.getString(R.string.error_consecutive_identical)
        false
    } else if (!pin.matches(regex.toRegex())) {
        errorString = resources.getString(R.string.error_identical)
    }
}
```

```

false
} else if (checkConsecutive(pin)) {
    errorString = resources.getString(R.string.error_consecutive)
    false
} else {
    true
}

```

As part of internal work tickets VMA-1895: [iOS] and VMA-1896: [Android], this feature is being considered for implementation in VMA by February 2024. The following are sample screens. Four security levels are proposed to be shown: Gray for no password, Red for insecure passwords (less than 48 bits of entropy), Yellow for passwords of moderate (48 to 64 bits) entropy, and Green for high (more than 64 bits) entropy.



Verify that there are no password composition rules limiting the type of characters permitted. There should be no requirement for upper or lower case or numbers or special characters.

1. The Voatz Administration Portal permits UTF-8 encoded characters but does not limit the type of characters permitted via password composition rules.
2. This is not applicable in VMA for devices that support biometric identification, which is stronger than password-based authentication, so VMA aligns with the spirit of this requirement. However, as part of internal work tickets VMA-1895: [iOS] and VMA-1896: [Android], this feature is being considered for implementation in VMA by February 2024. However, this feature may not be implemented for accessibility reasons.

Verify that there are no periodic credential rotation or password history requirements.

1. This is fully supported on all customer facing mobile platforms in accordance with the updated guidance from NIST in October 2020. Notable aspects of the NIST recommendations:
NIST 800-63B, Section 5.1.1.2: Memorized Secret Verifiers:
Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically).
2. This is fully supported on all customer facing web platforms in accordance with the updated guidance from NIST in October 2020. Notable aspects of the NIST recommendations include the following.
NIST 800-63B, Section 5.1.1.2: Memorized Secret Verifiers:
Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically).

Comments: This feature is currently implemented.

Verify that "paste" functionality, browser password helpers, and external password managers are permitted.

1. Internal work ticket VAP-343, which will improve the login process of the Voatz Administration Portal, is in active development and will implement such functionality in alignment with the updated NIST guidelines from October 2020. Notable aspects of the NIST recommendations include the following.
 - Allow integration/paste with password managers.
2. This is not applicable in VMA for devices that support biometric identification, which is stronger than password-based authentication, so VMA aligns with the spirit of this requirement. However, as part of internal work tickets VMA-1895: [iOS] and VMA-1896: [Android], this feature is being considered for implementation in VMA by February 2024, though we may not allow paste functionality for security reasons.

Verify that the user can choose to either temporarily view the entire masked password, or temporarily view the last typed character of the password on platforms that do not have this as a built-in functionality.

1. Onboarding and Identity Verification with Voatz: See time 0:53.
<https://vimeo.com/533255584/1baa3910a6>
2. Blockchain Technology Pilot Using Mobile Voting, City of Chandler, AZ: See time 0:53.
<https://www.chandleraz.gov/government/elections-and-voting/blockchain-mobile-voting>

Comments: Please see the above videos for reference.

General Authenticator Requirements

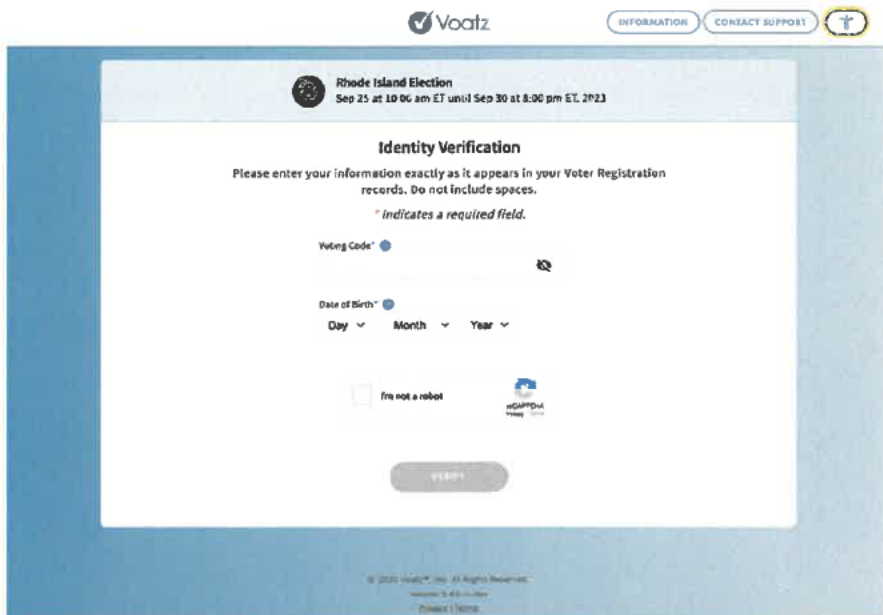
Verify that anti-automation controls are effective at mitigating breached credential testing, brute force, and account lockout attacks. Such controls include blocking the most common breached passwords, soft lockouts, rate limiting, CAPTCHA, ever increasing delays between attempts, IP address restrictions, or

risk-based restrictions such as location, first login on a device, recent attempts to unlock the account, or similar. Verify that no more than 100 failed attempts per hour is possible on a single account.

1. The following is from a voter authentication module.

```
val throttleInterval = if(attempt == 0){
  30 + Cryptography.issueRandomInt(1, 10)
}else{
  val min = (attempt)*10
  val max = (attempt+1)*10
  latestInterval + Cryptography.issueRandomInt(min, max)
```

2. The following image shows a CAPTCHA element on a test deployment of VWA; this element is standard on all registration and verification pages.



Comments: VMA is throttled at each failed login attempt, initially at least 30 seconds, with successive login attempts being throttled by at least 40 seconds, 60 seconds, 90 seconds, and so on. There can be no more than 12 failed attempts per hour.

Verify that the use of weak authenticators (such as SMS and email) is limited to secondary verification and transaction approval and not as a replacement for more secure authentication methods. Verify that stronger methods are offered before weak methods, users are aware of the risks, or that proper measures are in place to limit the risks of account compromise.

1. Onboarding and Identity Verification with Voatz
<https://vimeo.com/533255584/1baa3910a6>
2. Blockchain Technology Pilot Using Mobile Voting, City of Chandler, AZ: See time 0:53.
<https://www.chandleraz.gov/government/elections-and-voting/blockchain-mobile-voting>

Comments: Biometric verification is included as a standard feature on the Voatz platform and is offered as the first choice before the users are permitted to change to a weaker method. Additionally, identity verification and liveness checks along with digital signature verification are used as well.

Verify that secure notifications are sent to users after updates to authentication details, such as credential resets, email or address changes, logging in from unknown or risky locations. The use of push notifications - rather than SMS or email - is preferred, but in the absence of push notifications, SMS or email is acceptable as long as no sensitive information is disclosed in the notification.

1. The following is from the onboarding module.

```
def notifyIfEmailOrMobileDifferentFromSignup(mobileNumber, emailAddress,...)
val preRegEmail = customerPreregSnap.emailAddress.trim
val reqEmail = emailAddress.trim
val emailMatch = preRegEmail.equalsIgnoreCase(reqEmail)

RegistrationSettingsAsync.getRegistrationSettings() map{registrationSettings =>
if(!emailMatch){
MessagingAsync.sendReregisterWithDiffEmailOrMobileNotification(preRegEmail, reqEmail, ...
```

2. The following is from the onboarding module.

```
val preRegMobile = customerPreregSnap.mobileNumber
val mobileMatch = preRegMobile.equalsIgnoreCase(mobileNumber)
RegistrationSettingsAsync.getRegistrationSettings() map{registrationSettings =>
if(!mobileMatch){
MessagingAsync.sendReregisterWithDiffEmailOrMobileNotification(preRegEmail,
mobileNumber,...
```

Comments: The user is notified of any updates to authentication details via well-defined notifications.

Authenticator Lifecycle Requirements

Verify system generated initial passwords or activation codes SHOULD be securely randomly generated, SHOULD be at least 6 characters long, and MAY contain letters and numbers, and expire after a short period of time. These initial secrets must not be permitted to become the long term password.

1. The following is from a voter onboarding module.

```
val totp = Totp(Cryptography.issueRandomKey(256).getBytes, 6, SHA256, 30).generate()
```

2. Onboarding and Identity Verification, see example at 0:33:
<https://vimeo.com/533255584/1baa3910a6>

Comments: Two 6-digit activation codes are sent: one is a push notification sent to the mobile device and another to the supplied email address. Both codes are only valid for a small interval of time beyond which new codes must be requested. Neither code is long enough to be the long-term password.

Credential Recovery Requirements

Verify that a system generated initial activation or recovery secret is not sent in clear text to the user.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application: Encryption shall be implemented in:
 - Storage on disk, particularly secrets and user passwords.

Comments: The initial activation is sent via push notification, which is transmitted over https.

Verify password hints or knowledge-based authentication (so-called "secret questions") are not present.

1. Onboarding and Identity Verification with Voatz.
<https://vimeo.com/533255584/1baa3910a6>
2. Blockchain Technology Pilot Using Mobile Voting, City of Chandler, AZ.
<https://www.chandleraz.gov/government/elections-and-voting/blockchain-mobile-voting>

Comments: The examples above show that no such hints are present in the app.

Verify password credential recovery does not reveal the current password in any way.

Voatz applications require the user to go through a well defined reset process involving a set of encrypted APIs and two-factor authentication that never reveal the current password of the user.

Verify shared or default accounts are not present (e.g. "root", "admin", or "sa").

The Voatz Administration Portal employs role based access control which clearly defines and restricts the scope of information that a user has access to. Such checks are enforced on each API access and it is impossible to elevate privileges to an admin. Voatz aims for a layered approach to manage access to corporate and elections assets which includes:

Not permitting the use of shared account credentials.

Verify that if an authentication factor is changed or replaced, that the user is notified of this event.

1. One notification is sent to the user using a primary factor (e.g. email or mobile number) along with information for remediation in case of malicious activity detection.

Two-Factor Authentication Disabled

The Two Factor Authentication for your account has been deactivated.

We recommend that you add Two-Factor Authentication for additional security. If you wish to reactivate TFA, please proceed to TFA settings in your Accounts page.

[TFA settings](#)

2. A secondary notification is delivered using an alternate channel (e.g. push notification or backup email/mobile number).

Verify forgotten password, and other recovery paths use a secure recovery mechanism, such as time-based OTP (TOTP) or other soft token, mobile push, or another offline recovery mechanism. (C6)

The Voatz Administration Portal uses a two step process with two-factor authentication involving TOTP.

Out of Band Verifier Requirements

Verify that clear text out of band (NIST "restricted") authenticators, such as SMS or PSTN, are not offered by default, and stronger alternatives such as push notifications are offered first.

Voatz applications request the user for Push Notification permissions as part of the onboarding process.

Comments: Push notifications are offered as a first preference by default.

Verify that the out of band verifier expires out of band authentication requests, codes, or tokens after 10 minutes.

1. The following is from the voter onboarding module.

```
def checkOtpTimestamp(otpTimestamp: String, nowTimestamp: String, onBoardingLimits:
OnboardingLimitsSnapshot): Boolean = {
    val now = OffsetDateTime.parse(nowTimestamp)
    val otpts = OffsetDateTime.parse(otpTimestamp)
    val differenceInSeconds = ChronoUnit.SECONDS.between(otpts, now)
    differenceInSeconds <= onBoardingLimits.timeout
```

2. Voter onboarding limits are defined as follows.

```
(timeout: Int, consecutiveRequestInterval: Int, consecutiveRequests: Int,
consecutiveRequestFromSameIpInterval: Int,
consecutiveRequestsFromSameIp: Int)
```

Here, timeout is configured as 600 (10 minutes * 60 seconds).

Comments: The timeout for out of band auth requests, codes or tokens can be set as desired by the jurisdiction.

Verify that the out of band verifier authentication requests, codes, or tokens are only usable once, and only for the original authentication request.

1. The following is from a voter onboarding module.

```
else if(otpAlreadyVerified){
log.error(s"otp $otp already verified for mobile $mobileNumber")
val err = "An error occurred. Please request new codes."
```

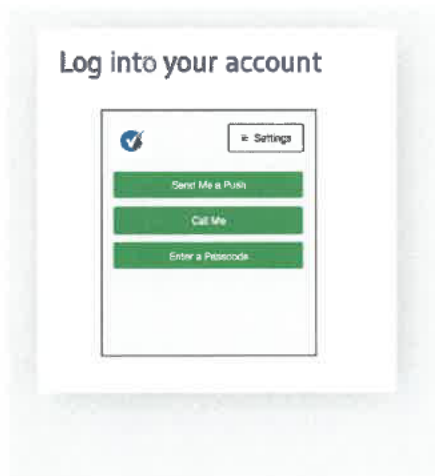
2. The following is from a voter onboarding module.

```
else if(!sideChannelSnapOpt.isDefined || !sideChannelSnapOpt.get.otpTimestamp.isDefined) {
log.error(s"No verified side channel otp record found for preRegisterId $preRegId")
```

Comments: The server checks if the one-time push notification had been already used or if it came from a different mobile phone. In either case, the verification request is rejected.

Verify that the out of band authenticator and verifier communicates over a secure independent channel.

1. Push notifications are used as one of the secure independent communication channels.
2. Authentication codes in the Voatz Admin Portal web app or code delivery via SMS or voice calls are used as additional independent channels.



Comments: Voatz integrates seamlessly with third party services such as the Duo App, Google Authenticator, Authy App, Microsoft Authenticator, etc.

Single or Multi-factor One Time Verifier Requirements

Verify that time-based OTPs have a defined lifetime before expiring.

1. The following is from a voter onboarding module.

```
if(checkOtpTimestamp(snapshot.timestamp, timestamp, onBoardingLimits)){
    ...
}
else{
    val error = "Otp timeout was exceeded"
```

2. The following is from an API routing module.

```
setCookie(HttpCookie(SessionCookie, session.sessionCookie, httpOnly = true, secure =
config.HttpConfig.UseHttps, expires =
```

Comments: Time-based OTPs have a well-defined timeout setting to eliminate any possibility of expired OTPs from being used.

Fundamental Session Management Requirements

Verify the application never reveals session tokens in URL parameters.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application: Encryption shall be implemented in:
 - All web applications via the most secure version of SSL/TLS.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application: Encryption shall be implemented in:
 - App/client communication with back end infrastructure.

Comments: Session tokens are transmitted via well defined HTTP headers, not as URL parameters.

Session Binding Requirements

Verify the application generates a new session token on user authentication.

1. The following is from a voter authentication module.

```
def createSession(customerId: Int): HttpApiSession = {
    val userHash = Cryptography.hash64(customerId.toString())
    val sessionCookie = Cryptography.hash64(Cryptography.issueRandomKey(64) + userHash)
    val csrfToken = Cryptography.issueRandomKey(64)
```

2. The following is from an API routing module.

```
respondWithHeader(RawHeader(CsrfToken, session.csrfToken)) {
```

```
setCookie(HttpCookie(SessionCookie, session.sessionCookie, httpOnly = true, secure =
config.HttpConfig.UseHttps,
expires =
```

Comments: A new session token is generated with each authentication call to the Voatz API.

Verify that session tokens possess at least 64 bits of entropy.

1. The following is from an authentication module.

```
csrfToken = Cryptography.issueRandomKey(64)
```

2. Policy A.10: Acceptable Encryption Policy, Section 3.6: Key Management, Key generation must be seeded from an industry standard random number generator.

Comments: Session tokens are derived from a cryptographically secure random number generator and have 256 bits of entropy.

Verify the application only stores session tokens in the browser using secure methods such as appropriately secured cookies (see section 3.4) or HTML 5 session storage.

1. The following is from an API router module.

```
validate(sessionCookie, csrfToken)
```

2. The following is from an API router module.

```
HttpApiSession(sessionCookie: String, csrfToken: String, customerId: Int, lastUse: Long)
```

Comments: Further, on a mobile device, the session token is stored in restricted device storage.

Session Logout and Timeout Requirements

Verify that logout and expiration invalidate the session token, such that the back button or a downstream relying party does not resume an authenticated session, including across relying parties.

Logout APIs invalidate the session token making it impossible to resume an authenticated session. For example, the following code is executed on the Voatz web app. when a session expires:

```
killSession() {
  this.mte.needHandshake = true;
  this.mte.clientDecoder?.uninstantiate;
  this.mte.clientDecoder?.destruct;
  this.mte.clientEncoder?.uninstantiate;
  this.mte.clientEncoder?.destruct;
```



```

    this.storer.sessionDecoder?.uninstantiate;
    this.storer.sessionDecoder?.destruct;
    this.storer.sessionEncoder?.uninstantiate;
    this.storer.sessionEncoder?.destruct;
    this.storer.removeAll()
    this.cookieService.remove('__HOST-SESSION_EXPIRY');
    this.currentSessionSubject.next(null)
  }

```

Comments: Upon logging out, the session is deleted from the session store, so it cannot be resumed.

Cookie Based Session Management

1. The following is from a voter authentication module.

```
setCookie(HttpCookie(SessionCookie, session.sessionCookie, httpOnly = true, secure = true
```

2. The following is from the Voatz Web App backend. Cookie-based session data is sent in the response to the frontend.

```

export function addSessionToResponse(response: APIGatewayProxyStructuredResultV2, session:
Session) {
  response.cookies = [cookie.serialize(sessionIdCookieName, session.id, {
    expires: session.expiresAt,
    httpOnly: true,
    path: '/',
    secure: true,
  }),
    cookie.serialize(sessionExpiryCookieName, session.expiresAt.toISOString(), {
    expires: session.expiresAt,
    path: '/',
    secure: true,
  }),
    cookie.serialize(csrfTokenCookieName, session.csrfToken, {
    expires: session.expiresAt,
    path: '/',
    secure: true,
  })];
  return response;
}

```

Verify that cookie-based session tokens have the 'Secure' attribute set.

1. The following is from a voter authentication module.

```
setCookie(HttpCookie(SessionCookie, session.sessionCookie, httpOnly = true, secure = true
```

2. The following is from an API configuration module.

```
val UseHttps = httpConfig.getBoolean("use-https")
```

Comments: All communications with Voatz servers are over https, so session tokens have the Secure attribute set.

Verify that cookie-based session tokens have the 'HttpOnly' attribute set.

1. The following is from a voter authentication module.

```
setCookie(HttpCookie(SessionCookie, session.sessionCookie, httpOnly = true,
```

2. The following shows the CSRF token from an organization authentication module.

```
HttpCookie(CsrfToken, payload.csrfToken, httpOnly = true, secure = true
```

Verify that cookie-based session tokens utilize the 'SameSite' attribute to limit exposure to cross-site request forgery attacks.

1. CSRF tokens are used to prevent such attacks. This code is from a voter authentication module.

```
respondWithHeader(RawHeader(CsrfToken, session.csrfToken))
```

2. General cookies sent to any client using the API router include a CSRF token. The following code is from an API session authentication module.

```
session = HttpApiSession(sessionCookie, csrfToken, customerId, System.currentTimeMillis)
```

Verify that cookie-based session tokens use "__Host-" prefix (see references) to provide session cookie confidentiality.

1. The following code is from the API routing module.

```
setCookie(HttpCookie(SessionCookie, session.sessionCookie, httpOnly = true, secure = true, expires = Some(DateTime.now.+(validInterval)), domain = domainVal, path = pathVal)) {
```

2. The following code is from another API routing module.

```
setCookie(HttpCookie(SessionCookie, payload.sessionCookie, httpOnly = true, secure = config.HttpConfig.UseHttps, expires = Some(DateTime.now.+(cookieExpiration)), domain = domainVal, path = pathVal),
```

Comments: The routing module sets the domain and path, along with the “Secure” prefix set to “true,” which offers the security offered by the Host prefix. Like the “Host” prefix, cookies with the “Secure” prefix can only be set from a secure connection.

Verify that if the application is published under a domain name with other applications that set or use session cookies that might override or disclose the session cookies, set the path attribute in cookie-based session tokens using the most precise path possible.

1. The following is from a voter API router module.

```
setCookie(HttpCookie(SessionCookie, session.sessionCookie, httpOnly = true, secure = true,
  expires = Some(DateTime.now.+(validInterval)), domain = domainVal, path = Some(pathVal)))
```

2. The following is from an API routing module.

```
def allRoutes: Route = encodeResponse {
  path(
```

Comments: The API router controls the path precision by defining the paths that are allowed for an HTTP request to make.

General Access Control Design

Verify that the application enforces access control rules on a trusted service layer, especially if client-side access control is present and could be bypassed.

The Voatz Administration Portal employs role based access control which clearly defines and restricts the scope of information that a user has access to. Such checks are enforced on each API access and it is impossible to bypass them.

Comments: Any operation which requires API calls to a Voatz server requires certain authentication and authorization, which are tightly controlled.

Verify that all user and data attributes and policy information used by access controls cannot be manipulated by end users unless specifically authorized.

The Voatz Administration Portal employs role based access control which clearly defines and restricts the scope of information that a user has access to. Such checks are enforced on each API access and it is impossible to bypass them. Policy information used by such access controls cannot be modified by end users as such information is never made available to client side APIs.

Verify that the principle of least privilege exists - users should only be able to access functions, data files, URLs, controllers, services, and other resources, for which they possess specific authorization. This implies protection against spoofing and elevation of privilege.

The Voatz Administration Portal employs fine grained role based access control which clearly defines and restricts the scope of information that a user has access to. Such checks are enforced on each API access and it is impossible to elevate privileges.

- The default approach taken is to assume that access to any service or resource is not required, rather than to assume that it is.
- Need to Know – access is only granted to the information required to perform a role, and no more. The default setting of the access control system is “deny-all”;

1. Policy AC9: Server Security, Section 4.2 General Configuration and Backup Guidelines:
Always use the least privilege to perform a function. Do not use root/admin accounts when a non-privileged account is sufficient.

Verify that the principle of deny by default exists whereby new users/roles start with minimal or no permissions and users/roles do not receive access to new features until access is explicitly assigned.

The Voatz Administration Portal employs fine grained, role based access control which clearly defines and restricts the scope of information that a user has access to. A new user is only assigned basic permissions which grant access to a minimal set of APIs. Once the user role is finalized, additional permissions are explicitly granted based on the user’s role.

Verify that access controls fail securely including when an exception occurs.

Voatz applications employ fine grained, role based access control. When an access control restriction fails or is violated, standard HTTP error codes such as 403 Forbidden are returned along with a generic error message without revealing any confidential information.

The following code checks whether a Voatz Administration portal user has permissions to enroll for access CVR (cast vote record) and returns a 403 error code if the user doesn’t have such a permission:

```
val validFut = OrganizationValidationsAsync.isEnrollForCVRRequestValid(request, sessionCookie)
validFut onComplete{
  case Success((valid, invalidReason, countyOpt, enrollingUserIdOpt)) => {
    if(!valid && invalidReason.matches("(?i)^Forbidden.*")){
      log.error(s"EnrollForCVR req failed: $invalidReason")
      theSender ! ForbiddenError(invalidReason)
    }
  }
}
```

The following code, corresponding to the Voatz Mobile App. Logout API, checks whether the session cookie supplied by a mobile app. user is valid, whether the request originated from a device belonging to the user, and whether the user requesting the logout and the user linked to the device requesting the logout are the same. If any of these pre-conditions are violated, a 403 Forbidden error code is returned:

```
val customerIdFromDevId = snapByDevIdOpt.get
```

```

        SessionAuthenticatorAsync.validateSessionAgainstUser(sessionCookieOpt,
customerIdFromDevId) onComplete{
        case Success(sessionValid) => {
            if(!sessionValid){
                theSender ! ForbiddenError("Session cookie provided does not authorize you to
perform this action")
            }else{
                if (customerIdFromDevId != apiRequest.customerId) {
                    theSender ! ForbiddenError("Supplied deviceId does not belong to requested
user")
                } else {

```

Operation Level Access Control

Verify that the sensitive data and APIs are protected against Insecure Direct Object Reference (IDOR) attacks targeting creation, reading, updating and deletion of records, such as creating or updating someone else's record, viewing everyone's records, or deleting all records.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application
Encryption shall be implemented in:
 - During sensitive data transmission over public networks
2. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application:
Encryption shall be implemented in:
 - All web applications via the most secure version of SSL/TLS.

Comments: All data in flight is encrypted via AES GCM with non repeating nonce. Additionally, each API is secured by authorization checks of the following form.

```

val snapByDevIdOptFut = UserModule.getByDeviceId(apiRequest.deviceId, sessionCookieOpt)
snapByDevIdOptFut onComplete {
    case Success(snapByDevIdOpt) => {
        if (!snapByDevIdOpt.isDefined) {
            val err = "user not found"
        }else {
            val userIdFromDevId = snapByDevIdOpt.get
            SessionAuthenticatorAsync.validateSessionAgainstUser(sessionCookieOpt, userIdFromDevId)
onComplete{
            case Success(sessionValid) => {
                if(!sessionValid){
                    theSender ! ForbiddenError("Session cookie provided does not authorize you to perform this
action")
                }else{
                    if (userIdFromDevId != request.userId) {
                        theSender ! ForbiddenError("Supplied deviceId does not belong to requested user")
                    } else {

```


In this code, the user attached to the device sending the request is validated against the user holding the current session, which in turn is matched against the user making the request, thus eliminating the possibility of IDOR (Insecure Direct Object Reference) attacks.

Verify that the application or framework enforces a strong anti-CSRF mechanism to protect authenticated functionality, and effective anti-automation or anti-CSRF protects unauthenticated functionality.

1. CSRF tokens are used to prevent such unauthenticated functionality. The following is from a voter authentication module.

```
respondWithHeader(RawHeader(CsrfToken, session.csrfToken))
```

2. General cookies sent to any client using the API router include a CSRF token. The following is from an API session authentication module.

```
session = HttpApiSession(sessionCookie, csrfToken, customerId, System.currentTimeMillis)
```

Comments: CSRF tokens disallow all unauthenticated functionality.

Other Access Control Considerations

Verify administrative interfaces use appropriate multi-factor authentication to prevent unauthorized use.

1. Policy AC.5: Password Policy, Section 5.1: General:
Whenever possible, use 2-Factor Authentication (2FA) or 2-Step Authentication.
2. Policy A.9: Access Control Policy, Section 6.4 - Use of Privileged Utility Programs:
 - Where possible and appropriate privileged utility programs must incorporate multi-factor authentication (MFA).

Comments: The election administration portal ensures that 2FA via 3rd party 2FA apps such as DUO is enforced on every authentication attempt.

Verify that directory browsing is disabled unless deliberately desired. Additionally, applications should not allow discovery or disclosure of files or directory metadata such as Thumbs.db, .DS_Store, .git or .svn folders.

1. Policy A.9: Access Control Policy, Section 3.0:
This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.
 - Defense in Depth – security shall not depend upon any single control but be the sum of a number of complementary controls;
 - Least Privilege – the default approach taken shall be to assume that access is not required, rather than to assume that it is. In particular, all access to election

infrastructure and election data shall be limited to those individuals that have a business or legal need-to-know;

2. Policy A.9: Access Control Policy, Section 3.0:

This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.

- Need to Know – access is only granted to the information required to perform a role, and no more. The default setting of the access control system has to be “deny-all”;
- Need to Use – users will only be able to access physical and logical facilities required for their role.

Comments: Directory browsing is not allowed and applications do not allow unauthorized users to view files and directory metadata.

Input Validation Requirements

Verify that the application has defenses against HTTP parameter pollution attacks, particularly if the application framework makes no distinction about the source of request parameters (GET, POST, cookies, headers, or environment variables).

1. N/A

2. N/A

Comments: N/A: Voatz applications and servers never concatenate unvalidated user input into URLs, thus eliminating the possibility of injecting query string delimiters to perform a malicious task via an HPP (HTTP Parameter Pollution) attack. No API endpoints make use of unvalidated query string parameters and all APIs use HTTP POST with strongly typed request validation.

Verify that frameworks protect against mass parameter assignment attacks, or that the application has countermeasures to protect against unsafe parameter assignment, such as marking fields private or similar.

1. N/A

2. N/A

Comments: N/A: Voatz applications and servers use explicit parameter assignment with strong type checking, eliminating the possibility of any mass parameter assignment attacks.

Verify that all input (HTML form fields, REST requests, URL parameters, HTTP headers, cookies, batch files, RSS feeds, etc) is validated using positive validation (allow lists).

1. The following is from an API routing module.

```
def allRoutes: Route = encodeResponse {
  path{
```

2. The following is from an API routing module.

```
private def heartbeat() = post {
```

Comments: The API router has explicit validation of requests as above, only accepting post requests and only to white-listed APIs.

Verify that structured data is strongly typed and validated against a defined schema including allowed characters, length and pattern (e.g. credit card numbers or telephone, or validating that two related fields are reasonable, such as checking that suburb and zip/postcode match).

1. Server code is written in Scala, which is a strongly typed language.
2. All data is checked for allowed characters, length, etc. as appropriate. Data from addresses taken from official ID is checked against USPS data for validity. One example of input validation is checking the date of birth. The following example is from the Voatz Web App backend.

```
const dateOfBirthRegex = /^d\d\d\d\d\d\d\d$/
export const dateOfBirthSchema = z.string().regex(dateOfBirthRegex, 'INVALID_DOB')
.refine(d => Date.parse(d) < Date.now(), { message: 'Date of birth is in the future.' });
```

Verify that URL redirects and forwards only allow destinations which appear on an allow list, or show a warning when redirecting to potentially untrusted content.

1. The following is from an API routing module.

```
def allRoutes: Route = encodeResponse {
  path{
```

2. The following is from an API routing module.

```
handleNotFound {
  log.error(s"The requested resource could not be found.")
```

Comments: The API router has explicit validation of requests as above, only accepting post requests and only to whitelisted APIs. Any request to a non-whitelisted resource is denied.

Sanitization and Sandboxing Requirements

Verify that all untrusted HTML input from WYSIWYG editors or similar is properly sanitized with an HTML sanitizer library or framework feature.

DOMPurify on the web client side and Java HTML sanitizer on the server side are used. Additionally, any untrusted user input is sanitized before being passed further along the validation pipeline.

The following code from the Voatz Web App. backend normalizes user supplied email addresses:

```
function sanitizeEmail(email: string): string {
    return email.replace(/[@_]/g, "").trim().toUpperCase();
}
```

The following code from the Voatz Web App. backend sanitizes input and ensures that it confirms to a well defined type schema:

```
import * as z from 'zod';

function parseBallotPackage(b: any): BallotPackage {
  const signatureRequired = process.env.REQUIRE_AFFIDAVIT === 'true';
  return z.object({
    ballot: ballotSchema,
    signature: signatureRequired ? signatureSchema : signatureSchema.optional(),
    locale: z.string().optional(),
  }).parse(b);
}
```

Verify that unstructured data is sanitized to enforce safety measures such as allowed characters and length.

The following code from the user write-in validation module sanitizes unstructured user input to ensure that it contains allowed characters and is limited to a certain length:

```
val pattern = Pattern.compile("[\\{\\}\\|\\(\\)\\[\\]\\+\\|\\*\\|\\?\\|\\^\\|\\$\\|\\\\\\\\\\|\\]|")
    val voteWriteInChoices = voteChoices.filter(vc => eventWriteInChoices.find(ewc => ewc.choiceId.equals(vc.choiceId)).isDefined)
    if(voteWriteInChoices.find(vwc => vwc.description.length > 255).isDefined){
        (false, "A write in choice with length greater than 255 characters found")
    }else if(voteWriteInChoices.find(vwc => pattern.matcher(vwc.description).find()).isDefined){
        (false, "A write in choice with one or more special characters found")
    }else{
        (true, "")
    }
}
```

Verify that the application sanitizes user input before passing to mail systems to protect against SMTP or IMAP injection.

The following code from the user validation module uses strict validation to protect against injection attacks:

```
val (emailValid, emailInvalidErr) = try {
```

```
JMail.strictValidator().disallowQuotedIdentifiers().disallowReservedDomains().enforceValid(emailAddress)
    (true, "")
} catch {
    case e: Exception => {
        val msg = s"Email address $emailAddress is invalid"
        log.error(msg + " " + e.getMessage)
        (false, msg)
    }
}
```

Verify that the application avoids the use of `eval()` or other dynamic code execution features. Where there is no alternative, any user input being included must be sanitized or sandboxed before being executed.

Voatz web applications do not employ `eval` or any other dynamic code execution features.

Verify that the application protects against template injection attacks by ensuring that any user input being included is sanitized or sandboxed.

DOMPurify on the web client side and Java HTML sanitizer on the server side are used.

Verify that the application protects against SSRF attacks, by validating or sanitizing untrusted data or HTTP file metadata, such as filenames and URL input fields, use whitelisting of protocols, domains, paths and ports.

Voatz web applications wrap untrusted data with multiple validators and verifiers before passing such data further. The following code from web app backend performs several validations:

```
startElectorSession: ApiHandler = async (event: APIGatewayProxyEventV2) => {
    return this.electionTimeCheck.wrap(
        this.requestAuthenticator.wrapOptional((e, session?: Session) =>
            this.mteDecoder.wrap(
                this.jsonDecoder.wrap(
                    this.captchaVerifier.wrap(
                        this.electorFormValidator.wrap(
```

Verify that the application sanitizes, disables, or sandboxes user-supplied SVG scriptable content, especially as they relate to XSS resulting from inline scripts, and `foreignObject`.

Voatz web applications do not permit user supplied scriptable content.

Verify that the application sanitizes, disables, or sandboxes user-supplied scriptable or expression template language content, such as Markdown, CSS or XSL stylesheets, BBCode, or similar.

Voatz web applications do not permit user supplied scriptable content.

Output Encoding and Injection Prevention Requirements

Verify that output encoding is relevant for the interpreter and context required. For example, use encoders specifically for HTML values, HTML attributes, JavaScript, URL Parameters, HTTP headers, SMTP, and others as the context requires, especially from untrusted inputs (e.g. names with Unicode or apostrophes, such as ねこ or O'Hara).

1. The following is from an API routing module.

```
import org.owasp.encoder.Encode
ctx.complete(HttpResponse(StatusCodes.OK, Encode.forHtml(e.getMessage()), getAllHeaders))
```

2. The following is from an API routing module.

```
import org.owasp.encoder.Encode
ctx.complete(HttpResponse(StatusCodes.InternalServerError, Encode.forHtml(e.getMessage()),
getAllHeaders))
complete(HttpResponse(BadRequest, Encode.forHtml(message), getAllHeaders))
```

Comments: Voatz servers use relevant output encoders from OWASP.

Verify that output encoding preserves the user's chosen character set and locale, such that any Unicode character point is valid and safely handled.

1. The following is from the validations module.

```
def convertStringFromHexToUTF8(str: String) = {
  val matcher = Pattern.compile("\\\\x[a-z0-9]{2}\\\\x[a-z0-9]{2}").matcher(str)
  val sb = new StringBuilder
  try {
    while (matcher.find) {
      sb.append(matcher.group(0))
    }
  } catch {
    case e: Exception => log.error(s"error matching $str with pattern for unicode hex strings: " +
e.getMessage)
  }
  val s = sb.toString
  str.replace(s, new String(getHexBytes(s), StandardCharsets.UTF_8))
}

import org.apache.commons.codec.binary.Hex

def getHexBytes(s: String) = {
```

```

try {
  Hex.decodeHex(s.replace("\\x", ""))
}catch{
  case e: Exception => log.error(s"error converting $s from hex to byte array" +
e.getMessage());Array[Byte]()
}
}

```

2. The following is from the validations module.

```

import org.apache.commons.lang3.StringUtils
StringUtils.stripAccents(lastNameFromUser).replace("`", "")
.replace("'", "").replace(" ", "").replace(".", "")

```

Comments: Voatz applications and servers preserve the user's chosen character set and locale by using standard Apache Commons libraries.

Verify that context-aware, preferably automated - or at worst, manual - output escaping protects against reflected, stored, and DOM based XSS.

1. The following response is from the user verification API.

```

< Content-Type: application/json; charset=UTF-8
< Content-Length: 3888
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block

```

2. The following response is from the user login API.

```

< Csrft-Token: nvhSh0hGnM+ufTna+4zoc0u+yeTmQ3zxX0LfWgnTpbk=
< Content-Type: text/plain; charset=UTF-8
< Content-Length: 2
< Set-Cookie: WS=MZdQe4iMq/6HJJjkDvbBgjo655tM1JNBBWBQYj8Xkel=; Expires..

```

Comments: Voatz applications and servers filter and sanitize input on arrival, properly encode data on output to prevent it from being interpreted as active content, and use appropriate response headers such as Content-Type to ensure that responses are interpreted as intended.

Verify that data selection or database queries (e.g. SQL, HQL, ORM, NoSQL) use parameterized queries, ORMs, entity frameworks, or are otherwise protected from database injection attacks.

Voatz web applications employ AWS managed web acls to prevent SQL injection attacks using the following rule set:

```
aws-waf:managed:aws:sql-database:SQLi_Body
```

```
awsaf:managed:aws:sql-database:SQLi_QueryArguments
awsaf:managed:aws:sql-database:SQLi_Cookie
awsaf:managed:aws:sql-database:SQLiExtendedPatterns_QueryArguments
awsaf:managed:aws:sql-database:SQLi_URIPath
```

Comments: Voatz applications and servers never create parameterized queries with unvalidated user input, thus eliminating the possibility of any database injection attacks.

Verify that where parameterized or safer mechanisms are not present, context-specific output encoding is used to protect against injection attacks, such as the use of SQL escaping to protect against SQL injection.

Voatz applications employ SQL escaping via strongly typed string interpolation and SQL syntax where any value supplied as part of a query is not treated as a binding parameter but appended as part of validated SQL

Verify that the application protects against JavaScript or JSON injection attacks, including for eval attacks, remote JavaScript includes, CSP bypasses, DOM XSS, and JavaScript expression evaluation.

Voatz web applications use interpolation in Angular templates to escape and sanitize data. The following code from the Voatz Web app. uses `{{}}` to encapsulate untrusted data:

```
<label class="vtz-form__radio-container vtz-text"
  [ngClass]="{'vtz-form__radio-container--selected': checkedIndex === i ,
'focused': focusStates[i]} "
  *ngFor="let ballot of ballots; index as i">
  <span class="tz-text vtz-text--ballot " style="white-space: pre-line "
    i18n>{{ballot.voteEventData.description + '\n' +
    ballot.voteEventData.description1 + '\n' +
    ballot.voteEventData.description2
  }}
</label>
```

Additionally, any unsafe attributes are removed from safe elements by binding to the `innerHTML` property which removes any unsafe tags or specific attributes/child elements:

```
<div *ngIf="description1HasHtml" [innerHTML]="description1"></div>
```

Verify that the application protects against LDAP Injection vulnerabilities, or that specific security controls to prevent LDAP Injection have been implemented.

Voatz applications are strictly sandboxed and have no linkage to any internal or external LDAP system.

Verify that the application protects against OS command injection and that operating system calls use parameterized OS queries or use contextual command line output encoding.

Voatz web and mobile applications are strictly sandboxed and do not invoke any OS commands, calls or queries directly and never interact with the command line. All OS calls are delegated to vetted libraries that are regularly audited using for e.g. npm audit and Guardsquare App Sweep.

Verify that the application protects against Local File Inclusion (LFI) or Remote File Inclusion (RFI) attacks.

Voatz mobile and web applications never upload or read files directly to/from Voatz servers. Live document images and selfies are captured via the device camera and uploaded with strict file naming conventions to secure file storage, with well-defined IAM roles and bucket policies, where micro services process the files further thus eliminating the possibility of disclosure, creation, updating or removal of local files.

Verify that the application protects against XPath injection or XML injection attacks.

Voatz applications employ JSON based REST APIs exclusively and do not use XPath, XML parsing or legacy SOAP APIs in front/backend code or when invoking third party services via well defined integrations. If a request does not contain valid JSON, it is rejected at the routing layer and not passed further along the validation pipeline.

Deserialization Prevention Requirements

Verify that serialized objects use integrity checks or are encrypted to prevent hostile object creation or data tampering.

Voatz applications use AES-GCM encryption with non repeating nonce to encrypt all client/server communication at the application layer in addition to employing TLS/SSL at the transport layer. All serialized objects are subject to rigorous type safety and schema validation checks. If any such checks are violated, the request is rejected.

Verify that the application correctly restricts XML parsers to only use the most restrictive configuration possible and to ensure that unsafe features such as resolving external entities are disabled to prevent XML eXternal Entity (XXE) attacks.

Voatz applications employ JSON based REST APIs exclusively and do not use XPath, XML parsing or legacy SOAP APIs in front/backend code or when invoking third party services via well defined integrations. If a request does not contain valid JSON, it is rejected at the routing layer and not passed further along the validation pipeline.

Verify that desterialization of untrusted data is avoided or protected in both custom code and third-party libraries (such as JSON, XML and YAML parsers).

Voatz applications employ strongly typed schema validation in client side code and backend APIs. Such validations are performed at the routing layer before any application logic is invoked thus preventing any validation failures from reaching the application layer.

Verify that when parsing JSON in browsers or JavaScript-based backends, JSON.parse is used to parse the JSON document. Do not use eval() to parse JSON.

Voatz Web applications use `Json.parse` to decode requests. The following code from the Voatz web app backend decodes and validates input:

```
const api = sessionManager.then( session_manager => new VoatzWebBackend(
  new JsonRequestDecoder(JSON.parse),
  new JsonValidator(parseFrontendPubKeys),
  new MteDecoderWrapper(session_manager.mte),
  new JsonValidator(parseElectorForm),
  new JsonValidator(parseVerifyPrecinctUserForm),
  new JsonValidator(parseBallotPackage),
```

Algorithms

Verify that all cryptographic modules fail securely, and errors are handled in a way that does not enable Padding Oracle attacks.

Voatz applications use ECDH key exchange and AES GCM with non repeating nonce to protect data in transit. Modes of operation such as ECB or CBC that are vulnerable to padding oracle attacks are not employed. When a cryptographic module fails to decrypt a response or encrypt a request, the standard behavior is to return an HTTP error code and force a new key exchange. Thus it is impossible to discover any erroneous intermediate state and replay captured cipher text to discover plain text.

Log Content Requirements

Verify that the application does not log credentials or payment details. Session tokens should only be stored in logs in an irreversible, hashed form.

Voatz backend code employs AWS CloudWatch to log errors that arise as a result of exceptional/unexpected situations. These error logs do not include any user credentials or session tokens and are primarily used to log exception stack traces. The following code logs an error to CloudWatch when an expected resource cannot be accessed:

```
let featureSetDB: FeatureSetResponse | undefined = await
this.db.getOrgFeatureSet(localeCode);
if(featureSetDB){
  return featureSetDB;
} else {
  console.error("Failed to get org feature set from db")
  const response = await this.post(getOrgFeatureSetPath, {locale: localeCode})
  .then(catchErrorResponse('Get Org Feature Set'));
```

Front end code only logs in rare situations involving exceptions that are unexpected/uncaught and cannot be handled by well defined application error modals. The following code only logs an error that does not match expected error paths that invoke error modals :


```

if ((error.error_type !== undefined && error.error_type === 'INVALID_IDENTITY') ||
    (error.error !== undefined && error.error.error_type === 'INVALID_IDENTITY')) {
    this.errorText = this.invalidIdentityErrorText;
    this.errorTitle = $localize`Registration not found.`
    this.isErrorModalOpen.next(true);
    this.focusErrorDialog()
} else if ((error.error_type !== undefined && error.error_type === 'ALREADY_REGISTERED')
||
    (error.error !== undefined && error.error.error_type === 'ALREADY_REGISTERED')) {
    this.errorText = this.alreadyRegisteredErrorText;
    this.errorTitle = $localize`Already registered.`
    this.isErrorModalOpen.next(true);
    this.focusErrorDialog()
} else if ((error.error_type !== undefined && error.error_type === 'DECODER_STATE_LOST')
||
    (error.error !== undefined && error.error.error_type === 'DECODER_STATE_LOST')) {
    this.errorText = $localize`Your session timed out. Please try to submit again or refresh
page.`;
    this.errorTitle = $localize`Registration Not Successful.`
    this.isErrorModalOpen.next(true);
    this.focusErrorDialog()
} else {
    console.error(error);
    this.errorText = $localize`An unknown error occurred. Please contact us.`
    this.isErrorModalOpen.next(true);
    this.errorTitle = $localize`Unknown error.`
    this.focusErrorDialog()
}

```

Verify that the application does not log other sensitive data as defined under local privacy laws or relevant security policy.

Voatz applications use AWS CloudWatch log groups created in the geographical region where the application is hosted in accordance with local privacy laws. Such logs are only used to log exception stack traces to provide Voatz support personnel with additional information to resolve an unexpected error or exception.

Error Handling

Verify that a generic message is shown when an unexpected or security sensitive error occurs, potentially with a unique ID which support personnel can use to investigate.

1. The following example is from a voter onboarding module on the server.

```
if(threatDetected){
```

```
val threatName = threatNameOpt.get
val err = s"Voatz has detected an unexpected issue with your device. Please quit the app and try
again, or contact Voatz."
log.error(s"Threat $threatName detected on device $deviceIdFromReq")
```

2. Policy AC.9: Server Security Policy, Section 4.3 Monitoring

All security-related events on critical or sensitive systems must be logged and audit trails saved as follows:

- Logs are accessible only to appropriate applications and trusted users.

Comments: Generic error messages are shown to the user in order to stymie attempts by an attacker to deduce internal app and server logic.

Client-side Data Protection

Verify the application sets sufficient anti-caching headers so that sensitive data is not cached in modern browsers.

Voatz web applications employ the following by default:

```
<meta http-equiv="Cache-Control" content="no-store, no-cache, must-revalidate">
```

Verify that data stored in client side storage (such as HTML5 local storage, session storage, IndexedDB, regular cookies or Flash cookies) does not contain sensitive data or PII.

Voatz web applications do not store any PII in local storage. Any application metadata that is persisted in local storage is encrypted in flight and at rest, and cleared when the user's session ends.

Verify that authenticated data is cleared from client storage, such as the browser DOM, after the client or session is terminated.

Voatz web applications employ the following sequence of operations when a session is terminated, clearing, un-instantiating and destroying all data:

```
killSession() {
  this.mte.needHandshake = true;
  this.mte.clientDecoder?.uninstantiate;
  this.mte.clientDecoder?.destruct;
  this.mte.clientEncoder?.uninstantiate;
  this.mte.clientEncoder?.destruct;
  this.storer.sessionDecoder?.uninstantiate;
  this.storer.sessionDecoder?.destruct;
  this.storer.sessionEncoder?.uninstantiate;
  this.storer.sessionEncoder?.destruct;
  this.storer.removeAll()
```

```

    this.cookieService.remove('__HOST-SESSION_EXPIRY');
    this.currentSessionSubject.next(null)
  }

```

Sensitive Private Data

Verify that sensitive data is sent to the server in the HTTP message body or headers, and that query string parameters from any HTTP verb do not contain sensitive data.

Voatz mobile and web applications only use HTTP POST requests where all data is encrypted and sent as part of the message body.

Verify that users have a method to remove or export their data on demand.

1. The following is from the user management module.

```

ApiClientleanupUserRequest(emailAddress: String, org: Option[Boolean], basic: Option[Boolean])

```

2. The following is from the user management module.

```

if(request.basic.getOrElse(false) && request.org.getOrElse(false)){
  Signups.deleteAllForCustomer(customerId)
  OrgVerificationProfile.deleteAllForCustomer(customerId)
  DatastoreKeyMap.deleteAllForCustomer(customerId)
  ControlNumber.deleteAllForCustomer(customerId)
  deleteAllBasicIdoRecords(customerId)
  ...
else if(request.basic.getOrElse(false)){
  deleteAllBasicIdoRecords(customerId)
  deleteAllBasicIdvRecords(customerId)
}

```

Comments: Voatz provides users with well-defined options to remove their data on demand. However, access to such API/resources is strictly controlled to eliminate the possibility of misuse.

Verify that users are provided clear language regarding collection and use of supplied personal information and that users have provided opt-in consent for the use of that data before it is used in any way.

Comments: This information is provided to the users in the EULA and in the app permissions from the Google Play Store and the Apple App Store. Additionally, app. users are requested to confirm Voatz privacy policies as part of the onboarding process.

Verify that all sensitive data created and processed by the application has been identified, and ensure that a policy is in place on how to deal with sensitive data.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE-1: **Pass**, June 19, 2020.

2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE-1: Pass, June 19, 2020.

Comments: All sensitive data is encrypted both in transit and at rest. Once sensitive data is no longer needed, it is deleted.

Deployed Application Integrity Controls

Verify that if the application has a client or server auto-update feature, updates should be obtained over secure channels and digitally signed. The update code must validate the digital signature of the update before installing or executing the update.

Voatz mobile applications can only be updated via formally reviewed and approved releases on Apple App. store and Google Play store. Each such release is digitally signed via the Voatz app. signing key and validated by the respective app store before the user can install it. If the application binary is obtained from an untrusted third party, the user is denied onboarding/login.

Verify that the application employs integrity protections, such as code signing or sub-resource integrity. The application must not load or execute code from untrusted sources, such as loading includes, modules, plugins, code, or libraries from untrusted sources or the Internet.

Voatz applications employ integrity protection for both iOS and Android. The Android application implements Play Integrity to ensure that the application is licensed and passes strong integrity checks. This involves a three way check between the device->Google, device->Voatz servers, and Google->Voatz servers. An attempt to bypass this results in failure to onboard and/or login. The following code describes the integrity checks recommended by Google:

```
val appIntegrity = decodeIntegrityTokenResponse.getTokenPayloadExternal().getAppIntegrity()
    val appRecognitionVerdict = appIntegrity.getAppRecognitionVerdict
    val integrityPackageName = appIntegrity.getPackageName
    if(!appRecognitionVerdict.equalsIgnoreCase("PLAY_RECOGNIZED")) {
        val error = s"The certificate or package name does not match Google Play records, app
recognition verdict: $appRecognitionVerdict"
        log.error(s"Play Integrity app integrity check failed for $deviceId, $nonce: $error")
        PlayIntegrityStatusAsync.setAppIntegrityError(deviceId, nonce, error)
        (false, error)
    }else if (!integrityPackageName.equalsIgnoreCase("com.voatz.vma")) {
        val error = s"App package name mismatch, received: $integrityPackageName"
        log.error(s"Play Integrity app integrity check failed for $deviceId, $nonce: $error")
        PlayIntegrityStatusAsync.setAppIntegrityError(deviceId, nonce, error)
        (false, error)
    }else{
        val deviceIntegrity =
decodeIntegrityTokenResponse.getTokenPayloadExternal().getDeviceIntegrity()
        val deviceRecognitionVerdict = deviceIntegrity.getDeviceRecognitionVerdict()
```

```

    if(!deviceRecognitionVerdict.contains("MEETS_STRONG_INTEGRITY")) {
      import scala.collection.JavaConverters._
      val deviceRecognitionVerdictList =
asScalaBuffer(deviceRecognitionVerdict).toList.mkString(";")
      val error = s"Device integrity check failed, device recognition verdict:
$deviceRecognitionVerdictList"
      log.error(s"Play Integrity device integrity check failed for $deviceId, $nonce: $error")
      PlayIntegrityStatusAsync.setDeviceIntegrityError(deviceId, nonce, error)
      (false, error)
    }

```

Verify that the application has protection from sub-domain takeovers if the application relies upon DNS entries or DNS sub-domains, such as expired domain names, out of date DNS pointers or CNAMEs, expired projects at public source code repos, or transient cloud APIs, serverless functions, or storage buckets (autogen-bucket-id.cloud.example.com) or similar. Protections can include ensuring that DNS names used by applications are regularly checked for expiry or change.

1. Voatz has domain/sub-domain protection measures enabled at each of its registrars that include periodic rotation of DNSSEC keys.
2. Voatz has set up weekly alerts at its DNS registrars that provide summary updates on any upcoming expiries or attempts to tamper with its DNS entries.

Business Logic Security Requirements

Verify the application will only process business logic flows for the same user in sequential step order and without skipping steps.

1. The following is from the screen order service on the Voatz Web App. This module directs the sequential step flow of VWA for a given user in a given election.

```

export class ScreenOrderService {
  constructor(...,
    private readonly ballotListService: BallotListService,
    private readonly declarationService: DeclarationService
  ) {}
  ...
  nextRoute(currentScreenName: ScreenName) {
    ...
  }
  ...
}

```

2. Walker, Michael and Owens, Wendy. Test Report for Test and Evaluation of the Voatz Remote Accessible Ballot Delivery, Marking and Return (RABDMR) System, Pro V&V (July 17, 2020).

Verify the application will only process business logic flows with all steps being processed in realistic human time, i.e. transactions are not submitted too quickly.

1. The following is from the transaction validations module.

```
private def resolvePreconditions: Future[(Boolean, String)] = {
  resolveUser map{ userValidation =>
    if(!userValidation.valid){
      Future{Future{(false, userValidation.reason)}}
    }else{
      resolveEvent map{ eventValidation =>
        if(!eventValidation.valid){
          Future{(false, eventValidation.reason)}
        }else{
          resolveControlNumber map{ controlNumberValidation =>
            if(!controlNumberValidation.valid){
              (false, controlNumberValidation.reason)
            }else{
              (true, "")
            }
          }
        }
      }
    }
  } flatMap(x => x) flatMap(y => y)
}
```

2. The following is from the transaction validations module.

```
private def resolveCustomer: Future[ValidReasonPair] = {
  UserModule.getByUserId(...)
  ...
  private def resolveEvent: Future[ValidReasonPair] = {
    EventModule.getByEventId(...)
    ...
  }
  private def resolveControlNumber: Future[ValidReasonPair] = {
    ControlNumberModule.controlNumberValidAsFut(userId, eventId)
    ...
  }
}
```

Comments: Critical transactions are split into multiple steps as part of a state machine where subsequent steps check and validate successful completion of preceding steps and any failure detected results in rollback. This ensures that no transaction is submitted too eagerly.

Verify the application has appropriate limits for specific business actions or transactions which are correctly enforced on a per user basis.

1. The following is from the Voatz Web App backend. This prevents a voter from voting more than once in an election.

```

async vote(elector: Elector, ballot: Ballot, signatureUrl?: string, locale?: string):
    Promise<string | AlreadyVotedError | MissingDocumentError> {
    ...
    if (result.status === 412) {
        const message = await result.text();
        if (/CNERR01/.test(message)) {
            return new AlreadyVotedError();
        } else if (/CNERR04/.test(message)) {
            return new MissingDocumentError();
        }
    }
    ...
}

```

2. The following core server code retrieves the elections that a user has voted in; this prevents VMA users from voting more than once in a given election.

```

case EncryptedGetVotedEventIds(apiRequest, sessionCookie) => {
    val theSender = sender()
    ...
}

```

Comments: Strict controls in the app code and server restrict such transactions. Most importantly, users are restricted to a single cast ballot per election.

Verify the application has sufficient anti-automation controls to detect and protect against data exfiltration, excessive business logic requests, excessive file uploads or denial of service attacks.

1. The following web app backend code restricts the size of document uploads to 3MB and restricts the type of image files to a fixed set of standard file formats.

```

export type SupportedImageType = 'image/jpeg' | 'image/png' | 'image/heic';
const maxDocumentSize = 3 * 1024 * 1024;
...
export class ImageValidator implements ApiWrapperFactory<Representation> {
    wrap(handler: EnrichedApiHandler<Representation>): ApiHandler {
        return (event) => {
            if (hasSupportedContentType(event)) {
                const contentType = event.headers['content-type'] as SupportedImageType;
                if (event.body !== undefined) {
                    if (event.body.length <= maxDocumentSize) {
                        ...
                    }
                }
            }
        }
    }
}

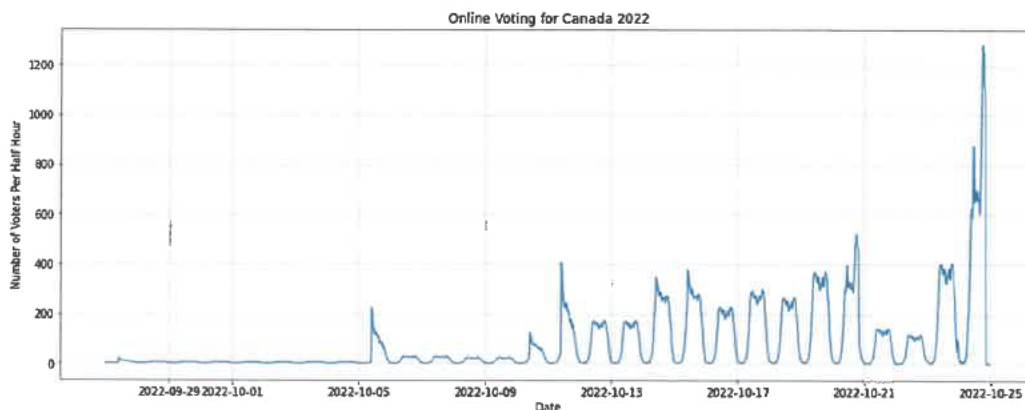
```

2. Voatz Mobile Voting Platform An Overview: Security, Identity, Auditability, Voatz (2020). <https://voatz.com/wp-content/uploads/2020/07/voatz-security-whitepaper.pdf>

Comments: Denial of Service attacks are prevented using multiple layers of defense, including DDoS attack mitigation provided through Cloudflare.

Verify the application has business logic limits or validation to protect against likely business risks or threats, identified using threat modeling or similar methodologies.

1. The following model was developed to predict voting patterns for the October 2022 Canadian municipal elections. Any significant deviation from this model would alert us to a potential threat. The model was extremely accurate.



2. Validations are used throughout all APIs. The following example prevents activity outside of an election window in VWA.

```
if(time_now <= election_start){
  return(ElectionNotOpenYetError);
} else if(election_close <= time_now){
  return(ElectionClosedError);
}
```

File Upload Requirements

Verify that the application will not accept large files that could fill up storage or cause a denial of service.

1. The following web app backend code restricts the size of document uploads to 3MB and restricts the type of image files to a fixed set of standard file formats. Similar code exists on the frontend as well.

```
export type SupportedImageType = 'image/jpeg' | 'image/png' | 'image/heic';
const maxDocumentSize = 3 * 1024 * 1024;
...
export class ImageValidator implements ApiWrapperFactory<Representation> {
  wrap(handler: EnrichedApiHandler<Representation>): ApiHandler {
    return (event) => {
      if (hasSupportedContentType(event)) {
        const contentType = event.headers['content-type'] as SupportedImageType;
```

```

if (event.body !== undefined) {
  if (event.body.length <= maxDocumentSize) {
    ...
  }
}

```

2. The following core server code similarly restricts the size of images uploaded from VMA. Similar code exists in the app code itself as well.

```

let router = RestApiRouter.submitUserImage(parameters: parameters)
  restApiManager.makeHttpRequest(router: router) { response in
    let currentStatus = UploadStatus(image: nil, url: url, position: .faceDetection, type:
      .imageToServer)
  }

```

Comments: Voatz Mobile Apps only send HTTPS urls of user and document images uploaded to secure cloud storage with well-defined IAM roles and bucket policies and never upload any files to the application/web tier directly, so there is no possibility of a denial of service.

File Execution Requirements

Verify that user-submitted filename metadata is not used directly with system or framework file and URL API to protect against path traversal.

Voatz applications never upload files directly to Voatz servers. Live document images and selfies are captured via the device camera and uploaded with strict file naming conventions to secure file storage, with well-defined IAM roles and bucket policies, where micro services process the files further thus eliminating the possibility of path traversal attacks.

Verify that user-submitted filename metadata is validated or ignored to prevent the disclosure, creation, updating or removal of local files (LFI).

Voatz applications never upload files directly to Voatz servers. Live document images and selfies are captured via the device camera and uploaded with strict file naming conventions to secure file storage, with well-defined IAM roles and bucket policies, where micro services process the files further thus eliminating the possibility of disclosure, creation, updating or removal of local files.

Verify that user-submitted filename metadata is validated or ignored to prevent the disclosure or execution of remote files (RFI), which may also lead to SSRF.

Voatz applications never upload files directly to Voatz servers. Live document images and selfies are limited to jpeg/png file types and are captured via the device camera and uploaded with strict file naming conventions to secure file storage, with well-defined IAM roles and bucket policies, where micro services process the files further thus eliminating the possibility of disclosure or execution of remote files.

Verify that the application protects against reflective file download (RFD) by validating or ignoring user-submitted filenames in a JSON, JSONP, or URL parameter, the response Content-Type header should be set to text/plain, and the Content-Disposition header should have a fixed filename.

All Voatz APIs that allow any file download employ strict filename conventions and use Content-Disposition header. For example the code below allows a Voatz Administration Portal user to download an affidavit pdf:

```
val request = GetAffidavit(apiRequest, sessionCookie)
    onComplete(forwardRequest(request)) {
        case Success(payload: AffidavitResult) => {
            respondWithHeader(`Access-Control-Expose-Headers`("Content-Disposition")) {
                respondWithHeader(`Content-Disposition`(attachment, Map(("filename",
payload.fileName)))) {
                    complete(HttpEntity.apply(MediaTypes.`application/pdf`, payload.fileBytes.toArray) )
                }
            }
        }
    }
```

Voatz mobile apps never upload/download files directly to/from Voatz servers. Live document images and selfies are captured via the device camera and uploaded with strict file naming conventions to secure file storage, with well-defined IAM roles and bucket policies, where micro services process the files further thus eliminating the possibility of RFD attacks.

Verify that untrusted file metadata is not used directly with system API or libraries, to protect against OS command injection.

No file metadata is ever used directly with system APIs. All file uploads employ secure cloud storage with well-defined IAM roles and bucket policies eliminating any possibility of OS command injection.

File Storage Requirements

Verify that files obtained from untrusted sources are stored outside the web root, with limited permissions, preferably with strong validation.

Voatz applications are strictly sandboxed and files are never uploaded to the web server or application server directly. All file uploads employ secure cloud storage with well-defined IAM roles and bucket policies.

Verify that files obtained from untrusted sources are scanned by antivirus scanners to prevent upload of known malicious content.

1. Policy A.12: Operations Security Policy, Section 5.0: Protection from malware
Anti-virus: this shall be installed on all key devices including firewalls, proxy servers, all servers and all end user devices.

2. Voatz Mobile Voting Platform An Overview: Security, Identity, Auditability, Voatz (2020).
<https://voatz.com/wp-content/uploads/2020/07/voatz-security-whitepaper.pdf>

Comments: All file uploads employ secure cloud storage where files are processed via dedicated micro services that sanitize and scan uploads via Sophos, then analyze them for type, size and other metadata.

File Download Requirements

Verify that the web tier is configured to serve only files with specific file extensions to prevent unintentional information and source code leakage. For example, backup files (e.g. .bak), temporary working files (e.g. .swp), compressed files (.zip, .tar.gz, etc) and other extensions commonly used by editors should be blocked unless required.

1. The following is from the web configuration file. The web server configuration explicitly denies file extensions to prevent unintentional information leakage.

```
<Files ~ "\.(bak|swp|zip|gz|cgi|shtml|phtml|php3?)$">
  Require all denied
</Files>
```

2. Policy A.9: Access Control Policy, Section 6.5 - Access Control to Program Source Code: Source code shall be managed using on-premise repositories that segregate the code into the major parts of the service.
 The following principles must be adhered to where possible:
 - o Program source code must not be held in operational systems where possible.

Verify that direct requests to uploaded files will never be executed as HTML/JavaScript content.

1. N/A
2. N/A

Comments: N/A: End users are never allowed to upload any files directly to Voatz servers. Files are uploaded internally via well-defined micro services that communicate with secure cloud storage using private and restricted APIs employing strict access control and AES GCM encryption with non-repeating nonce. Any files uploaded via mobile apps to secure cloud storage such as live document images and selfies for identity verification are secured via specific IAM roles and bucket policies.

SSRF Protection Requirements

Verify that the web or application server is configured with an allow list of resources or systems to which the server can send requests or load data/files from.

1. Policy A.13: Network Security Policy, Section 4.0:

All the traffic to and from a particular list of IPs and Domains (Blacklisted) should be denied at the Firewall Level.

2. Policy AC.9: Server Security Policy, Section 4.2 General Configuration and Backup Guidelines: Operating System configuration should be in accordance with approved Voatz, Inc. IT guidelines. Services and applications that will not be used must be disabled where practical.

Comments: Voatz web servers only communicate with whitelisted APIs on the application servers. Files are never uploaded to the web server or application server directly. All file uploads employ secure cloud storage with well-defined IAM roles and bucket policies.

Generic Web Service Security Verification Requirements

Verify that all application components use the same encodings and parsers to avoid parsing attacks that exploit different URI or file parsing behavior that could be used in SSRF and RFI attacks.

1. The following is from an API routing module (used by all application components).

```
// Defines all of the routing supported.
def allRoutes: Route = encodeResponse {
```

2. The following is from an API routing module for receipts (used by all application components).

```
routes = handleExceptions(exceptionHandler) {
  withRequestTimeoutResponse(request => timeoutResponse) {
```

Verify that access to administration and management functions is limited to authorized administrators.

1. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction: Voatz follows least privilege when granting access to information and application system functionality.

Additionally, access to information/data deemed sensitive or confidential is limited based on a need to know.

2. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software: Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.

Comments: These restrictions apply to election administrators and Voatz system administrators alike.

Verify API URLs do not expose sensitive information, such as the API key, session tokens etc.

Voatz API URLs are obfuscated with random UUIDs that are rotated periodically, eliminating the possibility of exposing sensitive information.

RESTful Web Service Verification Requirements

Verify that enabled RESTful HTTP methods are a valid choice for the user or action, such as preventing normal users using DELETE or PUT on protected API or resources.

1. The following is from an API routing module.

```
private def heartbeat() = post {
  requireAuthorization(Customer.View) {
```

2. The following is from an API routing module.

```
private def updateOrganizationUser() = put {
  requireAuthorization(OrganizationUser.Edit) {
    entity(as[OrganizationUserSnapshot]) { apiRequest =>
```

Comments: Each API request is either a DELETE, POST, or PUT request. Each request requires appropriate authentication. For example, put and delete requests require authentication that identifies the request as originating from an organization user, i.e., an election administrator. Every other request requires similar authentication, depending on the nature of the API request.

Verify that JSON schema validation is in place and verified before accepting input.

Voatz APIs perform strict JSON validation for parameter types and mandatory and optional attributes. Such checks are automatically enforced by the routing layer and any requests that fail such checks are prevented from reaching the application layer.

Verify that RESTful web services that utilize cookies are protected from Cross-Site Request Forgery via the use of at least one or more of the following: triple or double submit cookie pattern (see references), CSRF nonces, or Origin request header checks.

1. The following is from the core server API routing module.

```
protected val CsrfToken = "Csrf-Token"
protected val SessionCookie = "WS"

authenticateOrRejectWithChallenge[HttpApiSession](None =>
  SessionAuthenticatorAsync.validate(sessionCookie, csrfToken)
```

2. The following is from the Voatz Web App API authentication module.

```
const sessionId = getSessionId(event);
const csrfToken = event.headers[csrfTokenHeaderName.toLowerCase()];
if (sessionId) {
  if (csrfToken) {
    const s = await this.sessionManager.authenticate(sessionId, csrfToken);
```

Comments: CSRF tokens and nonces are used and origin request headers and are checked in a CORS (Cross-Origin Resource Sharing) router inside the API router.

SOAP Web Service Verification Requirements

Verify that XSD schema validation takes place to ensure a properly formed XML document, followed by validation of each input field before any processing of that data takes place.

1. N/A
2. N/A

Comments: N/A: No SOAP based web services are used internally in any Voatz applications. No external SOAP based web services are invoked by any Voatz applications.

Dependency

Verify that all components are up to date, preferably using a dependency checker during build or compile time.

Voatz uses OWASP's dependency check scanner that catalogs all open source components used in Voatz applications. Dependency check is integrated into SBT, Gradle and Jenkins CI server to automate dependency checks on each build.

Verify that all unneeded features, documentation, samples, configurations are removed, such as sample applications, platform documentation, and default or example users.

Voatz front end and back end code does not use any sample applications, configurations or default users. Additionally, the code base is regularly evaluated to remove unused features and APIs.

Verify that if application assets, such as JavaScript libraries, CSS stylesheets or web fonts, are hosted externally on a content delivery network (CDN) or external provider, Subresource Integrity (SRI) is used to validate the integrity of the asset.

1. Voatz Admin Portal JavaScript resources are bundled by webpack at build time and subresource integrity is added to these files by adding the -subresource-integrity option to the build script as follows.

```
"ng build --subresource-integrity"
```

2. Once the flag is added, angular-cli automatically generates a hash for the application resources and adds to the index.html when running npm run build, as the following shows.

```
<script  
  type="text/javascript"  
  src="runtime.js"
```

```
integrity="sha.."
crossorigin="...">
</script>
```

Unintended Security Disclosure Requirements

Verify that web or application server and framework error messages are configured to deliver user actionable, customized responses to eliminate any unintended security disclosures.

Well-defined error codes are used to ensure that the error is actionable, eliminating any unintended security disclosures. The following is from the validations module.

```
val errorCodes: Map[String, String] = Map("DAERR01" -> "Org. verification failed. Please contact
your elections administrator",
"DAERR02" -> "Attendance record already exists",
"DCERR01" -> "A user account with the information provided already exists. Please login.")
```

Verify that web or application server and application framework debug modes are disabled in production to eliminate debug features, developer consoles, and unintended security disclosures.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE-4: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE-4: **Pass**, June 19, 2020.

Comments: The logback configuration is set to ERROR in production to eliminate unintended security disclosures.

Verify that the HTTP headers or any part of the HTTP response do not expose detailed version information of system components.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE-6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE-6: **Pass**, June 19, 2020.

Comments: No HTTP responses contain such information.

HTTP Security Headers Requirements

Verify that every HTTP response contains a Content-Type header. text/*, /+xml and application/xml content types should also specify a safe character set (e.g., UTF-8, ISO-8859-1).

1. The following HTTP response is from the verification API.


```
< Access-Control-Max-Age: 1728000
< Content-Type: application/json; charset=UTF-8
< Content-Length: 3888
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
```

2. The following HTTP response is from the Voatz Web App backend.

```
Content-Type: text/plain; charset=utf-8
```

Comments: Each HTTP response contains the "Content-Type" header. Accepted types include UTF-8.

Verify that all API responses contain Content-Disposition: attachment; filename="api.json" header (or other appropriate filename for the content type).

1. The following API response is from a routing module.

```
respondWithHeader(`Access-Control-Expose-Headers` (Seq("Content-Disposition"))) {
```

2. The following API response is from a routing module.

```
respondWithHeader(`Content-Disposition` ("attachment", Map(("filename", payload.fileName))))
{
  respondWithMediaType(MediaTypes.`application/pdf`) {
```

Comments: The Content-Disposition header is set, along with the Media Type, on REST API responses.

Verify that a Content Security Policy (CSP) response header is in place that helps mitigate impact for XSS attacks like HTML, DOM, JSON, and JavaScript injection vulnerabilities.

1. The following HTTP Response is from the verification API.

```
< Access-Control-Max-Age: 1728000
< Content-Type: application/json; charset=UTF-8
< Content-Length: 3888
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
```

2. ISO Policy A.14.2.5 - Secure System Engineering Principles:
Principles for engineering secure systems shall be established, documented, maintained and applied to any information system implementation efforts.

Comments: The X-XSS-Protection response header stops the response payload from loading when a reflected cross-site scripting (XSS) attack is detected.

Verify that all responses contain an X-Content-Type-Options: nosniff header.

1. The following HTTP Response is from the verification API.

```
< Access-Control-Max-Age: 1728000
< Content-Type: application/json; charset=UTF-8
< Content-Length: 3888
< X-Content-Type-Options: nosniff
< X-XSS-Protection: 1; mode=block
```

2. All REST API responses contain X-Content-Type-Options: nosniff.

Verify that a Strict-Transport-Security header is included on all responses and for all subdomains, such as Strict-Transport-Security: max-age=15724800; includeSubdomains.

1. The following HTTP Response is from the verification API.

```
< Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

2. All REST API responses contain Strict-Transport-Security: max-age=.; includeSubdomains.

Verify that a suitable "Referrer-Policy" header is included, such as "no-referrer" or "same-origin".

1. The following referrer policy is strictly enforced. For same-origin, send the full Referrer. For cross-origin, send only the origin, unless it's an HTTPS to HTTP request, in which case send nothing.
2. Voatz APIs enforce the strict-origin-when-cross-origin referrer policy in any pre-flight as well as subsequent real requests.

Verify that the content of a web application cannot be embedded in a third party site by default and that embedding of the exact resources is only allowed where necessary by using suitable Content-Security-Policy: frame-ancestors and X-Frame-Options response headers.

1. The Voatz Admin Portal employs the following tag to prevent access from a website independently from the request.

```
<meta http-equiv="X-Frame-Options" content="deny">
```

This prevents the page from being displayed in a frame, regardless of the site that tries to do it. In addition, CORS checks are used as well.

2. Voatz's voter-facing web application (i.e., VWA) employs the same checks described above, along with additional domain whitelisting.

Validate HTTP Request Header Requirements

Verify that the application server only accepts the HTTP methods in use by the application/API, including pre-flight OPTIONS and logs/alerts on any requests that are not valid for the application context.

1. The following is from an API router module, which whitelists those methods and requests that are accepted.

```
var allHeaders = `Access-Control-Allow-Methods`(HttpMethods.OPTIONS,HttpMethods.POST
...
```

2. The following is from an API router module, which logs an error when a request cannot be completed.

```
handleNotFound {
  log.error(s"The requested resource could not be found")
  complete(HttpResponse(NotFound, getAllHeaders, "The requested resource could not be found"))
}
```

Comments: All REST APIs enforce the Access-Control-Allow-Methods header.

Verify that the supplied Origin header is not used for authentication or access control decisions, as the Origin header can easily be changed by an attacker.

1. The following is from the core server API router.

```
validate(sessionCookie, csrfToken)
```

2. The following is from the Voatz Web App backend API router.

```
const s = await this.sessionManager.authenticate(sessionId, csrfToken);
```

Comments: Authentication and access control are based on session cookies and CSRF tokens.

Verify that the Cross Origin Resource Sharing (CORS) Access-Control-Allow-Origin header uses a strict allow list of trusted domains and subdomains to match against and does not support the "null" origin.

1. The following is from a CORS router module.

```
CORSSupport {
  private val allowOriginHeader = `Access-Control-Allow-Origin`.white_list
```

2. The following is from an API routing module.

```
if(originOpt.isDefined){
  if(originOpt.get.matches(" ... ")){
```

Comments: Trusted domains are whitelisted.

Select Controls from Open Web Application Security Project - Application Security Verification Standard 4.0.2 Level 2

Secure Software Development Lifecycle

Verify the use of a secure software development lifecycle that addresses security in all stages of development.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-10: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-10: **Pass**, June 19, 2020.

Verify documentation and justification of all the application's trust boundaries, components, and significant data flows.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-1: **Pass**, June 19, 2020.

Verify implementation of centralized, simple (economy of design), vetted, secure, and reusable security controls to avoid duplicate, missing, ineffective, or insecure controls.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-5: **Pass**, June 19, 2020.

Verify availability of a secure coding checklist, security requirements, guideline, or policy to all developers and testers.

1. Policy A.14: Software Development Security Policy, Section 3.2 Development Methodology:

To ensure that code is developed in a secure manner and that the security controls are implemented throughout every stage, Voatz, Inc. development methodology includes:

- Testing of the application's security features.
2. Policy A.14: Software Development Security Policy, Section 3.2 Development Methodology:

To ensure that code is developed in a secure manner and that the security controls are implemented throughout every stage, Voatz, Inc. development methodology includes:

- Ensuring secure coding standards are followed.

Comments: As part of Voatz' ISO 27001 and StateRAMP certification processes, such policies were established.

Authentication Architectural Requirements

Verify that communications between application components, including APIs, middleware and data layers, are authenticated. Components should have the least necessary privileges needed.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-AUTH-10: **Pass**, June 19, 2020.
2. Policy AC.9: Server Security Policy, Section 4.2 General Configuration and Backup Guidelines:
 - Always use the least privilege to perform a function.
 - Privileged access must be performed over secure channels, (e.g., encrypted network connections using SSH or IPSec).

Verify that the application uses a single vetted authentication mechanism that is known to be secure, can be extended to include strong authentication, and has sufficient logging and monitoring to detect abuse or breaches.

1. Policy A.13: Network Security Policy, 5.9 Encrypt sensitive network traffic:
 - Appropriate encryption and authentication methods should be used for the transmission of sensitive data and remote access.
2. Policy AC.9: Server Security Policy, Section 4.2 General Configuration and Backup Guidelines:
 - Access to services should be logged and protected through access-control methods.
3. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:
 - All security-related events on critical or sensitive systems must be logged and audit trails saved.

Comments: Voatz mobile apps employ a biometric (touch/face) element to authenticate with the app and strong identity verification is necessary in order to perform the most important task on the app: submitting a marked ballot. Similarly, admin portal users must enroll in 2FA. All security related events on the server such as authentication attempts are logged via well-defined audit transitions which log the source, type, and timestamp of the transaction.

Verify that all authentication pathways and identity management APIs implement consistent authentication security control strength, such that there are no weaker alternatives per the risk of the application.

1. The following is from a core server API router module.

```
authenticateOrRejectWithChallenge[HttpApiSession](None =>
  SessionAuthenticatorAsync.validate(sessionCookie, csrfToken))
```

2. The following is from a Voatz Web App authentication module.

```
const s = await this.sessionManager.authenticate(sessionId, csrfToken);
```

Comments: The same authentication pathway is implemented consistently across all REST APIs. Any password-based authentication involves slow hashing primitives such as B-crypt and no weak hashing primitives are employed. Any encryption is performed using AES GCM with a unique nonce per request and no insecure padding/block modes are employed.

Access Control Architectural Requirements

Verify enforcement of the principle of least privilege in functions, data files, URLs, controllers, services, and other resources. This implies protection against spoofing and elevation of privilege.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.
2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows least privilege when granting access to information and application system functionality.

Additionally, access to information/data deemed sensitive or confidential is limited based on a need to know.

Comments: The principle of least privilege applies to users as well as Voatz team members. Access to certain resources is only granted on a need-to-know basis.

Verify the application uses a single and well-vetted access control mechanism for accessing protected data and resources. All requests must pass through this single mechanism to avoid copy and paste or insecure alternative paths.

1. Voatz Mobile App users access functionality through an API router, which enforces access control.

```
requireAuthentication(User.View) {
  entity(as[ApiRequest]) { apiRequest =>
    optionalCookie("WS"){sessionCookie =>
```

```
val request = ApiRequest(apiRequest, sessionCookie)
```

The function `requireAuthentication` is defined as follows.

```
def requireAuthentication(directive: HttpApiSession => RequestContext => Unit):
  RequestContext => Unit = optionalHeaderValueByName(CsrfToken) {
    csrfToken =>
      optionalCookie(SessionCookie) { sessionCookie =>
        authenticate(SessionAuthenticatorAsync.validate(sessionCookie, csrfToken)) .
```

Additionally, every API is secured by authorization checks of the following form.

```
val snapByDevIdOptFut = UserModule.getByDeviceId(apiRequest.deviceId, sessionCookieOpt)
snapByDevIdOptFut onComplete {
  case Success(snapByDevIdOpt) => {
    if (!snapByDevIdOpt.isDefined) {
      val err = "user not found"
    } else {
      val userIdFromDevId = snapByDevIdOpt.get
      SessionAuthenticatorAsync.validateSessionAgainstUser(sessionCookieOpt, userIdFromDevId)
    } onComplete {
      case Success(sessionValid) => {
        if (!sessionValid) {
          theSender ! ForbiddenError("Session cookie provided does not authorize you to perform this
          action")
        } else {
          if (userIdFromDevId != request.userId) {
            theSender ! ForbiddenError("Supplied deviceId does not belong to requested user")
          } else {
```

Here, the user attached to the device sending the request is validated against the user holding the current session, which in turn is matched against the user making the request.

2. Voatz Web App users must pass an additional layer of authentication. All sensitive data is protected by the following authentication wrapper.

```
wrap = (action: AuthenticatedApiHandler) => this.wrapOptional(
  (e: APIGatewayProxyEventV2, s?: Session) =>
    ...
    return async event => {
      const sessionId = getSessionId(event);
      const csrfToken = event.headers[csrfTokenHeaderName.toLowerCase()];
      if (sessionId) {
        if (csrfToken) {
          const s = await this.sessionManager.authenticate(sessionId, csrfToken);
        }
      }
      ...
```

```

else {
  return {
    statusCode: 400,
    body: JSON.stringify(new MissingCsrfTokenError().toJSON()),
  };
}

```

Verify that attribute or feature-based access control is used whereby the code checks the user's authorization for a feature/data item rather than just their role. Permissions should still be allocated using roles.

1. The following is from the election administration user module.

```

val orgAdminUserFut = SessionAuthenticatorAsync.getUserIfOrgAdmin(sessionCookie,
  request.organizationId)
orgAdminUserFut onComplete {
  case Success((true, _, userId)) => {
    val validFut = OrganizationValidationsModule.isGetBallotsRequestValid(request, userId)
    validFut onComplete {
      case Success(validReasonPair) => {
        if(validReasonPair.valid){
          val resultFut = OrganizationModule.getBallots{...

```

Here, the initial call to `getUserIfOrgAdmin` checks whether the user has the appropriate role to access the requested resource and the subsequent call to `isGetBallotsRequestValid` which is defined as follows.

```

def isGetBallotsRequestValid(request: ApiGetBallotsRequest, userId: Int)(implicit db: Database):
Future[ValidReasonPair] = async{
  val orgId = request.organizationId
  val canUserGetBallots = await(OrganizationUserMod.canUserGetBallots(userId, orgId))
  if(!canUserGetBallots){
    ValidReasonPair(false, "Forbidden: user does not have permission to perform this action")

```

This checks whether the user with the Admin role has the more fine grained permission to access the specifically requested feature/resource.

2. Users are granted access using a matrix of roles (e.g., Org. Admin, Org. User, etc.) and fine-grained, feature-based access. For instance, the user below can enroll for fetching the cast vote record, but cannot access voter affidavits.

```

"org" : ..73, "user" : ...814, "enrollcvr" : true, "getaffidavits" : false, "getballots" : true,
"getballotstatus" : true, "getcvr" : true

```

But the user below can do both.

```
"org" : ..73, "user" : ...813, "enrollcvr" : true, "getaffidavits" : true, "getballots" : true,  
"getballotstatus" : true, "getcvr" : true
```

Input and Output Architectural Requirements

Verify that input and output requirements clearly define how to handle and process data based on type, content, and applicable laws, regulations, and other policy compliance.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM-2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM-2: **Pass**, June 19, 2020.

Verify that serialization is not used when communicating with untrusted clients. If this is not possible, ensure that adequate integrity controls (and possibly encryption if sensitive data is sent) are enforced to prevent deserialization attacks including object injection.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM-8: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM-8: **Pass**, June 19, 2020.

Comments: Voatz APIs employ AES GCM encryption with non-repeating nonce to prevent any deserialization attacks. Additionally, Voatz apps employ certificate pinning, state of the art code virtualization/obfuscation techniques and can only be used on non-rooted and non-jailbroken devices.

Cryptographic Architectural Requirements

Verify that there is an explicit policy for management of cryptographic keys and that a cryptographic key lifecycle follows a key management standard such as NIST SP 800-57.

1. Policy A.10: Cryptography, Section 3.6: Key Management:
 - **Generation:**
 - Cryptographic keys must be generated and stored in a secure manner that prevents loss, interception, theft, or compromise.
 - Key generation must be seeded from an industry standard random number generator.
2. Policy A.10: Cryptography, Section 3.6: Key Management:
 - **Key Removal:**

- Key removal shall occur after an archival phase and after adequate analysis to ensure that loss of that key will not correspond to loss of data or other keys.

Verify that consumers of cryptographic services protect key material and other secrets by using key vaults or API based alternatives.

1. Policy A.6.2.1: Mobile Device Management Policy, Section 3.1 Technology and Security Requirements:
If the user is storing passwords in the device, an encrypted password store must be used.
2. Policy A.10: Cryptography, Section 3.1 Encryption Method and Application:
All private keys for encryption must be password protected and not stored in the clear on systems.

Verify that all keys and passwords are replaceable and are part of a well-defined process to re-encrypt sensitive data.

1. Policy AC.10: Database Credentials Policy, Section 4.4 Access to Database Usernames and Passwords:
 - Developer groups must have a process in place to ensure that database passwords are controlled and changed in accordance with the Password Policy. This process must include a method for restricting knowledge of database passwords to a need-to-know basis.
2. Policy AC.5: Password Policy, Section 5.3: Password Protection Standards:
 - If an account or password is suspected to have been compromised, report the incident to Voatz, Inc. IT and change all passwords immediately.
 - Password cracking or guessing may be performed on a periodic or random basis by Voatz, Inc. IT Ops or its delegates. If a password is guessed or cracked during one of these scans, the user will be required to change it.

Verify that the architecture treats client-side secrets - such as symmetric keys, passwords, or API tokens - as insecure and never uses them to protect or access sensitive data.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE-1: **Pass**, June 19, 2020.

Errors, Logging and Auditing Architectural Requirements

Verify that a common logging format and approach is used across the system.

1. The following is from an application module.

```
import org.slf4j.{Logger, LoggerFactory}
```



```
private implicit val log: Logger = LoggerFactory.getLogger("...")
```

2. The following is from an API routing module.

```
import org.slf4j.LoggerFactory
private val log = LoggerFactory.getLogger("...")
```

Comments: Standard and consistent system loggers are used across the platform.

Verify that logs are securely transmitted to a preferably remote system for analysis, detection, alerting, and escalation.

1. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:
All security-related events on critical or sensitive systems must be logged and audit trails saved.
2. Policy A.11: Physical and Environmental Security Policy, Section 4.2 Physical Access Control:
Maintain physical access audit logs for data center(s) and/or sensitive facilities entry/exit points.

Data Protection and Privacy Architectural Requirements

Verify that all sensitive data is identified and classified into protection levels.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-4: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-4: **Pass**, June 19, 2020.

Verify that all protection levels have an associated set of protection requirements, such as encryption requirements, integrity requirements, retention, privacy and other confidentiality requirements, and that these are applied in the architecture.

1. Policy A.10: Acceptable Encryption Policy, Section 3.3: Data Protection:
 - Data at rest:
 - All Voatz, Inc. and customer data deemed a secret shall be stored in an encrypted manner.
 - Data in motion:
 - All data sent or received outside of Voatz, Inc. data centers and cloud environments (corporate and elections networks) shall be encrypted for transmission.
 - Where possible, data exchanged inside the Voatz, Inc. network should be encrypted.
2. Policy A.18: Information Security/Sensitivity, Section 3.2 Data Protection Standards
Each classification of data has different requirements for protection throughout the lifecycle of use.

Communications Architecture Requirements

Verify the application encrypts communications between components, particularly when these components are in different containers, systems, sites, or cloud providers.

1. Policy A.10: Acceptable Encryption Policy, Section 3.3: Data Protection:
 - Data in motion:
 - All data sent or received outside of Voatz, Inc. data centers and cloud environments (corporate and elections networks) shall be encrypted for transmission.
2. Policy A.10: Acceptable Encryption Policy, Section 3.3: Data Protection:
 - Data in motion:
 - Where possible, data exchanged inside the Voatz, Inc. network should be encrypted.

Comments: Communication between systems is encrypted via TLS v. 1.2 and in cases where it is appropriate, encrypted at the application layer as well.

Verify that application components verify the authenticity of each side in a communication link to prevent person-in-the-middle attacks. For example, application components should validate TLS certificates and chains.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK-2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK-2: **Pass**, June 19, 2020.

Comments: TLS certificates are checked.

Malicious Software Architectural Requirements

Verify that a source code control system is in use, with procedures to ensure that check-ins are accompanied by issues or change tickets. The source code control system should have access control and identifiable users to allow traceability of any changes.

1. Policy A.9: Access Control Policy, Section 3.0: Business Requirements:
Cloud-based collaboration systems (e.g. Jira, Dropbox, Confluence, Slack)
2. Policy A.14: Software Development Security Policy, Section 3.3 Code and Configuration Reviews:
During implementation and maintenance, all source code and system configurations must be reviewed and approved before deployment by a team member who did not participate in implementing the change. Exceptions must be documented in Jira and approved by Voatz CEO or CTO.

Comments: A version control system is in place for all application and server code. All changes are connected to tickets on a collective board and comments or code branches reflect the appropriate ticket or version. All changes and code branches are tracked.

Secure File Upload Architectural Requirements

Verify that user-uploaded files are stored outside of the web root.

1. Policy AC.9: Server Security Policy, Section 4.2: General Configuration and Backup Guidelines
Always use the least privilege to perform a function. Do not use root/admin accounts when a non-privileged account is sufficient.
2. Policy A.9: Access Control Policy, Section 6.1 Information Access Restriction
Voatz follows least privilege when granting access to information and application system functionality.

Comments: End users are never allowed to upload any files directly to Voatz servers. Files are uploaded internally via well-defined micro services that communicate with secure cloud storage using private and restricted APIs employing strict access control and AES GCM encryption with non-repeating nonce.

Verify that user-uploaded files - if required to be displayed or downloaded from the application - are served by either octet stream downloads, or from an unrelated domain, such as a cloud file storage bucket. Implement a suitable content security policy to reduce the risk from XSS vectors or other attacks from the uploaded file.

1. N/A
2. N/A

Comments: N/A: User uploaded files (e.g., identity document images, selfies, etc.) are never displayed or downloaded via the application and are never uploaded directly to Voatz servers. These images are placed in secure cloud storage controlled via well-defined IAM (Identity and Access Management) roles and bucket policies, processed via well-defined private micro services using AES GCM encryption with non-repeating nonce, and deleted from secure cloud storage after being processed for document/identity verification.

Configuration Architectural Requirements

Verify the segregation of components of differing trust levels through well-defined security controls, firewall rules, API gateways, reverse proxies, cloud-based security groups, or similar mechanisms.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-5: **Pass**, June 19, 2020.

Verify that the build pipeline warns of out-of-date or insecure components and takes appropriate actions.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK-6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK-6: **Pass**, June 19, 2020.

Comments: Voatz integrates Snyk with Atlassian to achieve DevSecOps within existing CI/CD (Continuous Integration / Continuous Delivery) development workflows. This enables software component analysis to track and analyze open source component dependencies, enforce security and compliance policies, and fix potential vulnerabilities at source.

Verify that the build pipeline contains a build step to automatically build and verify the secure deployment of the application, particularly if the application infrastructure is software defined, such as cloud environment build scripts.

1. Voatz integrates Snyk with Atlassian to achieve DevSecOps within existing CI/CD development workflows. This enables software component analysis to track and analyze open source component dependencies, enforce security and compliance policies, and fix potential vulnerabilities at source.
2. This integration also provides much needed context such as severity and exploit maturity for vulnerabilities, enables creation of fix pull requests to upgrade the direct dependencies or patch vulnerabilities, and prioritize fixes via priority scores while integrating with Jira issue tracking.

Verify that application deployments adequately sandbox, containerize and/or isolate at the network level to delay and deter attackers from attacking other applications, especially when they are performing sensitive or dangerous actions such as deserialization.

1. Voatz employs SELinux security policy functions as a whitelist for user and application behavior by isolating applications into specific SELinux domains that are tailored to the application's permitted behaviors. Access to files, local interprocess communications (IPC) mechanisms, the network, and various other system resources are restricted on a per-domain basis.
2. By ensuring that every service that listens on a network is run in a confined SELinux domain, an attacker's access to resources, ability to pivot, read, and write, and the possible damage they can do is limited.

Verify the application does not use unsupported, insecure, or deprecated client-side technologies such as NSAPI plugins, Flash, Shockwave, ActiveX, Silverlight, NACL, or client-side Java applets.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE-5: **Pass**, June 19, 2020.

2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE-5: **Pass**, June 19, 2020.

Comments: Voatz applications and servers only use secure third-party services via well-defined application dependencies or SDKs. No insecure or deprecated client-side plugins such as Flash, ActiveX, Silverlight, or Java applets are used anywhere in the Voatz architecture.

Credential Storage Requirements

Verify that passwords are stored in a form that is resistant to offline attacks. Passwords SHALL be salted and hashed using an approved one-way key derivation or password hashing function. Key derivation and password hashing functions take a password, a salt, and a cost factor as inputs when generating a password hash.

1. Policy A.10: Cryptography, Section 3.5:
 - Where applicable, user passwords shall be stored in a secure manner by using HMAC with hash.
 - All user passwords should be salted on a per secret basis.
 - The salt should be random.
 - The salt size should be the same length as the output of the hashing function.
 - The salt should not be considered a secret.
2. The following is from a cryptography module.

```
def hashPassword(password: String): PasswordHash = {
  val salt = generateSalt
  PasswordHash(password.bcrypt(salt))
}
```

Verify that the salt is at least 32 bits in length and be chosen arbitrarily to minimize salt value collisions among stored hashes. For each credential, a unique salt value and the resulting hash SHALL be stored.

1. Policy A.10: Cryptography, Section 3.5:
 - The salt should be random.
2. The following is from a cryptography module.

```
private static final int BCrypt_SALT_LEN = 16;
```

Comments: The salt is 128 bits (16 bytes) in length.

Verify that if PBKDF2 is used, the iteration count SHOULD be as large as verification server performance will allow, typically at least 100,000 iterations.

1. The following is from a cryptography module.

```
PasswordHash(password.bcrypt(salt))
```

2. The following is from a cryptography module.

```
password.isBcrypt(passwordHash.hash)
```

Comments: PBKDF2 is not used as the password hash; bcrypt is used instead.

Credential Recovery Requirements

Verify that if OTP or multi-factor authentication factors are lost, that evidence of identity proofing is performed at the same level as during enrollment.

1. The following is from an API router module.

```
authenticateOrRejectWithChallenge[HttpApiSession](None =>
SessionAuthenticatorAsync.validate(sessionCookie, csrfToken))
```

2. The following is from an API router module.

```
authorize(performAuthorization(authorizationKey, session))
```

Comments: When login credentials are lost, the account is reactivated using the same process as during activation, using the same authentication and authorization checks as above.

Look-up Secret Verifier Requirements

Verify that lookup secrets can be used only once.

Comments: N/A: No lookup secrets are used.

Verify that lookup secrets have sufficient randomness (112 bits of entropy), or if less than 112 bits of entropy, salted with a unique and random 32-bit salt and hashed with an approved one-way hash.

Comments: N/A: No lookup secrets are used.

Verify that lookup secrets are resistant to offline attacks, such as predictable values.

Comments: N/A: No lookup secrets are used.

Out of Band Verifier Requirements

Verify that the out of band verifier retains only a hashed version of the authentication code.

1. The following is from a database module.

```
val pinSet = set("pin", Cryptography.hashPassword(requestCustomerSnap.pin).hash)
```

2. The following is from a voter authentication module.

```
val customerPinHashOpt = await(getPin(customerId))
```

Verify that the initial authentication code is generated by a secure random number generator, containing at least 20 bits of entropy (typically a six digital random number is sufficient).

1. Policy A.10: Cryptography, Section 3.6:
 - Key generation must be seeded from an industry standard random number generator.
2. The following is from a voter authentication module.

```
val totp = Totp(Cryptography.issueRandomKey(256).getBytes, 6, SHA256, 30).generate()
```

3. The following is from a cryptography module.

```
import java.security.SecureRandom;
```

Single or Multi-factor One Time Verifier Requirements

Verify that symmetric keys used to verify submitted OTPs are highly protected, such as by using a hardware security module or secure operating system based key storage.

1. Policy A.10: Cryptography, Section 3.6: Key Management:
 - Cryptographic keys must be generated and stored in a secure manner that prevents loss, interception, theft, or compromise.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-STORAGE-1: Pass, June 19, 2020.

Verify that approved cryptographic algorithms are used in the generation, seeding, and verification of OTPs.

1. Policy A.10: Cryptography, Section 3.6: Key Management:
 - Key generation must be seeded from an industry standard random number generator.
2. The following is from a voter authentication module.

```
val totp = Totp(Cryptography.issueRandomKey(256).getBytes, 6, SHA256, 30).generate()
```

Verify that time-based OTP can be used only once within the validity period.

1. The following is from a voter authentication module.

```
else if(otpAlreadyVerified){
    log.error(s"otp $otp already verified for mobile $mobileNumber")
}
```

2. The following is from a voter authentication module.

```
if(checkOtpTimestamp(snapshot.timestamp, timestamp, onBoardingLimits)){
    ...
}
else{
    val error = "Otp timeout was exceeded"
```

Verify that if a time-based multi factor OTP token is re-used during the validity period, it is logged and rejected with secure notifications being sent to the holder of the device.

1. The following is from an API routing module.

```
log.error(s"EncryptVerifyOtp request failed: $err")
```

2. The following is from an API routing module.

```
HttpResponse(StatusCodes.InternalServerError, entity = "otp verification request failed")
```

Cryptographic Software and Devices Verifier Requirements

Verify that cryptographic keys used in verification are stored securely and protected against disclosure, such as using a Trusted Platform Module (TPM) or Hardware Security Module (HSM), or an OS service that can use this secure storage.

1. Policy A.10: Acceptable Encryption Policy, Section 3.6 Key Management:
Cryptographic keys must be generated and stored in a secure manner that prevents loss, interception, theft, or compromise.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-STORAGE-1: **Pass**, June 19, 2020.

Comments: All cryptographic keys on a mobile app are generated and stored in a TPM (Trusted Platform Module) if such is available. Otherwise, these keys are generated and stored in sandboxed storage only accessible by the Voatz Mobile App.

Verify that the challenge nonce is at least 64 bits in length, and statistically unique or unique over the lifetime of the cryptographic device.

1. The following is from a cryptography module.

```
val GCM_NONCE_LENGTH = 12
```

2. The following is from a cryptography module.

```
val nonce = (firstIV zip toUnsignedByteArray(sequenceNumber, 12))
```

Comments: All nonces are at least 96 bits (12 bytes) in length so it is statistically unlikely that a nonce will repeat.

Verify that approved cryptographic algorithms are used in the generation, seeding, and verification.

1. Policy A.10: Cryptography, Section 3.6:
Key generation must be seeded from an industry standard random number generator.
2. The following is from a cryptography module.

```
import java.security.SecureRandom;  
val ranGen = new SecureRandom()  
val key = new Array[Byte](size)  
ranGen.nextBytes(key)
```

Service Authentication Requirements

Verify that if passwords are required for service authentication, the service account used is not a default credential (e.g. root/root or admin/admin are default in some services during installation).

1. Policy A.9: Access Control Policy, Section 4.4 Management of Secret Authentication Information of Users
 - Where Voatz IT Ops issues initial passwords, these are sent out of band and users are asked to change their passwords and any default passwords.
2. Policy A.13: Network Security Policy, Section 5.2 Privilege Access Control:
 - Privileged access to prevent configuration changes should be restricted to authorized personnel only.
 - Passwords shall follow strong password mechanisms outlined under the Password Guidelines.
 - Access control shall be used to provide separate authentication, authorization, and accounting services for network-based access.
 - A Privileged Access Management solution can control credentials accessing the device and commands that can be executed when a session is initiated, providing a complete audit of both commands and sessions.

Verify that passwords are stored with sufficient protection to prevent offline recovery attacks, including local system access.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application:

Encryption shall be implemented in:

- storage on disk, particularly secrets and user passwords.
2. Policy A.10: Acceptable Encryption Policy, Section 3.5: User Passwords
 - Where applicable, user passwords shall be stored in a secure manner by using HMAC with hash.

Comments: Passwords are stored, salted and hashed, in secure device storage.

Session Binding Requirements

Verify that session tokens are generated using approved cryptographic algorithms.

1. Policy A.10: Cryptography, Section 3.1: All Voatz, Inc. encryption shall be done using approved cryptographic modules.
2. Policy A.10: Cryptography, Section 3.2: The use of proprietary encryption algorithms is not allowed for any purpose, unless reviewed by qualified experts and approved by Voatz, Inc. IT Ops.

Comments: Session tokens are derived from a cryptographically secure random number generator and have 256 bits of entropy.

Session Logout and Timeout Requirements

Verify that the application gives the option to terminate all other active sessions after a successful password change (including change via password reset/recovery), and that this is effective across the application, federated login (if present), and any relying parties.

1. The following routine is called as part of the password reset/recovery flow in the user management module.

```
deleteAllPreviousSessions(userId)
```

This code is responsible for deleting any active sessions for the user in cache.

2. All active sessions are terminated on successful password change and the user is forced to authenticate again with the new credentials before being able to access any features in the app.

Verify that users are able to view and (having re-entered login credentials) log out of any or all currently active sessions and devices.

1. The following is from an API routing module.

```
private def encryptedLogoutUser() = post {
```


2. Onboarding and Identity Verification with Voatz:

<https://vimeo.com/533255584/1baa3910a6>

Token Based Session Management

Verify the application uses session tokens rather than static API secrets and keys, except with legacy implementations.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH-3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH-3: **Pass**, June 19, 2020.

Comments: Voatz applications and servers employ CSRF tokens and session tokens generated via cryptographically secure random number generators. No static API secrets or keys are used in the Voatz architecture.

Verify that stateless session tokens use digital signatures, encryption, and other countermeasures to protect against tampering, enveloping, replay, null cipher, and key substitution attacks.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH-3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH-3: **Pass**, June 19, 2020.

Other Access Control Considerations

Verify the application has additional authorization (such as step up or adaptive authentication) for lower value systems and/or segregation of duties for high value applications to enforce anti-fraud controls as per the risk of application and past fraud.

1. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows least privilege when granting access to information and application system functionality. Additionally, access to information/data deemed sensitive or confidential is limited based on a need to know basis depending on the segregation of duties.
2. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.

Comments: These restrictions apply to election administrators and Voatz system administrators alike.

Data Classification

Verify that regulated private data is stored encrypted while at rest, such as Personally Identifiable Information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application: Encryption shall be implemented in:
 - storage on disk, particularly secrets and user passwords.

2. Policy A.10: Acceptable Encryption Policy, Section 3.3 Data Protection:

- Data at rest:

All Voatz, Inc. and customer data deemed a secret shall be stored in an encrypted manner.

Comments: All sensitive private data on disk is encrypted at rest using FIPS 140-2 approved cryptographic modules.

Algorithms

Verify that industry proven or government approved cryptographic algorithms, modes, and libraries are used, instead of custom coded cryptography.

1. Policy A.10: Cryptography, Section 3.1: All Voatz, Inc. encryption shall be done using approved cryptographic modules.
2. Policy A.10: Cryptography, Section 3.2: The use of proprietary encryption algorithms is not allowed for any purpose, unless reviewed by qualified experts and approved by Voatz, Inc. IT Ops.

Comments: Bulk encryption is done via AES GCM. ECC and ECDSA are used for public key encryption and digital signatures; NSA and NIST-approved elliptic curves are used for both in all apps. SHA-256 and bcrypt are the secure hash algorithms used.

Verify that encryption initialization vector, cipher configuration, and block modes are configured securely using the latest advice.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CRYPTO-3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CRYPTO-3: **Pass**, June 19, 2020.

Verify that random number, encryption or hashing algorithms, key lengths, rounds, ciphers or modes, can be reconfigured, upgraded, or swapped at any time, to protect against cryptographic breaks.

1. The code is sufficiently modularized in order to allow for ciphers, hash functions, keys, and other parameters to be changed as necessary. In particular, we are cognizant of the anticipated need

to adopt post-quantum cryptographic algorithms and will adopt the new post-quantum NIST standards when implementations of these algorithms are complete and tested.

2. The following example is from a cryptographic module to demonstrate this modularity.

```
def getX509EncPublicKey(encodedPublicKey: String, keyAlgorithm: String)
```

```
def generatePublicKeys(numOfKeys: Option[Int], keySizeOpt: Option[Int])(implicit
  cryptoAlgorithm: CryptoAlgorithm)
```

Verify that known insecure block modes (i.e. ECB, etc.), padding modes (i.e. PKCS#1 v1.5, etc.), ciphers with small block sizes (i.e. Triple-DES, Blowfish, etc.), and weak hashing algorithms (i.e. MD5, SHA1, etc.) are not used unless required for backwards compatibility.

1. Policy A.10: Cryptography, Section 3.1:

All Voatz, Inc. encryption shall be done using approved cryptographic modules. Symmetric cryptosystem key lengths must be at least 256 bits. Asymmetric crypto-system keys must be of a length that yields equivalent strength. Voatz, Inc.'s key length requirements shall be reviewed annually as part of the yearly security review and upgraded as technology allows. All private keys for encryption must be password protected and not stored in the clear on systems.

2. The following is from a cryptographic module.

```
val cipher = Cipher.getInstance("AES/GCM/NoPadding", "SunJCE")
```

Verify that nonces, initialization vectors, and other single use numbers must not be used more than once with a given encryption key. The method of generation must be appropriate for the algorithm being used.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CRYPTO-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CRYPTO-5: **Pass**, June 19, 2020.

Comments: Every new request is encrypted with a new, random, 12-byte (96 bit) nonce.

```
def getNewNonce() = {
  val GCM_NONCE_LENGTH = 12
  val random = SecureRandom.getInstanceStrong()
  val nonce = new Array[Byte](GCM_NONCE_LENGTH)
  random.nextBytes(nonce)
  nonce
}
```

Similarly, every response is encrypted with a new random nonce obtained via the above method.

Random Values

Verify that all random numbers, random file names, random GUIDs, and random strings are generated using the cryptographic module's approved cryptographically secure random number generator when these random values are intended to be not guessable by an attacker.

1. Policy A.10: Acceptable Encryption Policy, Section 3.6: Key Management
Key generation must be seeded from an industry standard random number generator.
2. The following is from a cryptographic module, taken from java.util.Random.

```
val random = SecureRandom.getInstanceStrong()
```

Comments: All random values are generated using a cryptographically secure random number generator from entropy gathered by the operating system. In particular, random numbers on the server are generated using the SecureRandom class in the java.security library.

Verify that random GUIDs are created using the GUID v4 algorithm, and a Cryptographically-secure Pseudo-random Number Generator (CSPRNG). GUIDs created using other pseudo-random number generators may be predictable.

1. Policy A.10: Acceptable Encryption Policy, Section 3.6: Key Management
Key generation must be seeded from an industry standard random number generator.
2. The following is from a cryptographic module, taken from java.util.Random.

```
val random = SecureRandom.getInstanceStrong()
```

Comments: All GUIDs (i.e., UUIDs) are generated using the aforementioned random number generator.

Secret Management

Verify that a secrets management solution such as a key vault is used to securely create, store, control access to and destroy secrets.

1. Policy A.10: Acceptable Encryption Policy, Section 3.6: Key Management:
Key management should be planned which will include secure key generation, use, storage, distribution, recovery, and removal.
2. Policy AC.5: Password Policy, Section 2.0: Purpose:
The purpose of this policy is to establish a standard for creation of strong passwords, the protection of those passwords, and the frequency of change.

Verify that key material is not exposed to the application but instead uses an isolated security module like a vault for cryptographic operations.

1. Policy A.10: Acceptable Encryption Policy, Section 3.6: Key Management:

Key management should be planned which will include secure key generation, use, storage, distribution, recovery, and removal.

2. Policy AC.5: Password Policy, Section 2.0: Purpose:

The purpose of this policy is to establish a standard for creation of strong passwords, the protection of those passwords, and the frequency of change.

Log Content Requirements

Verify that the application logs security relevant events including successful and failed authentication events, access control failures, deserialization failures and input validation failures.

1. Policy AC.9: Server Security Policy, Section 4.2 General Configuration and Backup Guidelines: Operating System configuration should be in accordance with approved Voatz, Inc. IT guidelines. Services and applications that will not be used must be disabled where practical.
 - Access to services should be logged and protected through access-control methods.

2. The following is an example of a logged error in an API router.

```
val logError = s"Authentication failed for customer $customerId with error: $reason"
log.error(logError)
```

Comments: The server logs all security related events via well-defined audit transactions that record the source, type, and timestamp of the transaction.

Verify that each log event includes necessary information that would allow for a detailed investigation of the timeline when an event happens.

1. Policy A.12: Operations Security Policy, Section 7.0 Logging and Monitoring

Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.

2. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:

All security-related events on critical or sensitive systems must be logged and audit trails saved.

Comments: Server logs include timestamps and all relevant information necessary to trace intrusion attempts. These logs have enabled Voatz to identify and mitigate threats in real time.

Log Processing Requirements

Verify that all authentication decisions are logged, without storing sensitive session identifiers or passwords. This should include requests with relevant metadata needed for security investigations.

1. Policy A.12: Operations Security Policy, Section 7.0 Logging and Monitoring
Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.

2. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:
All security-related events on critical or sensitive systems must be logged and audit trails saved.

Comments: All authentication requests are logged using well-defined audit transactions that log the source, type and timestamp along with relevant non-sensitive metadata such as IP address, device id, etc.

Verify that all access control decisions can be logged and all failed decisions are logged. This should include requests with relevant metadata needed for security investigations.

1. Policy A.12: Operations Security Policy, Section 7.0 Logging and Monitoring
Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.
2. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:
All security-related events on critical or sensitive systems must be logged and audit trails saved.

Comments: All access control decisions (including failures) are logged using well-defined audit transactions along with relevant metadata such as IP address, device id, host, etc.

Log Protection Requirements

Verify that the application appropriately encodes user-supplied data to prevent log injection.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM-2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM-2: **Pass**, June 19, 2020.

Verify that all events are protected from injection when viewed in log viewing software.

1. N/A
2. N/A

Comments: N/A: Log viewing is enabled via local micro services that access log files in read only mode, thus eliminating the possibility of any injection attacks. Additionally, logs are always archived with checksums.

Verify that security logs are protected from unauthorized access and modification.

1. Policy A.9: Access Control Policy, Section 3.0:

This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.

- Defense in Depth – security shall not depend upon any single control but be the sum of a number of complementary controls;
- Least Privilege – the default approach taken shall be to assume that access is not required, rather than to assume that it is. In particular, all access to election infrastructure and election data shall be limited to those individuals that have a business or legal need-to-know;

2. Policy A.9: Access Control Policy, Section 3.0:

This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.

- Need to Know – access is only granted to the information required to perform a role, and no more. The default setting of the access control system has to be “deny-all”;
- Need to Use – users will only be able to access physical and logical facilities required for their role.

Comments: Security logs are protected from unauthorized access by denying any external application access to the log files. Any log aggregation for viewing and inspection is done via local micro services with strict access control and no public interfaces.

Verify that time sources are synchronized to the correct time and time zone. Strongly consider logging only in UTC if systems are global to assist with post-incident forensic analysis.

1. The following is from the user onboarding module.

```
def checkOtpTimestamp(otpTimestamp: String, nowTimestamp: String, onBoardingLimits:
OnboardingLimitsSnapshot): Boolean = {
  val otps = OffsetDateTime.parse(otpTimestamp)
  val now = OffsetDateTime.parse(nowTimestamp)
  val differenceInSeconds = ChronoUnit.SECONDS.between(otps, now)
  differenceInSeconds <= onBoardingLimits.timeout
}
```

2. The following is from the voter file management module.

```
val recordToCreateWithTs = recordToCreate.:+(KeyValuePair("insertTimestamp",
OffsetDateTime.now.toString))
```

Comments: All timestamps are logged with an offset from UTC/Greenwich in the ISO-8601 calendar system.

Error Handling

Verify that a "last resort" error handler is defined which will catch all unhandled exceptions.

1. The following is from an API routing module.

```
implicit def exceptionHandler: ExceptionHandler =
  ExceptionHandler {
    case e => { ctx =>
      log.error("Unhandled api exception", e)
```

2. The following is from an API routing module.

```
implicit val myRejectionHandler: RejectionHandler = RejectionHandler.newBuilder()
...
.handleNotFound {
  log.error(s"The requested resource could not be found")
  complete(HttpResponse(NotFound, getAllHeaders, "The requested resource could
not be found"))
```

General Data Protection

Verify the application protects sensitive data from being cached in server components such as load balancers and application caches.

1. Policy A.10: Acceptable Encryption Policy, Section 3.3: Data Protection:
Data at rest:
 - All Voatz, Inc. and customer data deemed a secret shall be stored in an encrypted manner.
2. Policy A.10: Acceptable Encryption Policy, Section 3.3: Data Protection:
Data in motion:
 - All data sent or received outside of Voatz, Inc. data centers and cloud environments (corporate and elections networks) shall be encrypted for transmission.
 - Where possible, data exchanged inside the Voatz, Inc. network should be encrypted.

Comments: No sensitive data such as PII (Personally Identifying Information) or user credentials is cached in application caches or load balancers. As a rule of thumb, all information cached in application caches is short lived, relates to the current/active user session, and is purged automatically in accordance with well-defined TTL (time to live) settings. Additionally, such caches are not accessible via any public interface or API and protected with cryptographically secure random secrets known only to the process accessing the cache.

Verify that all cached or temporary copies of sensitive data stored on the server are protected from unauthorized access or purged/invalidated after the authorized user accesses the sensitive data.

1. Policy A.9: Access Control Policy, Section 3.0:

This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.

- Defense in Depth – security shall not depend upon any single control but be the sum of a number of complementary controls;
- Least Privilege – the default approach taken shall be to assume that access is not required, rather than to assume that it is. In particular, all access to election infrastructure and election data shall be limited to those individuals that have a business or legal need-to-know;

2. Policy A.9: Access Control Policy, Section 3.0:

This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.

- Need to Know – access is only granted to the information required to perform a role, and no more. The default setting of the access control system has to be “deny-all”;
- Need to Use – users will only be able to access physical and logical facilities required for their role.

Comments: All cached information on Voatz servers is protected via well-defined TTL (time to live) settings and such cached information is automatically purged after the TTL settings expire. Additionally, there is no public access to the caching layer via any API and it is protected by cryptographically secure random secrets known only to the process accessing the cache.

Verify the application minimizes the number of parameters in a request, such as hidden fields, Ajax variables, cookies and header values.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM-1: **Pass**, June 19, 2020.

Comments: Only those parameters that are necessary are collected.

Verify the application can detect and alert on abnormal numbers of requests, such as by IP, user, total per hour or day, or whatever makes sense for the application.

1. Policy A.12: Operations Security Policy, Section 7.0: Logging and Monitoring:
Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.
2. Policy A.16 Information Security Incident Management, Section 1.0: Overview:
It is critical to the organization that incidents that threaten the security or privacy of critical information are properly identified, contained, investigated, and remedied.

Comments: The following code from the user onboarding module tracks requests by IP, mobile, email, and deviceId within a certain configurable interval to detect abnormal number of requests from a source and block access, if necessary.

```
def maxRequestsExceededByIp(ipAddress: String, onBoardingLimits:
OnboardingLimitsSnapshot)(implicit db: Database, ec: ExecutionContext): Future[Boolean] = {
  val nowMillis = OffsetDateTime.now.toInstant.toEpochMilli
  val key = s"userpreregisterread$ipAddress"
  Cache.getListOfStringFromCache(key) map{ recordIpFromCacheOpt =>
    if(recordIpFromCacheOpt.isDefined){
      val exceeded =
recordIpFromCacheOpt.get.map(OffsetDateTime.parse(_).toInstant.toEpochMilli).filter { x =>
nowMillis-x < onBoardingLimits.consecutiveRequestInterval}.size >=
onBoardingLimits.consecutiveRequests
      if(exceeded){
        log.error(s"max. consecutive requests from same ip $ipAddress exceeded")
      }
      exceeded
    }else{
      false
    }
  }
}

def maxRequestsExceededByMobile(mobileNumber: String, onBoardingLimits:
OnboardingLimitsSnapshot)(implicit db: Database, ec: ExecutionContext): Future[Boolean] = {
  val nowMillis = OffsetDateTime.now.toInstant.toEpochMilli
  val key = s"userpreregisterread$mobileNumber"
  Cache.getListOfStringFromCache(key) map{ recordMobileFromCacheOpt =>
    if(recordMobileFromCacheOpt.isDefined){
      recordMobileFromCacheOpt.get.map(OffsetDateTime.parse(_).toInstant.toEpochMilli).filter
{x => nowMillis-x < onBoardingLimits.consecutiveRequestInterval}.size >=
onBoardingLimits.consecutiveRequests
    }else{
      false
    }
  }
}
```

Sensitive Private Data

Verify accessing sensitive data is audited (without logging the sensitive data itself), if the data is collected under relevant data protection directives or where logging of access is required.

1. Policy A.9: Access Control Policy, Section 4.1 User Registration and De-registration
Voatz IT Ops is responsible for registering and de-registering users. They shall establish and maintain an agreed process for the registration and de-registration of users. The process shall ensure that:
 - The request to register or de-register a user has come from someone with appropriate authority and shall be auditable.
 - A record of the request and the action taken is kept for audit purposes.
2. Policy A.9: Access Control Policy, Section 4.2 User Access Provisioning
Voatz IT Ops is responsible for assigning and revoking access rights for all user types to all systems and services. They shall establish and maintain agreed upon processes for managing the user access rights. These shall ensure that:
 - A record of the request and the action taken is kept for audit purposes.

Verify that sensitive information contained in memory is overwritten as soon as it is no longer required to mitigate memory dumping attacks, using zeroes or random data.

1. The following is from the iOS app API encryption module, which creates a wrapper around the Swift Data type to wipe out bytes on deallocation so secrets cannot be recovered and read by reverse engineering.

```
public class ClearableData {
    var value: Data

    deinit {
        value.deinitialize()
    }
    ....
    func replaceSubrange(_ subrange: Range<Data.Index>, with data: [UInt8]) {
        value.replaceSubrange(subrange, with: data)
    }
}
```

This is used to replace sensitive data with random, meaningless data.

```
var mask = ClearableData()
for counter in 0 ..< interval {
    tseed.replaceSubrange((tseed.count - 4) ..< tseed.count, with: pack(counter))
    mask.append(CC.digest(tseed, alg: digest))
}
mask.count = maskLength
return mask
```

2. The following is from the Android app API encryption module, which defines an extension to de-initialize byte arrays.

```
fun ByteArray?.deInitialize() {
    if (this != null && this.isNotEmpty()) {
```

```

    val size = this.size
    for (i in 0 until size) {
        this[i] = 0
    }
}
}

```

This is used to replace sensitive data that is no longer processed by the app.

```

val decryptedBytes: ByteArray?
try {
    decryptedBytes = cipher.doFinal(data)
    val decryptedBytesToString = String(decryptedBytes)
    decryptedBytes.delInitialize()
}

```

Comments: Voatz iOS and Android applications implement approaches to zeroing out or randomizing sensitive data in memory as soon as it is no longer required to mitigate memory dumping attacks.

Verify that sensitive or private information that is required to be encrypted, is encrypted using approved algorithms that provide both confidentiality and integrity.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application
 - All Voatz, Inc. encryption shall be done using approved cryptographic modules.
2. Policy A.10: Acceptable Encryption Policy, 3.2 Proprietary Encryption Algorithms
 - The use of proprietary encryption algorithms is not allowed for any purpose, unless reviewed by qualified experts and approved by Voatz, Inc. IT Ops.

Comments: Voatz applications and servers employ AES GCM with non repeating nonce to encrypt all sensitive information in flight. Additionally, all sensitive information on disk is encrypted at rest.

Verify that sensitive personal information is subject to data retention classification, such that old or out of date data is deleted automatically, on a schedule, or as the situation requires.

1. Policy AM.1: Equipment Disposal, Section 4.1 Technology Equipment Disposal:

Selection of data destruction methods should be based on the sensitivity of the data being destroyed. Methods may include those procured in-house use and the use of off-site secure data destruction services provided by vetted third parties as necessary.

 - Erasing/overwriting
 - Erasing by overwriting is an acceptable method of scrubbing data that is not sensitive or requires safeguarding. Multiple passes should be performed with random overwrite patterns – not just all zeros or another single character. A minimum of three (3) overwrites is required; additional overwriting is recommended depending upon the sensitivity of the data to be erased.

2. Policy A.18: Information Security/Sensitivity, Section 3.2 Data Protection Standards

Data Retention

- Data shall be retained for as long as is necessary to provide the service to the client. After the retention period is over or upon specific request by the client, data shall be destroyed in accordance with Destruction and Disposal defined in this policy.

Destruction and Disposal

- IT Ops maintains and shall enforce a detailed list of approved destruction methods appropriate for each type of information stored, whether in physical storage media or in database records or backup files. When systems containing data are decommissioned, they must follow the following destruction policy:
- Paper documents containing Level 1, Level 2 information shall be shredded using secure, locked consoles designated in each office from which waste shall be periodically picked up by a security screened personnel. (CIC provides)
- Level 1 and Level 2 information on electronic files and data on Reusable Electronic Storage Devices should be reliably erased or physically destroyed using DOD Standard for Secure Data Sanitation (DOD 5220.22M). Functional electronic media that can be overwritten using a secure erase tool then may be recycled or disposed of. Non-functional electronic media (damaged disk drives) must be physically destroyed.

Client Communications Security Requirements

Verify that secured TLS is used for all client connectivity, and does not fall back to insecure or unencrypted protocols.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application:
 - All web applications via the most secure version of SSL/TLS.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-NETWORK-1: **Pass**, June 19, 2020.

Comments: All HTTP traffic is secured using TLS v. 1.2 or newer.

Verify using online or up to date TLS testing tools that only strong algorithms, ciphers, and protocols are enabled, with the strongest algorithms and ciphers set as preferred.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application:
 - All web applications via the most secure version of SSL/TLS.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application:

All Voatz, Inc. encryption shall be done using approved cryptographic modules. Symmetric cryptosystem key lengths must be at least 256 bits. Asymmetric crypto-system keys must be of a length that yields equivalent strength. Voatz, Inc.'s key length requirements shall be reviewed annually as part of the yearly security review and upgraded as technology allows.

Verify that old versions of SSL and TLS protocols, algorithms, ciphers, and configuration are disabled, such as SSLv2, SSLv3, or TLS 1.0 and TLS 1.1. The latest version of TLS should be the preferred cipher suite.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application:
 - All web applications via the most secure version of SSL/TLS.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-NETWORK-1: **Pass**, June 19, 2020.

Comments: TLS v. 1.2 (or newer) is used with an approved and standard cipher suite.

Server Communications Security Requirements

Verify that connections to and from the server use trusted TLS certificates. Where internally generated or self-signed certificates are used, the server must be configured to only trust specific internal CAs and specific self-signed certificates. All others should be rejected.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-NETWORK-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-NETWORK-2: **Pass**, June 19, 2020.
3. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-NETWORK-3: **Pass**, June 19, 2020.

Comments: All HTTP traffic is secured using TLS v. 1.2 or newer.

Verify that encrypted communications such as TLS is used for all inbound and outbound connections, including for management ports, monitoring, authentication, API, or web service calls, database, cloud, serverless, mainframe, external, and partner connections. The server must not fall back to insecure or unencrypted protocols.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application:
 - All web applications via the most secure version of SSL/TLS.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-NETWORK-1: **Pass**, June 19, 2020.

Comments: All HTTP traffic is secured using TLS v. 1.2 or newer.

Verify that all encrypted connections to external systems that involve sensitive information or functions are authenticated.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH-1: **Pass**, June 19, 2020.

2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH-1: **Pass**, June 19, 2020.

Verify that proper certification revocation, such as Online Certificate Status Protocol (OCSP) Stapling, is enabled and configured.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK-2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK-2: **Pass**, June 19, 2020.

Malicious Code Search

Verify that the application source code and third party libraries do not contain unauthorized phone home or data collection capabilities. Where such functionality exists, obtain the user's permission for it to operate before collecting any data.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE-12: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE-12: **Pass**, June 19, 2020.

Comments: The user gives permission for Voatz servers and apps to store only as much information as is needed to verify identity, receive a ballot, and submit a ballot. All unnecessary user information is deleted when it is no longer needed.

Verify that the application does not ask for unnecessary or excessive permissions to privacy related features or sensors, such as contacts, cameras, microphones, or location.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM-1: **Pass**, June 19, 2020.

Comments: Voatz Mobile Apps only request permissions that are absolutely necessary to conduct secure operations on the device. Voatz Mobile Apps never request permission for contact list or microphones, and they never track user location in the background. Voatz Mobile Apps request camera permission to enable the user to take a picture of their identity document to prove eligibility for an election.

Business Logic Security Requirements

Verify the application has configurable alerting when automated attacks or unusual activity is detected.

1. Policy A.12: Operations Security Policy, Section 5.0 Protection from Malware

A defense in depth approach shall be taken to protect Voatz from malware including, as a minimum:

- Firewalls: these shall be installed at all points where internal networks are connected to the internet. Further requirements on firewall configuration and maintenance are contained in the Communications Security Policy.
- Anti-virus: this shall be installed on all key devices including firewalls, proxy servers, all servers and all end-user devices.
- Threat monitoring: information about emerging threats shall be obtained from appropriate sources and necessary action taken, e.g. ensuring that users are alerted proactively of potential attacks.

2. Policy A.12: Operations Security Policy, Section 7.0 Logging and Monitoring

Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.

File Integrity Requirements

Verify that the files obtained from untrusted sources are validated to be of expected type based on the file's content.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM-2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM-2: **Pass**, June 19, 2020.

Comments: Voatz applications and servers never accept files from untrusted sources. Additionally, files are never uploaded directly to the servers and are instead processed via secure cloud storage configured with relevant IAM roles and permissions.

File Execution Requirements

Verify that the application does not include and execute functionality from untrusted sources, such as unverified content distribution networks, JavaScript libraries, node npm libraries, or server-side DLLs.

1. N/A
2. N/A

Comments: N/A: Voatz applications and servers rely on a well-defined set of external libraries from trusted and verified sources and these libraries have been demonstrated and proven to function consistently across numerous open source projects. Voatz applications and servers never use unverified CDNs. Additionally, Voatz servers run on Linux and do not employ any server side DLLs; such driver files only run on machines running a Windows operating system.

RESTful Web Service Verification Requirements

Verify that the message headers and payload are trustworthy and not modified in transit. Requiring strong encryption for transport (TLS only) may be sufficient in many cases as it provides both confidentiality and integrity protection. Per-message digital signatures can provide additional assurance on top of the transport protections for high-security applications but bring with them additional complexity and risks to weigh against the benefits.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application: Encryption shall be implemented in:
 - All web applications via the most secure version of SSL/TLS.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application: Encryption shall be implemented in:
 - App/client communication with backend infrastructure.

Comments: All HTTP traffic is secured using TLS v. 1.2 or newer.

SOAP Web Service Verification Requirements

Verify that the message payload is signed using WS-Security to ensure reliable transport between client and service.

1. N/A
2. N/A

Comments: N/A: Voatz does not use SOAP-based web services.

GraphQL and other Web Service Data Layer Security Requirements

Verify that a query allow list or a combination of depth limiting and amount limiting should be used to prevent GraphQL or data layer expression Denial of Service (DoS) as a result of expensive, nested queries. For more advanced scenarios, query cost analysis should be used.

1. N/A
2. N/A

Comments: N/A: No GraphQL is used in Voatz applications. All APIs that depend on queries run on data sources enforce strict pagination to control and limit the amount of data queried from the underlying data source.

Build

Verify that the application build and deployment processes are performed in a secure and repeatable way, such as CI / CD automation, automated configuration management, and automated deployment scripts.

1. Voatz employs Atlassian Bitbucket pipelines enabling efficient storage and management of build configurations, thus allowing management of the entire development workflow from code to deployment.
2. Additionally, Voatz integrates Snyk into Bitbucket pipelines to achieve DevSecOps within existing CI/CD development workflows. This enables software component analysis to track and analyze open source component dependencies, enforce security and compliance policies, and fix potential vulnerabilities at source.

Verify that compiler flags are configured to enable all available buffer overflow protections and warnings, including stack randomization, data execution prevention, and to break the build if an unsafe pointer, memory, format string, integer, or string operations are found.

1. For Swift (iOS) in XCode, -fobjc-arc is used for automatic ref counting, -PIE is used for position independent execution to ensure address space layout randomization, and -fstack-protector-all for stack smashing protection is enabled by default. For Objective C code, these are enabled explicitly.
2. For Kotlin (Android), concurrent mark and sweep GC is used by default, -Werror=format-security is used for format string vulnerability protections, mmap_min_addr is used to mitigate null pointer dereference privilege escalation, ASLR randomizes key locations in memory, dmesg_restrict and kptr_restrict is used to avoid leaking kernel addresses.

Verify that server configuration is hardened as per the recommendations of the application server and frameworks in use.

1. Voatz integrates Snyk into Bitbucket pipelines to achieve DevSecOps within existing CI/CD development workflows. This enables software component analysis to track and analyze component dependencies, enforce security and compliance policies, and fix potential vulnerabilities at source.
2. Vulnerability fixes are prioritized via priority scores and Jira tickets. This enables hardening of the server configuration regularly based on thorough analysis of component dependencies. Additionally, CVSS scores for all server components and dependencies are tracked to ensure compliance with security policies.

Verify that the application, configuration, and all dependencies can be re-deployed using automated deployment scripts, built from a documented and tested runbook in a reasonable time, or restored from backups in a timely fashion.

1. Voatz employs Atlassian Bitbucket pipelines to automate build, testing, and deployment of code.
2. This process is based on well-defined pipeline configuration files.

Dependency

Verify that third party components come from pre-defined, trusted and continually maintained repositories.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE-5: **Pass**, June 19, 2020.

Comments: A comprehensive list of the various third party components and their corresponding repositories used is available upon request. They are not listed here. Notable components and repositories include Amazon S3 for secure cloud storage, Redis for application cache, and Mongo and MySQL for secure databases.

Verify that the attack surface is reduced by sandboxing or encapsulating third party libraries to expose only the required behavior into the application.

1. Third party libraries are always encapsulated to introduce only the required behavior into the application. For example, the AWS SDK for Java exposes all AWS services but by pulling only the selected module for AWS storage and IAM, Voatz apps and servers ensure that any issues related to other unrelated AWS services such as EC2, S3, DynamoDB, etc. do not impact the application.
2. Such selective encapsulation is applied on all third party libraries used by mobile and web apps, as well as servers.

Validate HTTP Request Header Requirements

Verify that HTTP headers added by a trusted proxy or SSO devices, such as bearer token, are authenticated by the application.

1. Voatz applications and servers do not employ HTTP header-based authentication schemes such as bearer tokens. Any CSRF tokens or session cookies issued via cryptographically secure random generators and added to HTTP request and response headers are strictly validated at the routing layer.
2. The following code snippet demonstrates how such token cookie headers are validated before granting access to any APIs.

```
def validate(sessionCookieOpt: Option[HttpCookie], csrfTokenOpt: Option[String])(implicit ec:
ExecutionContext): Future[Authentication[HttpApiSession]] = (sessionCookieOpt, csrfTokenOpt)
match {
  case (Some(cookie), Some(token)) => {
    getSession(cookie.content) map { sessionOpt =>
      sessionOpt match {
```

```
case Some(HttpApiSession(sessionCookie, csrfToken, userId, lastUse)) => {
  if (csrfToken != csrfTokenOpt.get) {
    Left(AuthenticationFailedRejection(CredentialsRejected, List()))
  }
}
```

Select Controls from Open Web Application Security Project - Application Security Verification Standard 4.0.2 Level 3

General Authenticator Requirements

Verify impersonation resistance against phishing, such as the use of multi-factor authentication, cryptographic devices with intent (such as connected keys with a push to authenticate), or at higher AAL levels, client-side certificates.

1. Andrae, Philip; Braseth, Hilary; and Sawhney, Nimit. State-of-the-Art Security Performs First-Rate Threat Mitigation in Convention Elections, Voatz (2020).
<https://voatz.com/wp-content/uploads/2020/07/DR-V-Security-Whitepaper.pdf>
2. Voatz Mobile Voting Platform An Overview: Security, Identity, Auditability, Voatz (2020).
<https://voatz.com/wp-content/uploads/2020/07/voatz-security-whitepaper.pdf>

Comments: The most effective means to thwart phishing attempts for VMA is to use biometric identification. VMA authenticates a voter by requiring an official photo identification, along with a live video of the voter's face; the two images must match. MFA authentication occurs on the administrative end with two administrators needing their authenticator device in order to access the cast ballot record.

Verify that where a credential service provider (CSP) and the application verifying authentication are separated, mutually authenticated TLS is in place between the two endpoints.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application: Encryption shall be implemented in:
 - All web applications via the most secure version of SSL/TLS.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application: Encryption shall be implemented in:
 - App/client communication with back end infrastructure.

Comments: TLS v. 1.2 or newer is used to securely transmit information.

Session Logout and Timeout Requirements

If authenticators permit users to remain logged in, verify that re-authentication occurs periodically with 2FA both when actively used after 12 hours or after an idle period of 15 minutes

1. The following is from the session management module. User sessions are created with a TTL of 15 minutes, after which re-authentication is triggered.


```
def createMobileSession(sessionCookie: String, session: HttpApiSession) = {
  cache.set(s"mobilesession$sessionCookie", session, Some(Duration(900, TimeUnit.SECONDS)))
}
..
```

2. Voatz iOS and Android apps, including all authenticated interfaces, employ an in-app idle timer which automatically logs out after x minutes of inactivity. The following code from the Android app shows a warning dialog box and logs out when the timer expires.

```
private fun warningTimer(
  context: Context,
  isScreenOn: Boolean,
  dialog: MaterialDialog?
){
  warningTimer =
  object : CountDownTimer(remainingIdleTime.toLong(), 100) {
    override fun onTick(millisUntilFinished: Long) {
      if (isScreenOn) {
        if (TimeUnit.MILLISECONDS.toSeconds(millisUntilFinished).toInt() == 16) {
          (dialog)?.contentView?.announceForAccessibility(
            String.format(context.getString(R.string.logout_timeout_message), "15")
          )
        }
      }
      (dialog)?.setContent(
        String.format(
          context.getString(R.string.logout_timeout_message),
          TimeUnit.MILLISECONDS.toSeconds(
            millisUntilFinished

```

Comments: Voatz iOS and Android employ in-app idle timers that call log out and redirect the user to login or authenticate when the idle timer expires.

Algorithms

Verify that encrypted data is authenticated via signatures, authenticated cipher modes, or HMAC to ensure that ciphertext is not altered by an unauthorized party

1. The following is from a cryptographic module.

```
def verifySignature(publicKey: PublicKey, signedData: String, signature: String,
  signatureAlgorithm: String)
```

2. The following is from a cryptographic module.

```
def aes256GCMDecryptUsingKey(encryptedBytes: Array[Byte], key: Array[Byte], nonce:
  Array[Byte], tag: Array[Byte])
```

Comments: Galois Counter Mode (GCM) is used as the message authenticator for AES to ensure data integrity. ECDSA with an NSA and NIST-approved elliptic curve is used for digital signatures.

General Data Protection

Verify that regular backups of important data are performed and that test restoration of data is performed.

1. Backup and Restoration Policy, Section 3.0: Policy:
 - All user-level and system-level information maintained by Voatz, Inc. shall be backed up periodically. The backup media (if it exists) and backup copies shall be stored with sufficient protection and proper environmental conditions.
2. Backup and Restoration Policy, Section 3.1 Review and Testing
Voatz will carry out annual Testing to verify the effectiveness of the plan, train the DR team on what to do in actual disaster scenarios and highlight areas where the plan needs to be improved. Backup restores are also tested annually to ensure the integrity of the backups in case of a disaster.

Verify that backups are stored securely to prevent data from being stolen or corrupted.

1. Backup and Restoration Policy, Section 3.0: Policy:
 - Physical access controls implemented at offsite backup storage locations (if applicable) must meet or exceed the physical access controls of the source systems. Additionally, backup media must be protected in accordance with the highest sensitivity level of information stored.
2. Backup and Restoration Policy, Section 3.0: Policy:
 - All backup data must be stored encrypted using strong encryption mechanisms.

Code Integrity Controls

Verify that a code analysis tool is in use that can detect potentially malicious code, such as time functions, unsafe file operations and network connections.

1. Voatz Mobile App – Android Version 1.1.322 was scanned with AppSweep by Guardsquare on August 25, 2023 and found no such issues.
2. Voatz Mobile App – iOS was scanned with AppSweep by Guardsquare on August 4, 2023 and found no such issues.
3. Voatz uses the following tools for static code analysis:
 - AppSweep (Android, iOS)
 - Ktlint (Android)
 - XCode Analyzer (iOS)
 - Eslint (Web Apps)

Comments: This was submitted on September 13, 2023 as part of “Artifact #25 – Source Static Analysis” for Voatz’ StateRAMP certification.

Malicious Code Search

Verify that the application source code and third party libraries do not contain back doors, such as hard-coded or additional undocumented accounts or keys, code obfuscation, undocumented binary blobs, rootkits, or anti-debugging, insecure debugging features, or otherwise out of date, insecure, or hidden functionality that could be used maliciously if discovered.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE-5: **Pass**, June 19, 2020.

Comments: All code is reviewed internally and web, app, and server code is audited regularly by trusted third-party security experts to identify any such unwanted functionality.

Verify that the application source code and third party libraries do not contain time bombs by searching for date and time related functions.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE-5: **Pass**, June 19, 2020.

Comments: All code is reviewed internally and web, mobile app, and server code is audited regularly by trusted third-party security experts to identify any such unwanted functionality.

Verify that the application source code and third party libraries do not contain malicious code, such as salami attacks, logic bypasses, or logic bombs.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE-5: **Pass**, June 19, 2020.

Comments: All code is reviewed internally and web, mobile app, and server code is audited regularly by trusted third-party security experts to identify any such unwanted functionality.

Verify that the application source code and third party libraries do not contain Easter eggs or any other potentially unwanted functionality.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE-5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE-5: **Pass**, June 19, 2020.

Comments: All code is reviewed internally and web, mobile app, and server code is audited regularly by trusted third-party security experts to identify any such unwanted functionality.

2. Attachment C: OWASP Mobile Application Level Security Verification: Table

| | |
|---|--------------------|
| Attachment C: OWASP Mobile Application Level Security Verification | Yes/No/ Partial |
| Open Web Application Security Project - Mobile Application Security Verification Standard | |
| 1.2 Level 1 | |
| Architecture, design, and threat modeling | |
| All app components are identified and known to be needed. | Yes |
| Security controls are never enforced only on the client side, but on the respective remote endpoints. | Yes |
| A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture. | Yes |
| Data considered sensitive in the context of the mobile app is clearly identified. | Yes |
| The app should comply with privacy laws and regulations. | Yes |
| Data Storage and Privacy | |
| System credential storage facilities need to be used to store sensitive data, such as PII, user credentials, or cryptographic keys. | Yes |
| No sensitive data should be stored outside of the app container or system credential storage facilities. | Yes |
| No sensitive data is written to application logs. | Yes |
| No sensitive data is shared with third parties unless it is a necessary part of the architecture. | Yes |
| The keyboard cache is disabled on text inputs that process sensitive data. | Yes |
| No sensitive data is exposed via IPC mechanisms. | Yes |
| No sensitive data, such as passwords or pins, is exposed through the user interface. | Yes |
| Cryptography | |
| The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption. | Yes |
| The app uses proven implementations of cryptographic primitives. | Yes |
| The app uses cryptographic primitives that are appropriate for the particular use case with parameters that adhere to industry best practices. | Yes |
| The app does not use cryptographic protocols or algorithms that are widely considered deprecated for security purposes. | Yes |
| The app doesn't re-use the same cryptographic key for multiple purposes. | Yes |
| All random values are generated using a sufficiently secure random number generator. | Yes |
| Authentication and Session Management | |

| | |
|---|-----|
| If the app provides users access to a remote service, some form of authentication, such as username/password authentication, is performed at the remote endpoint. | Yes |
| If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials. | Yes |
| If stateless token-based authentication is used, the server provides a token that has been signed using a secure algorithm. | Yes |
| The remote endpoint terminates the existing session when the user logs out. | Yes |
| A password policy exists and is enforced at the remote endpoint. | Yes |
| The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times. | Yes |
| Sessions are invalidated at the remote endpoint after a predefined period of inactivity and access tokens expire. | Yes |
| Network Communication | |
| Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app. | Yes |
| The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards. | Yes |
| The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted. | Yes |
| Platform Interaction | |
| The app only requests the minimum set of permissions necessary. | Yes |
| All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources. | Yes |
| The app does not export sensitive functionality via custom URL schemes, unless these mechanisms are properly protected. | Yes |
| The app does not export sensitive functionality through IPC facilities, unless these mechanisms are properly protected. | Yes |
| JavaScript is disabled in WebViews unless explicitly required. | Yes |
| WebViews are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled. | Yes |
| If native methods of the app are exposed to a WebView, verify that the WebView only renders JavaScript contained within the app package. | Yes |
| Object deserialization, if any, is implemented using safe serialization APIs. | Yes |
| Code Quality and Build Setting | |
| The app is signed and provisioned with a valid certificate, of which the private key is properly protected. | Yes |
| The app has been built in release mode, with settings appropriate for a release build (e.g. | Yes |

| | |
|---|-----|
| non-debuggable). | |
| Debugging symbols have been removed from native binaries. | Yes |
| Debugging code and developer assistance code (e.g. test code, backdoors, hidden settings) have been removed. The app does not log verbose errors or debugging messages. | Yes |
| All third party components used by the mobile app, such as libraries and frameworks, are identified, and checked for known vulnerabilities. | Yes |
| The app catches and handles possible exceptions. | Yes |
| Error handling logic in security controls denies access by default. | Yes |
| In unmanaged code, memory is allocated, freed and used securely. | Yes |
| Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated. | Yes |
| Select Controls from Open Web Application Security Project - Mobile Application Security Verification Standard 1.2 Level 2 | |
| Architecture, design, and threat modeling | |
| All app components are defined in terms of the business functions and/or security functions they provide. | Yes |
| A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures. | Yes |
| All security controls have a centralized implementation. | Yes |
| There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57. | Yes |
| A mechanism for enforcing updates of the mobile app exists. | Yes |
| Security is addressed within all parts of the software development lifecycle. | Yes |
| A responsible disclosure policy is in place and effectively applied. | Yes |
| Data Storage and Privacy | |
| No sensitive data is included in backups generated by the mobile operating system. | Yes |
| The app removes sensitive data from views when moved to the background. | Yes |
| The app does not hold sensitive data in memory longer than necessary, and memory is cleared explicitly after use. | Yes |
| The app educates the user about the types of personally identifiable information processed, as well as security best practices the user should follow in using the app. | Yes |
| No sensitive data should be stored locally on the mobile device. Instead, data should be retrieved from a remote endpoint when needed and only be kept in memory. | Yes |
| Authentication and Session Management | |
| A second factor of authentication exists at the remote endpoint and the 2FA requirement is consistently enforced. | Yes |
| Network Communication | |

| | |
|--|-----|
| The app either uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish connections with endpoints that offer a different certificate or key, even if signed by a trusted CA. | Yes |
| The app doesn't rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery. | Yes |
| The app only depends on up-to-date connectivity and security libraries. | Yes |
| Platform Interaction | |
| The app protects itself against screen overlay attacks. (Android only) | Yes |
| A WebView's cache, storage, and loaded resources (JavaScript, etc.) should be cleared before the WebView is destroyed. | Yes |
| Verify that the app prevents usage of custom third-party keyboards whenever sensitive data is entered (iOS only). | Yes |
| Select Controls from Open Web Application Security Project - Mobile Application Security Verification Standard 1.2 Resiliency against Reverse Engineering | |
| Impede Dynamic Analysis and Tampering | |
| The app detects, and responds to, the presence of a rooted or jailbroken device either by alerting the user or terminating the app. | Yes |
| The app prevents debugging and/or detects, and responds to, a debugger being attached. All available debugging protocols must be covered. | Yes |
| Obfuscation is applied to programmatic defenses, which in turn impede de-obfuscation via dynamic analysis. | Yes |
| Impede Comprehension | |
| All executable files and libraries belonging to the app are either encrypted on the file level and/or important code and data segments inside the executables are encrypted or packed. | |
| Trivial static analysis does not reveal important code or data. | Yes |
| Impede Eavesdropping | |
| As a defense in depth, next to having solid hardening of the communicating parties, application level payload encryption can be applied to further impede eavesdropping. | Yes |

Attachment C: Supporting Documentation

Open Web Application Security Project - Mobile Application Security Verification Standard 1.2 Level 1

Note: Both the Android and iOS versions of the Voatz Mobile App were developed with these very security properties in mind, along with numerous other security considerations from OWASP. A “Pass” indication for the Android and iOS apps indicate that the “requirement is applicable to the mobile app and implemented according to best practices.”

Architecture, design, and threat modeling

All app components are identified and known to be needed.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH1: **Pass**, June 19, 2020.

Security controls are never enforced only on the client side, but on the respective remote endpoints.

1. The following is from a voter onboarding module on the server.

```
if(threatDetected){
    val threatName = threatNameOpt.get
    val err = s"Voatz has detected an unexpected issue with your device. Please quit the
    app and try again, or contact Voatz."
    log.error(s"Threat $threatName detected on device $deviceIdFromReq")
}
```

2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-ARCH-2: **Pass**, June 19, 2020.

A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH3: **Pass**, June 19, 2020.

Data considered sensitive in the context of the mobile app is clearly identified.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-4: **Pass**, June 19, 2020.

2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-4: **Pass**, June 19, 2020.

The app should comply with privacy laws and regulations.

1. Walker, Michael and Owens, Wendy. Test Report for Test and Evaluation of the Voatz Remote Accessible Ballot Delivery, Marking and Return (RABDMR) System, Pro V&V (July 17, 2020).

Req. 3.2.a.1.C: The voting system shall ensure that any notification required under this paragraph preserves the privacy of the voter and the confidentiality of the ballot. (Pass)

2. Req. 3.2.7: HAVA Section 301 (a)(4) states that the voting system shall provide alternative language accessibility pursuant to the requirements of Section 203 of the Voting Rights Act of 1965 (42 U.S.C. 1973aa-1a). Ideally every voter would be able to vote independently and privately, regardless of language. As a practical matter, alternative language access is mandated under the Voting Rights Act of 1975, subject to certain thresholds (e.g., if the language group exceeds 5% of the voting age population). Thus, election officials must ensure that the voting system they deploy is capable of handling the languages meeting the legal threshold within their districts. (Pass)

Data Storage and Privacy

System credential storage facilities need to be used to store sensitive data, such as PII, user credentials, or cryptographic keys.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE1: **Pass**, June 19, 2020.

Comments: Such sensitive data is stored in a TPM (Trusted Platform Module) or dedicated app storage.

No sensitive data should be stored outside of the app container or system credential storage facilities.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE2: **Pass**, June 19, 2020.

No sensitive data is written to application logs.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE3: **Pass**, June 19, 2020.

2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE3: **Pass**, June 19, 2020.

No sensitive data is shared with third parties unless it is a necessary part of the architecture.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE4: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE4: **Pass**, June 19, 2020.

The keyboard cache is disabled on text inputs that process sensitive data.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE5: **Pass**, June 19, 2020.

No sensitive data is exposed via IPC mechanisms.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE6: **Pass**, June 19, 2020.

No sensitive data, such as passwords or pins, is exposed through the user interface.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE7: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE7: **Pass**, June 19, 2020.

Cryptography

The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CRYPTO1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CRYPTO1: **Pass**, June 19, 2020.

Comments: Symmetric cipher keys (i.e., AES) are ephemeral and created with the server in a secure ECDH-based key agreement protocol that utilizes a NIST-approved elliptic curve.

The app uses proven implementations of cryptographic primitives.

1. Policy A.10: Cryptography, Section 3.1: All Voatz, Inc. encryption shall be done using approved cryptographic modules.
2. Policy A.10: Cryptography, Section 3.2: The use of proprietary encryption algorithms is not allowed for any purpose, unless reviewed by qualified experts and approved by Voatz, Inc. IT Ops.

The app uses cryptographic primitives that are appropriate for the particular use case with parameters that adhere to industry best practices.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CRYPTO3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CRYPTO3: **Pass**, June 19, 2020.

The app does not use cryptographic protocols or algorithms that are widely considered deprecated for security purposes.

1. Policy A.10: Cryptography, Section 3.1: All Voatz, Inc. encryption shall be done using approved cryptographic modules.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-CRYPTO4: **Pass**, June 19, 2020.

The app doesn't re-use the same cryptographic key for multiple purposes.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CRYPTO5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CRYPTO5: **Pass**, June 19, 2020.

All random values are generated using a sufficiently secure random number generator.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CRYPTO6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CRYPTO6: **Pass**, June 19, 2020.

Comments: All random numbers are generated using a cryptographically secure random number generator using device-collected entropy.

Authentication and Session Management

If the app provides users access to a remote service, some form of authentication, such as username/password authentication, is performed at the remote endpoint.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-AUTH1: **Pass**, June 19, 2020.
2. Authentication on the core server in an API routing module.

```
case EncryptedAuthenticateCustomer(request, ipAddress, hostName) => {
```

Comments: A user-defined PIN and random counter that is incremented on every successful login is used as part of the authentication.

```
val nextKeyErrorFut = UserModule.verifyPinAndPassword(customerId, apiRequest.pin,
    apiRequest.lastKey)
nextKeyErrorFut onComplete {
    case Success((Some(nextKey), None, None, deltaOpt)) => {
        val session = SessionAuthenticatorAsync.createSession(customerId)
```

If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-AUTH2: **Pass**, June 19, 2020.
2. Issuance of random session identification in an API router module.

```
val userHash = Cryptography.hash64(customerId.toString())
val randomKey = Cryptography.issueRandomKey(64)
val sessionCookie = Cryptography.hash64(randomKey + userHash)
val csrfToken = randomKey
val session = HttpApiSession(sessionCookie, csrfToken, customerId,
    System.currentTimeMillis)
Cache.createMobileSession(sessionCookie, session)
```

If stateless token-based authentication is used, the server provides a token that has been signed using a secure algorithm.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH3: **Pass**, June 19, 2020.

The remote endpoint terminates the existing session when the user logs out.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH4: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH4: **Pass**, June 19, 2020.

Comments: All sessions are created with a TTL (time to live) setting and such sessions are automatically cleared when the TTL expires, even if the remote logout endpoint is not explicitly called.

A password policy exists and is enforced at the remote endpoint.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH5: **Pass**, June 19, 2020.

The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH6: **Pass**, June 19, 2020.

Comments: After an unsuccessful login attempt, the user is throttled and cannot successfully login for at least 31 seconds. Throttle intervals increase at least 10 seconds for each additional unsuccessful login attempt. Less than 12 unsuccessful login attempts are allowed per hour.

Sessions are invalidated at the remote endpoint after a predefined period of inactivity and access tokens expire.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH7: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH7: **Pass**, June 19, 2020.

Comments: All sessions are created with a TTL (time to live) setting and such sessions are automatically cleared when the TTL setting expires. Additionally, Voatz iOS and Android apps implement in-app inactivity/idle timers that explicitly call logout on the remote endpoint when the idle timer expires.

Network Communication

Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application:
 - All web applications via the most secure version of SSL/TLS.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-NETWORK1: **Pass**, June 19, 2020.

The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK2: **Pass**, June 19, 2020.

Comments: Currently, TLS v. 1.2 is used.

The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK3: **Pass**, June 19, 2020.

Comments: Voatz iOS and Android apps use certificate pinning to verify the validity of remote endpoints. Additionally, both apps employ state of the art root/jailbreak detection, obfuscation, and code virtualization techniques to ensure that all communication with remote endpoints is secure.

Platform Interaction

The app only requests the minimum set of permissions necessary.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM1: **Pass**, June 19, 2020.

All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM2: **Pass**, June 19, 2020.

The app does not export sensitive functionality via custom URL schemes, unless these mechanisms are properly protected.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM3: **Pass**, June 19, 2020.

The app does not export sensitive functionality through IPC facilities, unless these mechanisms are properly protected.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM4: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM4: **Pass**, June 19, 2020.

JavaScript is disabled in WebViews unless explicitly required.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM5: **Pass**, June 19, 2020.

WebViews are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM6: **Pass**, June 19, 2020.

If native methods of the app are exposed to a WebView, verify that the WebView only renders JavaScript contained within the app package.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM7: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM7: **Pass**, June 19, 2020.

Object deserialization, if any, is implemented using safe serialization APIs.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-PLATFORM8: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-PLATFORM8: **Pass**, June 19, 2020.

Code Quality and Build Setting

The app is signed and provisioned with a valid certificate, of which the private key is properly protected.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE1: **Pass**, June 19, 2020.

Comments: The certificate is provided by Google Play Store and the Apple App Store for the Android and iOS versions of VMA, respectively.

The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE2: **Pass**, June 19, 2020.

Voatz mobile application binaries are submitted to the Apple App. store and Google Play store in release mode with obfuscation enabled and debugging disabled. Any attempt to run the app. with a debugger is blocked and results in an application crash.

Debugging symbols have been removed from native binaries.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE3: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE3: **Pass**, June 19, 2020.

Binaries submitted to the Apple App. store and Google Play store are obfuscated and do not contain debug symbols.

Debugging code and developer assistance code (e.g. test code, backdoors, hidden settings) have been removed. The app does not log verbose errors or debugging messages.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE4: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE4: **Pass**, June 19, 2020.

Voatz mobile apps use Google Crashlytics to log errors and exceptions. Such logs contain exception stack traces and do not reveal any user information.

All third party components used by the mobile app, such as libraries and frameworks, are identified, and checked for known vulnerabilities.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE5: **Pass**, June 19, 2020.

Voatz mobile app binaries are regularly tested using GuardSquare AppSweep to check for known vulnerabilities. This includes all libraries and frameworks used.

The app catches and handles possible exceptions.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE6: **Pass**, June 19, 2020.

Error handling logic in security controls denies access by default.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE7: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE7: **Pass**, June 19, 2020.

In unmanaged code, memory is allocated, freed and used securely.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE8: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE8: **Pass**, June 19, 2020.

Voatz mobile apps use managed code developed in Kotlin for Android and Swift for iOS.

Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-CODE9: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-CODE9: **Pass**, June 19, 2020.

Select Controls from Open Web Application Security Project - Mobile Application Security Verification Standard 1.2 Level 2

Architecture, design, and threat modeling

All app components are defined in terms of the business functions and/or security functions they provide.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH5: **Pass**, June 19, 2020.

A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH6: **Pass**, June 19, 2020.

All security controls have a centralized implementation.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH7: **N/A**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH7: **Pass**, June 19, 2020.

There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57.

1. Policy A.10: Acceptable Encryption Policy, Section 3.6: Key Management:
 - Key management should be planned which will include secure key generation, use, storage, distribution, recovery, and removal.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android and iOS, MSTG-ARCH8: **Pass**, June 19, 2020.

A mechanism for enforcing updates of the mobile app exists.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH9: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH9: **Pass**, June 19, 2020.

Security is addressed within all parts of the software development lifecycle.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH10: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH10: **Pass**, June 19, 2020.

A responsible disclosure policy is in place and effectively applied.

1. Policy A.16: Incident Response Policy
2. Policy A.18: Voatz, Inc. Information Security/Sensitivity, Section 3.2: Data Protection Standards: Incident Reporting
 - Any unauthorized disclosure or loss of Level 1 or Level 2 information must be reported to Voatz IT Ops.

Data Storage and Privacy

No sensitive data is included in backups generated by the mobile operating system.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE8: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE8: **Pass**, June 19, 2020.

Comments: On Android, backups are explicitly disabled by setting `android:allowBackup="false"` in the app's manifest. On iOS, all sensitive data is encrypted and no sensitive data is stored outside of the device's secure enclave to prevent any accidental leakage of information in plaintext via iCloud backups.

The app removes sensitive data from views when moved to the background.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE9: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE9: **Pass**, June 19, 2020.

Comments: Voatz iOS and Android apps call logout explicitly and redirect the user to the login view as soon as the app is moved to the background, thus invalidating the current user session.

The app does not hold sensitive data in memory longer than necessary, and memory is cleared explicitly after use.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE10: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE10: **Pass**, June 19, 2020.

Comments: Voatz iOS and Android apps explicitly zero out sensitive data from memory as soon as feasible, as seen from the code snippets below: The iOS app creates a wrapper around the Swift Data type to wipe out bytes on deallocation so secrets cannot be brought back to life by reverse engineering.

```
public class ClearableData {
    var value: Data

    deinit {
        value.deinitialize()
    }
    ....
    func replaceSubrange(_ subrange: Range<Data.Index>, with data: [UInt8]) {
        value.replaceSubrange(subrange, with: data)
    }
}
```

This is used to replace sensitive data with random, meaningless data.

```
var mask = ClearableData()
for counter in 0 ..< interval {
    tseed.replaceSubrange((tseed.count - 4) ..< tseed.count, with: pack(counter))
    mask.append(CC.digest(tseed, alg: digest))
}
```

```

    }
    mask.count = maskLength
    return mask

```

This is from the Android app API encryption module, which defines an extension to de-initialize byte arrays.

```

fun ByteArray?.deInitialize() {
    if (this != null && this.isNotEmpty()) {
        val size = this.size
        for (i in 0 until size) {
            this[i] = 0
        }
    }
}

```

This is used to replace sensitive data that is no longer processed by the app.

```

val decryptedBytes: ByteArray?
try {
    decryptedBytes = cipher.doFinal(data)
    val decryptedBytesToString = String(decryptedBytes)
    decryptedBytes.deInitialize()
}

```

The app educates the user about the types of personally identifiable information processed, as well as security best practices the user should follow in using the app.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-STORAGE12: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-STORAGE12: **Pass**, June 19, 2020.

No sensitive data should be stored locally on the mobile device. Instead, data should be retrieved from a remote endpoint when needed and only be kept in memory.

1. Policy AC.4: Mobile Device Encryption, Section 4.2 Cell phones and Tablets
Any sensitive Voatz, Inc. data stored on a cell phone or tablet must be saved to an encrypted file system using Voatz, Inc.-approved (in the future) software.
2. Some sensitive data is stored on the iPhone Keychain. The accessibility is set to “kSecAttrAccessibleWhenUnlockedThisDeviceOnly”, so they are only available when the device is unlocked, and will never move to a new device. Thus, after restoring from a backup of a different device, these items will not be present.

Authentication and Session Management

A second factor of authentication exists at the remote endpoint and the 2FA requirement is consistently enforced.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-AUTH9: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-AUTH9: **Pass**, June 19, 2020.

Network Communication

The app either uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish connections with endpoints that offer a different certificate or key, even if signed by a trusted CA.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK4: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK4: **Pass**, June 19, 2020.

The app doesn't rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK5: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK5: **Pass**, June 19, 2020.

The app only depends on up-to-date connectivity and security libraries.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-NETWORK6: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-NETWORK6: **Pass**, June 19, 2020.

Platform Interaction

The app protects itself against screen overlay attacks. (Android only)

1. The Toast Overlay attack on Android targets versions prior to Android 8. The Voatz Mobile App does not run on these older Android versions, so this attack does not apply in that case.

2. For versions 9 and above, the “Draw on Top” permission must be explicitly granted by the user in Settings in order to perform a screen overlay attack. The Voatz Mobile App for Android never requests for such a permission. Specifically, VMA for Android does not ask for the `android.permission.SYSTEM_ALERT_WINDOW` permission needed to draw on top. In general, it is considered best practice to disable this permission and only enable it, if absolutely necessary, for trusted apps.

A WebView's cache, storage, and loaded resources (JavaScript, etc.) should be cleared before the WebView is destroyed.

1. iOS: Not applicable. WKWebView in select components of VMA for iOS: Terms and Conditions, Resources URL, and WebUrl on candidates when desired by the client.
2. Android: Not applicable. WKWebView in select components of VMA for Android: Terms and Conditions, Resources URL, and WebUrl on candidates when desired by the client.

Comments: WebViews are only used for informational purposes to display official election authority/agency approved web resources (HTTPS only) where the user can find additional information about the election. Core application functionality does not rely on WebViews.

Verify that the app prevents usage of custom third-party keyboards whenever sensitive data is entered (iOS only).

1. Voatz Software Development Policy: Section 4.9 Security Vulnerabilities: Security vulnerabilities shall be identified, managed, and minimized by code fixes or configuration changes. Application security vulnerability scans and penetration tests are recommended to be based on the OWASP Top 10 Vulnerabilities. Example, OWASP A06:2021: Vulnerable and Outdated Components.
2. The following code from a user interface module prevents custom keyboards from being used in VMA.

```
func application(_ application: UIApplication, shouldAllowExtensionPointIdentifier
extensionPointIdentifier: UIApplication.ExtensionPointIdentifier) -> Bool {
    return extensionPointIdentifier != .keyboard
}
```

Select Controls from Open Web Application Security Project - Mobile Application Security Verification Standard 1.2 Resiliency against Reverse Engineering

Impede Dynamic Analysis and Tampering

The app detects, and responds to, the presence of a rooted or jailbroken device either by alerting the user or terminating the app.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-RESILIENCE1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-RESILIENCE1: **Pass**, June 19, 2020.

Comments: Voatz iOS and Android apps employ state of the art quantum secure root/jailbreak detection techniques. If the device is rooted or jailbroken, then the user will be unable to authenticate with the app., initiate identity verification, or receive a ballot.

The app prevents debugging and/or detects, and responds to, a debugger being attached. All available debugging protocols must be covered.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-RESILIENCE2: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-RESILIENCE2: **Pass**, June 19, 2020.

Comments: Voatz iOS and Android apps employ state of the art debugger detection, code obfuscation and virtualization techniques. Additionally, on Android, all devices must successfully complete Google license verification and Google's Play Integrity checks which explicitly disable debugging.

Obfuscation is applied to programmatic defenses, which in turn impede de-obfuscation via dynamic analysis.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-RESILIENCE9: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-RESILIENCE9: **Pass**, June 19, 2020.

Comments: Voatz Android and iOS apps employ state of the art code obfuscation and virtualization techniques which severely impede dynamic analysis. This process is non deterministic by design, resulting in randomized byte code on each successive build.

Impede Comprehension

All executable files and libraries belonging to the app are either encrypted on the file level and/or important code and data segments inside the executables are encrypted or packed. Trivial static analysis does not reveal important code or data.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-RESILIENCE11: **Pass**, June 19, 2020.

2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-RESILIENCE11: **Pass**, June 19, 2020.

Comments: All application files are encrypted using file level obfuscation to prevent static analysis. Additionally, specific code and data segments deemed critical for the secure functioning of the app are secured via another round of encryption. Application binaries submitted to Apple App. store and Google Play store are fully obfuscated with randomized/non deterministic code virtualization.

Impede Eavesdropping

As a defense in depth, next to having solid hardening of the communicating parties, application level payload encryption can be applied to further impede eavesdropping.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application: Encryption shall be implemented in:
 - App/client source code to secure voter data and selections (Android and iOS)
2. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application: Encryption shall be implemented in:
 - App/client communication with back end infrastructure. (Android and iOS)

Comments: All application data payloads are encrypted in flight using AES-GCM, with a non-repeating nonce. Ephemeral key exchange is facilitated by ECDH-based key agreement protocols employing NIST-approved elliptic curves.

3. Attachment D: Security Requirements for Databases: Table

| Attachment D: Select Controls from Department of Defense - Security Requirements Guide for Databases (Moderate Controls) | Yes/No |
|--|--------|
| The DBMS must limit the number of concurrent sessions to an organization-defined number per user for all accounts and/or account types. | Yes |
| The DBMS must protect against a user falsely repudiating having performed organization-defined actions. | Yes |
| The DBMS must be able to generate audit records when privileges/permissions are retrieved. | Yes |
| The DBMS must be able to generate audit records when unsuccessful attempts to retrieve privileges/permissions occur. | Yes |
| The DBMS must initiate session auditing upon startup. | Yes |
| The DBMS must produce audit records containing sufficient information to establish what type of events occurred. | Yes |
| The DBMS must produce audit records containing time stamps to establish when the events occurred. | Yes |
| The DBMS must produce audit records containing sufficient information to establish where the events occurred. | Yes |
| The DBMS must produce audit records containing sufficient information to establish the sources (origins) of the events. | Yes |
| The DBMS must produce audit records containing sufficient information to establish the outcome (success or failure) of the events. | Yes |
| The DBMS must produce audit records containing sufficient information to establish the identity of any user/subject or process associated with the event. | Yes |
| The DBMS must include additional, more detailed, organization-defined information in the audit records for audit events identified by type, location, or subject. | Yes |
| The DBMS must by default shut down upon audit failure, to include the unavailability of space for more audit log records; or must be configurable to shut down upon audit failure. | Yes |
| The DBMS must be configurable to overwrite audit log records, oldest first (First-In-First-Out - FIFO), in the event of unavailability of space for more audit log records. | Yes |
| The DBMS must use system clocks to generate time stamps for use in audit records and application data. | Yes |
| The audit information produced by the DBMS must be protected from unauthorized read access. | Yes |
| The audit information produced by the DBMS must be protected from unauthorized modification. | Yes |
| The audit information produced by the DBMS must be protected from unauthorized deletion. | Yes |

| | |
|---|-----|
| The DBMS must protect its audit features from unauthorized access. | Yes |
| The DBMS must protect its audit configuration from unauthorized modification. | Yes |
| The DBMS must protect its audit features from unauthorized removal. | Yes |
| The DBMS must limit privileges to change software modules, to include stored procedures, functions and triggers, and links to software external to the DBMS. | Yes |
| The DBMS software installation account must be restricted to authorized users. | Yes |
| Database software, including DBMS configuration files, must be stored in dedicated directories, or DASD pools, separate from the host OS and other applications. | Yes |
| Database objects (including but not limited to tables, indexes, storage, stored procedures, functions, triggers, links to software external to the DBMS, etc.) must be owned by database/DBMS principals authorized for ownership. | Yes |
| The role(s)/group(s) used to modify database structure (including but not necessarily limited to tables, indexes, storage, etc.) and logic modules (stored procedures, functions, triggers, links to software external to the DBMS, etc.) must be restricted to authorized users. | Yes |
| Default demonstration and sample databases, database objects, and applications must be removed. | Yes |
| Unused database components, DBMS software, and database objects must be removed. | Yes |
| Unused database components that are integrated in the DBMS and cannot be uninstalled must be disabled. | Yes |
| Access to external executables must be disabled or restricted. | Yes |
| The DBMS must uniquely identify and authenticate organizational users (or processes acting on behalf of organizational users). | Yes |
| If passwords are used for authentication, the DBMS must store only hashed, salted representations of passwords. | Yes |
| If passwords are used for authentication, the DBMS must transmit only encrypted representations of passwords. | Yes |
| The DBMS must obscure feedback of authentication information during the authentication process to protect the information from possible exploitation/use by unauthorized individuals. | Yes |
| The DBMS must use NIST FIPS 140-2 validated cryptographic modules for cryptographic operations. | Yes |
| The DBMS must separate user functionality (including user interface services) from database management functionality. | Yes |
| The DBMS must invalidate session identifiers upon user logout or other session termination. | Yes |
| The DBMS must recognize only system-generated session identifiers. | Yes |
| The DBMS must maintain the authenticity of communications sessions by guarding against man-in-the-middle attacks that guess at Session ID values. | Yes |
| The DBMS must fail to a secure state if system initialization fails, shutdown fails, or aborts fail. | Yes |
| In the event of a system failure, the DBMS must preserve any information necessary to determine cause of failure and any information necessary to return to operations with least | Yes |

| | |
|---|-----|
| disruption to mission processes. | |
| The DBMS must protect the confidentiality and integrity of all information at rest. | Yes |
| The DBMS must isolate security functions from non-security functions. | Yes |

Attachment D: Supporting Documentation

The DBMS must limit the number of concurrent sessions to an organization-defined number per user for all accounts and/or account types.

1. N/A
2. N/A

Comments: N/A: Voatz application logic limits a single user to one active session at any given time. Therefore, it is not possible to have multiple concurrent sessions and any attempt to establish a new session invalidates the previous session.

The DBMS must protect against a user falsely repudiating having performed organization-defined actions.

1. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:
All security-related events on critical or sensitive systems must be logged and audit trails saved.
2. Policy A.13: Network Security Policy, Section 5.2 Privilege Access Control:
 - Privileged access to prevent configuration changes should be restricted to authorized personnel only.
 - Access control shall be used to provide separate authentication, authorization, and accounting services for network-based access.
 - A Privileged Access Management solution can control credentials accessing the device and commands that can be executed when a session is initiated, providing a complete audit of both commands and sessions.

The DBMS must be able to generate audit records when privileges/permissions are retrieved.

1. The following is an example of an audit log entry generated when a user access level is retrieved to list signups in a certain election.

```
"txnId" : NumberLong(948126), "txnType" : "accesslevel", "txnStatus" : "SUCCESS",
"auditSourceId" : "800000040", "timestamp" : "2023-11-05T14:50:01.045450Z", "ipAddress" :
"172.58.21.118", "action" : "GETSIGNUPS", "userId" : 900000199, "channel": "VAP"
```

- The following is an example of an audit log entry generated when a user access level is retrieved to list affidavits in a certain election.

```
"txnId" : NumberLong(948393), "txnType" : "accesslevel", "txnStatus" : "SUCCESS",
"auditSourceId" : "800000053", "timestamp" : "2023-11-05T23:31:58.654373Z", "ipAddress" :
"68.110.100.196", "action" : "GETAFFIDAVITS", "userId" : 900000810, "channel" : "VAP"
```

Comments: An audit log entry is written to the database whenever an access level is retrieved to determine whether the user has the necessary privileges to execute a certain action for e.g. get list of signups, get list of affidavits, etc.

The DBMS must be able to generate audit records when unsuccessful attempts to retrieve privileges/permissions occur.

- The following is an example of an audit log entry generated when an attempt to perform an action that the user does not have permissions for is detected.

```
"txnId" : NumberLong(943250), "txnType" : "accesslevel", "txnStatus" : "FAILURE",
"auditSourceId" : "800000043", "timestamp" : "2023-10-15T13:17:09.036230Z", "ipAddress" :
"94.59.239.90", "action" : "GETSIGNUPS", "userId" : 900000190, "channel" : "VAP"
```

- An audit log entry is generated whenever an unsuccessful attempt to retrieve permissions is detected.

The DBMS must initiate session auditing upon startup.

- Voatz servers employ event sourcing to automatically initiate auditing on startup where log entries corresponding to each event type are logged.
- Along with the event source, Voatz servers initiate the event channel, event timestamp, and other metadata relevant for the specific transaction upon startup.

The DBMS must produce audit records containing sufficient information to establish what type of events occurred.

- The following is an example of an audit log entry indicating authentication (login/logout) requests from a certain Voatz Mobile App user.

```
"txnId" : NumberLong(813767), "txnType" : "login", "txnStatus" : "SUCCESS", "userId" :
1730153853, "channel" : "VMA", "timestamp" : "2023-11-05T19:51:04.596Z", "ipAddress" :
"198.241.2.100", "deviceId" : "ios-EE2B7626-EB6D..."
```

```
"txnId" : NumberLong(813766), "txnType" : "logout", "txnStatus" : "SUCCESS", "userId" :
1699377507, "channel" : "VMA", "timestamp" : "2023-11-05T19:29:50.347Z", "ipAddress" :
"71.223.122.232", "deviceId" : "ios-C1581EB3-89B3-..."
```


- The following is an example of an audit log entry indicating creation of a ballot (event) for a certain election.

```
"txnId" : NumberLong(951029), "txnType" : "event", "txnStatus" : "SUCCESS", "auditSourceId" :
"500014250", "timestamp" : "2023-11-05T06:37:01.117621Z", "ipAddress" : "73.219.188.197",
"action" : "CREATE", "userId" : 900000001, "channel" : "VAP"
```

Comments: Audit log entries use the transaction type attribute (txnType) to clearly indicate the type of audit event being logged.

The DBMS must produce audit records containing time stamps to establish when the events occurred.

- The following is an example of an audit log entry generated when a new Voatz Mobile App user is created.

```
txnId" : NumberLong(948186), "txnType" : "customer", "txnStatus" : "SUCCESS", "auditSourceId" :
"1689187950", "timestamp" : "2023-11-05T01:13:24.267Z", "ipAddress" : "172.58.21.118",
"action" : "CREATE", "channel": "VMA"
```

- The following is an example of an audit log entry generated when a user attempts to login to the Voatz Admin Portal.

```
"txnId" : NumberLong(947431), "txnType" : "orguser", "txnStatus" : "SUCCESS", "auditSourceId" :
"800000042", "timestamp" : "2023-11-05T01:17:15.667266Z", "ipAddress" : "135.84.57.144",
"action" : "LOGIN", "userId" : 900000171, "channel": "VAP"
```

Comments: All audit log entries contain a timestamp with an offset from UTC/Greenwich in the ISO-8601 calendar system.

The DBMS must produce audit records containing sufficient information to establish where the events occurred.

- The following is an example of an audit log entry generated on an attempt to verify a user for an election via Voatz Mobile App. (VMA).

```
"txnId" : NumberLong(954201), "txnType" : "orgidv", "txnStatus" : "Success", "authData" : [ {
"key" : "oidvSource", "value" : "datastore" }, { "key" : "oidvReferenceId", "value" :
"oidv_100080_800000073_1636333827" }, { "key" : "Text", "value" : "14802485495" } ], "userId" :
1645667116, "channel" : "VMA", "timestamp" : "2023-11-05T20:39:08.178284Z", "ipAddress" :
"135.84.57.36", "deviceId" : "ios-6D83B064-E97B..."
```

- The following is an example of an audit log entry generated when a user attempts to login to Voatz Admin Portal (VAP).

```
"txnId" : NumberLong(947431), "txnType" : "orguser", "txnStatus" : "SUCCESS", "auditSourceId" :
"800000023", "timestamp" : "2023-11-04T01:17:15.667266Z", "ipAddress" : "135.84.57.144",
"action" : "LOGIN", "userId" : 900000451, "channel": "VAP"
```

Comments: All audit log entries contain a channel attribute that clearly indicates where the log event occurred.

The DBMS must produce audit records containing sufficient information to establish the sources (origins) of the events.

1. The following is an example of audit log entry generated when a new Voatz Mobile App user is created.

```
txnid" : NumberLong(948186), "txnType" : "customer", "txnStatus" : "SUCCESS",
"auditSourceId" : "1689187950", "timestamp" : "2023-11-05T01:13:24.267Z", "ipAddress" :
"172.58.21.118", "action" : "CREATE", "channel": "VMA"
```

2. The following is an example of an audit log entry that was generated.

```
"txnid" : NumberLong(947431), "txnType" : "orguser", "txnStatus" : "SUCCESS", "auditSourceId"
: "800000042", "timestamp" : "2023-11-05T01:17:15.667266Z", "ipAddress" : "135.84.57.144",
"action" : "LOGIN", "userId" : 900000171, "channel": "VAP"
```

Comments: All audit log entries contain an audit source id (auditSourceId) attribute which together with the transaction type and channel uniquely identifies the origin of the audit log event.

The DBMS must produce audit records containing sufficient information to establish the outcome (success or failure) of the events.

1. The following is an example of an audit log entry generated during an attempt to verify a user's identity via the Voatz Mobile App.

```
"txnid" : NumberLong(954210), "txnType" : "basicidoidv", "txnStatus" : "PENDING", "authData"
: [ { "key" : "idvReferenceId", "value" : "Auto" }, { "key" : "idvSource", "value" : "Blockscore" }, {
"key" : "voatzIdvStatus", "value" : "SUCCESS" }, { "key" : "extAPIIdvStatus", "value" : "valid" }, {
"key" : "voatzIdvTimestamp", "value" : "2023-11-05T22:25:43.512Z" }, { "key" :
"extAPIIdvTimestamp", "value" : "2023-11-05T22:25:43.512Z" }, { "key" :
"voatzIdvFailureReason", "value" : "" }, { "key" : "extAPIIdvFailureReason", "value" : "" }, { "key" :
"idoReferenceId", "value" : "5cede519-9472-462b-a9e2-f2e7adca96e4" }, { "key" : "idoSource",
"value" : "VOATZ" }, { "key" : "voatzIdoStatus", "value" : "PENDING" }, { "key" : "extAPIIdoStatus",
"value" : "" }, { "key" : "voatzIdoTimestamp", "value" : "2023-11-05T22:25:43.880Z" } ]
```

2. All audit log entries contain a transaction status attribute which, together with the authData attribute (only relevant for certain transaction types), clearly establishes the outcome of the event generating the audit log entry.

The DBMS must produce audit records containing sufficient information to establish the identity of any user/subject or process associated with the event.

1. The following is an audit log entry generated when a Voatz Mobile App user attempts to log in.

```
"txnId" : NumberLong(954213), "txnType" : "login", "txnStatus" : "SUCCESS", "userId" :
1697138693, "channel" : "VMA", "timestamp" : "2023-11-05T22:40:10.460Z", "ipAddress" :
"24.23.61.46", "deviceId" : "ios-6D83B064-E9.."
```

- The following is an audit log entry generated when a Voatz Mobile App user attempts to log out.

```
"txnId" : NumberLong(813766), "txnType" : "logout", "txnStatus" : "SUCCESS", "userId" :
1699377507, "channel" : "VMA", "timestamp" : "2023-11-05T19:29:50.347Z", "ipAddress" :
"71.223.122.232", "deviceId" : "ios-C1581EB3-89B3-..."
```

Comments: Audit log entries contain a user id (userId) attribute that uniquely identifies the subject associated with the log entry event.

The DBMS must include additional, more detailed, organization-defined information in the audit records for audit events identified by type, location, or subject.

- The following is an example of an audit log entry generated on an attempt to verify a user for an election via Voatz Mobile App (VMA).

```
"txnId" : NumberLong(954201), "txnType" : "orgidv", "txnStatus" : "Success", "authData" : [ {
"key" : "oidvSource", "value" : "datastore" }, { "key" : "oidvReferenceId", "value" :
"oidv_100080_800000073_1636333827" }, { "key" : "Text", "value" : "14802485495" } ],
"userId" : 1645667116, "channel" : "VMA", "timestamp" : "2023-11-05T20:39:08.178284Z",
"ipAddress" : "135.84.57.36", "deviceId" : "ios-6D83B064-E97B..."
```

- Audit log entries contain an organization specific authorization data (authData) attribute which allows an organization to define/configure its own audit attributes. In the example above, the datastore reference **oidv_100080_800000073_1636333827** defines a mandatory field, **Text**, which must be supplied by and associated with the verifying entity as part of their verification workflow. Additionally, attributes such as location, etc. can be easily configured to be logged as part of audit log entries.

The DBMS must by default shut down upon audit failure, to include the unavailability of space for more audit log records; or must be configurable to shut down upon audit failure.

- N/A
- N/A

Comments: N/A: Voatz databases cannot be shut down due to the global nature of the application. Log entries are periodically archived and such archives are encrypted via cryptographically secure random secrets. Due to the nature of the archival process on dedicated backup environments, unavailability of disk space in production environments is never an issue.

The DBMS must be configurable to overwrite audit log records, oldest first (First-In-First-Out - FIFO), in the event of unavailability of space for more audit log records.

1. N/A
2. N/A

Comments: N/A: Voatz audit log entries are never overwritten or updated. These log entries are generated via append only event sourcing which eliminates any possibility of mutable updates. Log entries are periodically archived and such archives are encrypted via cryptographically secure random secrets. Due to the nature of the archival process on dedicated backup environments, unavailability of disk space in production environments is never an issue.

The DBMS must use system clocks to generate timestamps for use in audit records and application data.

1. The following is from a database module, which is an example of a timestamp being put on a record.

```
val createTsSet = set("createTs", OffsetDateTime.now.toString())
```

2. The following is the OffsetDateTime module used, from the aforementioned database module.

```
import java.time.OffsetDateTime
```

Comments: All audit log entries contain a timestamp with an offset from UTC/Greenwich in the ISO-8601 calendar system.

The audit information produced by the DBMS must be protected from unauthorized read access.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.
2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows the principle of least privilege when granting access to information and application system functionality.

Additionally, access to information/data deemed sensitive or confidential is limited, based on a need to know.

Comments: Audit log entries are not available via any public APIs and are only available in read only mode via local micro services with strict access control.

The audit information produced by the DBMS must be protected from unauthorized modification.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.

2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows the principle of least privilege when granting access to information and application system functionality.

Additionally, access to information/data deemed sensitive or confidential is limited, based on a need to know.

Comments: Audit log entries cannot be created/updated/deleted via any public APIs and are only available in read only mode via local micro services with strict access control.

The audit information produced by the DBMS must be protected from unauthorized deletion.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.
2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows the principle of least privilege when granting access to information and application system functionality.

Additionally, access to information/data deemed sensitive or confidential is limited based on a need to know.

Comments: Audit log entries cannot be created/updated/deleted via any public APIs and are only available in read only mode via local micro services with strict access control.

The DBMS must protect its audit features from unauthorized access.

1. Policy AC10: Database Credentials Policy, Section 4.1: General
Database credentials must not be stored in a location that can be accessed by unauthorized parties.
2. Policy AC10: Database Credentials Policy, Section 4.2: Storage of Database Credentials
Database credentials shall be stored only in an encrypted manner.

Comments: Audit logging features are strictly controlled by Voatz server modules for transactions that are deemed relevant for auditing purposes. Audit log features cannot be created, updated, or deleted via any public APIs.

The DBMS must protect its audit configuration from unauthorized modification.

1. Policy AC10: Database Credentials Policy, Section 4.1: General
Database credentials must not be stored in a location that can be accessed by unauthorized parties.
2. Policy AC10: Database Credentials Policy, Section 4.2: Storage of Database Credentials
Database credentials shall be stored only in an encrypted manner.

Comments: Audit logging configuration is strictly controlled by Voatz server modules. Audit log configuration cannot be created, updated, or deleted via any public APIs.

The DBMS must protect its audit features from unauthorized removal.

1. Policy AC10: Database Credentials Policy, Section 4.1: General
Database credentials must not be stored in a location that can be accessed by unauthorized parties.
2. Policy AC10: Database Credentials Policy, Section 4.2: Storage of Database Credentials
Database credentials shall be stored only in an encrypted manner.

Comments: Audit logging features are strictly controlled by Voatz server modules for transactions that are deemed relevant for auditing purposes. Audit log features cannot be created, updated, or deleted via any public APIs.

The DBMS must limit privileges to change software modules, to include stored procedures, functions and triggers, and links to software external to the DBMS.

1. Policy A.9: Access Control Policy, Section 4.2 User Access Provisioning:
Access to critical corporate and election systems shall be managed and controlled by the Voatz IT Ops Super User(s).
2. Policy A.9: Access Control Policy, Section 6.1 Information Access Restriction
Read-only access is granted to users based on their roles, while functions that allow users to create, modify, or delete information is restricted to designated super users/admins. (E.g., Only admins can create or delete code repositories, while developers are restricted to adding new code, and editing existing code. Only a super admin or designated alternate with super admin access can push code to the live/production environment.

Comments: Any software modules required for the DBMS to function can only be installed and updated by a super user. There is no public access to such features and private access is limited and controlled strictly by role.

The DBMS software installation account must be restricted to authorized users.

1. Policy A.9: Access Control Policy, Section 4.2 User Access Provisioning:
Access to critical corporate and election systems shall be managed and controlled by the Voatz IT Ops Super User(s).
2. Policy A.9: Access Control Policy, Section 6.1 Information Access Restriction
Only a super admin or designated alternate with super admin access can push code to the live/production environment.

Comments: Any software modules required for the DBMS to function can only be installed and updated by a super user. There is no public access to such features and private access is limited and controlled strictly by role.

Database software, including DBMS configuration files, must be stored in dedicated directories, or DASD pools, separate from the host OS and other applications.

1. Policy A.9: Access Control Policy, Section 6.5 - Access Control to Program Source Code
 - Source code shall be managed using on-premise repositories that segregate the code into the major parts of the service - Frontend (mobile apps, web application), Backend (server, database, blockchain).
2. Policy OM.2: Information Security Management System (ISMS) Policy, Section 4.0: General ISMS Policy Statements:
 The Voatz Mobile Platform hosts political and non-political elections and is committed to protecting its data and that of its customers. To achieve this, the company has implemented security and privacy protections throughout its corporate and elections infrastructure. Voatz ISMS covers the following components of the Voatz Mobile Elections Platform:
 - Corporate Infrastructure which includes the employees involved with and systems used create, deploy, maintain and store the source code and proprietary information pertaining to Voatz mobile apps, web based portals, and other Voatz products
 - Elections Infrastructure which consists of software applications that support elections operations including, client applications (ios, android apps), core server, web servers, database, and blockchain.

Comments: Voatz servers employ separate partitions for host OS and DBMS related files. As seen from the snapshot below, all DBMS related features are installed under /var, separate from the OS kernel under /boot and all other OS files and applications under /.

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/sda2 | 30G | 23G | 7.0G | 77% | / |
| /dev/sda1 | 497M | 106M | 391M | 22% | /boot |
| /dev/sdb1 | 252G | 4G | 248G | 2% | /var |

Database objects (including but not limited to tables, indexes, storage, stored procedures, functions, triggers, links to software external to the DBMS, etc.) must be owned by database/DBMS principals authorized for ownership.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
 Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.
2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
 Voatz follows the principle of least privilege when granting access to information and application system functionality.

Comments: Voatz application logic uses roles and granular, feature-based access levels to limit access to the DBMS. User roles are clearly distinguished from Admin roles and any APIs consumed by user roles never provide ownership to DBMS objects.

The role(s)/group(s) used to modify database structure (including but not necessarily limited to tables, indexes, storage, etc.) and logic modules (stored procedures, functions, triggers, links to software external to the DBMS, etc.) must be restricted to authorized users.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.
2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows the principle of least privilege when granting access to information and application system functionality.

Comments: Voatz application logic uses roles and granular, feature-based access levels to limit access to the DBMS. User roles are clearly distinguished from Admin roles and any APIs consumed by user roles cannot access any features that modify the DBMS structure.

Default demonstration and sample databases, database objects, and applications must be removed.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-1: **Pass**, June 19, 2020.

Comments: No demo or sample databases and objects are used in any Voatz environment.

Unused database components, DBMS software, and database objects must be removed.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-1: **Pass**, June 19, 2020.

Unused database components that are integrated in the DBMS and cannot be uninstalled must be disabled.

1. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – Android, MSTG-ARCH-1: **Pass**, June 19, 2020.
2. OWASP Mobile Application Security Checklist Evaluation, Security Requirements – iOS, MSTG-ARCH-1: **Pass**, June 19, 2020.

Access to external executables must be disabled or restricted.

1. Policy AC.10: Database Credentials Policy, Section 1.0: Purpose

Systems running on Voatz, Inc.'s managed networks often require the use of internal databases. In order to access any internal databases, a system must authenticate to the database by presenting acceptable credentials. Database usernames and passwords (i.e., database credentials) must be securely stored and retrieved.

2. Policy AC.10: Database Credentials Policy, Section 4.1 General:

In order to maintain the security of Voatz, Inc.'s internal databases, access by software programs must be granted only when authenticated. The credentials used for this authentication must not reside in the program's source code in clear text. Database credentials must not be stored in a location that can be accessed by unauthorized parties.

The DBMS must uniquely identify and authenticate organizational users (or processes acting on behalf of organizational users).

1. Policy AC.10: Database Credentials Policy, Section 1.0: Purpose

Systems running on Voatz, Inc.'s managed networks often require the use of internal databases. In order to access any internal databases, a system must authenticate to the database by presenting acceptable credentials. Database usernames and passwords (i.e., database credentials) must be securely stored and retrieved.

2. Policy AC.10: Database Credentials Policy, Section 4.1 General:

In order to maintain the security of Voatz, Inc.'s internal databases, access by software programs must be granted only when authenticated. The credentials used for this authentication must not reside in the program's source code in clear text. Database credentials must not be stored in a location that can be accessed by unauthorized parties.

Comments: Each organization created in Voatz databases controls its own set of organization users, where a new organization user must be created with an existing organizationId, and can only access resources available to that organization. Voatz application logic uses roles and granular, feature-based access levels to ensure that an organization user will never be able to access any resources outside of their organization.

If passwords are used for authentication, the DBMS must store only hashed, salted representations of passwords.

1. Policy A.10: Acceptable Encryption Policy, Section 3.5: User Passwords

- Where applicable, user passwords shall be stored in a secure manner by using HMAC with hash.
- All user passwords should be salted on a per secret basis.

2. Policy AC10: Database Credentials Policy, Section 4.2: Storage of Database Credentials
Passwords or passphrases used to access a database must adhere to the Password Policy.

Comments: All user credentials are hashed using slow hashing primitives such as B-Crypt.

If passwords are used for authentication, the DBMS must transmit only encrypted representations of passwords.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application
All private keys for encryption must be password protected and not stored in the clear on systems.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application
Encryption shall be implemented in:
 - During sensitive data transmission over public networks

Comments: Any data transmitted by Voatz applications is encrypted via AES GCM with non-repeating nonce. Specifically, any credentials used for authenticating against databases are never transmitted in cleartext.

The DBMS must obscure feedback of authentication information during the authentication process to protect the information from possible exploitation/use by unauthorized individuals.

1. Policy A.9: Access Control Policy, Section 6.2 - Secure Log-on Procedures
Where required by the access control policy, access to systems and applications shall be controlled by a secure log-on procedure.
2. Policy A.10: Cryptography: Section 3.1: Encryption Method and Application:
Encryption shall be implemented in:
 - All web applications via the most secure version of SSL/TLS,
 - Device/file encryption i.e. partial disk encryption and full disk encryption,
 - During data transmission between and among infrastructure components,
 - During sensitive data transmission over public networks,
 - storage on disk, particularly secrets and user passwords.

The DBMS must use NIST FIPS 140-2 validated cryptographic modules for cryptographic operations.

1. Policy A.10: Cryptography, Section 3.6: Key Management:
 - Key generation must be seeded from an industry standard random number generator.
2. Policy A.10: Cryptography, Section 3.1: All Voatz, Inc. encryption shall be done using approved cryptographic modules.

Comments: Sample of NIST FIPS validated modules used for encryption at rest:

```
SHOW GLOBAL VARIABLES LIKE 'have_openssl';
have_openssl    YES
SHOW GLOBAL VARIABLES LIKE 'version_ssl_library';
version_ssl_library    OpenSSL 1.x.y-fips ...
```


The DBMS must separate user functionality (including user interface services) from database management functionality.

1. Policy A.9: Access Control Policy, Section 3.0: This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.
 - Least Privilege – the default approach taken shall be to assume that access is not required, rather than to assume that it is. In particular, all access to election infrastructure and election data shall be limited to those individuals that have a business or legal need-to-know;
 - Need to Know – access is only granted to the information required to perform a role, and no more. The default setting of the access control system has to be “deny-all”;
 - Need to Use – users will only be able to access physical and logical facilities required for their role.
2. Policy AC9: Server Security, Section 4.2 General Configuration and Backup Guidelines: Always use the least privilege to perform a function. Do not use root/admin accounts when a non-privileged account is sufficient.

Comments: Organization users can only access DBMS through well-defined APIs subject to roles and granular, feature-based access levels. These APIs do not offer any database management functionality.

The DBMS must invalidate session identifiers upon user logout or other session termination.

1. All logout requests explicitly delete the user's session :

```
case EncryptedLogoutCustomer(request, sessionCookieOpt, ipAddress) =>{
...
case Some(sessionCookie) => {
  SessionAuthenticator.deleteSession(sessionCookie.content) match {
```
2. Additionally, all sessions are created with a TTL (time to live) setting such that the session is automatically cleared on TTL expiry, even if logout is not explicitly called:

```
def createMobileSession(sessionCookie: String, session: HttpApiSession) = {
  Cache.set(s"mobilesession$sessionCookie", session, Some(Duration(900, TimeUnit.SECONDS)))
}
```

Comments: Session identifiers are generated using cryptographically secure random generators and are invalidated on logout. Additionally, sessions are created with a TTL (time to live) setting and are automatically cleared on TTL expiry.

The DBMS must recognize only system-generated session identifiers.

1. Session identifiers are generated via well-defined authentication APIs that create such identifiers using cryptographically secure random generators before persisting them.
2. It is not possible to create session identifiers outside of these APIs or directly in the DBMS.

The DBMS must maintain the authenticity of communications sessions by guarding against man-in-the-middle attacks that guess at Session ID values.

1. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application: Encryption shall be implemented in:
 - All web applications via the most secure version of SSL/TLS.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application: Encryption shall be implemented in:
 - During sensitive data transmission over public networks.

Comments: All communication to the servers and DBMS uses TLS v. 1.2. Additionally, all data in flight is encrypted with AES in Galois Counter Mode (GCM), with non-repeating nonce.

The DBMS must fail to a secure state if system initialization fails, shutdown fails, or aborts fail.

1. Voatz servers are architected with a clear separation of concerns.
2. This architecture ensures that the DBMS state is never compromised in the event of system failure.

In the event of a system failure, the DBMS must preserve any information necessary to determine cause of failure and any information necessary to return to operations with least disruption to mission processes.

1. Voatz servers employ event sourcing based on Akka persistence using databases separate from the core transactional databases.
2. This event sourcing makes it possible to restore to a previous stable state in the event of failure.

The DBMS must protect the confidentiality and integrity of all information at rest.

1. Policy A.10: Acceptable Encryption Policy, Section 3.3 Data Protection
 - Data at rest:
 - All Voatz, Inc. and customer data deemed a secret shall be stored in an encrypted manner.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1: Encryption Method and Application: Encryption shall be implemented in:
 - storage on disk, particularly secrets and user passwords.

Comments: All data at rest is protected using FIPS 140-2 cryptographic modules. Additionally, all archives and backups are also secured using such cryptographic modules in dedicated backup environments, separate from production environments.

The DBMS must isolate security functions from non-security functions.

1. Voatz servers are architected with a clear separation of concerns and employ separate databases for security functionality, such as access levels, authentication, and auditing.
2. For reasons of security, the aforementioned activities are never mixed with the core transactional databases.

4. Attachment E: Select Controls from the StateRAMP Moderate Baseline: Table

| Attachment E: Select Controls from the StateRAMP Moderate Baseline | Yes/No |
|--|--------|
| Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems). | Yes |
| Limit system access to the types of transactions and functions that authorized users are permitted to execute. | Yes |
| Monitor and control remote access sessions. | Yes |
| Employ cryptographic mechanisms to protect the confidentiality of remote access sessions. | Yes |
| Authorize wireless access prior to allowing such connections. | Yes |
| Protect wireless access using authentication and encryption. | Yes |
| Control connection of mobile devices. | Yes |
| Ensure that managers, system administrators, and users of organizational systems are made aware of the security risks associated with their activities and of the applicable policies, standards, and procedures related to the security of those systems. | Yes |
| Ensure that personnel are trained to carry out their assigned information security-related duties and responsibilities. | Yes |
| Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity. | Yes |
| Correlate audit record review, analysis, and reporting processes for investigation and response to indications of unlawful, unauthorized, suspicious, or unusual activity. | Yes |
| Establish and maintain baseline configurations and inventories of organizational systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles. | Yes |
| Establish and enforce security configuration settings for information technology products employed in organizational systems. | Yes |
| Define, document, approve, and enforce physical and logical access restrictions associated with changes to organizational systems. | Yes |
| Employ the principle of least functionality by configuring organizational systems to provide only essential capabilities. | Yes |
| Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services. | Yes |
| Apply deny-by-exception (blacklisting) policy to prevent the use of unauthorized software or deny-all, permit-by-exception (whitelisting) policy to allow the execution of authorized software. | Yes |
| Identify system users, processes acting on behalf of users, and devices. | Yes |

| | |
|--|-----|
| Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems. | Yes |
| Use multifactor authentication (MFA) for local and network access to privileged accounts and for network access to non-privileged accounts. | Yes |
| Store and transmit only cryptographically-protected passwords. | Yes |
| Establish an operational incident-handling capability for organizational systems that includes preparation, detection, analysis, containment, recovery, and user response activities. | Yes |
| Track, document, and report incidents to designated officials and/or authorities both internal and external to the organization. | Yes |
| Provide controls on the tools, techniques, mechanisms, and personnel used to conduct system maintenance. | Yes |
| Require multifactor authentication to establish nonlocal maintenance sessions via external network connections and terminate such connections when nonlocal maintenance is complete. | Yes |
| Control the use of removable media on system components. | Yes |
| Ensure that organizational systems containing sensitive data are protected during and after personnel actions such as terminations and transfers. | Yes |
| Limit physical access to organizational information systems, equipment, and the respective operating environments to authorized individuals. | Yes |
| Protect and monitor the physical facility and support infrastructure for organizational systems. | Yes |
| Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified. | Yes |
| Periodically assess the security controls in organizational systems to determine if the controls are effective in their application. | Yes |
| Monitor security controls on an ongoing basis to ensure the continued effectiveness of the controls. | Yes |
| Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems. | Yes |
| Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems. | Yes |
| Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks. | Yes |
| Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception). | Yes |
| Protect the authenticity of communications sessions. | Yes |
| Identify, report, and correct information and information system flaws in a timely manner. | Yes |
| Provide protection from malicious code at designated locations within organizational systems. | Yes |
| Monitor system security alerts and advisories and take action in response. | Yes |
| Update malicious code protection mechanisms when new releases are available. | Yes |

| | |
|---|-----|
| Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks. | Yes |
| Employ the principle of least privilege, including for specific security functions and privileged accounts. | Yes |
| Ensure that the actions of individual system users can be uniquely traced to those users so they can be held accountable for their actions. | Yes |
| Perform maintenance on organizational systems. | Yes |
| Check media containing diagnostic and test programs for malicious code before the media are used in organizational systems. | Yes |
| Prohibit the use of portable storage devices when such devices have no identifiable owner. | Yes |
| Screen individuals prior to authorizing access to organizational systems containing sensitive data. | Yes |
| Periodically assess the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, and individuals, resulting from the operation of organizational systems and the associated processing, storage, or transmission of sensitive data. | Yes |
| Develop and implement plans of action designed to correct deficiencies and reduce or eliminate vulnerabilities in organizational systems. | Yes |
| Implement cryptographic mechanisms to prevent unauthorized disclosure of sensitive information during transmission unless otherwise protected by alternative physical safeguards. | Yes |
| Employ FIPS-validated cryptography when used to protect the confidentiality of sensitive data. | Yes |
| Perform periodic scans of organizational systems and real-time scans of files from external sources as files are downloaded, opened, or executed. | Yes |
| Identify unauthorized use of organizational systems. | Yes |
| Separate the duties of individuals to reduce the risk of malevolent activity without collusion. | Yes |
| Prevent non-privileged users from executing privileged functions and capture the execution of such functions in audit logs. | Yes |
| Limit unsuccessful logon attempts. | Yes |
| Use session lock with pattern-hiding displays to prevent access and viewing of data after a period of inactivity. | Yes |
| Terminate (automatically) a user session after a defined condition. | Yes |
| Route remote access via managed access control points. | Yes |
| Verify and control/limit connections to and use of external systems. | Yes |
| Limit use of portable storage devices on external systems. | Yes |
| Provide security awareness training on recognizing and reporting potential indicators of insider threat. | Yes |
| Review and update logged events. | Yes |
| Alert in the event of an audit logging process failure. | Yes |
| Protect audit information and audit logging tools from unauthorized access, modification, and | Yes |

| | |
|--|-----|
| deletion. | |
| Limit management of audit logging functionality to a subset of privileged users. | Yes |
| Analyze the security impact of changes prior to implementation. | Yes |
| Control and monitor user-installed software. | Yes |
| Enforce a minimum password complexity and change of characters when new passwords are created. | Yes |
| Allow temporary password use for system logons with an immediate change to a permanent password. | Yes |
| Obscure feedback of authentication information. | Yes |
| Test the organizational incident response capability. | Yes |
| Supervise the maintenance activities of maintenance personnel without required access authorization. | Yes |
| Protect the confidentiality of backup sensitive data at storage locations. | Yes |
| Escort visitors and monitor visitor activity. | Yes |
| Remediate vulnerabilities in accordance with risk assessments. | Yes |
| Terminate network connections associated with communications sessions at the end of the sessions or after a defined period of inactivity. | Yes |
| Establish and manage cryptographic keys for cryptography employed in organizational systems. | Yes |
| Develop, document, and periodically update system security plans that describe system boundaries, system environments of operation, how security requirements are implemented, and the relationships with or connections to other systems. | Yes |

Attachment E: Supporting Documentation

Attachment E: Supporting Documentation

Limit system access to authorized users, processes acting on behalf of authorized users, and devices (including other systems).

1. Policy A.9: Access Control, Section 3.0:
Users shall only be provided with access to the network and network services that they have been specifically authorized to use.
2. ISO Control A.6.2.1: Mobile Device Management Policy, Section 3.1: Technology and Security Requirements:
When remote, devices may only access the corporate network and data through the Internet using a Secure Socket Layer (SSL) Virtual Private Network (VPN) connection.

Limit system access to the types of transactions and functions that authorized users are permitted to execute.

1. Policy A.9: Access Control, Section 3.0:
Users shall only be provided with access to the network and network services that they have been specifically authorized to use.
2. Policy AC.9: Server Security Policy, Section 4.1 Ownership and Responsibilities:
All servers deployed at Voatz, Inc. must be the responsibility of corporate IT Ops.

Monitor and control remote access sessions.

1. Policy AC.6: Remote Access Policy, Section 4.2: Requirements:
Secure remote access must be strictly controlled. Access must be approved by Voatz, Inc. IT Ops.
2. Policy A.12: Operations Security Policy, Section 8.0: Control of Operational Software:
Voatz monitors systems to detect unauthorized access and modification.

Employ cryptographic mechanisms to protect the confidentiality of remote access sessions.

1. Policy A.13: Network Security Policy, Section 5.9 Encrypt sensitive network traffic:
Appropriate encryption and authentication methods should be used for the transmission of sensitive data and remote access.
2. Policy AC.7: Virtual Private Network (VPN) Policy, Section 3.0: Policy:
The VPN should be controlled by a strong authentication method. Initial configuration will be done by Voatz, Inc. IT OPs and users will be responsible for selecting a strong password after initial login.

Authorize wireless access prior to allowing such connections.

1. Policy A.8: IT Asset Management Policy, Section 5.1: Asset Types:
The following asset types are subject to tagging and tracking:
 - Network appliances (e.g. firewalls, routers, switches, Uninterruptible Power Supplies (UPS), wireless access points, and storage)
2. Policy IC.7: Wireless Communication Policy, Section 4.4 Access to Wireless Networks:
Access shall be granted according to the user's role. Voatz, Inc. is responsible for maintaining the confidentiality, integrity, and availability of its resources and must manage the roles accordingly. Access to the Voatz, Inc. corporate network through any device not on site must use remote access authentication through the Voatz VPN.

Protect wireless access using authentication and encryption.

1. Policy A.13: Network Security Policy, Section 5.9 Encrypt sensitive network traffic:
Appropriate encryption and authentication methods should be used for the transmission of sensitive data and remote access.
2. Policy IC.7: Wireless Communication Policy, Section 4.1: General Requirements:
All wireless infrastructure devices that reside at Voatz, Inc. sites and connect to Voatz, Inc. networks, or provide access to information classified as Voatz, Inc. Confidential (requiring notification or otherwise) must:
 - Use Voatz, Inc. approved authentication protocols and infrastructure.
 - Use Voatz, Inc. approved encryption protocols.

Control connection of mobile devices.

1. Policy A.6.2.1: Mobile Device Management Policy, Section 3.1: Technology and Security Requirements
Only devices managed by IT or authorized by IT will be allowed to connect directly to the internal corporate network.
2. Policy A.6.2.1: Mobile Device Management Policy, Section 3.1: Technology and Security Requirements
When remote, devices may only access the corporate network and data through the Internet using a Secure Socket Layer (SSL) Virtual Private Network (VPN) connection.

Ensure that managers, system administrators, and users of organizational systems are made aware of the security risks associated with their activities and of the applicable policies, standards, and procedures related to the security of those systems.

1. From Voatz' Annual Security Awareness Training:
It's important to have security policies and processes in place in order to comply with our customers', partners', and election industry security and privacy requirements.

Additionally, security audits conducted by external entities often request documentation covering all areas of our mobile elections platform, so it's important for all employees and contractors to review these policies at least annually.

2. Example from Security Training Videos:

• **Training Videos**

ISO 27001 – <https://www.youtube.com/watch?v=io6w3Yw4q9w>

Comments: Such training has been implemented in accordance with ISO 27001 and StateRAMP certifications.

Ensure that personnel are trained to carry out their assigned information security-related duties and responsibilities.

1. Policy A.7: Personnel Security Policy, Section 4.4: Personnel Onboarding

The organization must upon personnel start:

- Provide onboarding guidance via handbook, intranet, etc.
- Obtain a signed NDA and prior inventions agreement.
- Obtain a signed employment agreement with security and privacy expectations.

2. Policy AC.7: Virtual Private Network (VPN) Policy:

Organization members understand when and how to use virtual private networks.

Create and retain system audit logs and records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful or unauthorized system activity.

1. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:

All security-related events on critical or sensitive systems must be logged and audit trails saved.

2. Policy A.11: Physical and Environmental Security Policy, Section 4.2 Physical Access Control:

Maintain physical access audit logs for data center(s) and/or sensitive facilities entry/exit points.

Correlate audit record review, analysis, and reporting processes for investigation and response to indications of unlawful, unauthorized, suspicious, or unusual activity.

1. Policy AC.9: Server Security Policy, Section 4.3 Monitoring:

Security-related events will be reported to Voatz, Inc. IT Ops, who will review logs and escalate incidents as necessary.

2. Policy A.16 Information Security Incident Management:

Voatz, Inc.'s response to situations that threaten the security or privacy of confidential information shall ensure events and activities associated with those events are properly identified, contained, investigated, and remedied.

Establish and maintain baseline configurations and inventories of organizational systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles.

1. Policy AC.9: Server Security Policy: Section 4.1: Ownership and Responsibilities:
Approved server configuration guides must be established and maintained by each operational group, based on business needs and approved by Voatz, Inc. IT Ops.
2. Policy A.8: Asset Management Policy: Section 1.0: Overview:
Asset management is the process of receiving, tagging, documenting, and disposing of equipment. It is important to maintain up to date inventory and asset controls to ensure that equipment locations and dispositions are understood. Lost or stolen equipment can contain sensitive data. Proper asset management procedures and protocols provide documentation that aid in recovery, replacement, criminal, and insurance activities.

Establish and enforce security configuration settings for information technology products employed in organizational systems.

1. Policy AC.9: Server Security Policy, Section 2.0: Scope:
Based on the type of servers, such as Cloud Managed Systems, Cloud Managed Instances or Physical Servers, the policy for handling patching, configuration, backups, and monitoring shall vary. Configuration and Backup of remote and on-premises servers managed by Voatz, Inc.
2. Policy A.10: Acceptable Encryption Policy, Section 3.1 Encryption Method and Application:
Encryption shall be implemented in:
 - Device/file encryption i.e. partial disk encryption and full disk encryption.

Define, document, approve, and enforce physical and logical access restrictions associated with changes to organizational systems.

1. Policy A.11: Physical and Environmental Security Policy, Section 4.2: Physical Access Control:
The organization will control entry/exit from the data center(s) and/or sensitive facilities using physical access control devices (e.g., keycard or keys) and/ or security guard(s).
2. Policy AC9: Server Security Policy, Section 4.1: Ownership and Responsibilities:
Physical security and maintenance of the infrastructure (hardware, software, networking, and facilities that run a server) is the responsibility of Voatz, Inc. in the case of on-premise physical servers being used.

Employ the principle of least functionality by configuring organizational systems to provide only essential capabilities.

1. Policy AC.9: Server Security Policy, Section 4.2 General Configuration and Backup Guidelines:
Operating System configuration should be in accordance with approved Voatz, Inc. IT guidelines. Services and applications that will not be used must be disabled where practical.
2. Policy AC.9: Server Security Policy:
The organization manages, configures and protects organization servers and hosts based on industry best practices.

Restrict, disable, or prevent the use of nonessential programs, functions, ports, protocols, and services.

1. ISO Control A.6.2.1: Mobile Device Management Policy, Section 3.2: User Requirements:
 - Users may only load corporate data that is essential to their role onto their mobile device(s).
2. ISO Control A.6.2.1: Mobile Device Management Policy, Section 3.2: User Requirements:
 - Applications must only be installed from official platform-owner approved sources. Installation of applications from untrusted sources is forbidden. If you are unsure if an application is from an approved source contact IT.

Apply deny-by-exception (blacklisting) policy to prevent the use of unauthorized software or deny-all, permit-by-exception (whitelisting) policy to allow the execution of authorized software.

1. Policy A.13: Network Security Policy, Section 4.0:
All the traffic to and from a particular list of IPs and Domains (Blacklisted) should be denied at the Firewall Level.
2. Policy AC.9: Server Security Policy, Section 4.2 General Configuration and Backup Guidelines:
Operating System configuration should be in accordance with approved Voatz, Inc. IT guidelines. Services and applications that will not be used must be disabled where practical.

Identify system users, processes acting on behalf of users, and devices.

1. Policy A.9 Access Control Policy: Section 1.0 Purpose
This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.
2. Policy A.8 Asset Management Policy: Section 2.0 Purpose
This policy provides procedures and protocols supporting effective organizational asset management specifically focused on electronic devices.

Authenticate (or verify) the identities of users, processes, or devices, as a prerequisite to allowing access to organizational systems.

1. Policy A.6.2.1: Mobile Device Management Policy, Section 3.1: Technology and Security Requirements:
Only devices managed by IT or authorized by IT will be allowed to connect directly to the internal corporate network.
2. Policy A.9: Access Control Policy, Section 3.0 Business Requirements
Users shall only be provided with access to the network and network services that they have been specifically authorized to use.

Use multifactor authentication (MFA) for local and network access to privileged accounts and for network access to non-privileged accounts.

1. Policy AC.5: Password Policy, Section 5.1: General:
Whenever possible, use 2-Factor Authentication (2FA) or 2-Step Authentication.
2. Policy A.9: Access Control Policy, Section 6.4 - Use of Privileged Utility Programs:
 - Where possible and appropriate privileged utility programs must incorporate multi-factor authentication (MFA).

Store and transmit only cryptographically-protected passwords.

1. Policy A.10: Cryptography: Section 3.3:
 - All data sent or received outside of Voatz, Inc. data centers and cloud environments (corporate and elections networks) shall be encrypted for transmission.
 - Where possible, data exchanged inside the Voatz, Inc. network should be encrypted.
2. Policy A.10: Cryptography: Section 3.5:
 - Where applicable, user passwords shall be stored in a secure manner by using HMAC with hash.

Establish an operational incident-handling capability for organizational systems that includes preparation, detection, analysis, containment, recovery, and user response activities.

1. Policy A.16 Information Security Incident Management Policy: Section 1.0: Overview

The purpose of this policy is to provide an effective response to incidents that threaten the confidentiality, integrity, and availability of digital assets, information systems, and the networks that deliver information.
2. Documentation for ISO Control A.16.1.4: Assessment of and Decision on Information Security Events: The purpose of the Voatz Information Security Incident Management Policy is to provide an effective response to incidents that threaten the confidentiality, integrity, and availability of digital assets, information systems, and the networks that deliver information. Supporting documentation includes guidance for assessing security events and making decisions about when to declare an incident, Incident Response Checklist, and Reporting/Notification Contact information.

Track, document, and report incidents to designated officials and/or authorities both internal and external to the organization.

1. Policy A.16 Information Security Incident Management Policy: Section 1.0: Overview

The purpose of this policy is to provide an effective response to incidents that threaten the confidentiality, integrity, and availability of digital assets, information systems, and the networks that deliver information.

2. Documentation for ISO control A.16.1.4 Assessment of and Decision on Information Security Events: The purpose of the Voatz Information Security Incident Management Policy is to provide an effective response to incidents that threaten the confidentiality, integrity, and availability of digital assets, information systems, and the networks that deliver information. Supporting documentation includes guidance for assessing security events and making decisions about when to declare an incident, Incident Response Checklist, and Reporting/Notification Contact information.

Provide controls on the tools, techniques, mechanisms, and personnel used to conduct system maintenance.

1. Policy AC9: Server Security Policy, Section 4.1: Ownership and Responsibilities: Physical security and maintenance of the infrastructure (hardware, software, networking, and facilities that run a server) is the responsibility of Voatz, Inc. in the case of on-premise physical servers being used.
2. Policy A.9: Access Control Policy, Section 3.0: Business Requirements:
Users shall only be provided with access to the network and network services that they have been specifically authorized to use.

Require multifactor authentication to establish nonlocal maintenance sessions via external network connections and terminate such connections when nonlocal maintenance is complete.

1. Policy AC5: Password Policy, Section 5.1: General:
Whenever possible, use 2-Factor Authentication (2FA) or 2-Step Authentication.
2. Policy A.9: Access Control Policy, Section 6.4: Use of Privileged Utility Programs:
Where possible and appropriate privileged utility programs must incorporate multi-factor authentication (MFA).

Control the use of removable media on system components.

1. Policy IC6: Removable Media Policy, Section 5.0: Policy:
 - Voatz, Inc. staff may only use Voatz, Inc. IT Ops approved removable media in their work computers.
2. Policy IC6: Removable Media Policy, Section 5.0: Policy:
 - Sensitive information should be stored on removable media only when required in the performance of your assigned duties or when providing information required by other state or federal agencies. When sensitive information is stored on removable media, it must be encrypted in accordance with the Voatz, Inc. Acceptable Encryption Policy.

Ensure that organizational systems containing sensitive data are protected during and after personnel actions such as terminations and transfers.

1. Policy A.7: Personnel Security Policy, Section 4.5 Personnel Termination:

The organization must upon personnel termination:

- Terminate information asset access.
- Conduct exit interviews.
- Retrieve all security-related organizational information system-related property.
- Retain access to organizational information and information assets formerly controlled by terminated personnel.

2. Policy A.7: Personnel Security Policy, Section 4.6 Personnel Transfer:

The organization must review logical and physical access to information assets and facilities when personnel are re-assigned or transferred to other positions within the organization.

Limit physical access to organizational information systems, equipment, and the respective operating environments to authorized individuals.

1. Policy A.11: Physical and Environmental Security Policy, Section 4.1 Physical Access

Authorization:

Develop, approve, and maintain a list of personnel with authorized access to the facility where information systems are physically located.

2. Policy AC9: Server Security, Section 4.2: General Configuration and Backup Guidelines:

Servers should be physically located in an access-controlled environment.

Protect and monitor the physical facility and support infrastructure for organizational systems.

1. Policy A.11: Physical and Environmental Security Policy, Section 4.2 Physical Access Control:

Employ guards and/or alarms to monitor physical access points to the data center(s) where the information system resides 24 hours per day, 7 days per week.

2. Policy A.11: Physical and Environmental Security Policy, Section 4.4 Monitoring Physical Access:

Review physical access logs at a defined frequency and upon occurrence of security incidents.

Comments: Physical security is provided by CIC.

Scan for vulnerabilities in organizational systems and applications periodically and when new vulnerabilities affecting those systems and applications are identified.

1. Laptops, desktops, and servers are protected with Sophos Endpoint, which scans for malware.
2. Policy A.6.2.1: Mobile Device Management Policy, Section 3.1: Technology and Security Requirements:
Devices must have endpoint protection/anti-malware installed. Where possible this is supplied by Voatz IT Ops.

Periodically assess the security controls in organizational systems to determine if the controls are effective in their application.

1. **Voatz Risk Assessment and Risk Treatment Methodology: Section 1.0 Overview**
An effective risk management process is an important component of a successful information security program. Risk management is the process of identifying risk, assessing risk and taking steps to reduce risk to an acceptable level. This guideline provides a foundation for the effective risk management program, containing both the definitions and the practical guidance necessary for assessing and mitigating risks identified within Voatz, Inc.
2. **Policy OM.2: Voatz Information Security Management System (ISMS) Policy: Section 2.0 Purpose**
The purpose of this document is to outline the approach Voatz will take to ensure the security of the processes, technology, data/information, and people involved in the development, operation, administration, maintenance and delivery of the Voatz mobile platform.

Comments: Voatz, Inc. utilizes the OneTrust (formerly Tugboat Logic) Virtual CISO Platform Platform to manage its ISMS. Voatz strives to take a methodical approach to identify risks associated with each of its infrastructure components, evaluate the potential for threats/threat actors to exploit flaws or vulnerabilities in these components, and take the necessary action to protect individual privacy as well as the confidentiality, integrity, and availability of Voatz, Inc. resources and data.

Monitor security controls on an ongoing basis to ensure the continued effectiveness of the controls.

1. **Policy OM.2: Voatz Information Security Management System (ISMS) Policy: Section 2.0 Purpose**
The purpose of this document is to outline the approach Voatz will take to ensure the security of the processes, technology, data/information, and people involved in the development, operation, administration, maintenance and delivery of the Voatz mobile platform.
2. **Voatz Risk Assessment and Risk Treatment Methodology: Section 1.0 Overview**
An effective risk management process is an important component of a successful information security program. Risk management is the process of identifying risk, assessing risk and taking steps to reduce risk to an acceptable level. This guideline provides a foundation for the effective risk management program, containing both the definitions and the practical guidance necessary for assessing and mitigating risks identified within Voatz, Inc.

Comments: Voatz, Inc. utilizes the OneTrust (formerly Tugboat Logic) Virtual CISO Platform Platform to manage InfoSec policies, provide security awareness training, implement and document security controls, and track compliance with customers, 3rd party vendors, independent auditors, and regulatory agencies.

Monitor, control, and protect communications (i.e., information transmitted or received by organizational systems) at the external boundaries and key internal boundaries of organizational systems.

1. **Policy A.13 Communications Security Policy: Section 1.0: Purpose**
The purpose of this policy is to ensure the availability and reliability of network devices to ensure safe and secure connections to the information assets owned by Voatz, Inc.

2. **Policy A.12 Operations Security Policy:** The purpose of the Voatz Communications and Network security policy is to ensure the availability and reliability of network devices to ensure safe and secure connections to the information assets owned by Voatz, Inc. The purpose of Operations policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy. This includes the stipulation that Voatz, Inc. will operate in a manner that meets its Information Security Objectives. As listed in the Information Security Policy, these objectives include:
 - Operating a voting platform that meets the confidentiality, integrity and availability requirements of jurisdictions around the world.
 - Continuously improving the Voatz service by keeping technology up to date, employing additional security controls where these make sense to maintain a strong security posture.
 - Proactively managing risks.

Voatz, Inc. processes are designed to meet information security requirements of its corporate and elections operations.

The following environments are involved with Voatz Operations:

- Corporate Environment (on-premise employee laptops and on-premise servers including Jira, Confluence, etc. and some test servers in AWS and Azure)
- Mobile/Elections Environment (including the app, cloud hosted core servers, databases, blockchain)

Employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational systems.

1. **Policy A.14 System Acquisition, Development and Maintenance: Section 1.0 Purpose**
This policy ensures that information security is an integral part of information systems across the entire software development lifecycle. This includes requirements for information systems which provide services over public networks and provides for secure methodologies during the design and implementation of Voatz, Inc. products and services.
2. **ISO Control A.14.2.1 Secure Development Policy**
Voatz Secure Development Policy provides for secure methodologies during the design and implementation of Voatz, Inc. products and services. It incorporates best practices in software and operations design and implementation processes with security and vulnerability reviews throughout their lifecycle so that implementation ensures that Voatz, Inc. products are only used as intended and are not vulnerable to attack or abuse.

Comments: Voatz policy ensures that information security is an integral part of information systems across the entire software development lifecycle. This includes requirements for information systems which provide services over public networks and provides for secure methodologies during the design and implementation of Voatz, Inc. products and services.

Implement subnetworks for publicly accessible system components that are physically or logically separated from internal networks.

1. Policy A.12 Operations Security Policy: Section 3.0: Business Requirements:
The following environments are involved with Voatz Operations:
 - Corporate Environment (on-premise employee laptops and on-premise servers including JIRA, Confluence, etc. and some test servers in AWS and Azure)
 - Mobile/Elections Environment (including the app, cloud hosted core servers, databases, blockchain)
2. ISO Control A.12.1.4 Separation of Development, Testing and Operational Environments: The purpose of Voatz Operations Security policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.

Deny network communications traffic by default and allow network communications traffic by exception (i.e., deny all, permit by exception).

1. The following is from an API routing module. This module defines all the permitted network communications. All other routes are implicitly denied.

```
def allRoutes: Route = encodeResponse {  
...  
}
```

2. Policy A.13: Network Security Policy, Section 4.0:
 - All traffic and protocols should be expressly denied except for those necessary for business purposes.
 - All the traffic to and from a particular list of IPs and Domains (Blacklisted) should be denied at the Firewall Level.

Protect the authenticity of communications sessions.

1. Policy A.13: Network Security Policy, Section 5.9: Encrypt sensitive network traffic:
Appropriate encryption and authentication methods should be used for the transmission of sensitive data and remote access.
2. Policy A.13: Network Security Policy, Section 5.2: Privilege Access Control:
Access control shall be used to provide separate authentication, authorization, and accounting services for network-based access.

Identify, report, and correct information and information system flaws in a timely manner.

1. Policy A.12 Operations Security Policy, Section 1.0: Purpose

The purpose of this policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.

2. ISO Control A.12.1.2 Change Management

Voatz Change Control and Configuration Process process in which change requestors create tickets, request approval by stakeholders, and implement changes to ensure changes to the application(s) and supporting infrastructure are documented, tested and approved by authorized personnel prior to implementation into the production environment in accordance with the change management process.

Comments: The purpose of the Voatz Operation Security policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.

Provide protection from malicious code at designated locations within organizational systems.

1. Policy A.12 Operation Security Policy, Section 1.0: Purpose

The purpose of this policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.

2. Policy A.12.2.1 - Controls Against Malware: The purpose of Voatz Operation Security policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy. A defense in depth approach shall be taken to protect Voatz from malware including, as a minimum:

- **Firewalls:** these shall be installed at all points where internal networks are connected to the internet. Further requirements on firewall configuration and maintenance are contained in the Communications Security Policy.
- **Anti-virus:** this shall be installed on all key devices including firewalls, proxy servers, all servers and all end-user devices
- **Threat monitoring:** information about emerging threats shall be obtained from appropriate sources and necessary action taken, e.g. ensuring that users are alerted proactively of potential attacks.
- **Security Awareness Training:** as phishing it is a major delivery mechanism for malware, Voatz conducts regular security awareness training.

Monitor system security alerts and advisories and take action in response.

1. Policy A.12: Operations Security Policy, Section 7.0: Logging and Monitoring:

Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.

2. Policy A.16 Information Security Incident Management, Section 1.0: Overview:
It is critical to the organization that incidents that threaten the security or privacy of critical information are properly identified, contained, investigated, and remedied.

Update malicious code protection mechanisms when new releases are available.

1. Policy A.12 Operations Security Policy, Section 1.0 Purpose
The purpose of this policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.
2. Policy A.12.2.1 Controls against malware:
Employee laptops have enterprise anti-malware/endpoint protection installed and servers are running a capability that detects changes as indicators of the presence of malware.

Comments: The purpose of Voatz Operations Security policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.

Monitor organizational systems, including inbound and outbound communications traffic, to detect attacks and indicators of potential attacks.

1. Policy A.13 Communications Security: Section 1.0: Purpose
The purpose of this policy is to ensure the availability and reliability of network devices to ensure safe and secure connections to the information assets owned by Voatz, Inc.
2. ISO Control A.13.1.2 - Security of Network Services
This policy is to ensure the availability and reliability of network devices to ensure safe and secure connections to the information assets owned by Voatz, Inc. The policy applies to all network devices, (includes switches, firewalls, and wireless access points) that are owned and managed by Voatz, Inc.

Employ the principle of least privilege, including for specific security functions and privileged accounts.

1. Policy A.9: Access Control Policy, Section 3.0:
This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.
 - Defense in Depth – security shall not depend upon any single control but be the sum of a number of complementary controls;

- Least Privilege – the default approach taken shall be to assume that access is not required, rather than to assume that it is. In particular, all access to election infrastructure and election data shall be limited to those individuals that have a business or legal need-to-know;
 - Need to Know – access is only granted to the information required to perform a role, and no more. The default setting of the access control system has to be “deny-all”;
 - Need to Use – users will only be able to access physical and logical facilities required for their role.
2. Policy AC9: Server Security, Section 4.2 General Configuration and Backup Guidelines:
Always use the least privilege to perform a function. Do not use root/admin accounts when a non-privileged account is sufficient.

Ensure that the actions of individual system users can be uniquely traced to those users so they can be held accountable for their actions.

1. Policy A.9 Access Control Policy, Section 1.0: Purpose:
This policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege.
2. Policy A.12 Operations Security Policy
The purpose of Voatz Operations Security policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy. Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure.

Comments: Voatz Access Control Policy ensures access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege. Under the policy Users shall only be provided with access to the network and network services that they have been specifically authorized to use.

Perform maintenance on organizational systems.

1. Policy AC.9: Server Security Policy, Section 4.1: Ownership and Responsibilities:
Physical security and maintenance of the infrastructure (hardware, software, networking, and facilities that run a server) is the responsibility of Voatz, Inc. in the case of on-premise physical servers being used.
2. Policy A.13: Network Security Policy, Section 4.0:
 - Every network device deployed in the Voatz, Inc. network shall be appropriately configured and meet security requirements for their individual purposes (internal, public facing).

Check media containing diagnostic and test programs for malicious code before the media are used in organizational systems.

1. Policy A.12 Operations Security Policy, Section 5.0: Protection from Malware:
A defense in depth approach shall be taken to protect Voatz from malware.
2. Policy IC.6: Removable Media Policy, Section 1.0: Overview:
Removable media is a well-known source of malware infections and has been directly tied to the loss of sensitive information in many organizations. It is important to utilize removable media carefully.

Comments: The purpose of Voatz Operations Security policy policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.

Prohibit the use of portable storage devices when such devices have no identifiable owner.

1. Policy A.12 Operations Security Policy
The purpose of Voatz Operations Security policy policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.
2. Policy IC.6: Removable Media Policy, Section 1.0: Overview:
Removable media is a well-known source of malware infections and has been directly tied to the loss of sensitive information in many organizations. It is important to utilize removable media carefully.

Screen individuals prior to authorizing access to organizational systems containing sensitive data.

1. Policy A.7: Personnel Security Policy, Section 4.3: Personnel Screening
The organization must screen individuals prior to access to sensitive information. In addition, the organization must rescreen individuals every seven years.
2. Policy A.9 Access Control Policy

The purpose of Voatz access control is to ensure access to information assets and systems within Voatz, Inc.'s corporate and election environments is actively managed in accordance with the principles of least privilege. In addition to the specific requirements, a number of general principles will be used when designing access controls for Voatz' systems and services. These are:

- Defense in Depth – security shall not depend upon any single control but be the sum of a number of complementary controls;
- Least Privilege – the default approach taken shall be to assume that access is not required, rather than to assume that it is. In particular, all access to election infrastructure and election data shall be limited to those individuals that have a business or legal need-to-know;

- Need to Know – access is only granted to the information required to perform a role, and no more. The default setting of the access control system has to be “deny-all”;
- Need to Use – users will only be able to access physical and logical facilities required for their role

Periodically assess the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, and individuals, resulting from the operation of organizational systems and the associated processing, storage, or transmission of sensitive data.

1. Policy A.7: Personnel Security Policy, Section 4.2: Position Categorization
Conduct performance evaluations and review and revise position risk designations on an annual basis.
2. Voatz Risk Assessment; ISO Clause 8.2-Information Security Risk Assessment

Develop and implement plans of action designed to correct deficiencies and reduce or eliminate vulnerabilities in organizational systems.

1. Policy A.12 Operations Security, A.14 System Acquisition, Development and Maintenance
2. Policy A.12.6.1 - Management of Technical Vulnerabilities

Comments: The purpose of the policy is to ensure that information security is an integral part of information systems across the entire software development lifecycle. This includes requirements for information systems which provide services over public networks and provides for secure methodologies during the design and implementation of Voatz, Inc. products and services.

Implement cryptographic mechanisms to prevent unauthorized disclosure of sensitive information during transmission unless otherwise protected by alternative physical safeguards.

1. Policy A.10: Cryptography: Section 3.3:
 - All data sent or received outside of Voatz, Inc. data centers and cloud environments (corporate and elections networks) shall be encrypted for transmission.
 - Where possible, data exchanged inside the Voatz, Inc. network should be encrypted.
2. Policy A.13, Section 5.9 Encrypt sensitive network traffic
Appropriate encryption and authentication methods should be used for the transmission of sensitive data and remote access.

Employ FIPS-validated cryptography when used to protect the confidentiality of sensitive data.

1. Policy A.10: Cryptography: Section 3.1: Encryption Method and Application:
All Voatz, Inc. encryption shall be done using approved cryptographic modules.
2. Policy A.10: Cryptography: Section 3.1: Encryption Method and Application:

Symmetric cryptosystem key lengths must be at least 256 bits. Asymmetric crypto-system keys must be of a length that yields equivalent strength. Voatz, Inc.'s key length requirements shall be reviewed annually as part of the yearly security review and upgraded as technology allows.

Perform periodic scans of organizational systems and real-time scans of files from external sources as files are downloaded, opened, or executed.

1. Policy A.12: Operations Security Policy, Section 9.0 Technical Vulnerability Management
Voatz conducts regular vulnerability scans for all internal and external assets to identify known vulnerabilities.
2. Policy A.12: Operations Security Policy, Section 10.0 Information Systems Audit
For hosts in production, scans and other audit activities must be scheduled to run during times that minimize disruption to live events.

Identify unauthorized use of organizational systems.

1. Policy A.12: Operations Security Policy, Section 7.0 Logging and Monitoring
Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.
2. Policy A.12: Operations Security Policy, Section 8.0: Control of Operational Software:
Voatz monitors systems to detect unauthorized access and modification.

Separate the duties of individuals to reduce the risk of malevolent activity without collusion.

1. Policy A.9: Access Control Policy, Section 3.0: Business Requirements:
Users shall only be provided with access to the network and network services that they have been specifically authorized to use.
2. Policy A.9: Access Control Policy, Section 6.1 Information Access Restriction:
All Voatz, Inc. systems must enforce approved authorizations for access to the system in accordance with applicable policy and users requiring access must receive approval directly from the system/application owner.
Voatz follows least privilege when granting access to information and application system functionality.

Additionally, access to information/data deemed sensitive or confidential is limited based on a need to know.

Read-only access is granted to users based on their roles, while functions that allow users to create, modify, or delete information is restricted to designated super users/admins. (E.g., Only admins can create or delete code repositories, while developers are restricted to adding new code, and editing existing code. Only a super admin or designated alternate with super admin access can push code to the live/production environment.

Prevent non-privileged users from executing privileged functions and capture the execution of such functions in audit logs.

1. Policy A.9: Access Control Policy, Section 6.4: Use of Privileged Utility Programs:
 - Utility programs must have secure authentication mechanisms that use identification, authentication and authorization and accounting/audit procedures.
2. Policy A.11: Physical and Environmental Security Policy, Section 4.2 Physical Access Control: The organization will control entry/exit from the data center(s) and/or sensitive facilities using physical access control devices (e.g., keycard or keys) and/ or security guard(s).
 - Maintain physical access audit logs for data center(s) and/or sensitive facilities entry/exit points.

Limit unsuccessful logon attempts.

1. The following is from a database authentication module.

```
def checkLast3LoginAttempts(userId: Int)(implicit db: MongoDB): Future[Boolean] =
  async{
    val (last3AttemptsFailed, lastAttemptTs) = await(last3LoginAttemptsFailed(userId))
    if(last3AttemptsFailed){
      val (latestInterval, attempt) =
        await(ThrottledLoginAsync.getLatestThrottleIntervalAndCount(userId))
      ThrottledLoginAsync.create(userId, lastAttemptTs, latestInterval, attempt)
      false
    }else{
      true
    }
  }
}
```

2. The following is from an authentication module.

```
val throttleInterval = if(attempt == 0){
  30 + Cryptography.issueRandomInt(1, 10)
}else{
  val min = (attempt)*10
  val max = (attempt+1)*10
  latestInterval + Cryptography.issueRandomInt(min, max)
}
```

Use session lock with pattern-hiding displays to prevent access and viewing of data after a period of inactivity.

1. The following is from the VMA Android app.

```
disposable = RxBus.instance.listen(RxEvent.InAppEventAdd::class.java)
  .subscribe {
```



```
InAppIdleTimer.getInstance()
    .showWarningDialogBox(this)
```

```
....
```

Here, showWarningDialogBox shows an inactivity logout warning to the user.

```
fun showWarningDialogBox(context: Context) {
    val dialog = MaterialDialog.Builder(context)
        .title(context.resources.getString(R.string.timeout_title))
        .content(String.format(context.resources.getString(R.string.logout_timeout_message), "30"))
```

2. The VMA iOS app has logic equivalent to the Android version of VMA. The following is the inactivity timer connected to a ballot screen; the voter will be logged out from this screen after a period of inactivity.

```
private let inactivityTimer: InactivityTimerManager
...
init(...) { inactivityTimer = InactivityTimerManager(includeWarning: false)
```

Terminate (automatically) a user session after a defined condition.

1. The following is from the VMA Android app.

```
if (User.isLoggedIn) {
    if (isScreenOn && (dialog)!!.isShowing) {
        dialog.contentView?.announceForAccessibility(
            context.getString(R.string.timeout_message)
        )
        User.logoutButtonClicked(context, apiService, compositeDisposable)
        dialog.dismiss()
```

2. The VMA iOS app has logic equivalent to the Android version of VMA. Here is one example of VMA logging out when the app goes into the background.

```
@objc func applicationDidEnterBackground() {
    appBackgroundedTime = Date()
    showLoginView()
    ...
    loginApiService.restApiManager.cancelRequests(with: URLs) { [weak self] in
        DispatchQueue.main.async {
            self.logout {
                self.backgroundLogoutDispatchGroup.leave()
            }
        }
    }
```

Route remote access via managed access control points.

1. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction

Voatz aims for a layered approach to manage access to corporate and elections assets which includes:

- Disabling all ports or services on critical assets not needed.
2. Policy AC.6: Remote Access Policy, Section 4.2: Requirements:
 - Voatz, Inc. employees and contractors with remote access privileges must ensure that their Voatz, Inc.-owned or personal computer or workstation, which is remotely connected to Voatz, Inc.'s corporate network, is not connected to any other network at the same time, with the exception of personal networks that are under the complete control of the user.

Verify and control/limit connections to and use of external systems.

1. Policy A.9: Access Control Policy, Section 3.0: Business Requirements:
The access control measures implemented shall be appropriate to the business requirements. These requirements may depend on factors such as:
 - Contractual obligations to external third parties;
2. For VMA, VWA, and VAP this is not applicable; VMA, VWA, and VAP do not connect to or use external information systems. The app may provide the user with a link to the election authority's official website where the user can find additional information, but such links will always be over HTTPS.

Limit use of portable storage devices on external systems.

1. Policy IC6: Removable Media Policy, Section 5.0: Policy:
 - Voatz, Inc. staff may only use Voatz, Inc. IT Ops approved removable media in their work computers.
2. Policy IC6: Removable Media Policy, Section 5.0: Policy:
 - Sensitive information should be stored on removable media only when required in the performance of your assigned duties or when providing information required by other state or federal agencies. When sensitive information is stored on removable media, it must be encrypted in accordance with the Voatz, Inc. Acceptable Encryption Policy.

Provide security awareness training on recognizing and reporting potential indicators of insider threat.

1. Policy A.6.2.1: Mobile Device Management Policy, Section 3.2 User Requirements:
If a user suspects that company data has been sent from a personal email account, either in the body text or as an attachment, they must notify Voatz, Inc. IT Ops immediately.
2. Policy A.8: IT Asset Management Policy, Section 7.0 Responsibilities
Voatz IT Ops is responsible for implementing this policy. This team will be responsible for:
 - Reporting any incorrect disposal or misuse of an IT asset.

Review and update logged events.

1. Policy AC.9: Server Security, Section 4.3: Monitoring:
Security-related events will be reported to Voatz, Inc. IT Ops, who will review logs and escalate incidents as necessary. Corrective measures will be prescribed as needed.
2. Policy A.12: Operations Security Policy, Section 7.0 Logging and Monitoring
Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.

Alert in the event of an audit logging process failure.

1. Policy A.12: Operations Security Policy, Section 7.0 Logging and Monitoring
Voatz centralized collection of logs captures events from sources on the corporate and elections infrastructure. Logging activities have also been configured to trigger alerts for unusual or suspicious activity.
2. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz monitors systems to detect unauthorized access and modification.

Protect audit information and audit logging tools from unauthorized access, modification, and deletion.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.
2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows the principle of least privilege when granting access to information and application system functionality. Additionally, access to information/data deemed sensitive or confidential is limited based on a need to know. Read-only access is granted to users based on their roles, while functions that allow users to create, modify, or delete information is restricted to designated super users/admins.

Limit management of audit logging functionality to a subset of privileged users.

1. Policy A.12: Operations Security Policy, 8.0 Control of Operational Software:
Voatz also follows the principle of least privilege to limit the access to critical systems and limit the ability to make changes.
2. Policy A.9: Access Control Policy, Section 6.1: Information Access Restriction:
Voatz follows least privilege when granting access to information and application system functionality.

Additionally, access to information/data deemed sensitive or confidential is limited based on a need to know.

Read-only access is granted to users based on their roles, while functions that allow users to create, modify, or delete information is restricted to designated super users/admins.

Analyze the security impact of changes prior to implementation.

1. Policy A.14 System Acquisition, Development and Maintenance, Section 1.0 Purpose
This policy ensures that information security is an integral part of information systems across the entire software development lifecycle. This includes requirements for information systems which provide services over public networks and provides for secure methodologies during the design and implementation of Voatz, Inc. products and services.
2. ISO Control A.14.2.2 - System Change Control Procedures: Voatz policy ensures that information security is an integral part of information systems across the entire software development lifecycle. This includes requirements for information systems which provide services over public networks and provides for secure methodologies during the design and implementation of Voatz, Inc. products and services.

Control and monitor user-installed software.

1. Policy A.6.2.1: Mobile Device Management Policy, Section 3.2: User Requirements:
Applications must only be installed from official platform-owner approved sources. Installation of applications from untrusted sources is forbidden. If you are unsure if an application is from an approved source contact IT.
2. Policy IC.8: Workstation Security Policy, Section 4.2 Operational Safeguards:
The following operational safeguards shall be implemented:
 - Ensuring that latest OS and software updates are installed.
 - Never installing unauthorized software on workstations.

Enforce a minimum password complexity and change of characters when new passwords are created.

1. Policy AC 5: Password Policy, Section 5.2.1: General Password Construction Guidelines
All users should be aware of how to select strong passwords.
Administrative passwords should be 15 (semi-random) characters long.
2. Policy AC 5: Password Policy, Section 5.2.2: Passphrases
Passphrases can be used as an alternative to passwords.

Allow temporary password use for system logons with an immediate change to a permanent password.

1. Policy A.9: Access Control Policy, Section 4.4 Management of Secret Authentication Information of Users:
Users select their own passwords. Where Voatz IT Ops issues initial passwords, these are sent out of band and users are asked to change their passwords and any default passwords.

2. Policy AC.5: Password Policy, Section 5.2.3. Use of Passwords and Passphrases for Remote Access Users:

Access to the Voatz, Inc. network via remote access is to be controlled using either a one-time password authentication or a public/private key system with a strong passphrase.

Obscure feedback of authentication information.

1. Policy A.9: Access Control Policy, Section 6.2 - Secure Log-on Procedures
Where required by the access control policy, access to systems and applications shall be controlled by a secure log-on procedure.
2. Policy A.10: Cryptography: Section 3.1: Encryption Method and Application:
Encryption shall be implemented in:
 - All web applications via the most secure version of SSL/TLS,
 - Device/file encryption i.e. partial disk encryption and full disk encryption,
 - During data transmission between and among infrastructure components,
 - During sensitive data transmission over public networks,
 - storage on disk, particularly secrets and user passwords.

Test the organizational incident response capability.

1. Policy A.16 Information Security Incident Management Policy: Voatz Information Security and Incident Management Policy's purpose is to provide an effective response to incidents that threaten the confidentiality, integrity, and availability of digital assets, information systems, and the networks that deliver information
2. Policy A.17 Information Security Aspects of Business Continuity Management: The purpose of Voatz Business Continuity policy is to ensure information security continuity is embedded in the organization's business operations and provides guidance on how to ensure continuous secure operations during any adverse event affecting Voatz, Inc. assets.

Supervise the maintenance activities of maintenance personnel without required access authorization.

1. Policy A.11: Physical and Environmental Security Policy, Section 4.2 Physical Access Control:
 - Establish a process to escort visitors and monitor their activity within the data center(s) and/or sensitive facilities.
2. Policy AC.1: Visitor and Contractor Premises Access Policy, Section 4.4 Visitor Badges
Visitor Badges issued must be worn at all times. Visitors requiring access to locked areas should arrange times with their sponsor and Voatz IT Ops.

Protect the confidentiality of backup sensitive data at storage locations.

1. Policy A.8: IT Asset Management Policy, Section 5.2 Asset Value,
Assets, which store data, regardless of cost, shall be tracked as part of a computing device or as a part of network attached storage. These assets include:

- Network Attached Storage (NAS), Storage Area Network (SAN) or other computer data storage.
 - Temporary storage drives.
 - Tape or optical media with data stored on them including system backup data.
2. Policy A.8: IT Asset Management Policy, Section 7.0: Responsibilities:
Voatz IT Ops is responsible for implementing this policy. This team will be responsible for:
- Coordinating IT asset audit activities and updating and maintaining the accuracy of the asset inventory
- Checking IT equipment is returned in the same configuration as expected;
 - Administrating the control and security of IT equipment held in stock for issuing and awaiting reissue or disposal;
 - Ensuring that all IT equipment is returned to Voatz upon replacement or when the holder leaves the organization
 - Ensuring that any IT asset that is retired is disposed of according to the Equipment Disposal policy
 - Giving appropriate advice to users on the correct handling of IT assets; and
 - Reporting any incorrect disposal or misuse of an IT asset.

Escort visitors and monitor visitor activity.

1. Policy AC.1: Visitor and Contractor Premises Access Policy, Section 4.3: Visitor Logs
Visitors will be asked to sign-in at visitor control. Voatz sponsors will also ensure visitors sign-in once they arrive at the Voatz office. Voatz will also keep a log of visitors.
2. Policy AC.1: Visitor and Contractor Premises Access Policy, Section 4.4 Visitor Badges
Visitor Badges issued must be worn at all times. Visitors requiring access to locked areas should arrange times with their sponsor and Voatz IT Ops.

Remediate vulnerabilities in accordance with risk assessments.

1. Policy A.16 Information Security Incident Management, Section 3.1: Voatz Election Infrastructure Policy:
Voatz leadership will direct remediation and recovery efforts and reach out to external incident response and investigation resources as necessary.
2. Policy A.16 Information Security Incident Management, Section 3.2: Voatz Corporate Network Policy:

Risk mitigation: Identify and recommend strategies to mitigate the risk of harm arising from this incident.

Terminate network connections associated with communications sessions at the end of the sessions or after a defined period of inactivity.

1. Policy AC.6: Remote Access Policy, Section 4.2 Requirements:
 - All remote access connections must include a “time-out” system. In accordance with Voatz, Inc.’s security policies, remote access sessions for all SSH sessions will time out after 30 minutes of inactivity and will terminate after unlimited hours of continuous connection. Both time-outs will require the user to reconnect and re-authenticate in order to re-enter company networks.
2. Policy A.13: Network Security Policy, Section 4.0 Policy
 - Network devices owned and managed by Voatz, Inc. must be configured securely and designed to secure network traffic between trusted and untrusted network zones.

Establish and manage cryptographic keys for cryptography employed in organizational systems.

1. Policy A.10: Acceptable Encryption Policy, Section 3.6: Key Management:
 - Key management should be planned which will include secure key generation, use, storage, distribution, recovery, and removal.
2. Policy AC.5: Password Policy, Section 5.1 General
 - Whenever possible, use 2-Factor Authentication (2FA) or 2-Step Authentication.
 - All production system-level passwords must be part of the Voatz, Inc. IT Ops administered, global password management database.

Develop, document, and periodically update system security plans that describe system boundaries, system environments of operation, how security requirements are implemented, and the relationships with or connections to other systems.

1. Risk Assessment and Risk Treatment Methodology, Section 8.0 Monitor and Review:

The risk management process should be an iterative process and should be the subject of a structured monitoring and review process.
2. Policy A.12: Operations Security Policy, Section 1.0 Purpose

The purpose of this policy is to guide Voatz, Inc. to plan, implement and control the operational processes needed to meet information security requirements, address risks, and implement plans to achieve information security objectives as identified in the Voatz Information Security Policy.

5. Attachment F: POA&M Tracker

There are no requirements that are lacking supporting documentation.

6. Attachment G: Voatz, Inc. Accessibility Conformance Report

Please see the following pages with the Accessibility Conformance Report.

Voatz, Inc. Accessibility Conformance Report

WCAG Edition

(Based on VPAT® Version 2.4)

Name of Product/Version:

Voatz™ Web App (browser-based)

Report Date:

September 2, 2022 (v. 1.01)

Product Description:

Remote Accessible Ballot Delivery, Marking & Return (RABDMR)

Contact Information:

Linda Hutchinson, Director of Quality Assurance & Compliance

"Voluntary Product Accessibility Template" and "VPAT" are registered service marks of the Information Technology Industry Council (ITI)

Page 1 of 8

Notes:

Voatz for the web is a secure, browser-based application and compliant with WCAG 2.1 Level AA. This ensures that it is accessible for users with sensory, mobility and/or cognitive disabilities.

Evaluation Methods Used:

1. Voatz partnered with the **National Center for Accessible Media** during the design process for their recommendations on accessibility and WCAG 2.1.
2. Once implemented, the resulting apps were **systematically evaluated by professional testers** with previous WCAG and voting systems experience. Compliance testing was performed using both an accessibility assessment tool and by manually testing with popular screenreaders.
3. The Voatz team continues to **incorporate feedback from users**, including users with disabilities, and continuously evaluates accessibility.

Applicable Standards/Guidelines

This report covers the degree of conformance for the following accessibility standard/guidelines:

| Standard/Guideline | Included in Report |
|--|---|
| Web Content Accessibility Guidelines 2.1 | Level A (Yes) Level AA (Yes) Level AAA (Not Evaluated) |

Terms

The terms used in the Conformance Level information are defined as follows:

Page 2 of 8

- **Supports:** The functionality of the product has at least one method that meets the criterion without known defects or meets with equivalent facilitation.
- **Partially Supports:** Some functionality of the product does not meet the criterion.
- **Does Not Support:** The majority of product functionality does not meet the criterion.
- **Not Applicable:** The criterion is not relevant to the product.
- **Not Evaluated:** The product has not been evaluated against the criterion. This can be used only in WCAG 2.0 Level AAA.

WCAG 2.x Report

Note: When reporting on conformance with the WCAG 2.x Success Criteria, they are scoped for full pages, complete processes, and accessibility-supported ways of using technology as documented in the [WCAG 2.0 Conformance Requirements](#).

Table 1: Success Criteria, Level A

Notes:

| Criteria | Conformance Level | Remarks and Explanations |
|--|--------------------------|---|
| 1.1.1 Non-text Content (Level A) | Supports | |
| 1.2.1 Audio-only and Video-only (Prerecorded) (Level A) | Not Applicable | There is no video content within the Voatz apps. |
| 1.2.2 Captions (Prerecorded) (Level A) | Not Applicable | |
| 1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A) | Not Applicable | |
| 1.3.1 Info and Relationships (Level A) | Supports | |
| 1.3.2 Meaningful Sequence (Level A) | Supports | |
| 1.3.3 Sensory Characteristics (Level A) | Supports | |
| 1.4.1 Use of Color (Level A) | Supports | |
| 1.4.2 Audio Control (Level A) | Not Applicable | |
| 2.1.1 Keyboard (Level A) | Supports with exception. | Using screenreaders such as NVDA, JAWS, and Apple VoiceOver. Exception: For jurisdictions requiring use of Voatz' ID verification, a sighted individual can assist a visually-impaired voter by scanning a photo ID and following video 'selfie' instructions. This can be performed prior to voting without compromising the privacy of the voter's ballot. |
| 2.1.2 No Keyboard Trap (Level A) | Supports | |
| 2.1.4 Character Key Shortcuts (Level A 2.1 only) | Not Applicable | |
| 2.2.1 Timing Adjustable (Level A) | Supports | Timeout can be extended. |
| 2.2.2 Pause, Stop, Hide (Level A) | Not Applicable | |
| 2.3.1 Three Flashes or Below Threshold (Level A) | Not Applicable | |

| Criteria | Conformance Level | Remarks and Explanations |
|---|-------------------|--------------------------|
| 2.4.1 Bypass Blocks (Level A) | Supports | |
| 2.4.2 Page Titled (Level A) | Supports | |
| 2.4.3 Focus Order (Level A) | Supports | |
| 2.4.4 Link Purpose (In Context) (Level A) | Supports | |
| 2.5.1 Pointer Gestures (Level A 2.1 only) | Supports | |
| 2.5.2 Pointer Cancellation (Level A 2.1 only) | Supports | |
| 2.5.3 Label in Name (Level A 2.1 only) | Supports | |
| 2.5.4 Motion Actuation (Level A 2.1 only) | Not Applicable | |
| 3.1.1 Language of Page (Level A) | Supports | |
| 3.2.1 On Focus (Level A) | Supports | |
| 3.2.2 On Input (Level A) | Supports | |
| 3.3.1 Error Identification (Level A) | Supports | |
| 3.3.2 Labels or Instructions (Level A) | Supports | |
| 4.1.1 Parsing (Level A) | Supports | |
| 4.1.2 Name, Role, Value (Level A) | Supports | |

Table 2: Success Criteria, Level AA

Notes:

| Criteria | Conformance Level | Remarks and Explanations |
|--|-------------------|--|
| 1.2.4 Captions (Live) (Level AA) | Not Applicable | |
| 1.2.5 Audio Description (Prerecorded) (Level AA) | Not Applicable | |
| 1.3.4 Orientation (Level AA 2.1 only) | Not Applicable | The list nature of ballots requires portrait orientation to reduce scrolling in order to comply with Voluntary Voting System Guidelines. |
| 1.3.5 Identify Input Purpose (Level AA 2.1 only) | Supports | |
| 1.4.3 Contrast (Minimum) (Level AA) | Supports | |
| 1.4.4 Resize text (Level AA) | Supports | |
| 1.4.5 Images of Text (Level AA) | Supports | |
| 1.4.10 Reflow (Level AA 2.1 only) | Supports | |
| 1.4.11 Non-text Contrast (Level AA 2.1 only) | Supports | |
| 1.4.12 Text Spacing (Level AA 2.1 only) | Supports | |
| 1.4.13 Content on Hover or Focus (Level AA 2.1 only) | Nor Applicable | |
| 2.4.5 Multiple Ways (Level AA) | Not Applicable | |
| 2.4.6 Headings and Labels (Level AA) | Supports | |
| 2.4.7 Focus Visible (Level AA) | Supports | |
| 3.1.2 Language of Parts (Level AA) | Supports | |
| 3.2.3 Consistent Navigation (Level AA) | Supports | |
| 3.2.4 Consistent Identification (Level AA) | Supports | |
| 3.3.3 Error Suggestion (Level AA) | Supports | |
| 3.3.4 Error Prevention (Legal, Financial, Data) (Level AA) | Supports | |
| 4.1.3 Status Messages (Level AA 2.1 only) | Not Applicable | |

Table 3: Success Criteria, Level AAA

Notes:

| Criteria | Conformance Level | Remarks and Explanations |
|--|-------------------|---|
| 1.2.6 Sign Language (Prerecorded) (Level AAA) | Not Applicable | |
| 1.2.7 Extended Audio Description (Prerecorded) (Level AAA) | Not Applicable | |
| 1.2.8 Media Alternative (Prerecorded) (Level AAA) | Not Applicable | |
| 1.2.9 Audio-only (Live) (Level AAA) | Not Applicable | |
| 1.3.6 Identify Purpose (Level AAA 2.1 only) | Not Evaluated | |
| 1.4.6 Contrast (Enhanced) (Level AAA) | Not Evaluated | |
| 1.4.7 Low or No Background Audio (Level AAA) | Not Evaluated | |
| 1.4.8 Visual Presentation (Level AAA) | Not Evaluated | |
| 1.4.9 Images of Text (No Exception) (Level AAA) | Not Evaluated | |
| 2.1.3 Keyboard (No Exception) (Level AAA) | Not Evaluated | |
| 2.2.3 No Timing (Level AAA) | Supports | |
| 2.2.4 Interruptions (Level AAA) | Not Evaluated | |
| 2.2.5 Re-authenticating (Level AAA) | Not Evaluated | |
| 2.2.6 Timeouts (Level AAA 2.1 only) | Supports | After 4 1/2 minutes of inactivity, users are alerted to continue their session. |
| 2.3.2 Three Flashes (Level AAA) | Not Applicable | |
| 2.3.3 Animation from Interactions (Level AAA 2.1 only) | Not Applicable | |
| 2.4.8 Location (Level AAA) | Not Evaluated | |
| 2.4.9 Link Purpose (Link Only) (Level AAA) | Not Evaluated | |
| 2.4.10 Section Headings (Level AAA) | Supports | |
| 2.5.5 Target Size (Level AAA 2.1 only) | Not Evaluated | |
| 2.5.6 Concurrent Input Mechanisms (Level AAA 2.1 only) | Not Evaluated | |
| 3.1.3 Unusual Words (Level AAA) | Not Evaluated | |

Page 7 of 8

| Criteria | Conformance Level | Remarks and Explanations |
|--|-------------------|--------------------------|
| 3.1.4 Abbreviations (Level AAA) | Not Evaluated | |
| 3.1.5 Reading Level (Level AAA) | Not Evaluated | |
| 3.1.6 Pronunciation (Level AAA) | Not Evaluated | |
| 3.2.5 Change on Request (Level AAA) | Not Evaluated | |
| 3.3.5 Help (Level AAA) | Supports | |
| 3.3.6 Error Prevention (All) (Level AAA) | Not Evaluated | |

Page 8 of 8

Voatz, Inc. Accessibility Conformance Report

WCAG Edition

(Based on VPAT® Version 2.4)

Name of Product/Version:

Voatz™ Mobile Apps (for iOS® and Android®)

Report Date:

November 18, 2021

Product Description:

Remote Accessible Ballot Delivery, Marking & Return (RABDMR)

Contact Information:

Linda Hutchinson, Director of Quality Assurance & Compliance

"Voluntary Product Accessibility Template" and "VPAT" are registered service marks of the Information Technology Industry Council (ITI)

Page 1 of 8

Notes:

Voatz for iOS and Android are native smartphone apps, not web applications, so where HTML structure is referenced in WCAG, this report considers the comparable programming conventions to ensure optimal support the following native smartphone assistive capabilities.

Evaluation Methods Used:

1. Voatz partnered with the **National Center for Accessible Media** during the design process for their recommendations on accessibility and WCAG 2.1.
2. Once implemented, the resulting apps were systematically evaluated by professional testers with previous WCAG and voting systems experience.
3. Voatz engaged an **independent voting system testing laboratory (VSTL)** certified in assessing Accessibility compliance to VVSG 1.1. Pro V&V's report dated May 28, 2020, concludes that the Voatz apps conform to applicable federal accessibility requirements.
4. The Voatz team continues to incorporate feedback from users, including users with disabilities, and performs and continuously evaluates accessibility.

Applicable Standards/Guidelines

This report covers the degree of conformance for the following accessibility standard/guidelines:

| Standard/Guideline | Included In Report |
|--|---|
| Web Content Accessibility Guidelines 2.1 | Level A (Yes) Level AA (Yes) Level AAA (Not Evaluated) |

Terms

The terms used in the Conformance Level information are defined as follows:

- **Supports:** The functionality of the product has at least one method that meets the criterion without known defects or meets with equivalent facilitation.
- **Partially Supports:** Some functionality of the product does not meet the criterion.
- **Does Not Support:** The majority of product functionality does not meet the criterion.
- **Not Applicable:** The criterion is not relevant to the product.
- **Not Evaluated:** The product has not been evaluated against the criterion. This can be used only in WCAG 2.0 Level AAA.

WCAG 2.x Report

Note: When reporting on conformance with the WCAG 2.x Success Criteria, they are scoped for full pages, complete processes, and accessibility-supported ways of using technology as documented in the [WCAG 2.0 Conformance Requirements](#).

Table 1: Success Criteria, Level A

Notes:

| Criteria | Conformance Level | Remarks and Explanations |
|--|--------------------------|--|
| 1.1.1 Non-text Content (Level A) | Supports | |
| 1.2.1 Audio-only and Video-only (Prerecorded) (Level A) | Not Applicable | There is no video content within the Voatz apps. |
| 1.2.2 Captions (Prerecorded) (Level A) | Not Applicable | |
| 1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A) | Not Applicable | |
| 1.3.1 Info and Relationships (Level A) | Supports | |
| 1.3.2 Meaningful Sequence (Level A) | Supports | |
| 1.3.3 Sensory Characteristics (Level A) | Supports | |
| 1.4.1 Use of Color (Level A) | Supports | |
| 1.4.2 Audio Control (Level A) | Not Applicable | |
| 2.1.1 Keyboard (Level A) | Supports with exception. | <p>With VoiceOver (iOS) or TalkBack (Android) screenreaders enabled on smartphones, standard gestures (swipe right/left, tap and double-tap), combined with visible navigation focus ensure that precise mouse-like inputs are not required. In addition, iOS Voice Control is supported for entirely hands-free operation.</p> <p>Exception: For jurisdictions requiring use of Voatz' ID verification, a sighted individual can assist a visually-impaired voter by scanning a photo ID and following video 'selfie' instructions. This can be performed prior to voting without compromising the privacy of the voter's ballot.</p> |

| Criteria | Conformance Level | Remarks and Explanations |
|--|-------------------|--------------------------|
| 2.1.2 No Keyboard Trap (Level A) | Supports | |
| 2.1.4 Character Key Shortcuts (Level A 2.1 only) | Not Applicable | |
| 2.2.1 Timing Adjustable (Level A) | Supports | |
| 2.2.2 Pause, Stop, Hide (Level A) | Not Applicable | |
| 2.3.1 Three Flashes or Below Threshold (Level A) | Not Applicable | |
| 2.4.1 Bypass Blocks (Level A) | Supports | |
| 2.4.2 Page Titled (Level A) | Supports | |
| 2.4.3 Focus Order (Level A) | Supports | |
| 2.4.4 Link Purpose (In Context) (Level A) | Supports | |
| 2.5.1 Pointer Gestures (Level A 2.1 only) | Supports | |
| 2.5.2 Pointer Cancellation (Level A 2.1 only) | Supports | |
| 2.5.3 Label in Name (Level A 2.1 only) | Supports | |
| 2.5.4 Motion Actuation (Level A 2.1 only) | Not Applicable | |
| 3.1.1 Language of Page (Level A) | Supports | |
| 3.2.1 On Focus (Level A) | Supports | |
| 3.2.2 On Input (Level A) | Supports | |
| 3.3.1 Error Identification (Level A) | Supports | |
| 3.3.2 Labels or Instructions (Level A) | Supports | |
| 4.1.1 Parsing (Level A) | Supports | |
| 4.1.2 Name, Role, Value (Level A) | Supports | |

Table 2: Success Criteria, Level AA

Notes:

| Criteria | Conformance Level | Remarks and Explanations |
|--|-------------------|---|
| 1.2.4 Captions (Live) (Level AA) | Not Applicable | |
| 1.2.5 Audio Description (Prerecorded) (Level AA) | Not Applicable | |
| 1.3.4 Orientation (Level AA 2.1 only) | Not Applicable | The list nature of ballots requires portrait orientation to reduce scrolling in order to comply with Voluntary Voting System Guidelines (see below) |
| 1.3.5 Identify Input Purpose (Level AA 2.1 only) | Supports | |
| 1.4.3 Contrast (Minimum) (Level AA) | Supports | |
| 1.4.4 Resize text (Level AA) | Supports | |
| 1.4.5 Images of Text (Level AA) | Supports | |
| 1.4.10 Reflow (Level AA 2.1 only) | Supports | |
| 1.4.11 Non-text Contrast (Level AA 2.1 only) | Supports | |
| 1.4.12 Text Spacing (Level AA 2.1 only) | Supports | |
| 1.4.13 Content on Hover or Focus (Level AA 2.1 only) | Nor Applicable | |
| 2.4.5 Multiple Ways (Level AA) | Not Applicable | |
| 2.4.6 Headings and Labels (Level AA) | Supports | |
| 2.4.7 Focus Visible (Level AA) | Supports | |
| 3.1.2 Language of Parts (Level AA) | Supports | |
| 3.2.3 Consistent Navigation (Level AA) | Supports | |
| 3.2.4 Consistent Identification (Level AA) | Supports | |
| 3.3.3 Error Suggestion (Level AA) | Supports | |
| 3.3.4 Error Prevention (Legal, Financial, Data) (Level AA) | Supports | |
| 4.1.3 Status Messages (Level AA 2.1 only) | Not Applicable | |

Table 3: Success Criteria, Level AAA

Notes:

| Criteria | Conformance Level | Remarks and Explanations |
|--|-------------------|---|
| 1.2.6 Sign Language (Prerecorded) (Level AAA) | Not Applicable | |
| 1.2.7 Extended Audio Description (Prerecorded) (Level AAA) | Not Applicable | |
| 1.2.8 Media Alternative (Prerecorded) (Level AAA) | Not Applicable | |
| 1.2.9 Audio-only (Live) (Level AAA) | Not Applicable | |
| 1.3.6 Identify Purpose (Level AAA 2.1 only) | Not Evaluated | |
| 1.4.6 Contrast (Enhanced) (Level AAA) | Not Evaluated | |
| 1.4.7 Low or No Background Audio (Level AAA) | Not Evaluated | |
| 1.4.8 Visual Presentation (Level AAA) | Not Evaluated | |
| 1.4.9 Images of Text (No Exception) (Level AAA) | Not Evaluated | |
| 2.1.3 Keyboard (No Exception) (Level AAA) | Not Evaluated | |
| 2.2.3 No Timing (Level AAA) | Supports | |
| 2.2.4 Interruptions (Level AAA) | Not Evaluated | |
| 2.2.5 Re-authenticating (Level AAA) | Not Evaluated | |
| 2.2.6 Timeouts (Level AAA 2.1 only) | Supports | After 4 1/2 minutes of inactivity, users are alerted to continue their session. |
| 2.3.2 Three Flashes (Level AAA) | Not Applicable | |
| 2.3.3 Animation from Interactions (Level AAA 2.1 only) | Not Applicable | |
| 2.4.8 Location (Level AAA) | Not Evaluated | |
| 2.4.9 Link Purpose (Link Only) (Level AAA) | Not Evaluated | |
| 2.4.10 Section Headings (Level AAA) | Supports | |
| 2.5.5 Target Size (Level AAA 2.1 only) | Not Evaluated | |
| 2.5.6 Concurrent Input Mechanisms (Level AAA 2.1 only) | Not Evaluated | |
| 3.1.3 Unusual Words (Level AAA) | Not Evaluated | |

Page 7 of 8

| Criteria | Conformance Level | Remarks and Explanations |
|--|-------------------|--------------------------|
| 3.1.4 Abbreviations (Level AAA) | Not Evaluated | |
| 3.1.5 Reading Level (Level AAA) | Not Evaluated | |
| 3.1.6 Pronunciation (Level AAA) | Not Evaluated | |
| 3.2.5 Change on Request (Level AAA) | Not Evaluated | |
| 3.3.5 Help (Level AAA) | Supports | |
| 3.3.6 Error Prevention (All) (Level AAA) | Not Evaluated | |

Page 8 of 8

The Voatz web apps (VWA) support these browser-compatible capabilities:

- NVDA, VoiceOver, TalkBack and JAWS screen readers
- Predictable layout and navigation
- Configurable font size
- Voice Control (on MacOS)
- Speech-to-Text (for write-ins)
- Flexible session timeout limitations

Please refer to the Accessibility Conformance Reports (VPAT) in the 'Additional Resources' section below for detailed WCAG information.

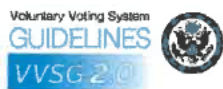
Development Methodology

Voatz incorporates accessibility and usability design throughout its development process:

- Voatz partnered with the National Center for Accessible Media for recommendations on usability and accessibility. Read more about the partnership [here](#).
- Voatz software is evaluated in-house by professional testers with previous WCAG 2.1 and certified voting systems experience. Read the VPAT statements below.
- Voatz engaged Pro V&V, an independent EAC-approved voting system testing laboratory (VSTL) certified in assessing accessibility compliance with the U.S. Election Assistance Commission's Voluntary Voting System Guidelines (VVSG 1.1). View Pro V&V's Accessibility and Usability Test Report [here](#).
- In addition to rigorous internal testing and remediation, Voatz continuously incorporates feedback from accessibility experts and voters with disabilities who have participated in elections using Voatz.

Additional Resources

- Pro V&V's [Accessibility and Usability Test Report](#)
- [VPAT Report \(Native Mobile Apps\)](#)
- [VPAT Report \(Browser App\)](#)
- An accessible PDF of the information on this page is available for download [here](#)





Voatz Elections Platform - Accessible Voting

The Voatz mobile elections platform provides robust support for all voters, including those with visual, cognitive, mobility, and dexterity disabilities. Hearing is not required for voting (unless using a screen reader). To the greatest extent possible, we comply with all applicable accessibility standards and guidelines and provide verification in independent reports.

Voatz follows these industry and regulatory accessibility standards:

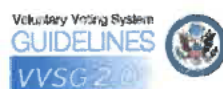
- EAC Voluntary Voting System Guidelines (VVSG v1.1) for usability and accessibility
- Worldwide Web Consortium's Web Content Accessibility Guidelines 2.1 level AA (WCAG 2.1/AA)
- Apple iOS, Android, MacOS and Windows best practices for accessibility

The Voatz mobile apps (VMA) support these native smartphone capabilities:

- VoiceOver and TalkBack screen readers
- Predictable layout and navigation
- Configurable font size
- Voice Control (on iOS)
- Speech-to-Text (for write-ins)
- Flexible session timeout limitations

Demonstration videos of accessibility features for blind or dexterity-impaired voters is available on our corporate website's Accessibility Statement [here](#).

Please refer to the Accessibility Conformance Reports (VPAT) in the 'Additional Resources' section below for detailed WCAG information.



Voatz Collaborates with WGBH's National Center for Accessible Media to Make Voting Accessible to Voters with Disabilities and Citizens Residing Overseas



NEWS PROVIDED BY
Voatz →
04 Nov, 2019, 10:58 ET

BOSTON, Nov. 4, 2019 /PRNewswire/ -- Voatz, a Boston-based elections company focused on secure mobile voting, announced a collaboration with the Carl and Ruth Shapiro Family National Center for Accessible Media at WGBH Educational Foundation (NCAM) to test the accessibility features of the company's secure mobile voting application.

The mobile voting application, available on compatible Android and iOS devices, allows deployed military personnel and overseas U.S. citizens, as well as people with disabilities, to conveniently and securely vote in elections with their smartphones from virtually anywhere in the world.

"We're proud to collaborate with NCAM to help make sure people with disabilities have accessible means to raise their voices in elections," said Nimit Sawhney, Voatz co-founder and CEO. "For too long, the needs of citizens with disabilities have largely been ignored in the perceived conflict between security and convenience. Voatz believes that citizens with disabilities deserve to take advantage of the advanced accessibility features available on modern smartphones. Democracy is at its best when all citizens can vote securely without limitation—physical or geographic."

Secured with blockchain technology and rigorously tested for ease of use, the app allows eligible users the option to forgo inaccessible paper ballots currently submitted by postal mail, facsimile or email. The Voatz app provides voters with an auditable confirmation and produces a fully marked paper ballot for tabulation, thereby providing unprecedented levels of end-to-end audibility and verifiability.

"In our tests, we have found Voatz's platform to be highly accessible," said Donna A. Danielewski, Ph.D., Senior Director of NCAM. "It allows individuals with disabilities to participate in the democratic voting process in a clear and accessible way. We look forward to continuing to work with Voatz in testing the platform as they work to bring it to more markets."

About NCAM

For nearly three decades, the National Center for Accessible Media (NCAM) has been a national leader in making digital media accessible for people with disabilities. The team in NCAM—with more than 150 years of combined experience in accessibility—are pioneers, inventors, and problem-solvers, frequently anticipating and creating solutions for tomorrow's technology challenges.

About WGBH

WGBH is America's preeminent public broadcaster and the largest producer of PBS content for TV and the Web, including *Frontline*, *Nova*, *American Experience*, *Masterpiece*, *Antiques Roadshow*, *Arthur*, and more than a dozen other prime-time, lifestyle, and children's series. WGBH also is a major source of digital content and programs for public radio through PRX, including *The World* and *Innovation Hub*; a leader in educational multimedia with PBS LearningMedia™, providing the nation's educators with free, curriculum-based digital content; and a pioneer in services that make media accessible to deaf, hard of hearing, blind and visually impaired audiences. WGBH has been recognized with hundreds of honors, including Emmys, Peabodys, duPont-Columbia Awards and Oscars. More info at www.wgbh.org.

About Voatz

Voatz is an award-winning mobile elections platform that leverages military-grade technology (including biometrics and a blockchain-based infrastructure) to increase accessibility and security in elections. Voatz has run more than 50 elections with state and local governments, universities, nonprofits, and both major state political parties for convention voting. Last year, Voatz partnered with [West Virginia](#) to empower deployed military and overseas citizens to vote, marking the first mobile votes in U.S. history. In 2019 Voatz expanded its pilots to [Denver](#) and [Utah](#), both of which held [citizen's public-facing audits](#), hosted by the National Cybersecurity Center. Recently, two counties in [Oregon](#) have also started to pilot the Voatz platform. All pilots have led to an increased turnout and in the case of Denver, 100% of voters responding to a post-election [survey](#) said they preferred this method of voting to any other. Learn more [here](#).

SOURCE Voatz



7. Attachment H: Full ProV&V Report

Summary of the report:

Letter Report



To: Linda Hutchinson - Voatz, Inc.
From: Wendy Owens - Pro V&V, Inc.
CC: Michael Walker - Pro V&V, Inc.
Date: July 16, 2020
Subject: Voatz Remote Accessible Ballot Delivery, Marking and Return (RABDMR) System

Dear Ms. Hutchinson:

Pro V&V is providing this letter to report the results of the evaluation of the Voatz Remote Accessible Ballot Delivery, Marking and Return (RABDMR) System to the applicable requirements in the U.S. Election Assistance Commission (EAC) 2015 Voluntary Voting System Guidelines (VVSG), Version 1.1 and the manufacturer-stated requirements set forth in the system documentation. The scope of the evaluation incorporated a sufficient spectrum of functional tests to verify that the RABDMR features and applications conform to the defined requirements.

The evaluation was conducted in two phases. Phase 1 consisted of a Usability and Accessibility Review. Phase 2 consisted of the following: 1) Verify that the RABDMR performs as documented in the provided system technical documentation, 2) Evaluate the RABDMR System as it relates to voter experience and transmission of the voter's selection to the jurisdiction, 3) Execute system use cases to evaluate system functionality, 4) Source Code Review, 5) Physical Configuration Audit (PCA), 6) Functional Configuration Audit (FCA), 7) System Integration Testing, including Accuracy Testing and Regression Testing, 8) Security Testing, and 9) Telecommunications Testing. All tests were performed to the VVSG 1.1 standards with the exception of the Accuracy Test, which was conducted on an abbreviated scale.

Based on the results obtained during the test campaign, Pro V&V determined the RABDMR, as presented for evaluation, meets the applicable requirements set forth for voting systems in the U.S. Election Assistance Commission (EAC) 2015 Voluntary Voting System Guidelines (VVSG), Version 1.1, with clarifications or exceptions noted in Section 4.0 of the final version of Pro V&V Test Report TR v. 01-02-VTZ-001-02.

Should you require additional information or would like to discuss this matter further, please contact me at 256-713-1111.

Sincerely,

Wendy Owens

Wendy Owens
VSTL Program Manager
wendy.owens@provandv.com

Please see the following pages for the full report.



6705 Odyssey Drive
Suite C
Huntsville, AL 35806
Phone (256)713-1111
Fax (256)713-1112

Test Report for Test and Evaluation of the VoatzTM Remote Accessible Ballot Marking (RABM) System

Phase 1
Version: 02
Date: 05/28/2020

SIGNATURES

Approved by:

Michael L. Walker
Michael Walker, VSTL Project Manager

05/28/2020

Date

Approved by:

Wendy Owens
Wendy Owens, VSTL Program Manager

05/28/2020

Date

REVISIONS

| Revision | Description | Date |
|----------|-----------------------------------|------------|
| 00 | Initial Release | 05/20/2020 |
| 01 | Updates per manufacturer comments | 05/22/2020 |
| 02 | Updates per manufacturer comments | 05/28/2020 |
| | | |
| | | |
| | | |

Table of Contents

| | | |
|------------|---|------------|
| 1.0 | INTRODUCTION..... | 1 |
| 1.1 | SCOPE..... | 1 |
| 1.2 | REFERENCES..... | 2 |
| 1.2 | TERMS AND ABBREVIATIONS..... | 2 |
| 2.0 | TESTING OVERVIEW..... | 3 |
| 2.1 | SYSTEM OVERVIEW..... | 3 |
| 2.2 | BLOCK DIAGRAMS..... | 4 |
| 2.3 | TEST CONFIGURATION..... | 5 |
| 3.0 | MATERIALS REQUIRED FOR TESTING..... | 5 |
| 3.1 | SOFTWARE..... | 5 |
| 3.2 | EQUIPMENT..... | 6 |
| 3.3 | TECHNICAL DATA PACKAGE..... | 6 |
| 3.4 | TEST SUPPORT MATERIALS..... | 6 |
| 4.0 | TEST PROCESS AND RESULTS..... | 7 |
| 4.1 | USABILITY AND ACCESSIBILITY TESTING..... | 7 |
| 5.0 | CONCLUSION..... | 9 |
| | ATTACHMENT A – REQUIREMENTS MATRIX..... | A-1 |
| | ATTACHMENT B – VOATZ EBDM RESPONSES..... | B-1 |

1.0 INTRODUCTION

The purpose of this Test Report is to document the procedures that Pro V&V, Inc. followed to perform the Phase 1 evaluation of the Voatz Remote Accessible Ballot Marking (RABM) System to the applicable requirements in the U.S. Election Assistance Commission (EAC) 2015 Voluntary Voting System Guidelines (VVSG), Version 1.1 and the manufacturer-stated requirements set forth in the system documentation. The RABM is categorized by the VVSG as an Electronically-assisted Ballot Marker (EBM) and is defined by the VVSG as follows:

Accessible Voting Station that produces an executed, human-readable paper ballot as a result, and that does not make any other lasting record of the voter's votes.

The test campaign is being conducted in two phases:

- Phase 1 assesses the RABM's Usability & Accessibility compliance in this Test Report.
- Phase 2 includes all elements of the approved Test Plan inclusive of Phase 1.

Pro V&V Test Plan TP v. 01-02-VTZ-001-01.01 was utilized as the guiding document during test performance. During testing, minor system modifications, such as revised system documentation or software versions, may have been incorporated. This test report reflects Phase 1 testing completed and details the final versions of all technical documentation and system components and supersedes the approved test plan.

Unless otherwise annotated, all testing was conducted at the Pro V&V test facility located in Huntsville, AL, by personnel verified by Pro V&V to be qualified to perform the test

1.1 Scope

The scope of the testing event incorporated a sufficient spectrum of functional tests to verify that the RABM features and applications conform to the defined requirements. Specifically, the testing event had the following goals:

Phase 1

- Usability and Accessibility Review

Phase 2

- Verify that the RABM performs as documented in the provided system technical documentation
- Evaluate the RABM System as it relates to voter experience and transmission of the voter's selection to the jurisdiction.

- Execute system use cases to evaluate system functionality.
- Source Code Review
- Physical Configuration Audit (PCA)
- Functional Configuration Audit (FCA)
- System Integration Testing, including Accuracy Testing and Regression Testing
- Security Testing
- Telecommunications Testing

1.2 References

- Voatz Mobile Elections Platform Proposed Test Plan
- Implementation Statement
- Election Assistance Commission 2015 Voluntary Voting System Guidelines (VVSG) Version 1.1, Volume I, "Voting System Performance Guidelines", and Volume II, "National Certification Testing Guidelines"
- Election Assistance Commission Testing and Certification Program Manual, Version 2.0
- Election Assistance Commission Voting System Test Laboratory Program Manual, Version 2.0
- National Voluntary Laboratory Accreditation Program NIST Handbook 150, 2016 Edition, "NVLAP Procedures and General Requirements (NIST Handbook 150)", dated July 2016
- National Voluntary Laboratory Accreditation Program NIST Handbook 150-22, 2008 Edition, "Voting System Testing (NIST Handbook 150-22)", dated May 2008
- United States 107th Congress Help America Vote Act (HAVA) of 2002 (Public Law 107-252), dated October 2002
- Pro V&V, Inc. Quality Assurance Manual, Revision 7.0
- [Hunt Engagement Summary](#)
- System Technical Data Package (*A listing of the documents submitted for this test campaign is listed in Section 4.6 of this Test Plan*)

1.2 Terms and Abbreviations

This subsection lists terms and abbreviations relevant to the hardware, the software, or this Test Plan.

"COTS" – Commercial Off-The-Shelf

"EAC" – United States Election Assistance Commission

"EBM" – Electronically-assisted Ballot Marker

- “FCA” – Functional Configuration Audit
- “HAVA” – Help America Vote Act
- “ISO” – International Organization for Standardization
- “PCA” – Physical Configuration Audit
- “QA” – Quality Assurance
- “TDP” – Technical Data Package
- “VSTL” – Voting System Test Laboratory
- “VVSG” – Voluntary Voting System Guidelines

2.0 TESTING OVERVIEW

The evaluation of the RABM System addressed each of the test goals in the following manner:

Table 2-1: Testing Overview

| Test Goal | Testing Response |
|---|--|
| Perform a Usability and Accessibility Review | A usability and accessibility review was performed on the submitted system. This review focused on the usability of the system as evaluated against the requirements matrix. |

2.1 System Overview

The following sections contain a product description and an overview of the design methodology of the RABM System, as taken from the system technical documentation.

The Voatz Mobile Elections Platform is a configurable, cloud-based Remote Accessible Ballot Marking (RABM) system that:

- Delivers blank ballots to eligible voters
- Allows voters to mark their selections accessibly and verify their selections
- Returns the marked ballots to the jurisdiction according to the statutory provisions of the voter’s state.

Jurisdictions can specify the method of returning voted ballots based on their state’s statutory requirements and include any state-specific legal documents (such as signed affidavits or checkbox waivers.)

To determine eligibility to vote, voter registration records are incorporated into the Voatz Platform by importing the data. Typically, an indicator in the voter file designates which subset of all voters are eligible to vote remotely. While the Voatz system can remotely confirm the validity of a voter’s credentials (e.g. driver’s license) and verify the person presenting their credentials is the same person on the credential, it does not by itself determine whether individual

voters are authorized to vote in the current election. This determination must be made by the jurisdiction and indicated in the voter records. Voatz then limits voting to those individuals so designated.

The Voatz RABM system uses all commercial off-the-shelf (COTS) hardware; no proprietary hardware is required. Ballots are marked securely and anonymously via a mobile app on the voter’s smartphone—the Voatz Mobile App (VMA).

Marked ballots are returned electronically to the jurisdiction, along with any other required documents. The jurisdiction then prints the returned marked ballots as optical scannable ballots on official ballot stock for tabulation by the primary voting system. Voters receive a password protected ballot receipt that lists their selections and contains an anonymous ID; the jurisdiction receives an identical anonymized copy. Only the voter knows the anonymous ID that is linked to them.

The RABM is categorized by the Voluntary Voting System Guidelines (VVSG) version 1.1 as an Electronically assisted Ballot Marker (EBM) and is defined as follows:

Accessible Voting Station that produces an executed, human-readable paper ballot as a result, and that does not make any other lasting record of the voter’s votes on the device.

Voatz has provided responses to a recently issued guidance to election officials regarding electronic ballot delivery and marking circulated by the EAC. The information on the RABM system functionality as provided by Voatz is presented in Attachment B.

2.2 Block Diagram

The process flow of the system is depicted in Figure 1-1.

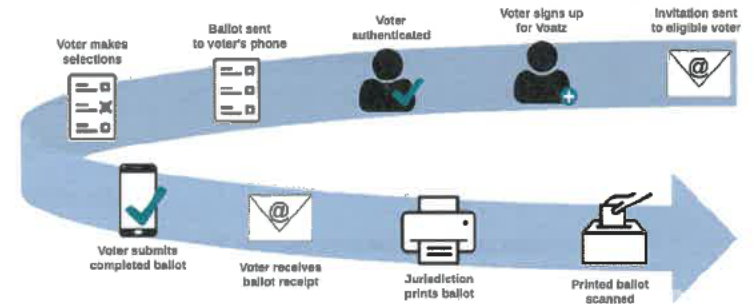
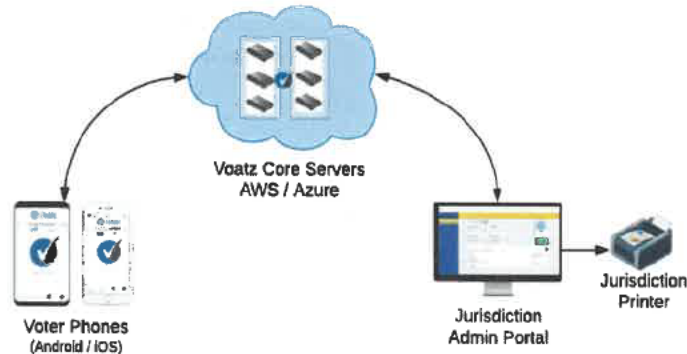


Figure 1-1. RABM Process Flow

2.3 Test Configuration

Voatz RABM Basic Architecture



The testing campaign utilized various models of cell phones, with the Voatz Mobile App (VMA) installed. To access the back-end environment, one Pro V&V laptop was utilized. Ballots generated during testing were printed as optical scannable ballots using an OKI printer.

3.0 MATERIALS REQUIRED FOR TESTING

The following sections list all materials required during the test engagement.

The materials required for testing of the RABM System included all materials to enable the test campaign to occur. This included the applicable hardware and software as well as the TDP, test support materials, and deliverable materials, as described in the following subsections.

3.1 Software

This subsection lists the proprietary and COTS software provided by the manufacturer as part of the test campaign.

Table 3-1. RABM System Software

| Firmware/Software | Version |
|------------------------------------|---------------|
| Voatz Mobile App (VMA) for iOS | 1.031 (186) |
| Voatz Mobile App (VMA) for Android | 1.1.120 (155) |
| Voatz Admin Portal | 1.0.42 |

3.2 Equipment

This subsection lists the COTS equipment provided by the manufacturer as part of the test campaign.

For COTS equipment, every effort was made to verify that the COTS equipment has not been modified for use. This was accomplished by performing research using the COTS equipment manufacturer's websites based on the serial numbers for each piece of equipment. Assigned test personnel evaluated COTS hardware, system software and communications components for proven performance in commercial applications other than voting. For smartphones, the device information was compared to the supported configurations in the campaign. Physical external and internal examination was also performed when the easily accessible without the possibility of equipment damage. Hard drives, RAM memory, and other components were examined to verify that the components match the information found on the COTS equipment manufacturer's websites. A factory reset was performed on smartphones prior to testing.

Table 3-2. RABM System Equipment

| Component | Model | Serial Number | OS |
|-----------------|------------|-----------------|---------------|
| iPhone 6s | MKRX2LL/A | C6KRL9HVGRY8 | iOS 13.3.1 |
| iPhone 6 | MG4Q2LL/A | FFMP901MG5MF | iOS 12.4.5 |
| LG Phoenix 4 | LM-X210APM | 912VTSM766174 | Android 8.1.0 |
| Moto e6 | XT2005-1PP | 352176100933789 | Android 9.0 |
| Okidata Printer | 432dn-B | AK88034459C0 | N/A |

3.3 Technical Data Package

A listing of all documents contained in the system TDP relevant to this Phase I report is provided in Table 3-3.

Table 3-3. TDP Documents

| Document | Description |
|--|---|
| Vendor Proposed VSTL Usability and Accessibility Test Plan | Provides suggested scope, approach, and criteria of intended testing activities. |
| System Hardware and Software Specification | Provides detailed specifications of the hardware and software components of the system. |

3.4 Test Support Materials

This subsection lists the test materials required to execute the required tests throughout the test campaign.

The following materials were supplied by Voatz to facilitate testing:

- Ballot Paper
- Printer Ink

- Other materials and equipment as required

4.0 TEST PROCESS AND RESULTS

Testing of the RABM System submitted for evaluation was performed to verify that the System conforms to the applicable requirements in the U.S. Election Assistance Commission (EAC) 2015 Voluntary Voting System Guidelines (VVSG), Version 1.1 and the manufacturer-stated requirements set forth in the system documentation. The VVSG 1.1 Requirements Matrix generated for this test campaign was used as a guide to determine the specific tests to be performed. Pro V&V developed test procedures designed to evaluate the system being tested against the stated requirements. The test cases were performed using three types of elections: Primary, General, and Ranked Choice voting. The test procedures were executed independently.

The evaluation area for this test engagement is summarized in the subsection below.

4.1 Usability and Accessibility Testing

The usability testing focuses on the usability of the system being tested. Usability is defined generally as a measure of the effectiveness, efficiency, and satisfaction achieved by a specified set of users with a given product in the performance of specified tasks. In the context of voting, the primary user is the voter, the product is the RABM, and the task is the correct recording of the voter ballot selections. Additional requirements for task performance are independence and privacy: the voter should normally be able to complete the voting task without assistance from others, and the voter selections should be private. Accessibility evaluates the requirements for accessibility. These requirements are intended to address HAVA 301 (a) (3) (B).

Usability and Accessibility Testing included system use cases for the following areas:

- Usability Support
- Cognitive Disabilities
- Perceptual Disabilities
- Interaction Disabilities
- System and Voter Wait Times
- Language Support
- General Accessibility Assistance
- Limited Vision Assistance
- Blind Voter Assistance
- Motor Control Difficulty Assistance
- Limited Hearing Assistance
- English Illiteracy Assistance
- Limited Speech Assistance

Summary Findings

The RABM System was evaluated against the VVSG 1.1 Requirements Matrix generated for this test campaign. Based on the results obtained, the RABM was determined to meet the applicable

Usability and Accessibility requirements. The following observations/exceptions were noted during the review:

Table 4-1 Summary Findings

| Requirement | Notes |
|--|---|
| Usability Review | |
| Volume I, Section 3.2.3.1 <i>Privacy at the polls</i> | |
| 3.2.3.1.a, 3.2.3.1.b, 3.2.3.1.c, 3.2.3.1.d, 3.2.3.1.e | These requirements are not applicable due to there being no polling place. |
| Volume I, Section 3.2.6.1 <i>Timing</i> | |
| 3.2.6.1.a, 3.2.6.1.b, 3.2.6.1.c | Note: Timing was tested concurrent with a simulated load of 500 other users voting. |
| 3.2.6.1.f | The timing of the alert and expiration of session pass. However, there is no poll worker, thus there is no poll worker intervention. The user is required to login again. |
| Volume I, Section 3.2.7a <i>Alternative Languages</i> | |
| 3.2.7a.i.v | Any external reports to be reviewed pending final TDP. |
| Accessibility Review | |
| Volume I, Section 3.3.2 <i>Enhanced visual interfaces</i> | |
| 3.3.2.c.i, 3.3.2.c.ii | Phone contains capability. |
| Volume I, Section 3.3.3 <i>Audio-tactile interfaces</i> | |
| 3.3.3.a & 3.3.3.a.i | Any external test reports to be reviewed pending final TDP. |
| 3.3.3.b, 3.3.3.b.i, 3.3.3.b.ii, 3.3.3.b.iii, 3.3.3.b.iv, 3.3.3.b.v, 3.3.3.c.i, 3.3.3.c.ii, 3.3.3.iii | The system does not contain an ATI; therefore these requirements are not applicable. |
| 3.3.3.c.iv, 3.3.3.c.v, 3.3.3.c.vi | These requirements relate to volume control and audio presentation and were not tested, as they are dependent upon the device being utilized. |
| 3.3.3.f, 3.3.3.g | These requirements relate to accessible voting stations and are not applicable. |
| Volume I, Section 3.3.4 <i>Enhanced input and control characteristics</i> | |
| 3.3.4.a, 3.3.4.d | These requirements relate to accessible voting stations and are not applicable. |
| 3.3.4.b | Partial Pass - Only on iPhone 6s & higher and iOS 13 & higher support hands-free operation using Voice Control. Android devices do not have this capability yet. However, both devices support limited dexterity with VoiceOver or TalkBack navigation (audio can be turned off.) |
| Volume I, Section 3.3.5 <i>Design for mobility aids</i> | |
| 3.3.5 | These requirements relate to accessible voting stations and are not applicable. |

Table 4-1 Summary Findings *(continued)*

| Requirement | Notes |
|---|---|
| Volume I, Section 3.3.6 <i>Enhanced auditory interfaces</i> | |
| 3.3.6.c | These requirements relate to volume control and audio presentation and were not tested, as they are dependent upon the device being utilized. |
| Volume I Section 3.3.10 <i>Summative Usability Report</i> | |
| 3.3.10.c.i, 3.3.10.a, 3.3.10.b, 3.3.10.c | Any external test reports to be reviewed pending final TDP. |

The results of the evaluation are presented in Attachment A. Each item listed in the table above will have an asterisk in Attachment A.

5.0 CONCLUSION

Based on the results obtained during the test campaign, Pro V&V determined the RABM, as presented for evaluation, meets the Usability and Accessibility requirements set forth for voting systems in the U.S. Election Assistance Commission (EAC) 2015 Voluntary Voting System Guidelines (VVSG), Version 1.1, with the clarifications or exceptions noted in Table 4-1.

ATTACHMENT A - REQUIREMENTS MATRIX

Table A-1 Usability Review Matrix Requirements

| Requirement | Description | Result |
|------------------|---|--------|
| Usability | | |
| 3 | Usability Requirements | |
| 3.1.1 | Purpose | |
| 3.1.1.a | All eligible voters shall have access to the voting process without discrimination. | Pass |
| 3.1.1.a.i | The voting process shall be accessible to individuals with disabilities. The voting process includes access to the polling place, instructions on how to vote, initiating the voting session, making ballot selections, review of the ballot, final submission of the ballot, and getting help when needed. | Pass |
| 3.1.1.b | Each cast ballot shall accurately capture the selections made by the voter. | Pass |
| 3.1.1.b.i | The ballot shall be presented to the voter in a manner that is clear and usable. Voters should encounter no difficulty or confusion regarding the process for recording their selections. | Pass |
| 3.1.1.c | The voting process shall preserve the secrecy of the ballot. | Pass |
| 3.1.1.c.i | The voting process shall preclude anyone else from determining the content of a voter's ballot without the voter's cooperation. If such a determination is made against the wishes of the voter, then his or her privacy has been violated. | Pass |
| 3.2 | General usability requirements | |
| 3.2.a and 3.2.b | The voting process shall provide a high level of usability for voters. Accordingly, voters shall be able to negotiate the process effectively, efficiently, and comfortably. The mandatory voting system standards mandated in HAVA Section 301 relate to the interaction between the voter and the voting system: | |
| 3.2.a.1.A | Except as provided in subparagraph (B), the voting system (including any lever voting system, optical scanning voting system, or direct recording electronic system) shall — | |
| 3.2.a.1.A.i | Permit the voter to verify (in a private and independent manner) the votes selected by the voter on the ballot before the ballot is cast and counted. | Pass |
| 3.2.a.1.A.ii | Provide the voter with the opportunity (in a private and independent manner) to change the ballot or correct any error before the ballot is cast and counted (including the opportunity to correct the error through the issuance of a replacement ballot if the voter was otherwise unable to change the ballot or correct any error). | Pass |
| 3.2.a.1.A.iii | If the voter selects votes for more than one candidate in a single office: | |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------------|--|--------|
| 3.2.a.1.A.iii.I | Notify the voter that the voter has selected more than one candidate for a single office on the ballot. | Pass |
| 3.2.a.1.A.iii.II | Notify the voter before the ballot is cast and counted of the effect of casting multiple votes for the office. | Pass |
| 3.2.a.1.A.iii.III | Provide the voter with the opportunity to correct the ballot before the ballot is cast and counted. | Pass |
| 3.2.a.1.B | A state or jurisdiction that uses a paper ballot voting system, a punch card voting system, or a central count voting system (including mail-in absentee ballots and mail-in ballots), may meet the requirements of subparagraph (A) (iii) by: | |
| 3.2.a.1.B.i | Establishing a voter education program specific to that voting system that notifies each voter of the effect of casting multiple votes for an office. | Pass |
| 3.2.a.1.B.ii | Providing the voter with instructions on how to correct the ballot before it is cast and counted (including instructions on how to correct the error through the issuance of a replacement ballot if the voter was otherwise unable to change the ballot or correct any error). | Pass |
| 3.2.a.1.C | The voting system shall ensure that any notification required under this paragraph preserves the privacy of the voter and the confidentiality of the ballot. | Pass |
| 3.2.1 | General usability | |
| 3.2.1.a | The voting system shall support voters in the task of effectively completing their ballots. | Pass |
| 3.2.1.b | The features of the voting system shall not contribute to the commission of voter error within the voting session. | Pass |
| 3.2.2 | Functional Capabilities | |
| | The usability of the voting process is enhanced by the presence of certain functional capabilities. These capabilities differ somewhat depending on whether or not the system presents an editable interface within which voters can easily change their votes (typically an electronic screen) or an interface in which voters must obtain a new ballot to make changes (typically a manually-marked paper ballot). | |
| 3.2.2.a | If the voter selects more than the allowable number of choices within a contest, the voting system shall notify the voter of the effect of this action before the ballot is cast and counted. | Pass |
| 3.2.2.c | The voting system shall provide the voter the opportunity to correct the ballot for either an undervote or overvote before the ballot is cast and counted. | Pass |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|---|--------|
| 3.2.2.b | The voting system shall allow the voter, at the voter's choice, to submit an undervoted ballot without correction. | Pass |
| 3.2.2.d | If and only if the voter successfully casts or prints the ballot, then the electronic ballot interface or PCOS system shall so notify the voter. | Pass |
| 3.2.2.1 | Editable electronic ballot interfaces | |
| | Voting systems such as DREs and EBMs present voters with an editable interface, allowing them to easily change their votes prior to final casting of the ballot. | |
| 3.2.2.1.a | The electronic ballot interface shall prevent voters from selecting more than the allowable number of choices for each contest. | Pass |
| 3.2.2.1.b | The electronic ballot interface shall provide feedback to the voter, before final casting or printing of the ballot, that identifies specific contests for which the voter has selected fewer than the allowable number of choices (i.e., undervotes). | Pass |
| 3.2.2.1.c | The electronic ballot interface shall provide the voter the opportunity to correct the ballot before it is cast or printed. The electronic ballot interface shall allow the voter to make these corrections without assistance. The corrections to be supported include modifying an undervote and changing a vote from one candidate to another. | Pass |
| 3.2.2.1.d | The electronic ballot interface shall allow the voter to change a vote within a contest before advancing to the next contest. | Pass |
| 3.2.2.1.e | The electronic ballot interface shall provide navigation controls that allow the voter to advance to the next contest or go back to the previous contest before completing a vote on the contest(s) currently being presented (whether visually or aurally). | Pass |
| 3.2.2.1.f | If the voter takes the appropriate action to cast a ballot, but the DRE does not accept and record it successfully, including failure to store the ballot image, then the DRE shall so notify the voter and provide clear instruction as to the steps the voter should take to cast the ballot. | Pass |
| 3.2.2.1.g | If the electronic ballot interface generates a paper record (or some other durable, human-readable record) that can be the official ballot or determinative vote record, then the voting system shall allow the voter to verify that record using the same access features used by the voter to vote the ballot. | Pass |
| 3.2.2.2 | Non-Editable ballot interfaces | |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|---------------|--|----------------|
| | Non-Editable interfaces, such as manually-marked paper ballots, do not have the same flexibility as do editable interfaces. Nonetheless, certain features are required, especially in the case of precinct-based optical scanners. | |
| 3.2.2.2.a | The PCOS system shall be capable of providing feedback to the voter that identifies specific contests for which the voter has made more than the allowable number of votes (i.e., overvotes). | Not Applicable |
| 3.2.2.2.b | The PCOS system shall be capable of providing feedback to the voter that identifies specific contests for which the voter has made fewer than the allowable number of votes (i.e., undervotes). The system shall provide a means for an authorized election official to deactivate this capability entirely and by contest. However, if a ballot is submitted with all the contests on one side left blank, notification to the voter is performed as described in requirement 3.2.2.2.c | Not Applicable |
| 3.2.2.2.c | The PCOS system shall be capable of notifying the voter that he or she has submitted a paper ballot that is blank on one or both sides. The system shall provide a means for an authorized election official to deactivate this capability. | Not Applicable |
| 3.2.2.2.d | If the PCOS system has notified the voter that a potential error condition (such as an overvote, undervote, or blank ballot) exists, the system shall then allow the voter to correct the ballot or to submit it as is. | Not Applicable |
| 3.2.2.2.e | Paper-based precinct tabulators shall be able to identify a ballot containing marginal marks. When such a ballot is detected, the tabulator shall: | Not Applicable |
| 3.2.2.2.e.i | Return the ballot to the voter; | Not Applicable |
| 3.2.2.2.e.ii | Provide feedback to the voter that identifies the specific contests for which a marginal mark was detected; and | Not Applicable |
| 3.2.2.2.e.iii | Allow the voter either to correct the ballot or to submit the ballot "as is" without correction, at the voter's choice. | Not Applicable |
| 3.2.2.2.f | Software used to format optical scan ballots shall constrain the size and contrast of all target areas to conform to the following requirements: | Not Applicable |
| 3.2.2.2.f.i | The target shall be no less than 3 mm across in any direction | Not Applicable |
| 3.2.2.2.f.ii | The contrast ratio between the target area boundaries and the surrounding space shall be no less than 10:1. | Not Applicable |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|---|------------------------|
| 3.2.2.2.g | If the voter takes the appropriate action to cast a ballot, but the PCOS system does not accept and record it successfully, including failure to read the ballot or to transport it into the ballot box, the PCOS shall so notify the voter. | <i>Not Applicable</i> |
| 3.2.3 | Privacy | |
| 3.2.3 | The voting process shall preclude anyone else from determining the content of a voter's ballot, without the voter's cooperation. | |
| 3.2.3.1 | Privacy at the polls | |
| 3.2.3.1.a | The voting system shall prevent others from determining the contents of a ballot. | <i>Not Applicable*</i> |
| 3.2.3.1.b | The voting system shall support ballot privacy during the voting session and ballot submission. | <i>Not Applicable*</i> |
| 3.2.3.1.c | During the voting session, the audio interface of the voting system shall be audible only to the voter. | <i>Not Applicable*</i> |
| 3.2.3.1.d | The voting system shall issue all warnings in a way that preserves the privacy of the voter and the confidentiality of the ballot. | <i>Not Applicable*</i> |
| 3.2.3.1.e | The voting system shall not issue a receipt to the voter that would provide proof to another of how the voter voted. | <i>Not Applicable*</i> |
| 3.2.3.2 | No recording of alternative format usage | |
| | When voters use non-typical ballot interfaces, such as large print or alternative languages, their anonymity may be vulnerable. To the extent possible, only the logical contents of their ballots should be recorded, not the special formats in which they were rendered. However, in the case of paper ballots, where the interface is the record, some format information is unavoidably preserved. | |
| 3.2.3.2.a | No information shall be kept within an electronic cast vote record that identifies any alternative language feature(s) used by a voter. | Pass |
| 3.2.3.2.b | No information shall be kept within an electronic cast vote record that identifies any accessibility feature(s) used by a voter. | Pass |
| 3.2.4 | Voter instructions, plain language, and information presentation | |
| | The features specified in this section are intended to minimize cognitive difficulties for voters. Voters should always be able to operate the voting system and understand the effect of their actions. Note that the "should" requirements in this section must be adhered to unless there is strong justification provided for making an exception. | |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|--|--------|
| 3.2.4.a | The voting system shall provide instructions for all its valid operations. | Pass |
| 3.2.4.b | The voting system shall provide a means for the voter to get help directly from the system at any time during the voting session. | Pass |
| 3.2.4.c | Instructional material for the voter shall conform to norms and best practices for plain language. | Pass |
| 3.2.4.c.i | Warnings and alerts issued by the voting system shall be distinguishable from other information and should clearly state: The nature of the problem; Whether the voter has performed or attempted an invalid operation or whether the voting system itself has malfunctioned in some way; and The set of responses available to the voter. | Pass |
| 3.2.4.c.ii | When an instruction is based on a condition, the condition should be stated first, and then the action to be performed. | Pass |
| 3.2.4.c.iii | The voting system should use familiar, common words and avoid technical or specialized words that voters are not likely to understand. | Pass |
| 3.2.4.c.iv | Each distinct instruction should be separated spatially from other instructions for visual or tactile interfaces, and temporally for auditory interfaces. | Pass |
| 3.2.4.c.v | The voting system should issue instructions on the correct way to perform actions, rather than telling voters what not to do. | Pass |
| 3.2.4.c.vi | The system's instructions should address the voter directly rather than use passive voice constructions. | Pass |
| 3.2.4.c.vii | The voting system should avoid the use of gender-based pronouns. | Pass |
| 3.2.4.d | Consistent with election law, the voting system shall support a process that does not introduce bias for or against any of the contest choices to be presented to the voter. In both visual and aural formats, the choices shall be presented in an equivalent manner. | Pass |
| 3.2.4.e | The voting system shall provide the capability to design a ballot with a high level of clarity and comprehensibility. | Pass |
| 3.2.4.e.i | The voting system should not visually present a single contest spread over two pages or two columns. | Pass |
| 3.2.4.e.ii | The ballot shall clearly indicate the maximum number of candidates for which one can vote within a single contest. | Pass |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|--|----------|
| 3.2.4.e.iii | The relationship between the name of a candidate and the mechanism used to vote for that candidate shall be consistent throughout the ballot. | Pass |
| 3.2.4.e.iv | The voting system should present instructions near to where they are needed. | Pass |
| 3.2.4.f | The use of color by the voting system shall agree with common conventions: (a) green, blue or white is used for general information or as a normal status indicator; (b) amber or yellow is used to indicate warnings or a marginal status; (c) red is used to indicate error conditions or a problem requiring immediate attention. | Pass |
| 3.2.4.g | When an icon is used to convey information, indicate an action, or prompt a response, it shall be accompanied by a corresponding linguistic label. | Pass |
| 3.2.5 | Visual display characteristics | |
| | The requirements of this section are designed to minimize perceptual difficulties for the voter. Some of these requirements are designed to assist voters with poor reading vision. These are voters who might have some difficulty in reading normal text, but are not typically classified as having a visual disability and thus might not be inclined to use the Acc-VS. | |
| 3.2.5.a | If the voting system uses an electronic display screen as the primary visual interface for the voter, the display shall have the following characteristics: | |
| 3.2.5.a.i | Flicker frequency NOT between 2 Hz and 55 Hz. Does not say "flashing elements" like NIST did? If so maybe one Dell reported concern in settings. | Untested |
| 3.2.5.a.ii | Minimum display brightness: 130 cd/m2 | Untested |
| 3.2.5.a.iii | Minimum display darkroom 7x7 checkerboard contrast: 150:1 | Untested |
| 3.2.5.a.iv | Minimum display pixel pitch: 85 pixels/inch (0.3 mm/pixel) | Untested |
| 3.2.5.a.v | Minimum display area 700 cm2 | Untested |
| 3.2.5.a.vi | Antiglare screen surface that shows no distinct virtual image of a light source | Untested |
| 3.2.5.a.vii | Minimum uniform diffuse ambient contrast ratio for 500 lx illuminance: 10:1 | Untested |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|--|----------|
| 3.2.5.b | Any aspect of the voting system voter interface that is adjustable by either the voter or poll worker, including font size, color, contrast, audio volume, or rate of speech, shall automatically reset to a standard default value upon completion of that voter's session. For the Acc-VS with an electronic image display, the aspects include synchronized audio/video mode and non-manual input mode. | Pass |
| 3.2.5.c | If any aspect of a voting system is adjustable by either the voter or poll worker, there shall be a mechanism to allow the voter to reset all such aspects to their default values while preserving the current votes. | Pass |
| 3.2.5.d | For all text intended for voters or poll workers, the voting system shall provide a font with the following characteristics | |
| 3.2.5.d.i | Height of capital letters at least: 3.0 mm | Untested |
| 3.2.5.d.ii | x-height of at least: 70% of cap height | Untested |
| 3.2.5.d.iii | Stroke width at least: 0.35 mm. | Untested |
| 3.2.5.e | A voting system that uses an electronic image display shall be capable of showing all information in at least two font sizes: | |
| 3.2.5.e.i | 3.0-4.0 mm cap height, with a corresponding x-height at least 70% of the cap height and a minimum stroke width of 0.35 mm; | Untested |
| 3.2.5.e.ii | 6.3-9.0 mm cap height, with a corresponding x-height at least 70% of the cap height and a minimum stroke width of 0.7 mm; under control of the voter. The system shall allow the voter to adjust font size throughout the voting session while preserving the current votes. | Untested |
| 3.2.5.f | Text intended for the voter should be presented in a sans serif font. | Pass |
| 3.2.5.g | Voting systems using paper ballots or paper verification records shall provide features that assist in the reading of such ballots and records by voters with poor reading vision. | |
| 3.2.5.g.i | The voting system may achieve legibility of paper records by supporting the printing of those records in at least two font sizes, 3.0-4.0mm and 6.3-9.0mm. | Untested |
| 3.2.5.g.ii | The system may achieve legibility of paper records by supporting magnification of those records. This magnification may be done by optical or electronic devices. The manufacturer may either: 1) provide the magnifier itself as part of the system, or 2) provide the make and model number of readily available magnifiers that are compatible with the system. | Untested |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|--|----------|
| 3.2.5.h | The colors in the default presentation shall support perception by voters and poll workers with color vision deficiencies, of all text, controls, and infographics or icons on the ballot or ballot interface. | |
| 3.2.5.h.i | The default visual display for voters and poll workers of a voting station with an electronic display shall have a luminosity contrast ratio between the foreground text and background color of at least 10:1 for all elements that visually convey information such as text, controls, and infographics or icons. For paper ballots, the contrast ratio shall be at least 10:1 as measured based on ambient lighting of at least 300 lx. | Untested |
| 3.2.5.h.ii | A voting station with an electronic display screen shall have a high contrast mode either as an initial setting or under the control of the voter. If the system allows the voter to adjust contrast during the voting session it shall preserve the current votes. High contrast is a luminosity contrast ratio between the foreground text and background color of at least 20:1. The high contrast mode shall use at least one of the following color combinations: o Black text on a white background o White text on a black background o Yellow text on a black background o Light cyan text on a black background | Untested |
| 3.2.5.i | Color coding shall not be used as the sole means of conveying information, indicating an action, prompting a response, or distinguishing a visual element. | Pass |
| 3.2.6 | Voter-interface interaction | |
| | The requirements of this section are designed to minimize interaction difficulties for the voter. | |
| 3.2.6.a | Voting machines with electronic image displays shall not require page scrolling by the voter. Discussion: This is not an intuitive operation for those unfamiliar with the use of computers. Even those experienced with computers often do not notice a scroll bar and miss information at the bottom of the "page." Voting systems may require voters to move to the next or previous "page." | Pass |
| 3.2.6.b | The voting machine shall provide unambiguous feedback regarding the voter's selection, such as displaying a checkmark beside the selected option or conspicuously changing its appearance. | Pass |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|---|---------------|
| 3.2.6.c | d. Input mechanisms shall be designed to minimize accidental activation. | Pass |
| 3.2.6.c.i | On touch screens, the sensitive touch areas shall have a minimum height of 0.5 inches and minimum width of 0.7 inches. The vertical distance between the centers of adjacent areas shall be at least 0.6 inches, and the horizontal distance at least 0.8 inches. | Untested |
| 3.2.6.c.ii | No key or control on a voting machine shall have a repetitive effect as a result of being held in its active position. | Pass |
| 3.2.6.l | Timing | |
| | These requirements address how long the system and voter wait for each other to interact. | |
| 3.2.6.l.a | The initial system response time of the electronic ballot interface shall be no greater than 0.5 seconds. | Pass* |
| 3.2.6.l.b | When the voter performs an action to record a single vote, the completed system response time of the electronic ballot interface shall be no greater than one second in the case of a visual response, and no greater than five seconds in the case of an audio response. | Pass* |
| 3.2.6.l.c | The completed system response time during a voter interaction with the visual display of the electronic ballot interface shall be no greater than 10 seconds. | Pass* |
| 3.2.6.l.d | If the electronic ballot interface has not completed its visual response within one second, it shall present to the voter, within 0.5 seconds of the voter's action, some indication that it is preparing its response. | Pass |
| 3.2.6.l.e | The electronic ballot interface shall detect and warn about lengthy voter inactivity during a voting session. Each electronic ballot interface shall have a defined and documented voter inactivity time, and that time shall be between two and five minutes. | Pass |
| 3.2.6.l.f | Upon expiration of the voter inactivity time, the electronic ballot interface shall issue an alert and provide a means by which the voter may receive additional time. The alert time shall be between 20 and 45 seconds. If the voter does not respond to the alert within the alert time, the electronic ballot interface shall go into an inactive state requiring poll worker intervention. | Partial Pass* |
| 3.2.7 | Alternative languages | |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|---|-----------------|
| | HAVA Section 301 (a)(4) states that the voting system shall provide alternative language accessibility pursuant to the requirements of Section 203 of the Voting Rights Act of 1965 (42 U.S.C. 1973aa-1a). Ideally every voter would be able to vote independently and privately, regardless of language. As a practical matter, alternative language access is mandated under the Voting Rights Act of 1975, subject to certain thresholds (e.g., if the language group exceeds 5% of the voting age population). Thus, election officials must ensure that the voting system they deploy is capable of handling the languages meeting the legal threshold within their districts. | |
| 3.2.7.a | The voting system shall be capable of presenting the ballot, contest choices, review screens, vote verification records, and voting instructions in any language declared by the manufacturer to be supported by the system. | Pass |
| 3.2.7.a.i | The electronic ballot interface should allow the voter to select among the available languages throughout the voting session while preserving the current votes. When presenting a choice of languages to the voter, the electronic ballot interface shall use the native name of each language. | Pass |
| 3.2.7.a.ii | Information presented to the voter in the typical case of English-literate voters (including instructions, warnings, messages, contest choices, and vote verification information) shall also be presented when an alternative language is being used, whether the language is written or an unwritten language presented aurally. | Pass |
| 3.2.7.a.iii | Any records, including paper ballots and paper verification records, shall have the information required to support auditing by poll workers and others who can read only English. | Pass |
| 3.2.7.a.iv | The manufacturer shall conduct summative usability tests for each of the voting system's supported languages, using subjects who are fluent in those languages but not fluent in English and shall report the test results, using the Common Industry Format, as part of the TDP. In addition, the usability test report shall be submitted to the EAC as part of the documentation manufacturers are required to file with the application to test a voting system. | Not Applicable* |
| 3.2.8 | Usability for poll workers | |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|-------------|---|----------------|
| | Voting systems are used not only by voters to record their votes, but also by poll workers who are responsible for set-up, operation while polls are open, light maintenance, and poll closing. Because of the wide variety of implementations, it is impossible to specify detailed design requirements for these functions. The requirements below describe general capabilities that all systems must support. | Untested |
| 3.2.8.a | Messages generated by the voting system for poll workers in support of the operation, maintenance, or safety of the system shall adhere to the requirements for clarity in Section 3.2.4 "Voter instructions, plain language and information presentation." | Untested |
| 3.2.8.1 | Operations | |
| | <p>Poll workers are responsible for opening polls, keeping the polls open and running smoothly during voting hours, and closing the polls afterwards. Operations may be categorized in three phases:</p> <p>Setup includes all the steps necessary to take the system from its state as normally delivered to the polling place, to the state in which it is ready to record votes. It does not include ballot definition.</p> <p>Polling includes such functions as:</p> <ul style="list-style-type: none"> • voter identification and authorization; • preparing the system for the next voter; • assistance to voters who wish to change their ballots or need other help; • system recovery in the case of voters who abandon the voting session without having cast a ballot; and routine hardware operations, such as installing a new roll of paper. <p>Shutdown includes all the steps necessary to take the system from the state in which it is ready to record votes to its normal completed state in which it has captured all the votes cast and the voting information cannot be further altered.</p> | |
| 3.2.8.1.a | Voting system setup, polling, and shutdown, as documented by the manufacturer, shall be reasonably easy for the typical poll worker to learn, understand, and perform. | Not Applicable |
| 3.2.8.1.b | The manufacturer shall conduct summative usability tests on the voting system using individuals who are representative of the general population and shall report the test results, using the Common Industry Format, as part of the TDP. | |

Table A-1 Usability Review Requirements Matrix (continued)

| Requirement | Description | Result |
|---------------|---|-----------------------|
| 3.2.8.1.b.i | The tasks to be covered in the test shall include setup, operation, and shutdown. | <i>Not Applicable</i> |
| 3.2.8.1.b.ii | In addition, the usability test report shall be submitted to the EAC as part of the documentation manufacturers are required to file with the application to test a voting system. | <i>Not Applicable</i> |
| 3.2.8.1.c | The voting system shall include clear, complete, and detailed instructions and messages for setup, polling, and shutdown. | <i>Not Applicable</i> |
| 3.2.8.1.c.i | The documentation required for normal voting system operation shall be presented at a level appropriate for poll workers who are not experts in voting system and computer technology. | <i>Not Applicable</i> |
| 3.2.8.1.c.ii | The documentation shall be in a format suitable for use in the polling place. | <i>Not Applicable</i> |
| 3.2.8.1.c.iii | The instructions and messages shall enable the poll worker to verify that the voting system <ul style="list-style-type: none"> o Has been set up correctly (setup); o Is in correct working order to record votes (polling); and o Has been shut down correctly (shutdown). | <i>Not Applicable</i> |
| 3.2.8.2 | Safety | |
| | All voting systems and their components must be designed so as to eliminate hazards to personnel or to the equipment itself. Hazards include, but are not limited to: <ul style="list-style-type: none"> • fire hazards; • electrical hazards; • potential for equipment tip-over (stability); • potential for cuts and scrapes (e.g., sharp edges); • potential for pinching (e.g., tight, spring-loaded closures); and • potential for hair or clothing entanglement. | |
| 3.2.8.2.a | Devices associated with the voting system shall be certified in accordance with the requirements of UL 60950-1, Information Technology Equipment – Safety – Part 1 by a certification organization accredited by the Department of Labor, Occupational Safety and Health Administration's Nationally Recognized Testing Laboratory program. | Pass |
| 3.2.8.2.b | The certification organization's scope of accreditation shall include IEC/UL 60950-1. | Pass |

Table A-2 Accessibility Review Matrix Requirements

| Requirement | Description | Result |
|-------------|---|-----------|
| | Accessibility | |
| 3.3 | Accessibility Requirements | pass/fail |
| 3.3.1 | General accessibility | |
| | The requirements of this section are relevant to a wide variety of disabilities. | |
| 3.3.1.a | The of this section are relevant to a wide variety of disabilities. Acc-VS shall be integrated into the manufacturer's complete voting system so as to support accessibility for disabled voters throughout the voting session. | |
| 3.3.1.a.i | The manufacturer shall supply documentation describing 1) recommended procedures that fully implement accessibility for voters with disabilities and 2) how the Acc-VS supports those procedures. | Pass |
| 3.3.1.b | When the provision of accessibility for the Acc-VS involves an alternative format for ballot presentation, then all information presented to non-disabled voters, including instructions, warnings, error and other messages, and contest choices, shall be presented in that alternative format. | Pass |
| 3.3.1.c | The support provided to voters with disabilities shall be intrinsic to the Acc-VS. Personal assistive devices of the voter shall not be necessary to operate the Acc-VS correctly. This does not apply to personal assistive technology required to comply with 3.3.4 b. | Pass |
| 3.3.1.d | If a voting system provides for voter identification or authentication by using biometric measures that require a voter to possess particular biological characteristics, then the Acc-VS shall provide a secondary means that does not depend on those characteristics. | Pass |
| 3.3.1.e | If the Acc-VS generates a paper record (or some other durable, human-readable record) that can be the official ballot or determinative vote record then the voting system shall allow the voter to verify that record using the same access features used by the voter to cast the ballot. | Pass |
| 3.3.2 | Enhanced visual interfaces | |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|-------------|--|--------|
| | <p>These requirements specify the features of the Acc-VS designed to make the visual interface easier to see, in particular for voters with vision deficiencies, and synchronized with audio for voters with various language, reading, or some cognitive disabilities.</p> <p>In general, low vision is defined as having a visual acuity worse than 20/70. Low (or partial) vision also includes dimness of vision, haziness, film over the eye, foggy vision, extreme near-sightedness or far-sightedness, distortion of vision, color distortion or blindness, visual field defects, spots before the eyes, tunnel vision, lack of peripheral vision, abnormal sensitivity to light or glare and night</p> <p>People with tunnel vision can see only a small part of the ballot at one time. For these users it is helpful to have letters at the lower end of the font size range in order to allow them to see more letters at the same time. Thus, there is a need to provide font sizes at both ends of the range.</p> <p>People with low vision or color blindness benefit from high contrast and from a selection of color combinations appropriate for their needs. Between 7% and 10% of all men have color vision deficiencies. Certain color combinations in particular cause problems. Therefore, use of color combinations with good contrast is required. Note also the general Requirement 3.2.5 h.i.</p> <p>However, some users are very sensitive to very bright displays and cannot use them for long. An overly bright background causes a visual white-out that makes these users unable to distinguish individual letters. Thus, use of non-saturated color options is an advantage for some people.</p> <p>It is important to note that some of the requirements in 3.2.5 "Visual display characteristics" also provide support for voters with certain kinds of vision problems.</p> | |
| 3.3.2.a | An Acc-VS with a color electronic image display shall allow the voter to adjust the color saturation throughout the voting session while preserving the current votes. | Pass |
| 3.3.2.a.i | At a minimum, two alternative display options listed shall be available: 1) black text on white background, 2) white text on black background, 3) yellow text on a black background, or 4) light cyan text on a black background. | Pass |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|---------------------|--|-----------------|
| 3.3.2.b | Groups of buttons and controls which perform different functions on the Acc-VS shall be distinguishable by both shape and color. This applies to buttons and controls implemented either "on-screen" or in hardware. This requirement does not apply to sizeable groups of keys in wide use by individuals with disabilities, such as a full alphabetic keyboard when used for purposes other than basic navigation and selection (e.g. entering a write-in candidate name). | Pass |
| 3.3.2.c | If the Acc-VS has an electronic image display, the Acc-VS shall provide synchronized audio output to convey the same information as that which is displayed on the screen | |
| 3.3.2.c.i | There shall be a means by which the voter can disable either the audio or the video output, resulting in a video-only or audio-only presentation, respectively. | Pass* |
| 3.3.2.c.ii | The system shall allow the voter to switch among the three modes (synchronized audio/video, video-only, or audio-only) throughout the voting session while preserving the current votes. | Pass* |
| 3.3.3 | Audio-tactile interfaces | |
| | These requirements specify the features of the Acc-VS designed to not only assist voters who are blind, but also those voters who would benefit from an auditory, rather than a purely visual, interface. | |
| 3.3.3.a & 3.3.3.a.i | The vendor shall conduct summative usability tests on the voting system using individuals who are blind. The vendor shall document the testing performed and report the test results using the Common Industry Format. This documentation shall be included in the Technical Data Package submitted to the EAC for national certification. | Not Applicable* |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|-------------|---|------------------------|
| 3.3.3.b | <p>The accessible voting station shall provide an audio-tactile interface (ATI) that supports the full functionality of the visual ballot interface, as specified in Subsection 2.3.3.</p> <ul style="list-style-type: none"> • Instructions and feedback on initial activation of the ballot (such as insertion of a smart card), if this is normally performed by the voter on comparable voting stations • Instructions and feedback to the voter on how to operate the accessible voting station, including settings and options (e.g., volume control, repetition) • Instructions and feedback for navigation of the ballot • Instructions and feedback for contest choices, including write-in candidates • Instructions and feedback on confirming and changing selections • Instructions and feedback on final submission of ballot | <i>Not Applicable*</i> |
| 3.3.3.b.i | <p>The ATI of the accessible voting station shall provide the same capabilities to vote and cast a ballot as are provided by other voting machines or by the visual interface of the standard voting machine.</p> <p>Discussion: For example, if a visual ballot supports voting a straight party ticket and then changing the choice in a single contest, so must the ATI.</p> | <i>Not Applicable*</i> |
| 3.3.3.b.ii | The ATI shall allow the voter to have any information provided by the voting system repeated. | <i>Not Applicable*</i> |
| 3.3.3.b.iii | The ATI shall allow the voter to pause and resume the audio presentation. | <i>Not Applicable*</i> |
| 3.3.3.b.iv | <p>The ATI shall allow the voter to skip to the next contest or return to previous contests.</p> <p>Discussion: This is analogous to the ability of sighted voters to move on to the next contest once they have made a selection or to abstain from voting on a contest altogether.</p> <p>v. The ATI shall allow the voter to skip over the reading of a referendum so as to be able to vote on it immediately.</p> | <i>Not Applicable*</i> |
| 3.3.3.b.v | The ATI shall allow the voter to skip over the reading of a referendum so as to be able to vote on it immediately. | <i>Not Applicable*</i> |
| 3.3.3.c | All voting stations that provide audio presentation of the ballot shall conform to the following requirements: | |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|--------------|--|------------------------|
| 3.3.3.c.i | The ATI shall provide its audio signal through an industry standard connector for private listening using a 3.5mm stereo headphone jack to allow voters to use their own audio assistive devices. | <i>Not Applicable*</i> |
| 3.3.3.c.ii | When a voting machine utilizes a telephone style handset or headphone to provide audio information, it shall provide a wireless T-Coil coupling for assistive hearing devices so as to provide access to that information for voters with partial hearing. That coupling shall achieve at least a category T4 rating as defined by American National Standard for Methods of Measurement of Compatibility between Wireless Communications Devices and Hearing Aids, ANSI C63.19. | <i>Not Applicable*</i> |
| 3.3.3.c.iii | A sanitized headphone or handset shall be made available to each voter. | <i>Not Applicable*</i> |
| 3.3.3.c.iv | The audio system shall set the initial volume for each voting session between 60 and 70 dB SPL. | Untested* |
| 3.3.3.c.v | The voting machine shall provide a volume control with an adjustable volume from a minimum of 20dB SPL up to a maximum of 100 dB SPL, in increments no greater than 10 dB. | Untested* |
| 3.3.3.c.vi | The audio system shall be able to reproduce frequencies over the audible speech range of 315 Hz to 10 KHz. | Untested* |
| 3.3.3.c.vii | The audio presentation of verbal information should be readily comprehensible by voters who have normal hearing and are proficient in the language. This includes such characteristics as proper enunciation, normal intonation, appropriate rate of speech, and low background noise. Candidate names should be pronounced as the candidate intends. | Pass |
| 3.3.3.c.viii | The audio system shall allow voters to control the rate of speech. The range of speeds supported should be at least 75% to 200% of the nominal rate. | Pass |
| 3.3.3.d | If the normal procedure is to have voters initialize the activation of the ballot, the accessible voting station shall provide features that enable voters who are blind to perform this activation. | Pass |
| 3.3.3.e | If the normal procedure is for voters to submit their own ballots, then the accessible voting station shall provide features that enable voters who are blind to perform this submission. | Pass |
| 3.3.3.f | All mechanically operated controls or keys on an accessible voting station shall be tactilely discernible without activating those controls or keys. | <i>Not Applicable</i> |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|-------------|---|------------------------|
| 3.3.3.g | On an accessible voting station, the status of all locking or toggle controls or keys (such as the "shift" key) shall be visually discernible, and discernible either through touch or sound. | <i>Not Applicable</i> |
| 3.3.4 | Enhanced input and control characteristics | |
| | These requirements specify the features of the Acc-VS designed to assist voters who lack fine motor control or use of their hands. | |
| 3.3.4.a | The Acc-VS shall provide a 3.5 mm industry standard jack used to connect a personal assistive technology switch to the Acc-VS. This jack shall allow only switch data to be transmitted to the voting system. The voting system shall accept switch input that is functionally equivalent to tactile input. All the functionality of the Acc-VS (e.g., straight party voting, write-in candidates) that is available through the conventional forms of input, such as tactile, shall also be available through this non-manual input mechanism. | <i>Not Applicable*</i> |
| 3.3.4.b | The Acc-VS shall provide features that enable voters who lack fine motor control or the use of their hands to submit their ballots privately and independently without manually handling the ballot. | Partial pass |
| 3.3.4.c | All keys and controls on the accessible voting station shall be operable with one hand and shall not require tight grasping, pinching, or twisting of the wrist. The force required to activate controls and keys shall be no greater 5 lbs. (22.2 N). | Pass |
| 3.3.4.d | The accessible voting station controls shall not require direct bodily contact or for the body to be part of any electrical circuit. | <i>Not Applicable*</i> |
| 3.3.5 | Design for mobility aids | |
| | These requirements specify the features of the Acc-VS designed to assist voters who use mobility aids, including wheelchairs. Many of the requirements of this section are based on the ADA Accessibility Guidelines for Buildings and Facilities (ADAAG). | |
| 3.3.5.a | The accessible voting station shall provide a clear floor space of 30 inches (760 mm) minimum by 48 inches (1220 mm) minimum for a stationary mobility aid. The clear floor space shall be level with no slope exceeding 1:48 and positioned for a forward approach or a parallel approach. | <i>Not Applicable*</i> |
| 3.3.5 | All controls, keys, audio jacks and any other part of the accessible voting station necessary for the voter to operate the voting machine shall be within reach as specified under the following sub-requirements: | |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|-----------------------------|--|------------------------|
| 3.3.5.1.a | If the accessible voting station has a forward approach with no forward reach obstruction then the high reach shall be 48 inches maximum and the low reach shall be 15 inches minimum. | <i>Not Applicable*</i> |
| 3.3.5.1.b.i 3.3.5.1.b.ii | If the accessible voting station has a forward approach with a forward reach obstruction, the following requirements apply: <ul style="list-style-type: none"> • The forward obstruction shall be no greater than 25 inches in depth, its top no higher than 34 inches and its bottom surface no lower than 27 inches. • If the obstruction is no more than 20 inches in depth, then the maximum high reach shall be 48 inches, otherwise it shall be 44 inches. | <i>Not Applicable*</i> |
| 3.3.5.1.b.iii | Space under the obstruction between the finish floor or ground and 9 inches (230 mm) above the finish floor or ground shall be considered toe clearance and shall comply with the following provisions: <ul style="list-style-type: none"> • Toe clearance shall extend 25 inches (635 mm) maximum under the obstruction • The minimum toe clearance under the obstruction shall be either 17 inches (430 mm) or the depth required to reach over the obstruction to operate the accessible voting station, whichever is greater • Toe clearance shall be 30 inches (760 mm) wide minimum | <i>Not Applicable*</i> |
| 3.3.5.1.b.iv | Space under the obstruction between 9 inches (230 mm) and 27 inches (685 mm) above the finish floor or ground shall be considered knee clearance and shall comply with the following provisions: <ul style="list-style-type: none"> • Knee clearance shall extend 25 inches (635 mm) maximum under the obstruction at 9 inches (230 mm) above the finish floor or ground. • The minimum knee clearance at 9 inches (230 mm) above the finish floor or ground shall be either 11 inches (280 mm) or 6 inches less than the toe clearance, whichever is greater. • Between 9 inches (230 mm) and 27 inches (685 mm) above the finish floor or ground, the knee clearance shall be permitted to reduce at a rate of 1 inch (25 mm) in depth for each 6 inches (150 mm) in height. • Knee clearance shall be 30 inches (760 mm) wide minimum. | <i>Not Applicable*</i> |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|-----------------------------|---|------------------------|
| 3.3.5.c | All labels, displays, controls, keys, audio jacks, and any other part of the accessible voting station necessary for the voter to operate the voting machine shall be easily legible and visible to a voter in a wheelchair with normal eyesight (no worse than 20/40, corrected) who is in an appropriate position and orientation with respect to the accessible voting station | <i>Not Applicable*</i> |
| 3.3.5.1.c | If the accessible voting station has a parallel approach with no side reach obstruction then the maximum high reach shall be 48 inches and the minimum low reach shall be 15 inches. | <i>Not Applicable*</i> |
| 3.3.5.1.d.i 3.3.5.1.d.ii | If the accessible voting station has a parallel approach with a side reach obstruction, the following sub-requirements apply. • The side obstruction shall be no greater than 24 inches in depth and its top no higher than 34 inches. • If the obstruction is no more than 10 inches in depth, then the maximum high reach shall be 48 inches, otherwise it shall be 46 inches. | <i>Not Applicable*</i> |
| 3.3.6 | Enhanced auditory interfaces | |
| | These requirements specify the features of the Acc-VS designed to assist voters with hearing disabilities. | |
| 3.3.6.a | The Acc-VS shall incorporate the features listed under Requirement 3.3.3.c for voting systems that provide audio presentation of the ballot. | Partial pass |
| 3.3.6.b | If voting equipment provides sound cues as a method to alert the voter, the tone shall be accompanied by a visual cue, unless the station is in audio-only mode. | Pass |
| 3.3.6.c | No voting device shall cause electromagnetic interference with assistive hearing devices that would substantially degrade the performance of those devices. The voting device, measured as if it were a wireless device, shall achieve at least a category T4 rating as defined by [ANSI01] American National Standard for Methods of Measurement of Compatibility between Wireless Communications Devices and Hearing Aids, ANSI C63.19. | Untested* |
| 3.3.8 | English proficiency | |
| 3.3.8.a | For voters who lack proficiency in reading English, the Acc-VS shall provide an audio interface for instructions and ballots as described in 3.3.3.b. | Pass |
| 3.3.9 | Speech not required | |
| 3.3.9.a | The voting system shall not require voter speech for its operation. | Pass |

Table A-2 Accessibility Review Matrix Requirements (continued)

| Requirement | Description | Result |
|---------------|--|------------------------|
| 3.3.10 | Summative Usability Report | |
| 3.3.10.a | The manufacturer shall submit a report of their summative usability tests on the voting system using individuals who are representative of the general population. | <i>Not Applicable*</i> |
| 3.3.10.a.i | The report shall be submitted in the Common Industry Format. | <i>Not Applicable*</i> |
| 3.3.10.a.ii | The report shall contain the results of the summative usability tests. | <i>Not Applicable*</i> |
| 3.3.10.b | The manufacturer shall conduct summative usability tests on the Acc-VS using individuals with low vision and shall report the test results, using the Common Industry Format, as part of the TDP. | <i>Not Applicable*</i> |
| 3.3.10.b.i | In addition, the usability test report shall be submitted to the EAC as part of the documentation manufacturers are required to file with the application to test a voting system. | <i>Not Applicable*</i> |
| 3.3.10.c | The manufacturer shall conduct summative usability tests on the Acc-VS using individuals lacking fine motor control and shall report the test results, using the Common Industry Format, as part of the TDP. | <i>Not Applicable*</i> |
| 3.3.10.c.i | The vendor shall conduct summative usability tests on the voting system using individuals lacking fine motor control. The vendor shall document the testing performed and report the test results using the Common Industry Format. This documentation shall be included in the Technical Data Package submitted to the EAC for national certification. | <i>Not Applicable*</i> |
| 3.3.10.c.i | In addition, the usability test report shall be submitted to the EAC as part of the documentation manufacturers are required to file with the application to test a voting system. | <i>Not Applicable*</i> |
| | Discussion: Voting system developers are required to conduct realistic usability tests on their product before submitting the system to conformance testing. This is to encourage early detection and resolution of usability problems. The manufacturer must submit the usability test report to the VSTL as part of their TDP. The VSTL will then check the technical data package to ensure that the report is present and reported in the Common Industry Format and contains the results from a summative usability test. | |



Electronic Ballot Delivery and Marking

Voatz Remote Accessible Ballot Marking System
Voatz, Inc.

In response to recently-issued guidance to election officials regarding electronic ballot delivery and marking circulated by the Federal Election Assistance Commission (EAC) *, Voatz provides the following information on Voatz functionality.

System Considerations

System Infrastructure

| Consideration | Voatz Implementation |
|---|---|
| How will the voting infrastructure be hosted (i.e., on servers at your facility, in the cloud, etc.)? | Voatz infrastructure is hosted at qualified, professional cloud services providers Microsoft and Amazon. |
| If you use the cloud, do you have awareness of where will the data be hosted (i.e. outside your jurisdiction, state, or the United States)? | All Voatz critical infrastructure is hosted within the United States at Microsoft Azure™ and Amazon Web Services™ data centers. |
| Does the system have the capacity to handle the increased load? | Cloud services enable spinning up additional servers on demand. Voatz runs monitored automated load tests daily to assess performance of the server(s) so that additional servers can be commissioned during heaviest load if needed. |
| What redundancies should be built into the system (i.e., backups, failover system, etc.)? | All Voatz cloud servers have hot standby replacements and are backed up daily. |
| Who will administer system configuration (i.e., security, load balancing, updates, patches, etc.)? | Voatz IT operations team, with guidance from the cloud providers, handles configuration of load balancing, security patches, etc. |

* See: Electronic Ballot Delivery and Marking Considerations for Election Officials (as posed by EAC)
https://www.eac.gov/sites/default/files/electionofficials/etm/etmBallot_Delivery.pdf



Electronic Interface / Usability & Accessibility

| Consideration | Voatz Implementation |
|--|---|
| Is the system accessible to voters with specific needs (i.e., visual impairments, disabilities, language, etc.)? | Yes. Voatz apps operate on iOS or Android smartphones and incorporate Apple's and Google's native <u>accessibility</u> features including VoiceOver™, TalkBack™, etc. The Voatz Interface and onscreen help are available in English and Spanish. Ballots can be translated into additional languages. |
| Is the system compatible with mobile devices? | Yes. Voatz apps are designed specifically to operate on Apple iOS v12.4 or higher or Android v8 or higher mobile phones (not other devices). |
| Is the system compatible with readily available screen readers? | Yes. Voatz runs as native iOS or Android smartphone apps which incorporate Apple's and Google's VoiceOver, TalkBack, as well as other accessibility features. |
| Is the system accessible with binary personal assistive technologies (i.e., jelly switches, sip-n-puff, etc.)? | Voatz for iOS supports Voice Control hands-free operation. Binary assistive technologies have not been tested but may be adaptable with Bluetooth™. |
| Do you provide help to voters directly through the electronic ballot delivery system? | Yes. Context dependent, onscreen Help is available in English and Spanish. In addition, Voatz detects if a screen reader is enabled, and provides modified help when necessary. |

Copyright 2020, Voatz, Inc. Visit us online at voatz.com. Updated 2020-05-26



Election Definition Files

| Consideration | Voatz Implementation |
|--|--|
| What file format does your system accept (i.e., HTML, PDF, CSV, etc.)? | Voatz interprets native Electionware™ and ClearVote™ election definition files to transform them into remote accessible ballots. For other vendors, specific scripting routines are used to transform PDF designs into remote accessible ballot formats. |
| Does your voting system produce ballots in the accepted file types, or do you need software to convert them? | Yes, for supported vendors, fully marked and scannable ballots are available for printing and tabulation by the primary voting system. |
| What type(s) of audio files does your system use? | Audio files are not required as Apple iOS and Android provide speech to text. |
| What languages do the ballots need to be presented in? | Determined by jurisdiction. Voatz currently supports English/Spanish EBM interface. |

Additional Supplies

| Consideration | Voatz Implementation |
|--|--|
| Do you need to supply additional affidavits and instructions to the voter who votes electronically? | Voatz offers an electronic affidavit with signature or checkbox if required by the jurisdiction- typically for UOCAVA voters. |
| Will your materials contain labels and self-folding envelopes to mail the ballots back? | No. Ballot return is electronic over secure network and auditable. |
| Will you provide printable privacy sleeves for the voter to protect the ballot? | Not applicable. Electronic affidavits are stored separately from the returned ballot for privacy and ballots are identified in an anonymized manner. |
| What auxiliary technologies are required for the voter to complete his or her ballot (i.e., Internet service, email, printer, fax service, specific software, etc.)? | Voatz requires a supported Apple or Android smartphone, an email account, and cellular or Wi-Fi connectivity. No paper, printer or postal service is involved. Auxiliary voting channels should be available for other voters. |

Copyright 2020, Voatz, Inc. Visit us online at voatz.com. Updated 2020-05-26



Ballot Duplication

| Recommendation | Voatz Implementation |
|---|---|
| Many ballots generated by an electronic ballot delivery system cannot be directly scanned and tabulated into your voting system. To tabulate ballots using the voting system, the ballots must be printed on paper stock meeting certain specifications. For those ballots, enough blank paper stock will need to be purchased in advance. Jurisdictions may need additional technology (i.e., ballot duplication system, ballot on demand system, etc.) or staff to duplicate electronically generated ballots onto a ballot that can be scanned and duplicated. Depending on the volume of ballots that require duplication, additional staff needs could be significant. | Voatz is rare in its ability to return the ballots electronically to the jurisdiction. After they confirm the voter's affidavit, election officials print marked, scannable ballots directly on ballot stock for tabulation. Election officials do not need to open paper mail. Nor is faxing or hand transcribing required. Voatz supports jurisdiction and/or public post-election auditing of ballots. |

Security Recommendations

Vulnerability Scan and Remote Penetration Testing

| Recommendation | Voatz Implementation |
|--|--|
| Because these systems are publicly facing, jurisdictions using an electronic ballot delivery system should request a vulnerability scan and remote penetration test be conducted on the system. Also, for vendor-provided systems, election officials should suggest that their vendor subject the system to a critical product evaluation. These services provide the situational awareness needed to make informed decisions to manage the risks associated with the system and are provided at no cost to election jurisdictions and their private sector partners. | In September 2019, the CISA Hunt and Incident Response Team (HIRT) conducted a proactive hunt operation that included the Voatz internal corporate network, as well as Amazon Web Services (AWS) and Microsoft Azure cloud networks that support the mobile-based election infrastructure. HIRT analysts did not detect threat actor behaviors or artifacts of past activities on the in-scope portions of the Voatz networks. HIRT commended Voatz for their proactive measures in the use of canaries, bug bounties, Shodan alerts, and active internal scanning and red teaming. For more, see: https://voatz.com/Hunt-Engagement-Summary-Voatz.pdf |

Copyright 2020, Voatz, Inc. Visit us online at voatz.com. Updated 2020-09-26



Web-Based Portals and File Servers

| Recommendation | Voatz Implementation |
|--|---|
| Use security best practices for web and network connected election systems, including two-factor authentication (2FA) for employees and voters. | Yes, Voatz Admin portal uses 2FA for election officials. Voters use Voatz app on smartphones with integrated security features. |
| Encrypt traffic using secure hypertext transfer protocol (HTTPS) or, if you use a file server, ensure it uses a secure file transfer protocol (SFTP) by supporting transport layer security (TLS) version 1.2. | Yes. |
| Obtain outside cybersecurity assessments, such as CISA vulnerability scanning and remote penetration testing. | Yes. |

Copyright 2020, Voatz, Inc. Visit us online at voatz.com. Updated 2020-09-26

8. Attachment I: Synack Assessment Report

Please see the following pages with the report.

Synack Executive Summary Report: March 08, 2022 - March 16, 2022

Summary

Voatz has completed an extensive penetration test performed by Synack for their Voatz Mobile Voting Application. The data below reflects testing done between March 08, 2022 and March 16, 2022. During the course of the assessment, **0 critical severity, 0 high severity, 0 medium severity and 0 low severity** vulnerabilities were exposed per CVSS standard specification.

| | | |
|----------------------------------|----------------------------|-----------------------------|
| 155 Active Researchers | 9 Research Hours | 0 Vulnerabilities |
|----------------------------------|----------------------------|-----------------------------|

METHODOLOGY

The Synack Red Team employs a variety of tools and techniques that closely mimic adversarial tradecraft. Synack researchers cover a full range of testing methodologies which hit known variants of web-based attacks to target issues pertaining to: configuration management, business logic, authentication, authorization, session management, data validation, web services and client-side code. These researchers address the most prevalent security threats facing organizations today, with specific attention being spent on the OWASP Top 10 Project and CWE/SANS Top 25 Most Dangerous Software Errors.

SYNACK RED TEAM (SRT)

The SRT is a diverse group of highly skilled security experts, which include top researchers from academia, government, and the private sector. The SRT represents 50 different countries around the world, and hundreds of years of combined testing experience. Many Synack researchers are top producers from over 60 vulnerability disclosure programs, and regularly speak at industry events such as DefCon, AppSec, and Black Hat.

ABOUT SYNACK

Synack was founded by former NSA cyber analysts on a mission to reinvent the penetration testing and security assessment market. Synack facilitates access to the most accomplished white-hat hackers in the world, and rewards them on a solely incentivized basis to discover high-risk vulnerabilities.



DUST-CustomReport2022

Date Generated:
April 06, 2022



Table of Contents

| | |
|--|---|
| Executive Overview | 3 |
| Missions Checklist Summary | 4 |
| Missions Checklist Summary by Category | 4 |
| Missions Checklist | 4 |
| Missions Details | 5 |
| About Synack | 8 |



Executive Overview

Voatz engaged Synack, Inc. to perform a penetration test for their applications. The testing was performed on the targets reported under scope section.

This report reflects testing done between March 08, 2022 - March 16, 2022.

Scope

Voatz Mobile Voting Application

Vulnerabilities Summary

| | | | | |
|-----------------------|-------------------|---------------|-----------------|--------------|
| 0 | 0 | 0 | 0 | 0 |
| Total Vulnerabilities | Critical Severity | High Severity | Medium Severity | Low Severity |

Missions Checklist Summary

| | | | |
|--------------------|--------|--------|-----|
| 1 | 0 | 1 | 0 |
| Missions Completed | Failed | Passed | N/A |



Missions Checklist Summary

In all of our penetration tests, Synack incentivizes SRT members to discover any security issue that might be negatively impacting the hardness of your attack surface. In addition to the diverse, open-ended tradecraft employed by our researchers, we guarantee all action items listed in the following checklists were completed.

Missions Checklist Summary by Category

- Aligned with NIST SP 800-53 R4 Control guidelines
- Total Checks - 1 with 1 mission passed for a 100% Success Rate

| TESTING CATEGORY | PASS | FAIL | N/A | SUCCESS RATE |
|---|------|------|-----|--------------|
| Client-side protection bypass (jailbreak detection and certificate pinning) | 1 | 0 | 0 | 100% |

Missions Checklist

The checklist is based on NIST SP 800-53 R4 Control guidelines.

| REF # | CONTROL FAMILY | TESTING CATEGORY | MISSION | COMPLETED | STATUS | FISMA APPLICABILITY | | |
|-------|----------------|---|--|-----------|--------|---------------------|------|------|
| | | | | | | LOW | MOD. | HIGH |
| 1 | N/A | Client side protection bypass (jailbreak detection and certificate pinning) | BROMINEDUST iOS client side protection bypass. | ✓ | Pass | | | |

9. Attachment J: Proposed team and Resumes of Key Personnel

This table lists key staff on the Voatz team designated to work on this project along with their project responsibilities.

| Role / Specialty | Experience | Project Responsibilities |
|---|--|--|
| InfoSec Engineer, Developer/Engineer | SW Engineer: 24 years Elections: 8 years MS: Computer Science, Carnegie Mellon | Information and cyber security, Platform and web services development, Mobile app development, Election data management, Administrative Portal development and management, Reporting. |
| Product Manager, Customer Success Engineer | Product Management: 3 years Elections: 3 years MBA: International Business and Finance, Hult International BBA: Finance, Florida Atlantic University | Primary Client point of contact, Develop/manage project plan and schedule, Project communications, Status updates and reporting, Election Business/Process Analyst Assessments, Training and support oversight, Solution deployment coordinator |
| Director, Quality Assurance and Certification, Project Manager | Quality Assurance: 37 years Elections: 8 years BA: IMJR, U. Connecticut | Quality Assurance Lead, Author Test Execution Reports for system and system components, Execute test plans, Logic & Accuracy testing, Accessibility testing, Gather requirements, Customization oversight |
| Director of Finance, Account Manager | Business Finance: 25 years Elections: 5 years MBA: Northeastern BA: Finance, Bridgewater State | Contract owner, Coordination with Client Legal and Financial, Manage licensing |
| Developer/ Engineer, Senior Scientist | SW Engineer: 3 years Elections: 3 years Academia: 12 years | Browser voting app development, Mobile app development, Administrative Portal development, |

| | | |
|--|--|---|
| | Ph.D.: Mathematics, U. Illinois BS, MS: Mathematics, Virginia Tech | Platform backend and web services development, Ballot operations, Cryptography and security research |
|--|--|---|

Please see the following pages with the Resumes of the proposed Key Personnel.

Contact

Top Skills

Mobile Payments
Mobile Commerce
Software Development

Languages

French (Elementary)
English (Native or Bilingual)
Hindi (Native or Bilingual)

Honors-Awards

Voatz selected for the Techstars Boston 2017 Accelerator Program

Vantiv Challenge Winner @ Money2020 Hackathon 2016

IBM Challenge Winner @ AngelHack Boston 2016

Voatz wins 'The Startup Most Likely to Develop a Cult Following' Award at MIT Enterprise Forum's Startup Spotlight

Voatz wins the Audience Favorite award at Mass Innovation Nights

Patents

System and methods for roaming subscribers to replenish stored value accounts

Nimit Sawhney

Entrepreneur, Co-Founder @ Voatz

Boston, Massachusetts

Experience

Voatz

Co-Founder, CEO

January 2015 - Present

Boston, MA

Voatz is a mobile elections platform changing the way the world votes by making it possible to vote with a smartphone or tablet. Powered by military-grade technology secured by blockchain, biometric identification and cryptography, Voatz ensures safe and secure elections while increasing accessibility and citizen engagement.

Since 2016, Voatz has worked with both major political parties, towns, cities, states, universities and non-profits to administer elections. In 2018, Voatz ran the first mobile blockchain vote in US Federal Election history. Specifically, Voatz partnered with the State of West Virginia to empower deployed military and overseas citizens to vote in the 2018 Primary Elections (2 counties) and the 2018 Mid-Term Elections (24 counties). In March 2019, Voatz was selected by the City/County of Denver CO for its 2019 Municipal General & Run-off Elections where its use led to a more than 2x increase in voter participation amongst deployed military and overseas citizen voters. More recently, Utah County (UT) and Jackson County (OR) have selected Voatz for their 2019 elections.

Oberthur Technologies

Director of R&D - Mobile Financial Services Business Line

June 2012 - December 2014 (2 years 7 months)

Waltham MA

- Led the expansion of the R&D team to a group of 25 engineers and managers with dedicated teams for core platform development, professional services, QA and IT support.
- Led the successful implementation and delivery of Czech Republic's first universal mobile payment service (Mobilo) in partnership with 4 leading banks and all 3 mobile operators in the country.

- Designed the 4 class projects based on the above & mentored a Senior Level Systems class of 100 students.

Worked on building networked (distributed) applications of the future (emphasis on the interaction of applications and networks) by making use of powerful network processors like the Intel IXP1200:

- Installed and configured the new IXA (Intel Internet Exchange Architecture) programming platform.
- Ported the IXP1200 PCI drivers to work on the Suse Linux platform.

Amdocs

Software Engineer

August 1999 - July 2001 (2 years)

Cyprus, Germany, Austria

- Performed a high-level requirements analysis for developing a new convergent telecom-billing engine.
- Developed routines for handling hierarchical billing of corporate customers, handling paper-bill suppression and Datawarehousing of daily usage data.
- Developed routines for migrating data from external systems (like SAP, Clarify) to the Amdocs data model.
- Trained new employees in C, Pro-C and Data-layer module programming.

Education

Carnegie Mellon University

MS, (Information Networking) Computer Science, Electrical Engg., Business Management, Public Policy · (2001 - 2003)

Harvard Business School

Young American Leaders Program (YALP) · (2019)

Harvard Law School

Program on Negotiation · (2007)

- Led the successful implementation and delivery of a geo-redundant mobile/landline prepaid recharge solution for Cable & Wireless International in several countries around the world.
- Led the successful implementation and delivery of Caribbean's first cross-border peer-to-peer airtime transfer solution covering 13 nations with 1 million active users.
- Led the successful implementation and delivery of Sierra Leone's first Mobile Money implementation (Splash Cash) – one of the first multi-bank, multi-operator solutions in Africa.

Sr. Software Architect
January 2003 - December 2007 (5 years)

Newton, MA

- Developed an XML-based generic reporting framework as part of a distributed inter-carrier financial settlement system in order to facilitate the real-time/batch generation of multi-format reports.
- Developed a Wi-Fi accounting and management system utilizing a Linux server as a WLAN router.
- Developed a .NET plug-in for a RADIUS-based accounting/authorization protocol running on TCP.
- Added real-time charging functionality to the Helix DNA Media Server for streaming pay-per-view audio/video streams to RTSP-compatible mobile-phones/PDAs.
- Designed and developed a .NET based barcode generation and security application for mobile phones.
- Developed a SMS-based account recharging system for the Nokia Series-60 phones using J2ME.
- Provided technical consultancy and support for the Nokia Payment Solution (a white-labeled MoreMagic service) deployed at T-Mobile in Hungary.

School of Computer Science, Carnegie Mellon University
Researcher
September 2001 - December 2002 (1 year 4 months)

Pittsburgh, PA

- Developed a non pre-emptive embedded micro-kernel for the Intel XScale 80200 processor.
- Ported the uCOS-II embedded operating system to the XScale 80200 platform.
- Developed device drivers and interrupt handler routines for the serial port, timer and the PCI sound card.

- Designed the 4 class projects based on the above & mentored a Senior Level Systems class of 100 students.

Worked on building networked (distributed) applications of the future (emphasis on the interaction of applications and networks) by making use of powerful network processors like the Intel IXP1200:

- Installed and configured the new IXA (Intel Internet Exchange Architecture) programming platform.
- Ported the IXP1200 PCI drivers to work on the Suse Linux platform.

Amdocs
Software Engineer
August 1999 - July 2001 (2 years)

Cyprus, Germany, Austria

- Performed a high-level requirements analysis for developing a new convergent telecom-billing engine.
- Developed routines for handling hierarchical billing of corporate customers, handling paper-bill suppression and Datawarehousing of daily usage data.
- Developed routines for migrating data from external systems (like SAP, Clarify) to the Amdocs data model.
- Trained new employees in C, Pro-C and Data-layer module programming.

Education

Carnegie Mellon University
MS, (Information Networking) Computer Science, Electrical Engg., Business Management, Public Policy · (2001 - 2003)

Harvard Business School
Young American Leaders Program (YALP) · (2019)

Harvard Law School
Program on Negotiation · (2007)

Orlando Alomá, FMVA®

Boston, MA | [REDACTED]
[REDACTED]
[REDACTED]

Professional Experience

VOATZ — Boston, USA

May 2019 – Present

Project Manager

- Served as a subject matter expert in products and services; often appeal to by senior management to provide training and support to clients and other team members
- Collaborated with Business Development team to deliver 15+ RFI & RFP for government clients and improved response times by 25% by creating a searchable guide
- Developed plans for mission-critical projects including all necessary product requirements & features, materials & equipment sourcing strategies, and proposed schedules
- Supervised an integrated product team of 4+ members and 3 distinct products
- Managed 10+ elections with a total of over 100k+ eligible voters including international elections in Canada & US.
- Communicated directly with subcontractors such as Gilmore Doculink and Smartmatic to coordinate information sharing procedures and development of documentation
- Presided communication to/from C-level management when isolated supply chain issues arose and communicated a solution to clients and/or support agents

Management Associate

- Spearheaded a project to identify an opportunity to collaborate with the 2020 Census Opportunity Project and pitched an idea of a secure electronic delivery of data service to the US Deputy Director of the Census Bureau
- Performed Product Demos to potential clients and government regulators such as Thomas Greg & Sons from Colombia and EAC Commissioner Donald Palmer
- Advised as Technical Lead of the Project Management Team about products and capabilities
- Teamed with the Product Owner to evaluate alternatives to make quick and informed decisions in difficult situations
- Led Quality Assurance Testing on iOS and Android mobile apps and implemented a Regression Testing spreadsheet with 500+ Test Cases resulting in a 20% decrease in testing time
- Executed 30+ election audits and recommended improvements to decreased time spent on audits by 40%

Professional Certificates

Financial Modeling & Valuation Analyst (FMVA®), Corporate Finance Institute (CFI®)

December 2020

- Relevant courses include: Advance Excel Analysis, Financial Modeling, Corporate Finance, Valuation, Budgeting & Forecasting, and Strategy

Blockchain Revolution Specialization, INSEAD

December 2019

- Relevant courses include: Introduction to Blockchain, Transacting on the Blockchain, Blockchain and Business Applications, and Blockchain Opportunity Analysis

Skills and Certifications

Other languages: English (Native) and Spanish (Native)

Mastered Tools: Excel, PowerPoint, Word, NetSuite, PitchBook, JIRA, Slack, Zoom, MS Teams, MS Office

Personal Interests: Aviation, Traveling, Technology, Financial Markets, Trading, Tennis, Rock Climbing

Education

Master of International Business & Finance

HULT INTERNATIONAL BUSINESS SCHOOL— Boston, USA

August 2019

Bachelor of Business Administration in Finance

FLORIDA ATLANTIC UNIVERSITY— Boca Raton, USA

August 2017

Contact

Top Skills

Software Project Management

SaaS

Testing

Certifications

Blockchain User Certification

Cyber Risk Management

Patents

System and methods for calculating and predicting near term production cost, incremental heat rate, capacity and emissions of electric generation power plants based on current operating and optionally atmospheric conditions.

Linda Hutchinson

Director of QA & Compliance

Boston, Massachusetts

Summary

Proven track record in the design, testing, scheduling and delivery of quality software products to enterprise, multinational and GovTech clients. Led all phases of productization from concept prototypes to successful SaaS and enterprise-wide client/server deployments. Real-world Cybersecurity experience and continuing education. Experienced technical manager and project leader.

Experience

Voatz

Director of Quality Assurance & Compliance

February 2019 - Present

Boston, Massachusetts

Direct testing of Voatz' mobile voting platform in use in five US states. Serve as Agile product owner and ensure compliance with federal voting standards and accessibility guidelines. Partner with scrum team in planning and scoping sprints. Collaborate with senior management on executing corporate growth strategies.

Clear Ballot Group

Director of Quality Engineering

October 2015 - January 2019 (3 years 4 months)

Greater Boston Area

Directed all testing of the ClearVote voting system which achieved the fastest ever initial US federal certification. Led QA team in establishing in house software and hardware testing methods for six GovTech products. Partnered with third party federal testing laboratories to ensure products met accuracy, usability, scalability, security and accessibility requirements of states' election officials. Clear Ballot Group was selected as a GovTech Top 100 winner in 2018 for its success in bringing innovation and transparency to the election industry.

IBM

Advisory Software Engineer

2006 - September 2015 (9 years)

Greater Boston Area

Led multiple quality engineering projects at Kenexa, an IBM company. Devised new test methodologies for BrassRing SaaS recruiting software serving global clients. Specified and validated migration of applicant tracking and resume data into new database platform. With development architect, migrated to open source search engine resulting in \$3MM in cost savings while increasing both reliability and performance. In partnership with a localization vendor, tested translation and internationalization of software for European and Asian markets. Ensured compliance with PII privacy regulations. Partnered with DevOps on seamless, staged deployments in US and European data centers.

BrassRing

Quality Engineering Architect consultant

2002 - 2006 (4 years)

Waltham, MA

Served as technical project leader for KenexaBrassRing SaaS recruiting solution serving global clients. Led teams of two to seven engineers in developing ten thousand new test cases for growing product. Served as cross-departmental project manager for migration of millions of resumes to new search platform. Established method of back-end validation of full-text search functionality. Designed automation for execution prior to deployments.

Lotus Development

Senior Quality Engineering Manager - Lotus Notes

1987 - 1996 (9 years)

Directed Lotus Notes Quality Engineering responsible for all testing of the software on multiple client/server platforms of product with 60 million users. Led test automation, network/platform certifications, scalability, performance benchmarking, localization, UI and system testing. Grew department(s) to 100 employees to efficiently meet rapidly growing product requirements on across multiple platforms while achieving highest employee satisfaction/retention. Helped ensure successful deployment at selected Fortune clients. Nominated by peer managers to direct corporate QA Managers Task Force. Partnered with Development (Iris Associates) and Product Management in scheduling and implementing new Notes releases. Established certification Lab and automation strategy. With software architect, devised method to accurately scale testing for each platform while minimizing risk and expediting new products to market.

Education

The University of Connecticut
Bachelors, IMJR

The Commonwealth Institute
Certificate, Entrepreneurship/Entrepreneurial Studies · (2000 - 2001)

Harvard Kennedy School
Certificate, Cybersecurity · (2018 - 2018)

Geoffrey J. Dickson

SUMMARY.

Experienced finance director and divisional controller MBA with a diverse background (both international and domestic) of over 25 years in professional services, including consulting (ERP software, mobile voting technology, SaaS and management consulting), franchising and communications (advertising and public relations) with a strong desire to continue to take on new and increased responsibilities.

EXPERIENCE.

4/17 – Present Voatz, Inc.

Weymouth/Boston, MA

DIRECTOR OF FINANCE and HUMAN RESOURCES

- Leading the Finance, Accounting and Human Resources functions for an exciting startup.
- Started with title of Financial Controller. Promoted to Director of Finance in 2021.
- Member of the Senior Leadership Team. Reporting directly to the CEO and cofounders.
- Ensuring timely and accurate month-end close cycle. Participate in quarterly Board Meetings, providing the finance and human resources update and answering related questions from Board Members. Participate in monthly investor calls providing similar updates.
- Designed and built out all internal financial reporting and KPI measurement tools. Perform all annual budgeting, monthly forecasting, cash planning, KPI review and flux analysis, expense reimbursement approval and processing.
- Designing and documenting internal financial policies and procedures, specifically in the areas of travel expense, capitalization, PTO, and credit policy, amongst others.
- Built and fine-tuned investor-facing financials package for purposes of closing a new financing round.
- Coordinate year-end audit and tax returns with outside CPA firm, review compliance issues with legal firm. Appointed Compliance Officer for company's Anti-Corruption Policy.
- Heading up the Human Resources function, handling contracting, onboarding, working with insurance brokers, reviewing and recommending benefits, assisting with immigration applications, reviewing and processing payroll, monitoring enrollment in company benefits and staying informed about changes in HR-related state and federal laws that can impact the company and our employees.
- Liaise with building management, assist with real estate matters.
- Regular interaction with the CEO and sales team to ensure accurate forecasting.

3/17 – 12/19 Robert Half Finance & Accounting

Boston, MA

SENIOR CONSULTANT

Project Assignment: Duck Creek Technologies (5 months)

- Internal audit project reporting to the Corporate Controller and VP of Finance
- Preparing professional services revenue for audit at fiscal year end.
- Working closely with 15 Delivery Leads on over 60 projects generating over \$35 million in revenue in the current fiscal year.

Project Assignment: Nanigans Inc. (12+ months)

- Assisting Corporate Controller with completion of 5 audits for business units located in US, UK, Australia, Singapore, and South Korea.
- Worked in tandem with local accountants in Singapore and UK to assemble audit workpapers and fulfill PBC lists, answer questions from auditors (RSM)

- Managed tax return calculation and filing of returns for sales tax (MA and NY), GST and VAT in other business units
- Project work involving clean up of the company's cap table
- Ad hoc projects as needed for Corporate Controller and CFO
- **Project Assignment: Current powered by G.E. (4 months)**
- Internal carve-out audit support for two G.E. companies, reporting to the CFO
- Worked in tandem with internal auditors and a business controller based in Cleveland, OH
- Using SAP and a number of proprietary G.E. systems, provided invoice and payment documentation
- Balance sheet analysis assisting with the calculation of net economic benefit of sale of business units

12/16 – 5/17 XR Media

Weymouth MA/Brooklyn, NY

SENIOR CONSULTANT

- Working part-time with the founders of XR on revising business plan to launch multimedia platform and secure funding for the launch
- Firm was able to partner with global network Trace TV to air its video content.

8/16 – 1/17 VT MAK

North Cambridge, MA

CONTROLLER

- Managing team of three (2 managers, 1 senior accountant) and reporting directly to the CFO in a software company specializing in modeling and simulation for the defense industry.
- Manage month-end close, present results to CFO.
- Responsible for extensive internal reporting to the parent company, Singapore Technologies.
- Daily application of 97-2 principles in revenue recognition and cost deferrals.
- Prepare and conduct periodic meetings with department heads to analyze actual vs. budget spending.
- Coordinating the year-end audit as well as tax planning with CPA firm.
- Member of investment committee.

4/13 – 6/16 MaidPro Franchise Corporation

Boston, MA

CONTROLLER

- Responsibility for MPF, three subsidiaries (MaidPro Inc., Fetch Storage, Rent An AC), and one non-profit (MaidPro Cares, Inc.) managing and actively mentoring a staff of two- Accounting Manager and Staff Accountant.
- Reported directly to CEO, indirectly to COO and President.
- Emphasis on process improvements in department since start, including focus on automating tasks and delegating tasks out to AM and SA.
- Created new annual departmental/company-wide budgeting process and templates and tracked actual results vs. budget across departments monthly.
- Authored corporate policies governing monthly billing/payments to/from Canadian franchisees and the handling of two different currencies.
- Authored internal services agreements between MPF and its subsidiaries with result of more simplified annual audits.
- Authored new company standard payment plans for franchisees, resulting in their renewed commitment to repay old outstanding balances.
- Extensive financial modeling (in Excel) related to the acquisitions of Fetch and RAC.
- Managed relationship with auditors with oversight of annual audits and tax preparation.

- Created internal reporting with purpose of forecasting cash flows allowing for better visibility into cash uses/sources on a daily/weekly basis.
- Managed relationships with bank, insurance agents, IRS, and other large vendors.
- Full responsibility for annual budgeting across departments with approval of CEO.

**8/12 – 9/12 Continuum Innovation West Newton, MA
INTERIM CONTROLLER (Contract)**

- Worked with Michael Page International in a temporary contract role. Primary responsibilities included cash management and collections, month-end reporting and management of staff of 4 accountants.

**6/11 – 6/12 Lodestone Management Consultants, Inc. Cambridge, MA
CONTROLLER – NORTH AMERICA**

- Full controllership responsibilities for Lodestone entities in the USA and Canada with combined revenues of \$16M.
- Report to US Partners as well as Managing Partner, CFO and Controlling Manager in company HQ located in Zurich, Switzerland.
- Responsible for controlling related to expense reimbursement, cash disbursements, accounts receivable, accounts payable.
- Collect, analyze and report results to HQ for the Americas region at each month-end.
- Work in tandem with outsourced accounting services firm to ensure timeliness of cash inflows and outflows in accordance with company policy.
- Responsible for multiple ad-hoc projects from a number of HQ departments, including Finance, Controlling, Human Resources, Legal, IT, and Global Mobility.
- Coordinate all month-end invoicing of services in SAP.
- Manage two operations associates (in areas of finance, ops and HR); one local (Cambridge) and one located in Atlanta, GA.
- Perform extensive strategic facilities analysis and benefits and payroll provider cost analysis (in the replacement of a PEO).
- Internal utilization reporting and analysis.

Position was eliminated to cut overhead costs, as LMC prepared to sell the firm (to Infosys in September 2012).

**5/05 – 6/11 Maconomy, Inc. Boston, MA
COUNTRY CONTROLLER / REGIONAL FINANCE DIRECTOR**

- Full financial and accounting responsibility (acting Finance Director as well as Financial Controller) for a \$10M (gross sales) software sales and consulting office.
- Report directly to the President of the US office and the CFO in Copenhagen, Denmark. Liaise regularly with the Director of Finance, Business Controller, and CFO in Copenhagen related to day to day business decisions and corporate policies. Frequently travel to corporate HQ.
- Month-end close and full Profit and Loss and Balance Sheet reporting to HQ, including variance analysis.
- Job costing, job creation and maintenance, and full time and expense responsibility. Month-end consulting revenue metrics tracking and analysis for the President, including utilization and realization stats. Responsible for managing WIP at month end.
- Extensive involvement in revenue recognition on both the software license side, as well as the consulting revenue side of the business.
- Full annual budgeting responsibility. Monthly present actuals vs. budget to CEO, COO, CFO and other top officers of company. Full year-end audit responsibility in tandem with outside auditors (PWC).
- Invoice all customers for license and consulting revenue. Accounts receivable management (DSO), collections, cash forecasting and management. Set up and execute international wire transfers to pay

intercompany balances. Established an interest-bearing companion savings account which has earned the company \$25K annually.

- Created, tracked, and archived all A/R contracts with new and existing customers for both license and consulting revenue.
- Full accounts payable cycle and negotiations on company leases and vendor contracts.
- Payroll documentation and processing on a semi-monthly basis with Paychex.
- Fixed assets accounting, including monthly depreciation, acquisitions and disposals.
- Expense report review and approval and the implementation, communication and enforcement of internal cost controls.
- Office management duties, including regular contact and coordination with real estate management company regarding access to the space, construction, and repairs.
- Performed all aspects of Human Resources function, including benefits coordination, new employee introductions, and the documentation of all HR-related data in the company's database. Coordination of all company off-site events. Saved the office \$13K in costs for the 2009/2010 plan year by switching representatives and implementing a new reimbursement plan.
- Formulated and managed all internal policy regarding credit notes and WIP write-offs and the documentation and reimbursement of travel expenses.
- Briefed and trained new employees on benefits, time and expense entry.
- Updated and rolled out internal policies related to time and expense to U.S. office.

With the acquisition of Maconomy by Deltek in 2010, all regional controllers were eliminated. I was offered a sales operations role but decided that I would entertain other options. Was recruited to join LMC in June.

**3/04 – 2/05 Eze Castle Software, Inc. Boston, MA
SENIOR ACCOUNTANT / BILLING MANAGER**

- Reported directly to the Controller and CFO.
- Responsible for full billing cycle for \$40M financial industry software company.
- Generated all monthly and quarterly bills for 225 clients and 6 product lines.
- Calculated and booked revenue accruals for quarterly billings.
- Reconciled revenue monthly for each product line.
- Updated and managed A/R aging, including a weekly calculation of DSO. Collected \$8M in receivables during November and December 2004- achieving a difficult collections goal.
- Reviewed contracts to ensure proper revenue recognition.
- Assisted Controller in year-end audit.

Position replaced by two junior accountants, as the majority of the work became more labor-intensive.

**2/03 – 2/04 Fuld & Company, Inc. Cambridge, MA
SENIOR ACCOUNTANT / BILLING MANAGER**

- Reported directly to the Corporate Controller in this second-most senior finance position.
- Supported Controller in daily activities such as communication with outside consultants, remote offices (UK), foreign clients (Germany, Singapore, UK, etc.), external auditors (Switzerland), and others.
- Managed the compilation and posting of all billable time and expense to clients and projects. Provided support and training to employees in the use of the time and expense system (Timekeeper).
- Liaised regularly with all vertical practice area Vice Presidents to ensure a steady and accurate billing cycle with multiple clients. Provided regular reporting and analysis to practice area VP's concerning work in process.

- Coordinated all billing- from setting up projects in MAS200 and Timekeeper to setting up billing schedules (in conjunction with VP's), to generating and sending invoices. Answered all inquiries which sporadically arise from billed clients.
- Provided analytical support to Controller during month end process, including report modeling, utilization reporting, all project-related reporting (monthly wins, backlog tracking, etc.), graphical analysis of statistically significant trends, as well as commentary related to these areas.
- Provided all initial training to the new accountant in London office in May 2003. In constant phone contact with her, providing technical and procedural support where needed.
- Supervised the work of a staff accountant who handled A/P, payroll, fixed assets, as well as collections activity. Delegated some tasks to her, when needed. Provided support for her when needed.
- Created many of the company's standardized reports in Crystal Reports.

Position eliminated with the re-hire of the prior Controller.

4/00 – 8/02 Weber Shandwick Worldwide ASSISTANT CONTROLLER

Cambridge, MA

- Worked directly with the Controller and CFO on a daily basis, performing a variety of tasks.
- Managed and completed the month-end close.
- Created and compiled all profit and loss reports for all ten U.S. offices of TWG, which were distributed monthly to office managers and senior management. Performed monthly expense account analysis.
- Responsible for consolidation of branch offices' P&L reports, as well as for variance analysis.
- Assisted Controller and CFO in annual budgeting across 10 corporate entities. Performed all necessary financial proforma modeling.
- Played a key role in G/L system conversion from Great Plains to Platinum. Made executive decisions with Controller regarding a more streamlined chart of accounts.
- Shared responsibility for completion of annual tax package reporting to corporate HQ with Controller.
- Monthly reconciliation of several domestic and international intercompany accounts, along with several other balance sheet accounts.
- Created monthly expense tracking (vs. budget) reports to be distributed to functional admin department heads, as well as CFO. Performed expense account analysis on monthly basis for CFO.
- Created a labor utilization analysis template and used it to perform monthly analysis for Director of Finance.
- Improvement in financial controls implemented by updating company's expense report (template) for 10 U.S. offices, which was posted to company intranet.

Laid off to cut costs as a result of the failure of a number of dot com firms in the company's portfolio.

EDUCATION.

MBA, Northeastern University, Boston MA. Areas of study included accounting, finance, corporate strategy, and general management.

BA, Bridgewater State College, Bridgewater MA. Major in political science, minor in history. Inducted into Pi Sigma Alpha, the National Political Science Honor Society. German language.

Computer Skills:

- Expert with Microsoft EXCEL 2016.
- Microsoft Office 2016.
- QuickBooks Pro.
- QuickBooks Online.

- Freshbooks Accounting.
- Xero Accounting Software and Online Bookkeeping.
- Salesforce.
- MAS500 with Timekeeper for time tracking.
- Zoho Mail and Expense.
- Slack.
- Zoom conferencing.
- Best Software certificate in Crystal Reports – 2003.
- Maconomy X+ project accounting software
- SAP P, FI, CO and BI modules.
- Business Objects (AnalytiX) training completed 2009.
- Paychex payroll software 2010 edition.
- Softrax Operations and Financials.
- Monthly close done using Platinum accounting package.
- Microsoft GP, with FRx as a report writer.
- FRx Software certificate – 16 CPE Credits - FRx Essentials I course completed August 25, 2004.
- Experience with Hyperion Pillar software using Retrieve in Excel.
- Bank of America CashPro.
- FAS Fixed Assets Software.
- MicroStrategy DSS Agent.
- A/P experience using Datatech and Computron in a Mac environment.
- Experience using proprietary G/L systems with USTrust and BBRG.
- Some experience with Deltek Vision.

Professional references furnished upon request.

Curriculum Vitae: Eric J. Landquist

Education

- B.S., Mathematics (Applied Discrete Mathematics Option), *Summa Cum Laude*
Minor: Computer Science, Virginia Tech, December 1998.
- M.S., Mathematics, Virginia Tech, May 2000.
- Ph.D., Mathematics, University of Illinois, January 2009.

Research Interests

- Cryptography
- Computational Algebraic Number Theory
- Number Theory
- Discrete Optimization
- Mathematical Modeling

Positions Held

- Senior Scientist, Voatz, Inc. (Boston, MA / remote), February 2021–Present.
- Associate Professor, Kutztown University (Kutztown, PA), August 2017–October 2022.
- Assistant Professor, Kutztown University, August 2009–August 2017.
- Post-Doctoral Fellow, University of Calgary (Calgary, AB, Canada), April–August 2009.
- Wissenschaftlicher Mitarbeiter (Research Assistant), Carl von Ossietzky Universität Oldenburg (Oldenburg, Germany), September 2008–February 2009.
- Research Assistant and Teaching Assistant, University of Illinois (Urbana, IL), August 2001–May 2008.
- Part-Time Faculty, Parkland College (Champaign, IL), Summer 2007.
- Visiting Researcher, University of Calgary, August 2004 and January–July 2005.
- Adjunct Faculty, New River Community College (Dublin, VA), August 2000–August 2001.
- Cryptologic Mathematician, Director's Summer Program, National Security Agency (Ft. Meade, MD), May–August 1999.
- Graduate Teaching Assistant, Virginia Tech (Blacksburg, VA), January 1999–August 2000.

RESEARCH

Selected Publications

- F. Fontein, E. Landquist, and R. Scheidler, *Class number and regulator computation in purely cubic function fields of unit rank two*. eprint arXiv:1001.4095, (2010), 1–14.
(Online at <http://arxiv.org/abs/1001.4095>.)
- E. Landquist, P. Rozenhart, R. Scheidler, J. Webster, and Q. Wu, *An explicit treatment of cubic function fields, with applications*. Canadian Journal of Mathematics. **62**(2010), 787–807.
- P. Wiltout and E. Landquist, *The Collatz Conjecture and integers of the form $2^kn - m$ and $3^kn - 1$* . Furman University Electronic Journal of Undergraduate Mathematics. **17**(2013), 1–5.
- F. Vasko, E. Landquist, G. Kresge, A. Tal, Y. Jiang, and X. Papademetris, *A simple and efficient strategy for solving very large-scale generalized Cable-Trench Problems*. Networks **67**(3), (2016) 199–208.
- E. Landquist, R. Scheidler, and A. Stein, *Class number and regulator computation in purely cubic function fields*. eprint arXiv:1601.03309, (2016), 1–30.
(Online at <http://arxiv.org/abs/1601.03309>.)
- K. Zyma, J. Girard, E. Landquist, G. Schaper, and F. Vasko, *Formulating and solving a radio astronomy antenna connection problem as a generalized Cable-Trench Problem: an empirical study*. International Transactions in Operational Research **24**(5), (2017) 943–957. DOI: 10.1111/itor.12312.
(Online at <http://onlinelibrary.wiley.com/doi/10.1111/itor.12312/full>.)
- E. Landquist, F. Vasko, G. Kresge, A. Tal, Y. Jiang, and X. Papademetris, *The Generalized Steiner Cable-Trench Problem with application to error correction in vascular image analysis*. In: Andreas Fink, Armin Fügenschuh, and Martin J. Geiger (eds.) Operations Research Proceedings 2016: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Helmut Schmidt Universität Hamburg, Springer, (2017) 391–397.
- E. Landquist, C. Reigle, and F. Vasko, *A final note on the Ones Assignment Method and its variants: they do not work*. International Journal of Industrial and Systems Engineering **29**(3), (2018) 405–412. DOI: 10.1504/IJISE.2018.093048.
- E. Koch and E. Landquist, *Secure Electronic Voting using the Paillier Cryptosystem*. Minnesota Journal of Undergraduate Mathematics, (2018). (Submitted)
- A. Douventzidis and E. Landquist, *Logarithms are Hot Stuff: A New Rating Scale for Chili Peppers*. PRIMUS, (2021) 1–11. <https://www.tandfonline.com/doi/full/10.1080/10511970.2021.1886206>
- I. Reiter and E. Landquist, *Determining Biases in the Card-Chameleon Cryptosystem*. CONTACT **2**(1), (2021). <https://research.library.kutztown.edu/contact/vol2/iss1/1/>
- E. Landquist, P. Andreae, and L. Hutchinson, *Ensuring Trustworthy Voting for Military and Overseas Voters*. National Association of Secretaries of State (NASS) 2021 Summer Conference, (2021). <https://www.nass.org/node/2286>
- E. Landquist and L. Hutchinson, *Fostering Trust and Trustworthiness in Election Infrastructure Using Trustless Technologies*. NASS 2021 Tech Talks, (2021).
- E. Landquist et al., *Parallel Internet and Paper Elections – a Practical PIPELine to Secure and Accessible Elections*. NASS 2022 Virtual Winter Conference, (2022). <https://www.nass.org/node/2362>

- P. Andreae, L. Hutchinson, and E. Landquist, *Towards Creating Standards for Remote Digital Voting*. NASS 2022 Summer Conference, (2022). <https://www.nass.org/node/2437>
- E. Landquist et al., *Take Me Out to the Blockchain*. NASS 2023 Summer Conference, (2023). <https://www.nass.org/node/2546>

External Grants

- Academy of Inquiry-Based Learning Small Grant, Category 2: *Development of an IBL Course in Cryptography* (\$2500), Educational Advancement Foundation; November 26, 2014.
- Preparation for Industrial Careers in Mathematical Sciences (PIC Math), (Approximately \$7000), Mathematical Association of America; April 27, 2016.
- Undergraduate Teaching in Mathematics with Open Software and Textbooks (UTMOST): Participating Institution, American Institute of Mathematics; October 31, 2017. (\$1,000).
- Preparation for Industrial Careers in Mathematical Sciences (PIC Math), (\$1,000), Mathematical Association of America; April 11, 2018.
- PASSHE Faculty Professional Development Council Annual Grant Program, *Open Educational Resource Development for Applied Calculus Courses*; February 14, 2019. (\$10,000).
- Preparation for Industrial Careers in Mathematical Sciences (PIC Math), Mathematical Association of America; March 25, 2019. (\$1,000).
- Emerging Mathematics And Computer Science (EMACS) Scholars Program (Co-PI), National Science Foundation; August 19, 2020. (\$999,758).

SERVICE

- College and University Committees and Activities
 - Academic Technology Committee (Spring 2012 - 2020); **Chair:** Fall 2018 - 2020.
 - College of Liberal Arts and Sciences Curriculum Committee (Spring 2015 - 2020); **Chair:** Fall 2017.
 - Affordable Learning Pennsylvania (ALPa) Campus Partner: Fall 2018 - 2020.
- Service to the Academic Community
 - Judge at the Cumberland Valley Math Modeling Challenge (2012–2017).
 - Judge for Moody's/Mathworks Mega Math Challenge, SIAM. (2015–Present).
 - Judge for the Mathematical Contest in Modeling (MCM), COMAP. (2019, 2021).
 - Supervisor for the event "Code Busters" at the Centrales Eastern Pennsylvania Regional Science Olympiad, Kutztown University, (2016–2019).
 - Co-organizer, MAA Careers in Mathematics Conference; Kutztown University; October 5, 2019.
- Other
 - Contributor, *Remote Election Technology Report*, a technical report to be published by the Remote Election Working Group of the Government Blockchain Association; (August 2021–Present).

Membership in Professional Organizations

- Government Blockchain Association 2021–Present.
- SIAM (Society for Industrial and Applied Mathematics) 2017–Present.

10. Attachment K: Sample Project Plan

This section presents a proposed Sample Project Schedule/draft Solution Implementation Plan for the project, which is typically included as part of Voatz's work plan to deliver a full working voting Solution. The schedule lists major work items, dependencies, the responsible party, and suggested dates. After its review with the Jurisdiction, it is expected that the dates and contents are finalized as part of the project kick-off.

Work items are typically grouped into these phases:

- **Phase I - Election Ready Solution:**
 - *Project Kick-off:* The project will be initiated, reviewing the schedule and the project requirements, and preparing the team members and the client team on expectations, communication, and collaboration for the project.
 - *Platform Integration and Customization Work:* During this phase the specific integration and customizations requirements will be agreed and, afterwards, will be implemented and tested, and, finally, accepted for its use in the election.
- **Phase II - Deployment in Election:**
 - *Pre-Election Staging and Support Work:* This phase, covers the configuration of the election, and the logic and accuracy testing of the configured data, before it goes live.
 - *Voting Window Work:* This phase covers the voting period (advanced voting and until election day), and includes all the support related activities, until the voting period closes and results are provided.
 - *Post-Election Work:* This phase covers all the post-election related activities, including lessons learned sessions, post-election audits, etc.

Assumptions

- Work items and dates in this sample schedule are approximate and can be adjusted following the project kickoff. Dates below are based on the information available from the RFP and, if not available, have been assumed by Voatz for initial discussion purposes.
- Not all items may apply to all jurisdictions.

Phase I - Solution Implementation

Project Kick-off

| Task | Dependencies | Responsible Party | Output/Deliverables | Proposed Time |
|--|---|--|---|---------------|
| Project Kick-off meeting | Contract signed by both parties | Project Manager | Introduce project team members and client team members. Review client expectations, and agree on communication and collaboration mechanisms for the project Review project schedule and project requirements | 1 Day |
| Customer project team intro to the Voatz platform | Demo and training session scheduled by Voatz in collaboration with the Customer | Project Manager, Product Manager | Customer project team gains understanding of features, capabilities, and use of the Voatz platform Training materials provided for ongoing reference by customer project team | 2 days |
| Review/finalize work plan and schedule | Key dates and deadlines confirmed by Customer | Project Manager, Account Manager | Formal project work plan and schedule document provided by Voatz to the Customer and project team | 1 week |
| Determine file and information sharing methods. Grant permissions. | Secure file and information sharing methods reviewed by Customer, method selected | Project Manager, Customer Success Engineer | Secure file and information sharing system established All authorized parties granted access and given appropriate permissions | 1 week |

| | | | | |
|--|--|-----------------|--|---------|
| Hold regular team meetings with Customer and Proposer to review project work and provide feedback. | Team meetings scheduled by Voatz in collaboration with the Customer Feedback provided by the Customer | Project Manager | Voatz and the Customer maintain an open line of communication, with updates and progress communicated regularly Meeting notes maintained by Voatz and provided to the Customer after each meeting | Ongoing |
| Implement problem resolution and cost and schedule control. | Project cost and schedule confirmed by the Customer | Project Manager | Formal document detailing problem resolution and cost and schedule control, provided by Voatz to the customer | 1 week |

Platform Integration and Customization Work

| Task | Dependencies | Responsible Party | Output/Deliverables | Proposed Time |
|---|---|----------------------------------|---|---------------|
| Collect requirements to determine required integration and customization to Voatz Platform | Platform requirements confirmed by the Customer | Product Manager | Report of required integration and customization to the Voatz Platform provided by Voatz to the Customer and project team | 1 week |
| Write test plans | Full understanding of and agreement on project requirements | Director of QA and Certification | Formal test plans provided by Voatz to the Customer and project team | 1 week |
| Implement defect tracking for SW defect resolution, software mishandling, or discrepancies between the built software and requirements. | Requirements document and project schedule | Director of QA and Certification | Epics and user stories for the development work and election are created in Voatz's Jira defect-tracking system. | 2 Days |
| Implement Integration with the proposed project components | Requirements document and project schedule | Product Manager | Report of the implemented integration provided by Voatz to the Customer and project team | 1 week |

| | | | | |
|--|---|----------------------------------|--|---------|
| Additional customizations | Additional app requirements confirmed by the Customer | Product Manager | Report of additional customization to the Voatz apps provided by Voatz to the Customer and project team | 1 week |
| Develop Release criteria | Product requirements | Director of QA and Certification | Specify testing criteria to determine product readiness | 1 Day |
| Testing - Internal | Test plans prepared (see above) | Director of QA and Certification | Internal testing conducted by the Voatz QA Team, report of findings provided by Voatz to the Customer and project team | 1 week |
| Develop Training and Support Materials | | Customer Success | <p>Finalized schedule for training and support dates</p> <p>Customer-tailored training plan and materials for target audiences</p> <p>Support materials (FAQs, videos, quick starts, agent scripts, etc.)</p> | 1 week |
| Customer acceptance testing (UAT) | Voatz creates a customer acceptance checklist. Customer reviews and approves checklist. | Project Manager | <p>Acceptance testing conducted by the Customer, report of findings provided by the Customer to Voatz for analysis and implementation</p> <p>Delivery of:</p> <ul style="list-style-type: none"> - an election-ready error-free version of the required solution to be deployed during the election event, - the SLA and - the deployment plan. | 2 weeks |

Phase II - Deployment in Election

Pre-Election Election Staging and Support Work

| Task | Dependencies | Responsible Party | Output/Deliverables | Proposed Times (Approximate) |
|-------------------------------------|---|-------------------|---|--|
| Create live election instance | Official Notifications | Product Manager | The election record is created in the Voatz database. The election can be accessed in the Voatz Admin Portal. | Around 65 days prior to Election Day (ENW) |
| Import voter list | Voter list provided by the Jurisdiction for secure import, VRS integration | Product Manager | Voter list finalized in the Voatz system (additions, changes, and deletions can be made at the Customer's discretion) | ENW + 5 to 10 days |
| Review imported voter data | Voter list imported, VRS integration | Customer | Voter data vetted by customer administrators for accuracy and currency | ENW + 10 to 12 days |
| Create live administration accounts | List of authorized administrators provided to Voatz by the Customer | Infosec Engineer | Authorized election administrators given access to the Voatz system via the Administrator Portal | ENW + 5 to 10 days |
| Import ballot definition data | Election data (PDFs and Ballot Definition Exports) provided by the Jurisdiction | Product Manager | Election data imported in the Voatz system for election preparation | ENW + 10 to 12 days |
| Generate ballots | Contest information and all ballot styles data from the Customer | Product Manager | Finalized digital ballots for review and approval by the Customer | ENW + 15 days |
| Ballot review/approval | Voatz to provide finalized ballots for review and approval by the Customer | Customer | Approved ballots | ENW + 15 to 17 days |

| | | | | |
|---|---|-----------------------------------|--|--------------------------|
| Deploy voter resources (landing page, training materials) | All Voatz-generated landing page materials approved by the Customer for publication Desired Customer-generated landing page materials (if any) provided to Voatz | Product Manager, Development Lead | Landing page published for external viewing on the web | ENW + 10 days |
| Provision Support Desk | Support Desk availability and contact channels approved by the Customer | Product Manager | Voatz Support Desk will be available during agreed-upon windows and via the agreed-upon channels to serve the needs of voters and administrators | ENW + 15 days |
| Train poll workers | Training session scheduled by Voatz in collaboration with the Customer | Customer Success | Poll workers fully trained on use of the Voatz platform Training materials provided for ongoing reference by election officials | ENW + 10 days to 17 days |
| Logic & Accuracy (L & A) testing | Approved ballots L&A Test Deck | Voatz /Customer | Customer completed end to end test of ballots and integrations | ENW + 17 days to 19 days |

Voting Window Work

| Task | Dependencies | Responsible Party | Output/Deliverables | Proposed Time |
|------|--------------|-------------------|---------------------|---------------|
|------|--------------|-------------------|---------------------|---------------|

| | | | | |
|-----------------------------|--|---|---|--|
| Live voting support | Voatz Support Desk deployed (see above) | Voatz Support Team | All voters who require assistance to participate in the election process will be aided by the Voatz Support Desk, to ensure maximum participation | Start of UOCAVA/Absentee Voting window till Election Day |
| Voter list updates | Corrections and updates to voter list data, as required and approved | Election officials & Voatz Support Team | Updated voter list | Start of UOCAVA/Absentee Voting window till Election Day |
| Election Closing Procedures | All authorized election officials have their access keys configured | Project Manager, Product Manager | County clerks close the election, unlock the digital lockbox and start printing the digital ballots for seamless tabulation | Election Day |

Post-Election Work

| Task | Dependencies | Responsible Party | Output/Deliverables | Proposed Time |
|-----------------------------------|---|--|---|--|
| After-action/improvement planning | Feedback from the Customer provided to Voatz for review | Project Manager, Account Manager | A consolidated project report including outcome analysis, feedback, areas for improvement, etc. | Election Day+1 day - Election Day+5 days |
| Post-election audit | Identify audit team | Voatz/Customer /Designated members of the public | Post-election audit report | TBD by Customer |

11. Attachment L: Detailed Voting Processes and Voter Guides

Attachment L-1: Detailed Voting processes

Attachment L-1-1: End to End Voting Process

Please see the following pages that include the description with images of the different steps of the voting process.

End to End Voting Process (Hybrid Election)

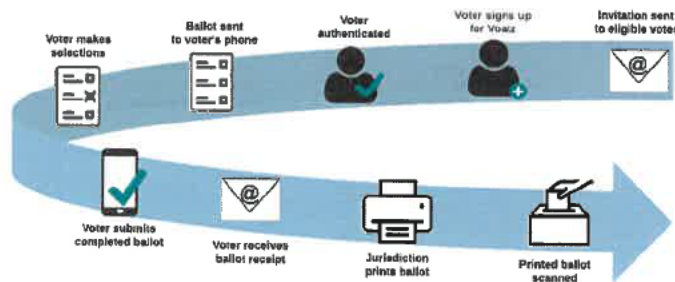
1. How does the voter receive their mail ballot electronically?

As a prerequisite to ballot delivery, Voatz requires a file that pairs each voter eligible to receive a ballot electronically with a precinct and ballot style (or access to an API that will output the ballot style for a given voter). This information is used for identity verification and to deliver the proper ballot style.

Once the election window has begun, a voter can receive his or her ballot via the **Voatz Mobile App (VMA)** or the **Voatz Web App (VWA)** for electronic marking and return. The Voatz Web App can also be used by voters who wish to print and mail their marked ballot.

The following screenshots demonstrate how voters identify themselves in each app, allowing the Voatz platform to identify their precinct and ballot style.

VMA: Voatz Mobile App

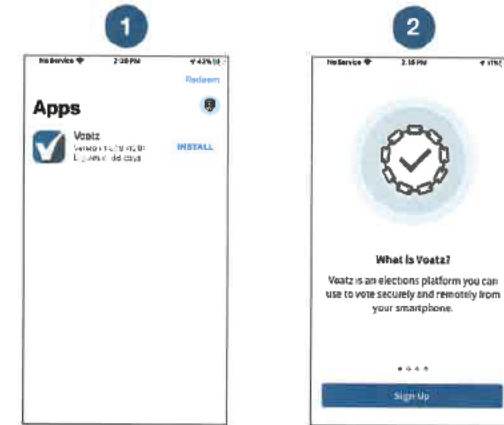


Voting Process in VMA (electronic ballot return)

The Voatz Mobile Application (VMA) can be downloaded to Apple and Android smartphones and tablets from their respective App/Play Stores. The steps below are illustrative of a potential voting workflow.

1. Download and access the Voatz Mobile App.

Downloading and accessing VMA are the first steps in the voting process and the first phase of Identity Proofing.



Download the Voatz app to your phone. (iOS 13.0+ or Android 8.0+ is required.)

Tap **Sign Up**. Make sure you have a strong and secure internet connection.

Download and Install VMA



Enter the mobile number and email address that you used to register for this election



Verify your mobile number by entering the SMS code you receive.

Verify Device and Phone Number



Follow the instructions on your device to create a Votz PIN that you'll remember (Android phones require 12 digits). It cannot contain 3 or more sequential (878) or repeating (333) numbers.

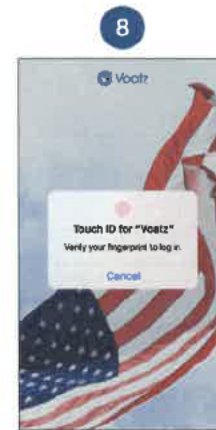


Scroll through the Terms and tap the box for "I have read and agreed to the above". Then tap [Continue](#).

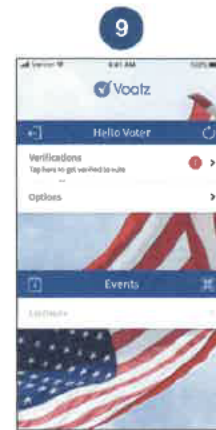


You have successfully signed up for Voatz!

Create a Voatz PIN (for use if biometric Identification is not configured)



Log In with your phone's Touch ID, Face ID, or the Voatz PIN you just created.



Tap [Verifications](#) to begin the verification process.

Log into VMA

VOTER VERIFICATION OPTIONS:

Voatz offers configurable identity verification options:

Option 1 is Manual identity Verification and requires the voter to enter a Verification Key and a Voting PIN provided by the jurisdiction and any other field required such as voter's date of birth or license number. Option 2 is Government Issued Photo ID verification.

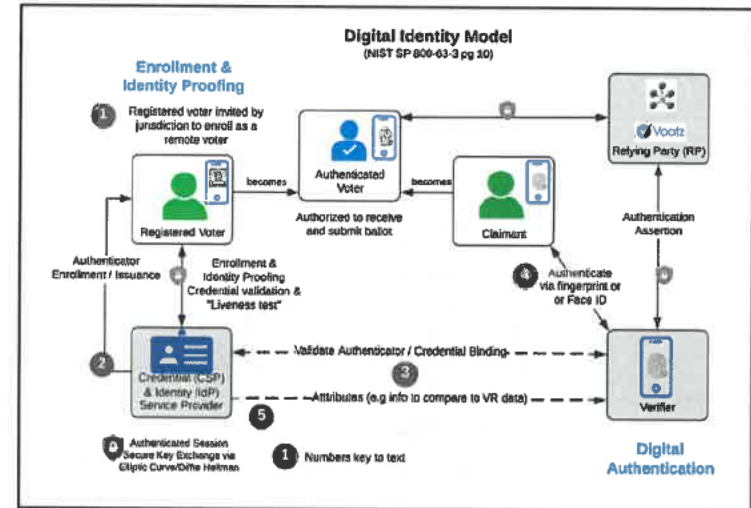
Option 1: Manual Identity Verification



Verification Option 1: Verification Key

Option 2: Remote Photo ID Verification

The Solution includes a comprehensive identity verification and voter data management system that supports multiple forms of ID (as simple as webform self-affirmation scaling to standard government issued documents) and secondary verification sources.



NIST Digital Identity Model

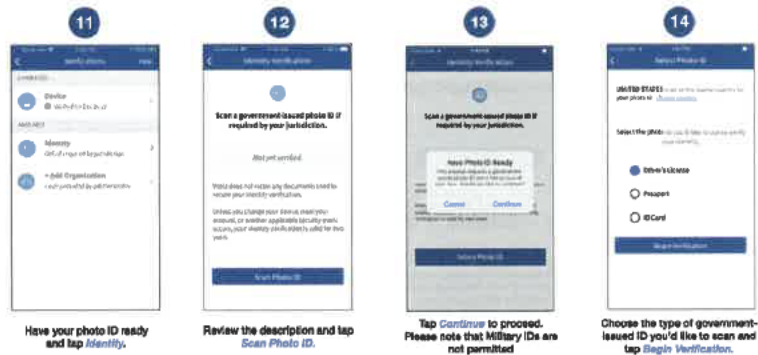
- 1. Remote Identity Proofing:** In the first phase, the registered voter presents an approved government-issued photo ID (the "credential") to the jurisdiction. An authorized Credential Service Provider (CSP) verifies the validity of the government-issued credential. An authorized Identity Provider (IdP) performs a biometric test for "liveness." Identity validation is done by comparing the photo image of the remote person to the picture in the government issued credential. The secure communication between a relying party and third party CSPs and IdPs is done through the process called "federation" described below.

Voatz uses the smartphone or tablet's digital camera to capture an approved credential (i.e. driver's license, passport, or government issued photo ID). The Voatz application integrates a third party CSP to perform both credential verification and "liveness" detection. Credential verification is a sophisticated process in which the CSP automatically examines the security artifacts of the voter's credential against their database of known credentials. Confirmation of "liveness" is performed biometrically by taking a video "selfie," where the voter is asked to nod their head and blink their eyes. Voatz has two options to perform the comparison of the video selfie to the photograph of the voter's credential: manually by Voatz personnel during the small pilots of governmental elections and automatically during a relatively large election. If the

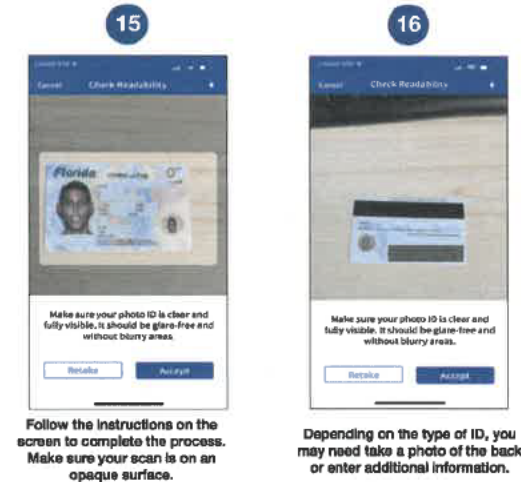
automatic verification fails, then an authorized human makes the decision; this may include a video chat with the requesting person (e.g. via Skype). Biometric data is retained only as long as it takes to verify the identity of the registered voter in the real world.

Note: There may be a delay in comparing the video selfie to the voter's credentials; until then the Voatz application will indicate a "pending status." The voter cannot open their ballot until notification of a successful resolution of identity is received by the application.

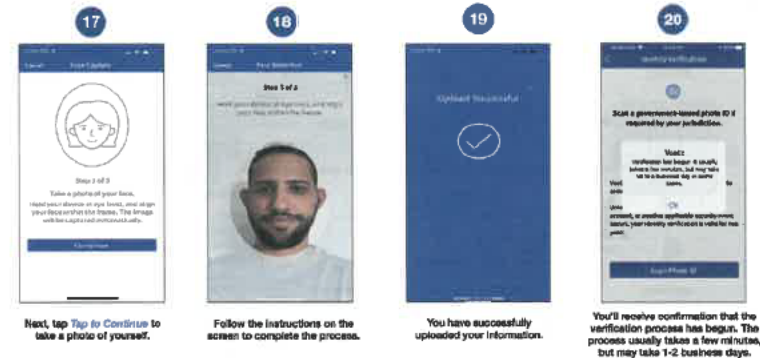
Voatz bases its identity proofing mechanisms on the requirements of the Client, and can support identity proofing mechanisms ranging from voting credentials distributed to voters via mail, to the government issued document verification process described above. While the latter process offers the greatest level of security, Voatz understands that each client has unique needs, and offers a range of identity proofing options to meet those needs.



Verification Option 2: Prepare to Scan ID



Verification Option 2: Scan Front and Back of ID



Verification Option 2: Take a Photo of Yourself

Identity proofing answers the question, "Is this person who they say they are and does that person exist in the real world?"

2. **Binding:** Binding is the final phase of identity proofing. Here, the user biometrically binds the identity-proofed person to their personal smartphone or tablet. Binding prevents an identity-proofed person from voting on another device and prevents another person from voting on the device that went through the identity-proofing process. For security reasons, it is critical that this step be done in the same session as the identity proofing step.

Binding is achieved when the Voatz application requests the user to re-authenticate themselves in the same way they did to gain access to their device – via fingerprint or Face ID if the device is configured for biometric authentication, and via VMA-specific PIN otherwise.

Binding prevents a remote voter from casting a ballot more than once and enables secure authentication.

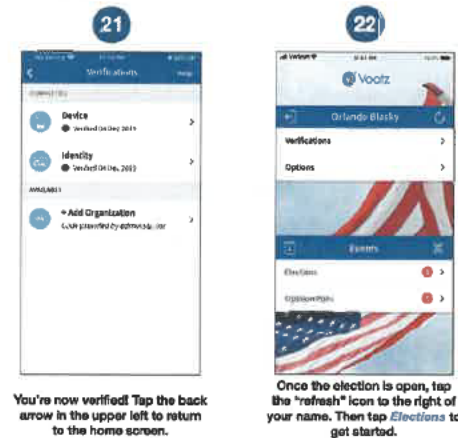
3. Access to ballot and voter information.

Access to the ballot is granted through the following processes:

Authentication: Voter authentication occurs in every election; it is unlike identity proofing which is episodic. Authentication is central to the process of associating the device to the identity of a remote voter just prior to their act of voting – but not how they voted. Authentication is performed by verifying that the voter possesses at least one method, called a “valid authenticator,” of ensuring that their identity can be confirmed at the time of voting. If a voter’s device is configured to authenticate the user with biometric capabilities (e.g., fingerprint, Touch ID, or Face ID), then that can serve as a valid authenticator. Otherwise, authentication is done with a 12-digit or 16-digit PIN created by the voter specifically for VMA.

Voatz uses the smartphone or tablet as the “valid authenticator.” In the Voatz application, authentication is requested twice in a voting session: once when the voter tries to open their blank ballot and again when the voter submits their voted ballot. Authentication – biometrically (if enabled on the device) or by PIN (otherwise) – provides a registered and identity-proofed voter with the authorization to access their blank ballot and the authorization to submit their voted ballot.

Voatz carefully follows the required protocols for certificate key exchange (using Elliptic Curve Diffie-Hellman Exchange) and standard bulk encryption (using AES 256). These standards provide for secure communications between the application and the voter.



Verification is complete. Voting can begin when the election opens.

Authentication provides assurance that the person who voted today on their personal device is the same person who verified their identity previously on that device.

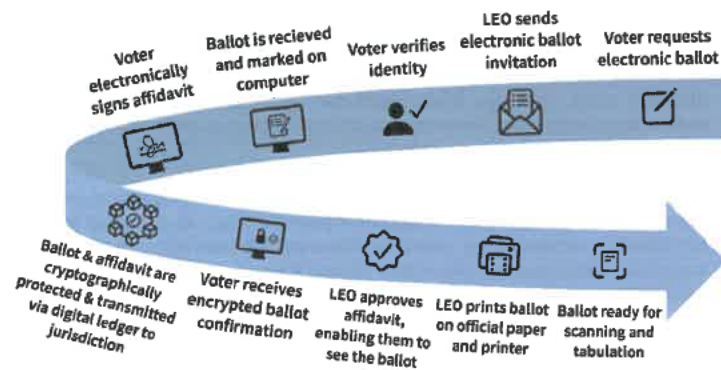
Federation: Federation is a process that allows for the secure conveyance of authentication and voter attribute information across networked systems. In a federated system, the Credential Service Provider (CSP) and the Identity Provider(s) (IdP), provide identity services to the application developer, called the Relying Party (RP). Federation requires relatively complex multiparty protocols that have subtle security and privacy requirements and require careful consideration.

Voatz implements credential validation and “liveness” detection using the NIST 800-63-3 or ISO/IEC 30107-3 (Biometric presentation attack detection) guidelines. Voatz carefully follows the required protocols for certificate key exchange using Elliptic Curve Diffie-Hellman Exchange and standard bulk encryption methods (using AES 256). These standards provide for secure communications between independent service providers, the relying party, the voter and the jurisdiction.

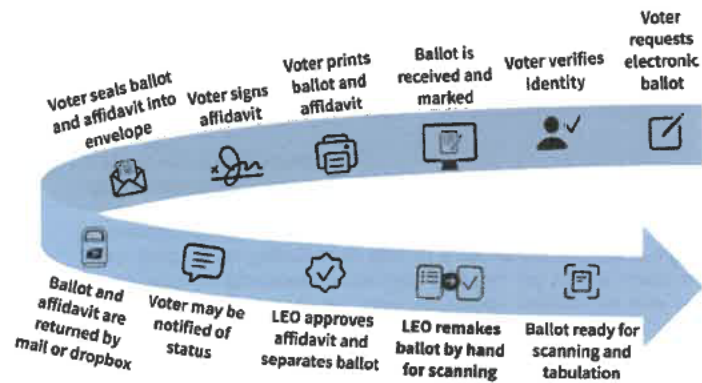
Federation speeds progress by enabling “best-of-breed” technologies to be securely integrated into a solution.

VWA: Voatz Web App

The workflow diagrams below present the two VWA voting options for ballot transmission from the local election office ('LEO') to the voter and its return.



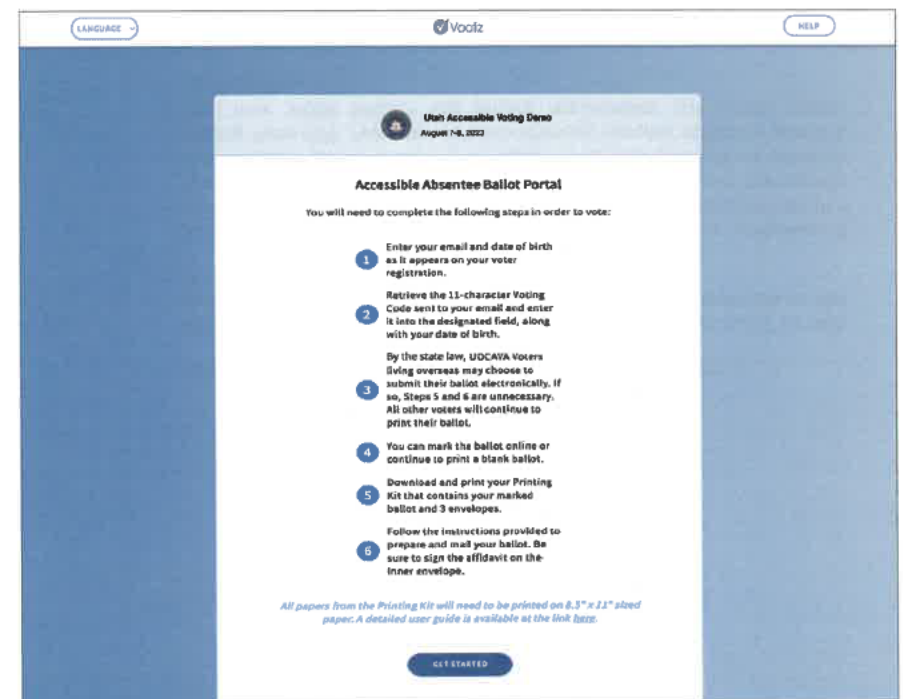
Voting Process in VWA (electronic return option)



Voting Process in VWA (voter-printed & mail return option)

The following steps in VWA allow voters to identify themselves and receive a ballot electronically.

1. Authorized voters access VWA via a URL provided to them for the election. The landing page may provide instructions on the steps needed to successfully cast a ballot. Voting must be done in one session (but can be restarted if interrupted.)



Sample Voatz Web App Landing Page with Language Selection Dropdown Menu

2. A voter verification screen prompts them to enter a mix of personal identification data as required by the jurisdiction, for example, name, date of birth, Voter ID, and email address. This information is validated against voter data in the Voatz database.

The screenshot shows the 'Voter Verification' screen of the 'Utah Accessible Voting Demo' (August 7-8, 2023). The page has a blue header with a 'LANGUAGE' dropdown menu on the left and a 'HELP' button on the right. The main content area is white and contains the following elements:

- Voter Verification** title.
- Instruction: 'Please enter your information exactly as it appears on your passport. All fields are required'.
- Form fields: 'First Name (id)', 'Last Name (id)', 'Date of Birth' (with a calendar icon), 'Email', and 'Confirm Email'.
- A 'Voter ID' dropdown menu.
- A 'VOTE' button at the bottom.
- A small icon and text 'It's not a robot' with a refresh button.

Sample VWA Voter Verification Screen with optional Language Dropdown Menu

3. **OPTIONAL SECURITY CONFIGURATION - PHOTO ID**
Voters upload the front and back images of an official photo ID.

The screenshot shows the 'ID Verification' screen of the 'Utah Accessible Voting Demo' (August 7-8, 2023). The page has a blue header with a 'Voatz' logo on the left and 'HELP' and 'LOG OUT' buttons on the right. The main content area is white and contains the following elements:

- ID Verification** title.
- Instruction: 'Attach an image of your ID. JPEG, PNG, and HEIC formats are allowed. Maximum file size is 4MB.'.
- Two upload areas: one with a 'CHOOSE FILE...' button and another with a dashed box and the text '...or drag your file here.'.
- Text: 'If you are not able to attach your document, you can still participate in person. For more information, visit [toll-free 800-448-8888](#)'.
- A 'VOTE' button at the bottom.

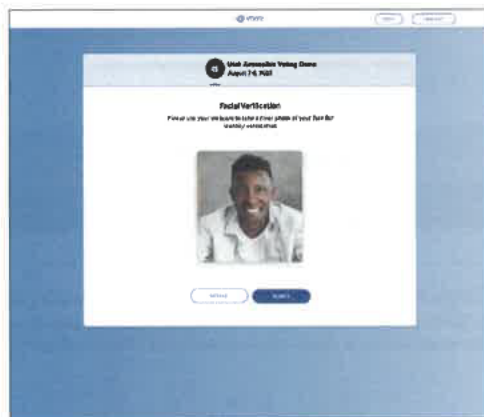
Optional VWA ID Verification Screen

4. **OPTIONAL SECURITY CONFIGURATION - SELFIE UPLOAD**

The voter takes a picture of his or her face in order to match the image on the photo ID. This requires the device to have a camera in order to proceed. (More details on this verification process were given in the corresponding description of VMA.)

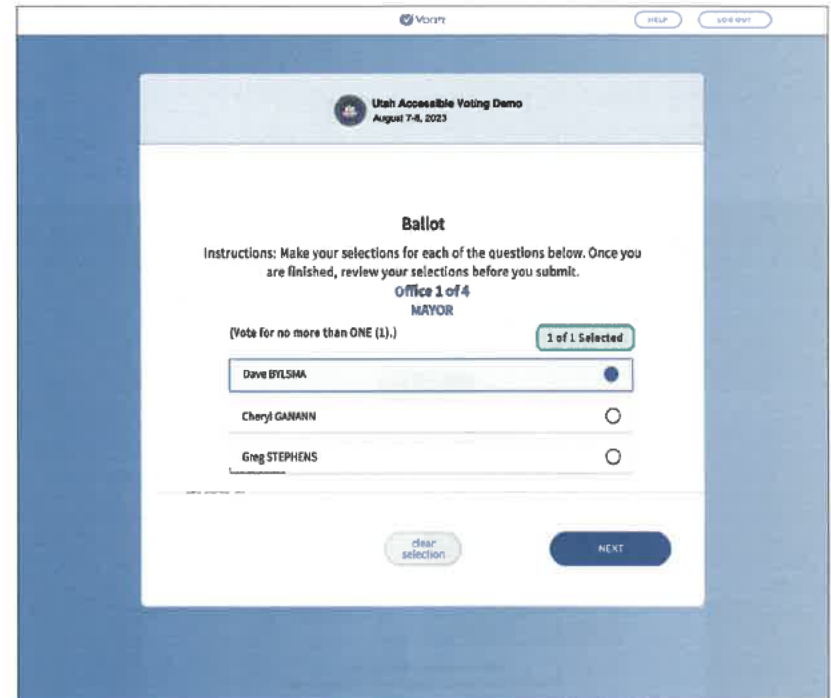


Optional Sample VWA Facial Validation Screen



Optional Sample VWA Facial Validation Screen

5. Once a voter's identity is confirmed – by pairing the photo ID to the live image of the voter, the voter is taken to the ballot.



VWA Contest on a Sample Ballot

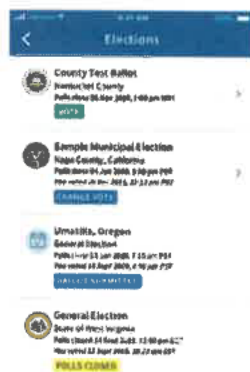
4. **How does a voter mark their mail ballot electronically?**

VMA: Voatz Mobile App

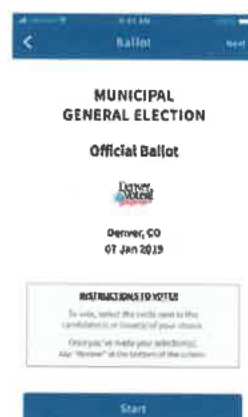
1. Voting the ballot (including undervote warnings and prevention of overvotes).

Selections for choices (candidates or ballot questions) are made one contest at a time by touching a candidate's name. A simple, familiar interface speeds the voting experience and provides instructions in the beginning of the process. The typical voting session is under two minutes. The Voatz application also supports the Voluntary Product Accessibility Template (VPAT) to support voters with disabilities (e.g. Screenreaders, font enlargement, flexible timeouts, etc.).

Voters are prevented from selecting more choices than allowed to ensure that only their allotted number of votes count. At any time before submission, the voter can review their choices and make changes if necessary (this feature is configurable). Once finished, the voter submits their ballot. Once submitted, all information is anonymized, routed via a "mixnet" and posted to the blockchain.



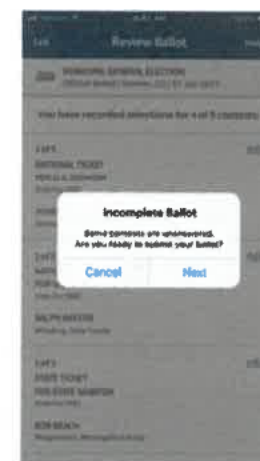
Available Elections



Voter Instructions



Contests on a Ballot



Undervote Warning

VWA: Voatz Web App

1. To make their choice(s), voters select the candidate(s) they wish to vote for by clicking on their name or anywhere in their designated box on the screen. (For keyboard only use, use of Enter or Space selects the choice.) If desired, voters can click the "Info" button to view additional information and resources for each candidate. The information found via the (i) button can be customized by the jurisdiction and can include embedded media and links to external resources.

Ballot

Instructions: Make your selections for each of the questions below. Once you are finished, review your selections before you submit.

Office 1 of 4
MAYOR

(Vote for no more than ONE (1).)

1 of 1 Selected

Dave BYLSMA ☒

Cheryl GANAMIN ☐

Greg STEPHENS ☐

clear selection NEXT

VWA Contest on a Sample Ballot

Ballot

Instructions: Make your selections for each of the questions below. Once you are finished, review your selections before you submit.

Office 1 of 4
MAYOR

(Vote for no more than ONE (1).)

2 of 1 Selected

Dave BYLSMA ☒

Cheryl GANAMIN ☒

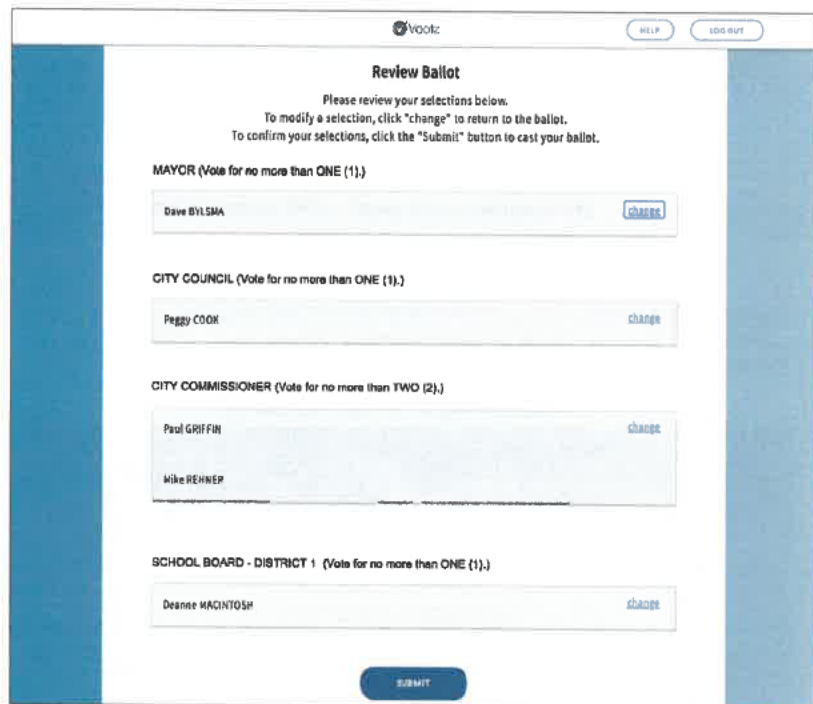
Greg STEPHENS ☐

The maximum number of selections for Mayor is 1. Please deselect at least 1 choice in order to continue.

clear selection NEXT

Sample Overvote Warning – the “Next” Button is Deactivated

2. After completing the ballot, voters are shown a review screen where they can see an overview of their selections. Voters can navigate back to any contest and change their selections by clicking the “change” link located next to each contest.



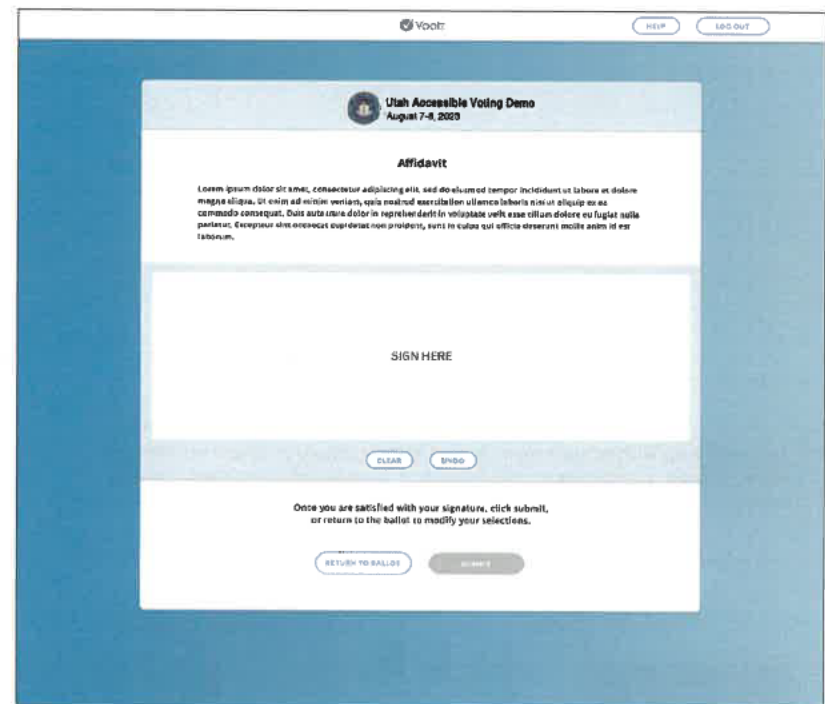
The screenshot shows the 'Review Ballot' screen in the Voatz application. At the top, there's a header with the Voatz logo, a 'HELP' button, and a 'LOG OUT' button. The main heading is 'Review Ballot'. Below it, instructions state: 'Please review your selections below. To modify a selection, click "change" to return to the ballot. To confirm your selections, click the "Submit" button to cast your ballot.'

The ballot contains four sections, each with a list of candidates and a 'change' button:

- MAYOR (Vote for no more than ONE (1).)**
 - Dave BYLSMA
- CITY COUNCIL (Vote for no more than ONE (1).)**
 - Peggy COOK
- CITY COMMISSIONER (Vote for no more than TWO (2).)**
 - Paul GRIFFIN
 - Mike REHNER
- SCHOOL BOARD - DISTRICT 1 (Vote for no more than ONE (1).)**
 - Deanne MAGINTOSH

At the bottom center, there is a large blue 'SUBMIT' button.

Sample VWA Review Screen



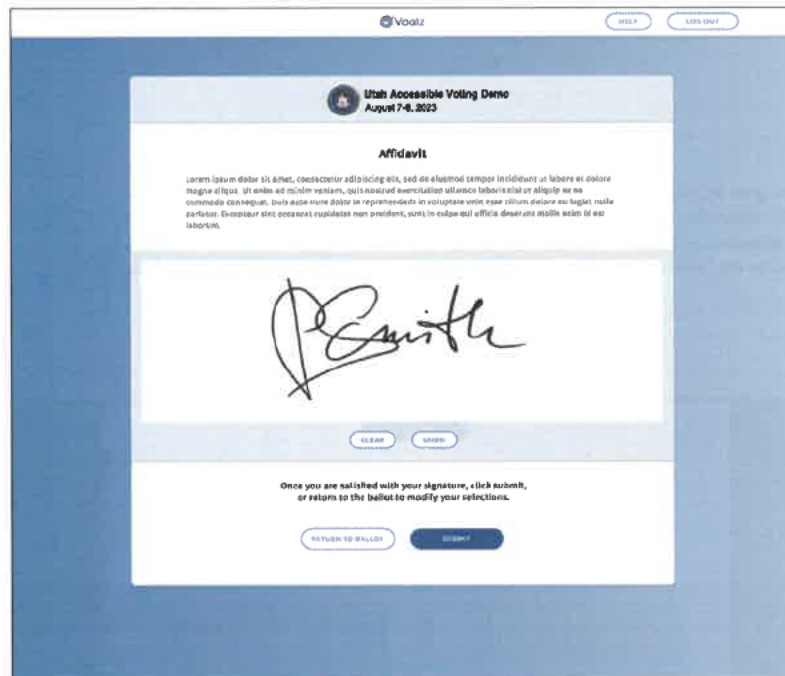
The screenshot shows the 'Affidavit' screen in the Voatz application. At the top, there's a header with the Voatz logo, a 'HELP' button, and a 'LOG OUT' button. The main heading is 'Affidavit'. Below it, there's a sub-header 'Utah Accessible Voting Demo' and the date 'August 7-8, 2023'.

The main content area contains a large text box for the affidavit, followed by a 'SIGN HERE' label and a large white box for the signature. Below the signature box, there are 'CLEAR' and 'END' buttons.

At the bottom, there's a message: 'Once you are satisfied with your signature, click submit, or return to the ballot to modify your selections.' Below this message, there are 'RETURN TO BALLOT' and 'SUBMIT' buttons.

Sample VWA Affidavit and Signature Screen

- Next, the voter must sign an affidavit affirming their eligibility to vote. We note that this signature will appear in the Voatz Administration Portal for election officials to compare to the signature on file; by accepting this signature as legitimate, the election official accepts the ballot, just as with physical mail-in ballots.



Sample VWA Affidavit and Signature Screen

4. Voters are prompted to confirm their intent to submit their ballot by clicking "Submit" when the popup appears as shown below. Until they click "Submit," voters still have the option of navigating back to their ballot and changing their selections, by clicking the "Cancel" and "Return to Ballot" buttons.

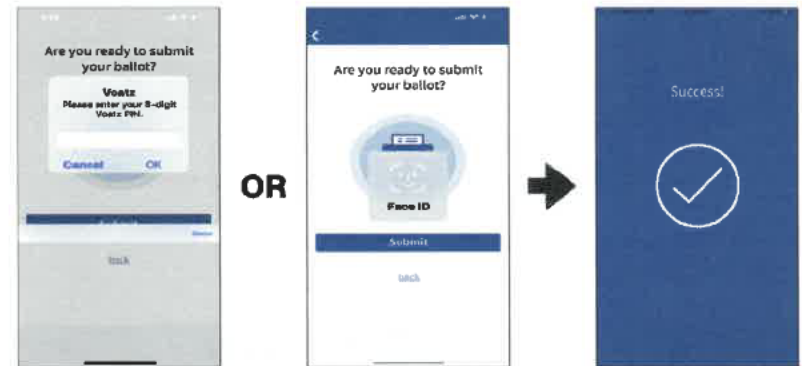
5. How do voters transmit [return] their ballot electronically?

VMA: Voatz Mobile App

1. Confirmation messages to voters.

The voters receive voting confirmation messages before and after a vote is cast. Before a ballot is submitted, the application confirms the voter's identity. Once a

ballot is successfully submitted, the application will let the voter know that their votes have been successfully recorded.

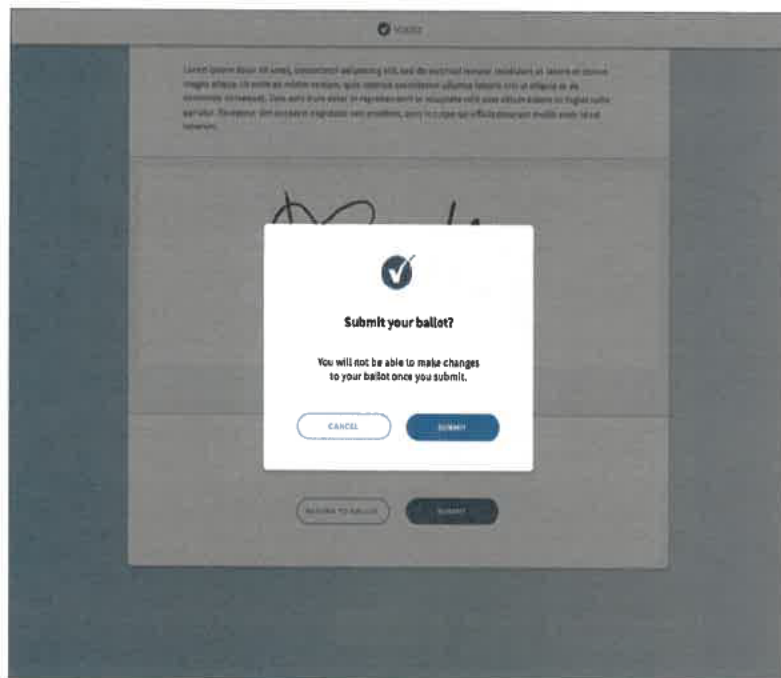


Biometric Confirmation OR PIN

Voting Confirmation

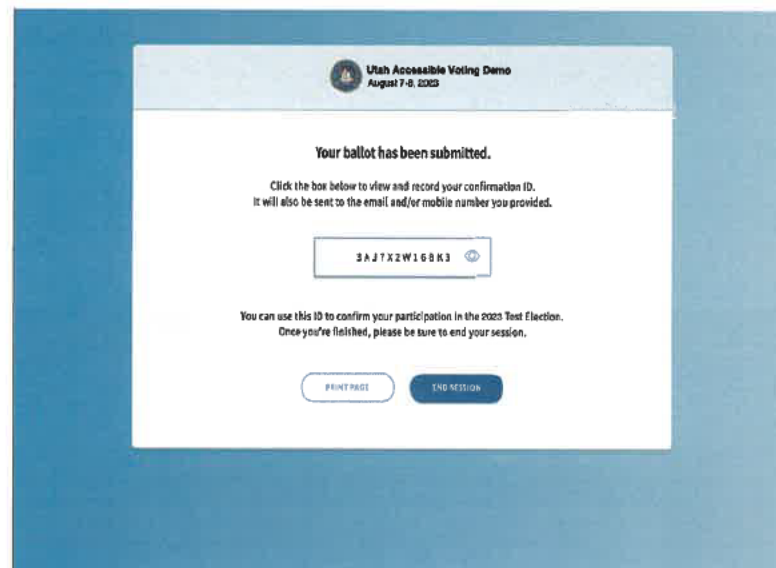
VWA: Voatz Web App

1. Voters are prompted to confirm their intent to submit their ballot by clicking "Submit" when the popup appears as shown below. Until they click "Submit," voters still have the option of navigating back to their ballot and changing their selections, by clicking the "Cancel" and "Return to Ballot" buttons.



Sample VWA Ballot Submission Screen

2. After submitting their ballot, voters are taken to a confirmation screen where they are provided with a confirmation number which they can print or email for their records. Optionally, encrypted vote receipts can be emailed or texted to the voter upon submission (in addition to the confirmation code provided on screen, or in place of the confirmation code).



VWA Ballot Submission Confirmation Screen

6. What assistive devices can voters use to mark their mail ballot?

Voters use their own COTS device such as an iOS or Android mobile phone, or desktop/laptop computer. These devices incorporate sophisticated assistive technologies such as screen readers, mouseless use, Voice Control (hands free use) among others. The Voatz mobile and web apps are designed to support these capabilities and meet WCAG 2.1 as described further in our Accessibility Compliance Statements (aka. VPATs) which accompany this proposal and also available on our website (<https://voatz.com/Accessibility-Statement/>).

7. How does a voter sign their ballot materials electronically?

VMA: Voatz Mobile App

Before submitting a ballot via VMA, a voter may be prompted to supply a digital rendering of a hand signature on the affidavit page. Voters can use a finger, a stylus, or other compatible assistive device to render a signature. This signature feature is accessible with screen readers enabled. After the ballot has been submitted and its contents recorded on the blockchain, the affidavit and signature is available in the Administrative Portal for election staff to confirm receipt and verify validity.

The image displays two side-by-side screenshots of the Voatz Mobile App (VMA) interface. Both screens show the 'Voter Affidavit' page for a 'General Election | I VOTED STICKER - No Recurrence'. The top status bar shows the time as 4:33 and 4:34, and the signal as 5G. The left screenshot shows the 'SIGN HERE' prompt, and the right screenshot shows a digital signature. Below the signature, there are buttons for 'Clear Signature' and 'Next'.

4:33 5G
Back Voter Affidavit Help
General Election | I VOTED STICKER - No Recurrence

By signing this affidavit, I am stating under penalty of law:

I am a registered voter qualified to vote in the precinct in which I am registered; this ballot belongs to me; I voted the ballot; and I am not a convicted felon currently incarcerated for the commission of a felony.

I understand that by electronically transmitting my voted ballot I am voluntarily waiving my right to a secret ballot.

SIGN HERE

Clear Signature Next

4:34 5G
Back Voter Affidavit Help
General Election | I VOTED STICKER - No Recurrence

By signing this affidavit, I am stating under penalty of law:

I am a registered voter qualified to vote in the precinct in which I am registered; this ballot belongs to me; I voted the ballot; and I am not a convicted felon currently incarcerated for the commission of a felony.

I understand that by electronically transmitting my voted ballot I am voluntarily waiving my right to a secret ballot.

SIGN HERE

Clear Signature Next

Sample VMA Affidavit and Signature Screen

VWA: Voatz Web App

Before submitting the ballot, the voter must sign an affidavit affirming their eligibility to vote. This signature may be written via mouse, a connected digital pen, by hand (if a user has a touch screen), or by another compatible assistive device.

The image shows a screenshot of the Voatz Web App (VWA) interface. The top navigation bar includes the Voatz logo, a 'HELP' button, and a 'SIGN OUT' button. The main content area displays the 'Utah Accessible Voting Demo' for 'August 7-8, 2023'. Below this, the 'Affidavit' section contains a paragraph of text and a 'SIGN HERE' prompt. At the bottom, there are buttons for 'CLEAR' and 'END'. A message at the bottom states: 'Once you are satisfied with your signature, click submit, or return to the ballot to modify your selections.' Below this message are buttons for 'RETURN TO BALLOT' and 'SUBMIT'.

Voatz HELP SIGN OUT

Utah Accessible Voting Demo
August 7-8, 2023

Affidavit

lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

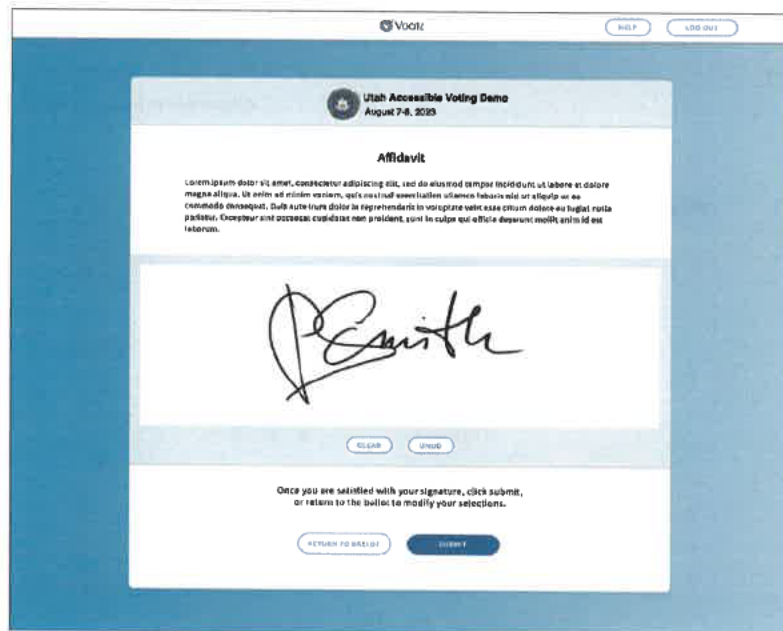
SIGN HERE

CLEAR END

Once you are satisfied with your signature, click submit, or return to the ballot to modify your selections.

RETURN TO BALLOT SUBMIT

Sample VWA Affidavit and Signature Screen



Sample VWA Affidavit and Signature Screen

8. How are ballots hosted before, during, and after transmission to the voter.

Ballots are stored in two formats: PDF files and JSON files. In addition, election definition files (EDFs) and intermediate ballot definition files (BDFs) can be used to create the ballot JSON files for each ballot style and precinct. After the secure transfer of ballot PDFs and EDFs from the State or counties, these files are stored in a secure folder on our network, which is hosted within the United States. One of our engineers, who is a United States citizen and who resides within the United States, will process the EDFs into JSONs. The ballot JSONs are then stored in an encrypted database on a Voatz server in Boston. The PDF ballots will also be transferred to an encrypted AWS S3 bucket, which will be hosted in the US-East-1 region in northern Virginia.

During transmission, marked ballots, in the form of JSON files, are encrypted from the user's device to AWS storage, to the Voatz relay server, and to the blockchain. Ballot PDFs to be mailed will be marked in AWS cloud servers, encrypted, and sent back to the voter's device.

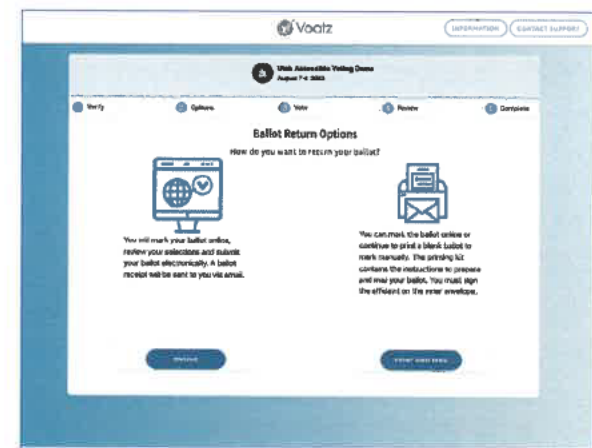
After transmission, all ballot data is erased from the voter's device, both in VMA and VWA.

9. How is the voter's ballot transmitted to the Elections office?

The Voatz Web App offers two methods for ballot submission: voters can either print their marked ballot and mail it along with a pre-addressed envelope and affidavit, or submit their marked ballot electronically after signing an affidavit.

Voter-printed ballot with mail delivery

If a voter chooses to mark and mail their ballot to the Elections office, the Voatz Web App prepares the marked ballot, pre-addressed envelope and affidavit for the voter to properly print and mail their ballot.



Sample VWA screen to select Ballot Return option



Sample VWA Printing Ballot & Envelopes Screen



Sample VWA Print Ballot/Envelopes dialog

Electronically-returned ballots

When the voter chooses to return their marked ballot electronically, the voter signs the oath affidavit (if applicable) on the device (either within VMA or VWA) and submits the ballot. The

ballot is then encrypted, anonymized and delivered to the digital ballot lockbox along with affidavits. These ballots and affidavits become available via the Voatz Administrative Portal (VAP) for processing by the designated election officials.

Attachment L-1-2: Election Administration Process

Please see the following pages that include the description with images of the different steps of the election administration process.

Election Administration

1. How does an Election Office transfer its ballot designs to Voatz?

Once the paper ballot designs are finalized, the election office will need to provide the final ballot PDF files, along with the associated ballot definition files via a secure file transfer (i.e., SFTP or secure Dropbox upload or encrypted email) to the Voatz IT Ops Team.

The above files are then ingested and processed and the digital ballot designs are available in the Voatz Admin Portal for random proofing (if desired) by the election office staff or the respective County election officials. The diagram below depicts the process.

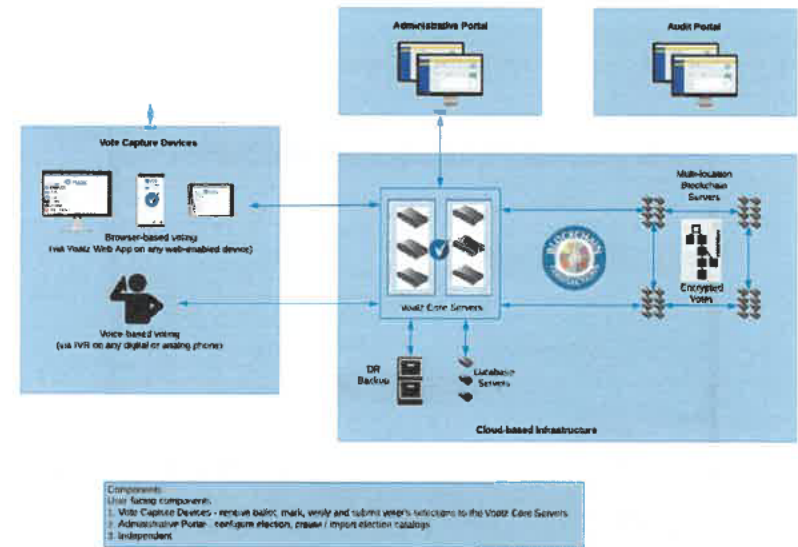
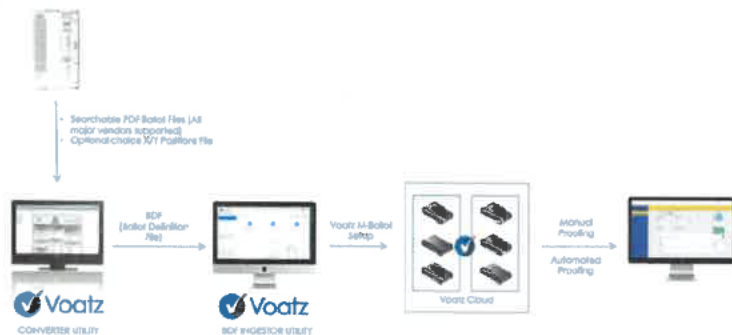


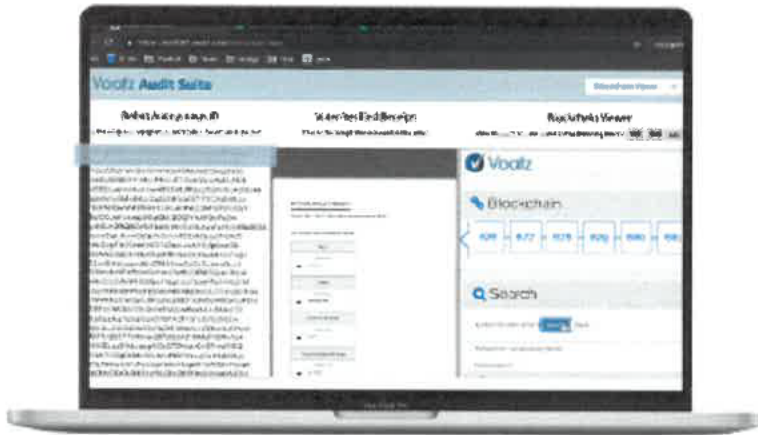
Image: Voatz Platform Component Architecture

2. How does Voatz create digital ballots?

Prior to the opening of the election, Voatz engineers will create a ballot definition file (BDF) from the Election Definition File (EDF), and from that BDF, a JSON file for each ballot style is created. During the election window, the appropriate ballot is transferred as a JSON file to the voter's device. If a voter chooses to print and mail their ballot, then the appropriate ballot PDF is marked and made available to print. All personal information and ballot data in transit is encrypted at the application level (via AES and ECC) and the transfer (TLS) level. No ballot information is retained on the voter's device for security purposes.

3. How does an Election Office approve affidavits and print ballots?

An election official completes voter authentication by accepting a voter's affidavit within VAP; this action accepts the voter's marked ballot and makes it ready for printing/tabulation. The following image shows a sample voter affidavit review screen within VAP.



Post-Election Ballot Audit Suite

All data at rest is encrypted with AES-256. This includes ballot data stored in databases used by VAP, VMA, and VWA on the AWS backend and on the core server. In VMA, sensitive data is encrypted using the secure hardware module on iOS and Android. In VWA, sensitive session storage information is encrypted. For the vote database we have a dual ACID-compliant database in which we store data in two different ways, Flat structure (MongoDB) and Relational database (MySQL).

7. What reporting options are available?

In the Voatz configuration described above, electronically returned ballots are printed at the election office and tallied along with in-person or mail ballots. Voter-printed ballots are also returned to the election office for tallying.

Election officials can use the Voatz Admin Portal to review summary use data for the election, including:

- Metrics of contests
- Eligible voters
- Signed up voters
- Incomplete voting statistics
- Contest completion statistics
- Ballots submitted
- Oath Affidavits received
- Oath Affidavits approved
- Votes by device
- Votes by jurisdiction
- Cybersecurity reports
- UAT and L&A Testing reports

Additional reports can be customized based on the needs of the election office.

Ballot Quality Control Process

Following Voatz internal quality control, the ballot acceptance testing occurs in the following phases:

1. Onscreen ballot comparison using the Voatz Administrative Portal (VAP).

Authorized State of Alabama officials and county reviewers login to VAP from using two-factor authentication from whitelisted IP addresses. Once in VAP, reviewers can view each ballot PDF to the electronic ballot configuration and individually accept or reject each ballot style. The manage ballot screen supports quality control verification of:

- Contest Name and description to match the PDF ballot
- Order of contests
- Vote Rule/Selections of contest (e.g. "vote for one", "vote for sixteen" including ranked choice)
- Order of candidates
- Spelling of candidates name, including transliteration in all covered languages
- Write-in options



Ballot style review screen in Voatz Administration Portal


2. Customer Acceptance Test

During this phase, the electronic ballots can be viewed interactively within the voting portal in a Staging environment. (See Voter Interface Portal, Ballot Contest displays section for additional detail.) During this phase, Voatz will provide a staging environment where the VWA voting portal can be used with test voters and thoroughly and test outline.

3. Logic & Accuracy Testing of UOCAVA & Accessible Absentee Ballots

Voatz provides support to the State of Municipal customer to perform their L&A Testing of accessible absentee ballots with the election management system for adjudication and tabulation.

SAMPLE LOGIC AND ACCURACY TEST CHECKLIST



Maricopa County Name: SAMPLE Election Date: _____

L&A Date: _____

Logic & Accuracy Test Software Versions

| Component | Expected Version | Installed Version | Notes |
|-----------|------------------|-------------------|-------|
| ADP | SAMPLE | SAMPLE | |
| DCD | SAMPLE | SAMPLE | |

Logic & Accuracy Test Procedure

1. Print or obtain blank _____ ballots for _____ unique ballot styles. ☐
2. Hand mark test deck ballots – Test deck ballots are marked using the pre-determined marking pattern covering every target (not on every contest). ☐
3. Power on the Tabulator and log in. ☐
4. Open Pkts for the Logic & Accuracy election. ☐
5. Generate the zero-vote report. ☐
6. Verify Election name, location, VoteStation ID, and that contest choices are zero votes (Reports are in English). ☐
7. Sign and date the Logic and Accuracy zero-vote report. ☐
8. Prepare the test deck ballot batch. ☐
9. Scan test deck ballots. ☐
10. Verify overvoted ballot handling (reject return ballot). Press Reject (red X) upon warning to reject an Overvoted ballot. (VM show in report in "No. of rejected ballots" and "rejected ballots" but not "counted ballots"). ☐
11. Verify overvoted ballot handling (accept). Reject all of Overvoted ballot. Press Accept (Green) to allow Overvoted ballot. (VM show in both "Yes" and "counted ballots", but overvoted contest will not count.) ☐
12. Pause election (to simulate overnight) which generates the Logic and Accuracy vote file's report for Day 1. ☐
13. Verify the expected ballot batches scanned on Day 1 (perchance Vote Totals). ☐
14. Sign and date the Day 1 report. ☐
15. Simulate Day 2 by opening (unpending) the election on Tabulator. ☐
16. Photograph or scan to send report to Voatz for integration on Unofficial total. ☐
17. Remove L&A ballots from ballot box beneath Tabulator and store with vote reports in a secure place. ☐
18. If L&A was successful, then use Pkts to "zero out" vote totals in preparation for the election. ☐
19. NOTE – Election day process is similar, however, you will also remove media (SD cards) from tabulators. ☐

Sample L&A Testing Checklist

Attachment L-2: Voter Guides

Attachment L-2-1: VWA (Voatz Web App) with Electronic Return Voter Guide

Please see the following pages that include the Voatz Web App (VWA) Voter Guide (with Electronic Return).



Voatz Web App Sample Instructions for Electronic Return Flow

Technical questions? Contact us at:
support@voatzsupport.zendesk.com

To expedite the process, please include:

1. Your name
2. The email address you used to sign up for this election
3. The make and model of your mobile

1

Utah Accessible Voting Demo
August 7-6, 2023

Accessible Absentee Ballot Portal

You will need to complete the following steps in order to vote:

- 1 Enter your email and date of birth as it appears on your voter registration.
- 2 Review the 11-character Voting Code sent to your email and enter it into the designated field, along with your date of birth.
- 3 By the state law, UOCAVA Voters Using overseas may choose to submit their ballot electronically. If so, Steps 5 and 6 are unnecessary. All other voters will continue to print their ballot.
- 4 You can mark the ballot online or continue to print a blank ballot.
- 5 Download and print your Printing Kit that contains your marked ballot and 3 envelopes.
- 6 Follow the instructions provided to prepare and mail your ballot. Be sure to sign the affidavit on the inner envelope.

All papers from the Printing Kit will need to be printed on 8.5" x 11" sized paper. A detailed user guide is available at the link below.

GET STARTED

2

Utah Accessible Voting Demo
August 7-6, 2023

Voter Verification

Please enter your information exactly as it appears on your passport. All fields are required.

First Name: [text input] Last Name: [text input]

Date of Birth: [Month] [Day] [Year]

Email: [text input]

CAPTCHA: [image]

VERIFY

Enter your information exactly as it appears on your voter registration. Be sure to complete all required fields and review it to make sure it is accurate and error-free. Complete the CAPTCHA, click **VERIFY** to continue to the next step.

3

Utah Accessible Voting Demo
August 7-6, 2023

ID Verification

Attach an image of your ID. JPEG, PNG, and GIF formats are allowed. Maximum file size is 4 MB.

Choose File...

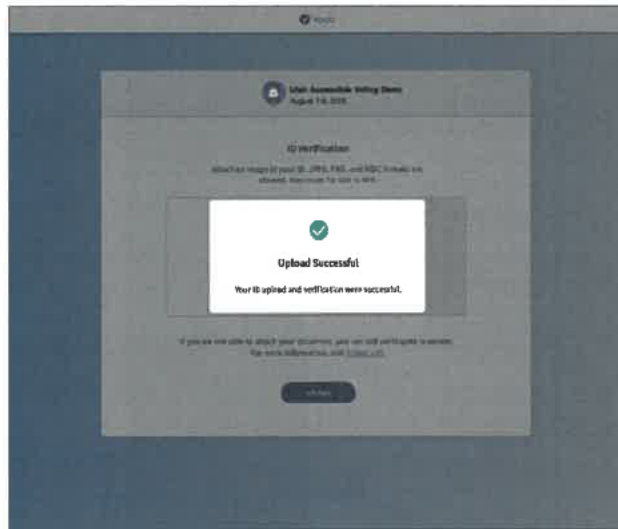
or drag your file here

If you are unable to attach your document, you can still participate in person. For more information, click [here](#).

PHOTO

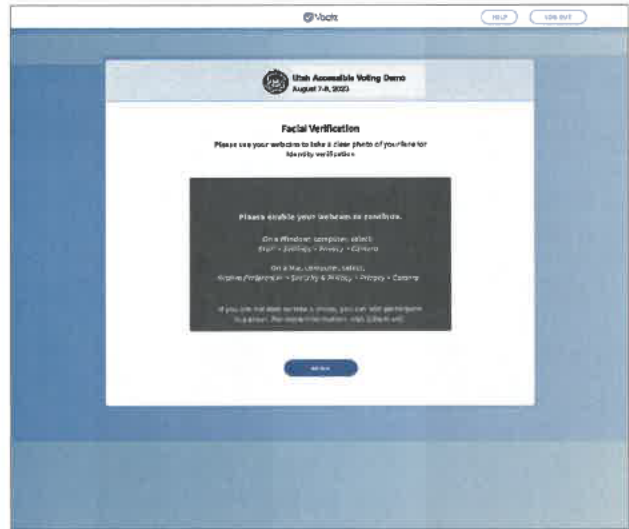
Upload a photo of your passport: Use the "CHOOSE FILE..." button or drag-and-drop a file into the gray area on the screen.

4



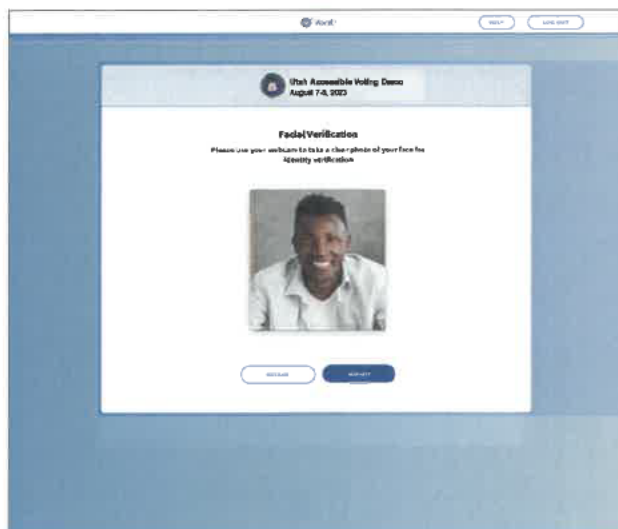
Upon successful upload and verification, you will automatically proceed to the facial verification stage.

5



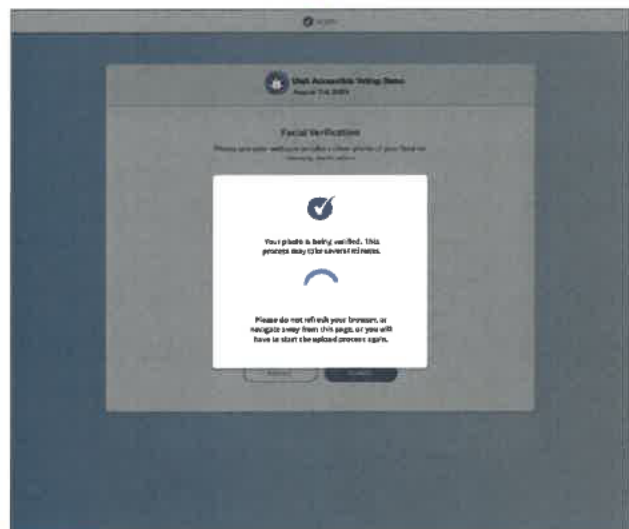
Make sure your device's camera is enabled, you may have to grant permission at the top of the browser screen or in your device settings. Review the photo tips and click [BEGIN](#) to take your photo.

6



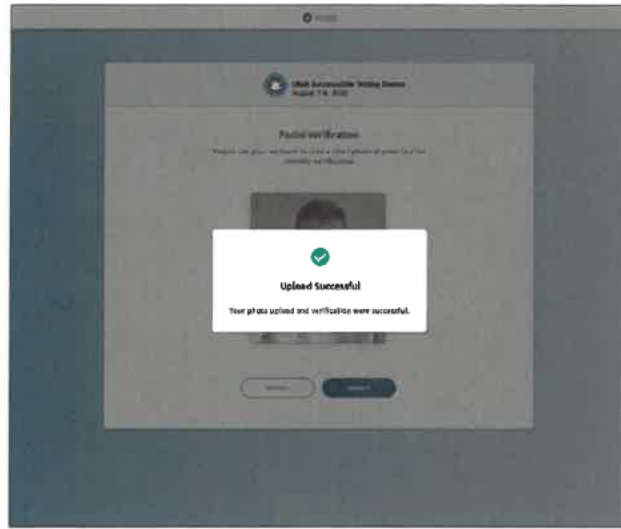
If you are satisfied with the photo, click [SUBMIT](#) to proceed, or click [RETAKE](#) to try again.

7



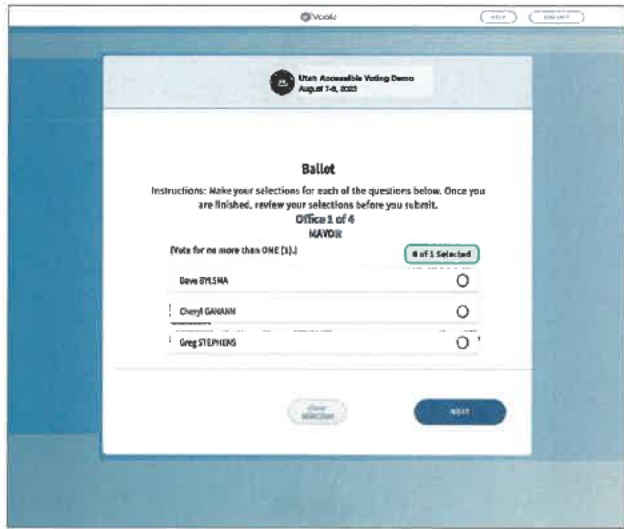
The system will confirm your identity; this process again may take several minutes. Do not refresh your browser or navigate away from the page, or you will have to start the upload process again.

8



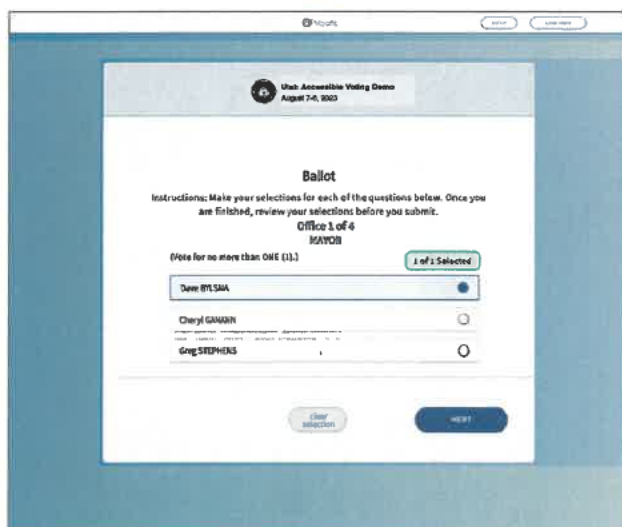
Upon successful upload and verification, you will automatically proceed to your ballot.

9



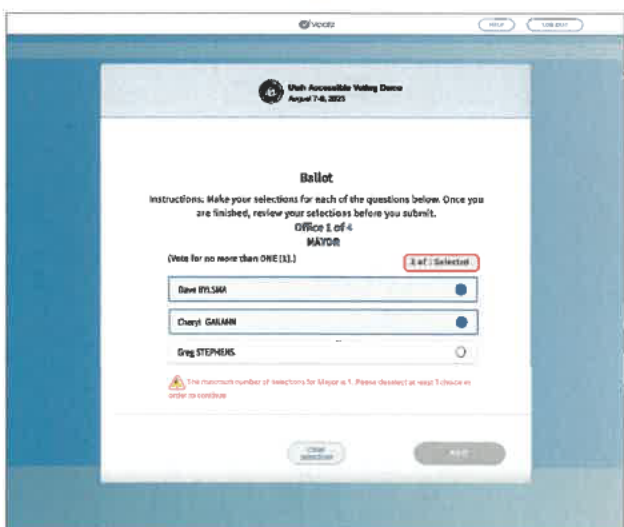
Make your selection for each question. Be sure to review your selections before you proceed.

10



When you are finished, click **NEXT**.

11



Overvotes are alerted and the **NEXT** button is disabled.

12

V Voatz

HELP
SIGN OUT

Review Ballot

Please review your selections below.

To modify a selection, click "change" to return to the ballot.

To confirm your selections, click the "Submit" button to cast your ballot.

MAYOR (Vote for no more than ONE (1))

Dave BILSBA

Peggy COOK

[change](#)

CITY COUNCIL (Vote for no more than ONE (1))

Paul GRIFFIN

Mike BURGER

[change](#)

CITY COMMISSIONER (Vote for no more than TWO (2))

Paul GRIFFIN

Mike BURGER

[change](#)

SCHOOL BOARD - DISTRICT 1 (Vote for no more than ONE (1))

Dorene WALKOTSH

[change](#)

SUBMIT

[illegible][illegible]

The screenshot shows a web browser window with a blue header bar. In the top left corner of the browser, there is a circular icon with the number 16. The browser's address bar shows a URL starting with 'vot'. The main content area has a blue background with a white rectangular box in the center. Inside this box, the text reads: 'Your ballot has been submitted.' followed by instructions to click a box to view a confirmation ID. Below this, a box displays the ID '1A J7 K 2 W 1 G B K 2'. At the bottom of the box are two buttons: 'PRINT PAGE' and 'END SESSION'.

16

1A J7 K 2 W 1 G B K 2

PRINT PAGE

END SESSION

Your ballot has been submitted! A confirmation code is available on this screen, and a confirmation receipt will be emailed to you as well. You may want to print this page for your records.

Attachment L-2-2: VMA (Voatz Mobile App) with Electronic Return Voter Guide

Please see the following pages that include the Voatz Mobile App (VMA) Voter Guide (with Electronic Return).



Voatz Mobile App

Sample Instructions for Electronic Return Flow

Technical questions? Contact us at:
support@voatzsupport.zendesk.com

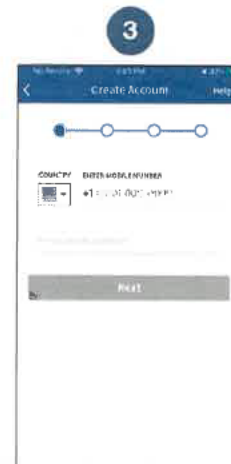
To expedite the process, please include your name, jurisdiction/county, and the make and model of your mobile phone.



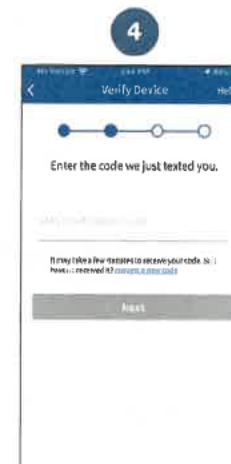
Download the Voatz app to your phone. (iOS 13.0+ or Android 8.0+ is required.)



Tap **Sign Up**. Make sure you have a strong and secure internet connection.



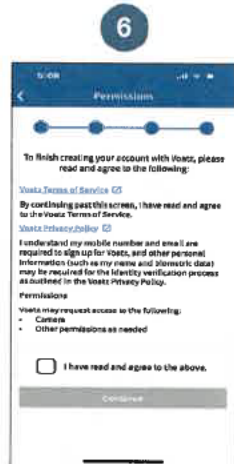
Enter the mobile number and email address that you used to register for this election



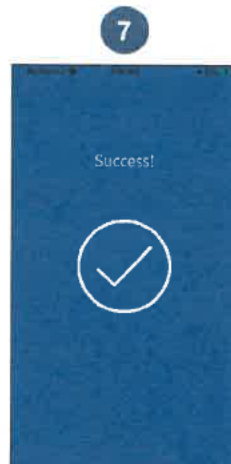
Verify your mobile number by entering the SMS code you receive.



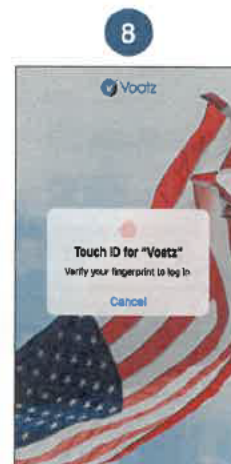
Follow the instructions on your device to create a Voatz PIN that you'll remember (Android phones require 12 digits). It cannot contain 3 or more sequential (678) or repeating (333) numbers.



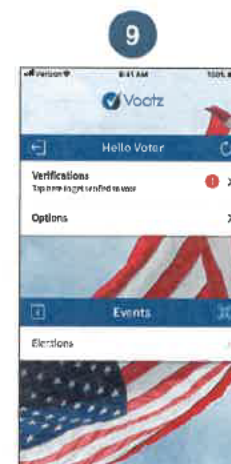
Scroll through the Terms and tap the box for "I have read and agreed to the above". Then tap **Continue**.



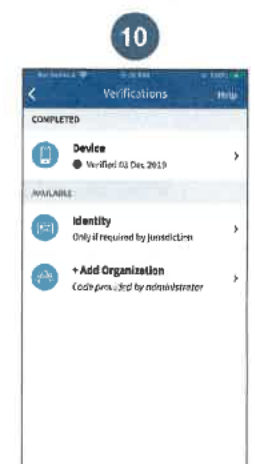
You have successfully signed up for Voatz!



Log in with your phone's Touch ID, Face ID, or the Voatz PIN you just created.

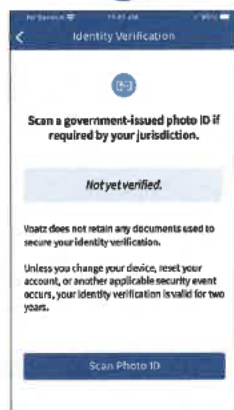


Tap **Verifications** to begin the verification process.



Have your photo ID ready and tap **Identity**.

11



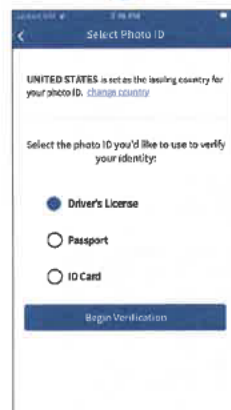
Review the description and tap **Scan Photo ID**.

12



Tap **Continue** to proceed. Please note that Military IDs are not permitted

13



Choose the type of government-issued ID you'd like to scan and tap **Begin Verification**.

14



Follow the instructions on the screen to complete the process. Make sure your scan is on an opaque surface.

15



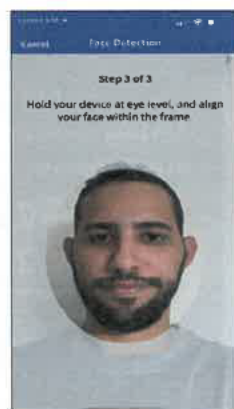
Depending on the type of ID, you may need take a photo of the back or enter additional information.

16



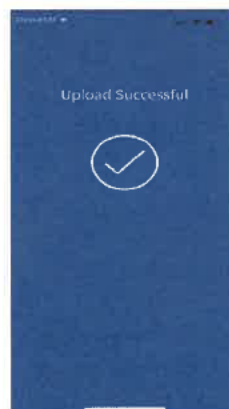
Next, tap **Continue** to take a photo of yourself.

17



Follow the instructions on the screen to complete the process.

18



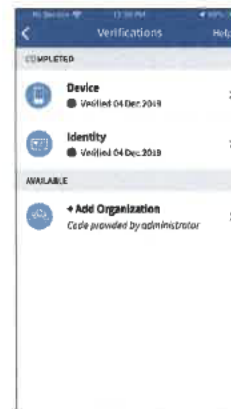
You have successfully uploaded your information.

19



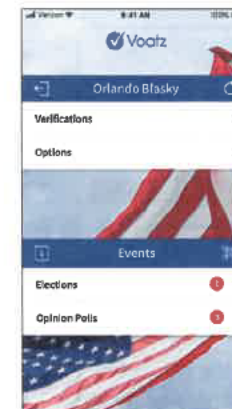
You'll receive confirmation that the verification process has begun. The process usually takes a few minutes, but may take 1-2 business days.

20



You're now verified! Tap the back arrow in the upper left to return to the home screen.

21



Once the election is open, tap the "refresh" icon to the right of your name. Then tap **Elections** to get started.

Attachment L-2-3: VWA (Voatz Web App) with Mark, Print and Mail Ballot Voter Guide

Please see the following pages that include the Voatz Mobile App (VMA) Voter Guide (with Mark, Print and Mail Ballot).

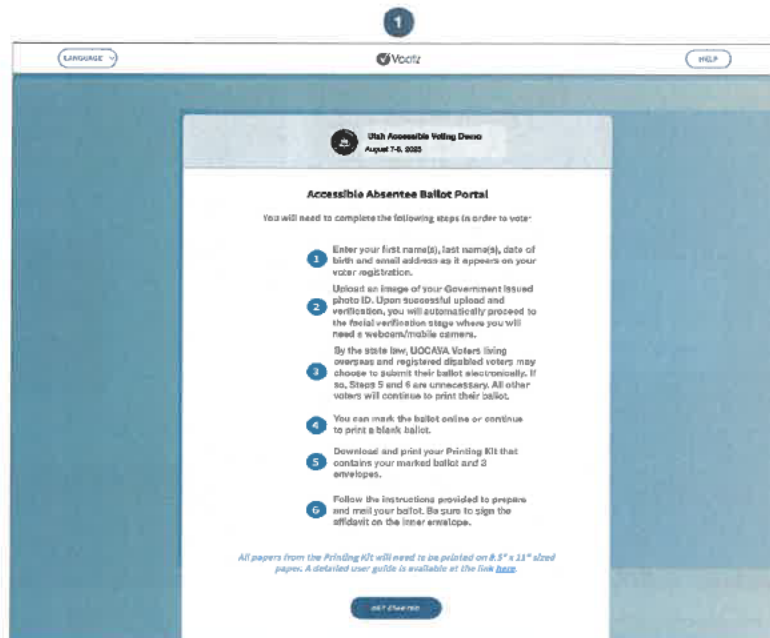


Voatz Web App Sample Instructions for Print Ballot Flow

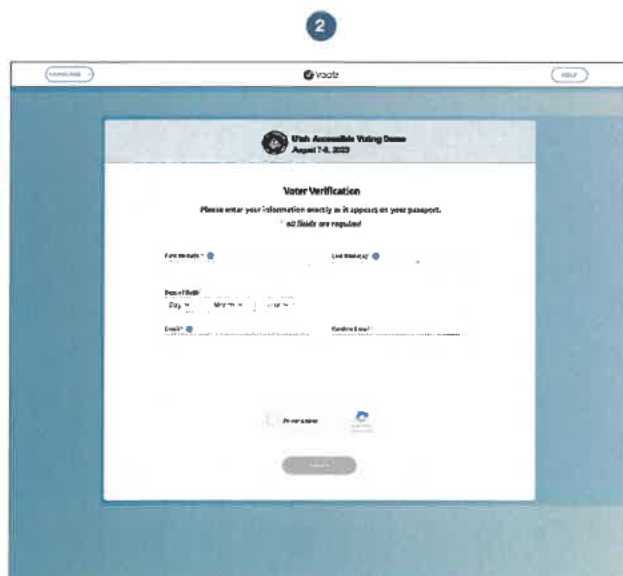
Technical questions? Contact us at
support@voatzsupport.zendesk.com

To expedite the process, please
include:

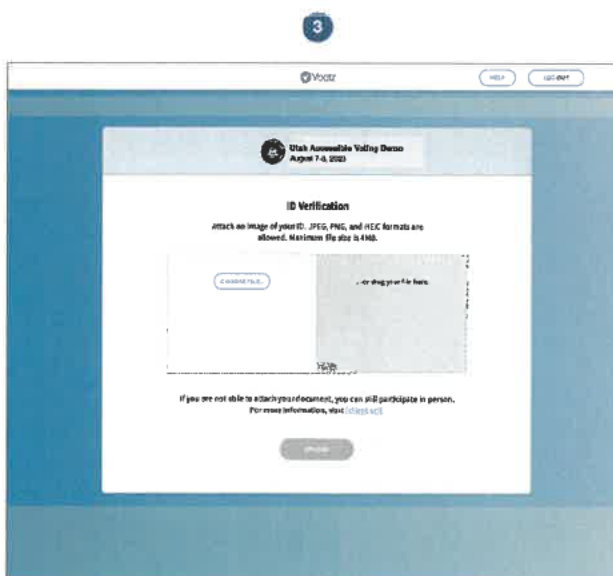
1. Your name
2. The email address you used to sign
up for this election
3. The make and model of your mobile



Navigate to the Voatz Web App in your web browser and have a
photo of your photo ID ready to upload.

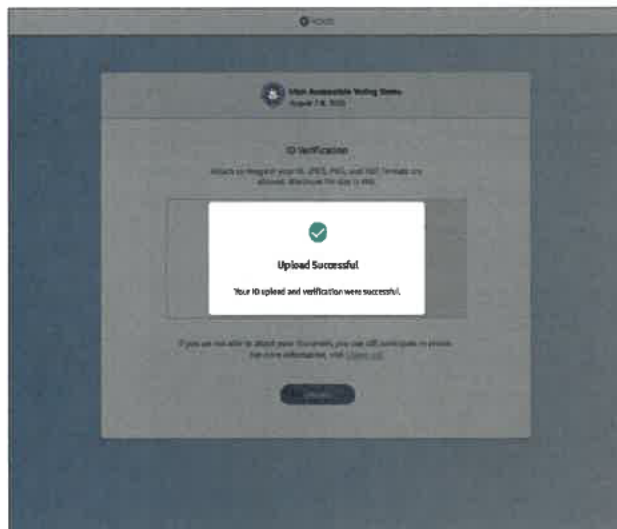


Enter your information exactly as it appears on your voter
registration. Be sure to complete all required fields and review it
to make sure it is accurate and error-free. Complete the
CAPTCHA, click **VERIFY** to continue to the next step.



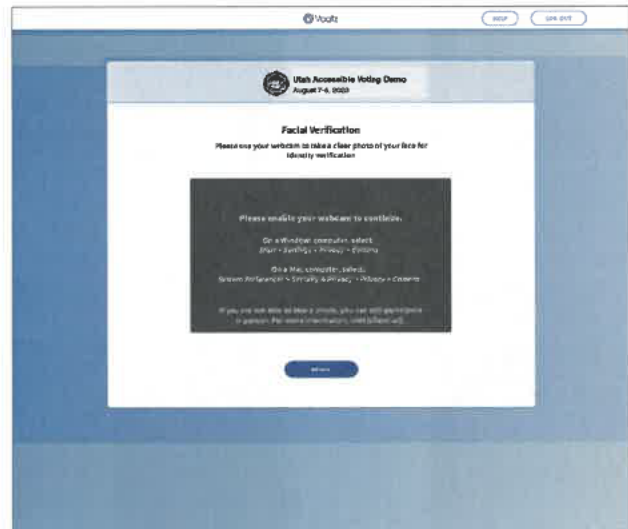
Upload a photo of your Government issued ID. Use the ***CHOOSE
FILE...** button or drag-and-drop a file into the gray area on the screen.

4



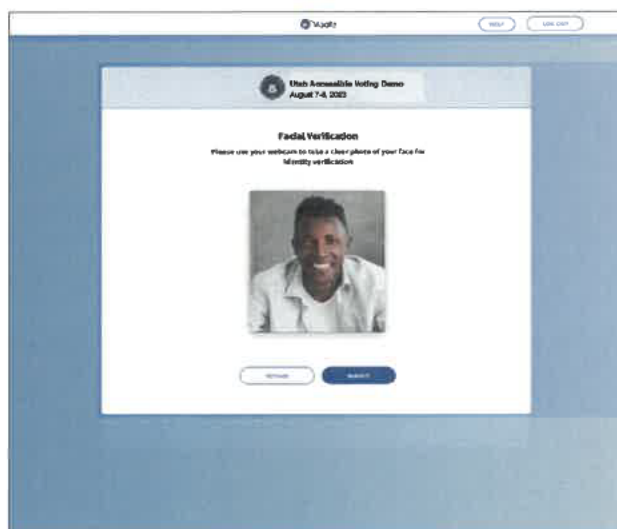
Upon successful upload and verification, you will automatically proceed to the facial verification stage.

5



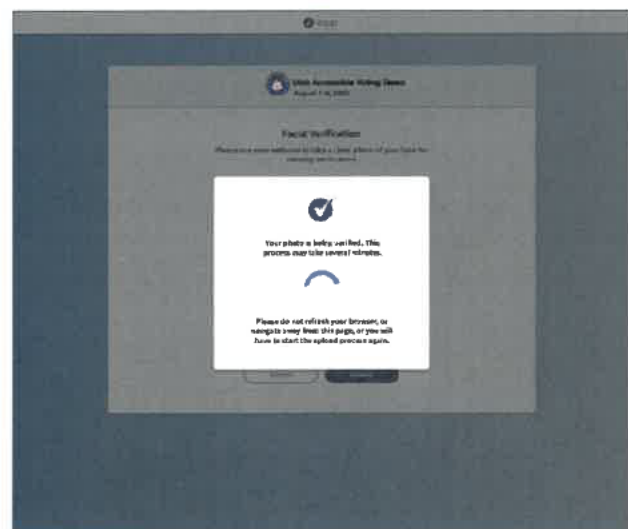
Make sure your device's camera is enabled, you may have to grant permission at the top of the browser screen or in your device settings. Review the photo tips and click [BEGIN](#) to take your photo.

6

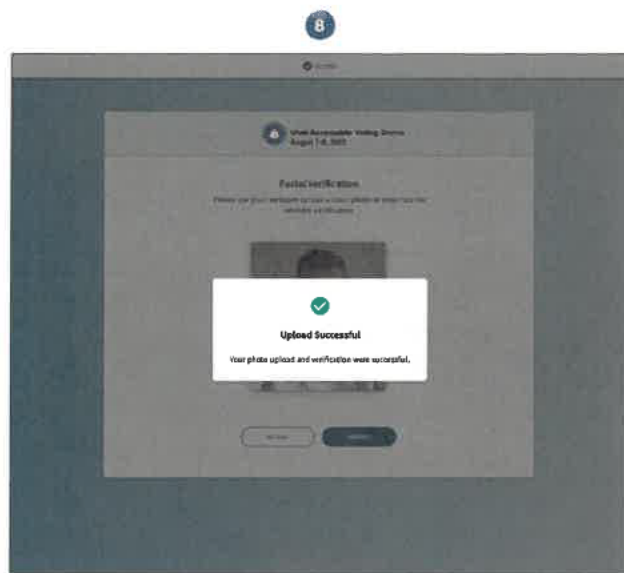


If you are satisfied with the photo, click [SUBMIT](#) to proceed, or click [RETAKE](#) to try again.

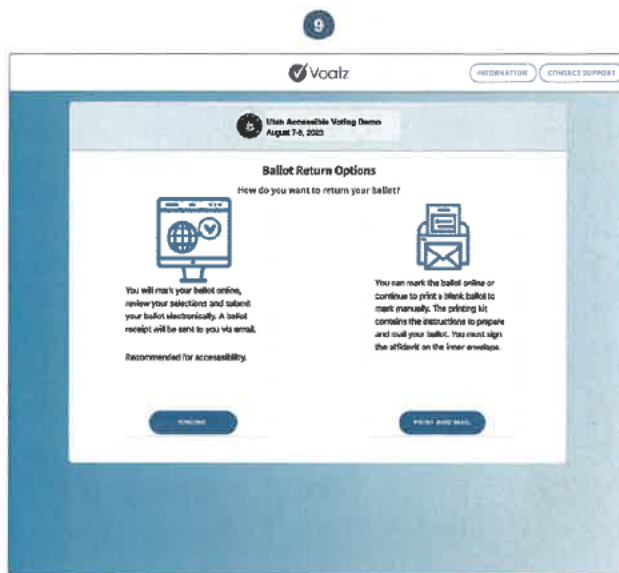
7



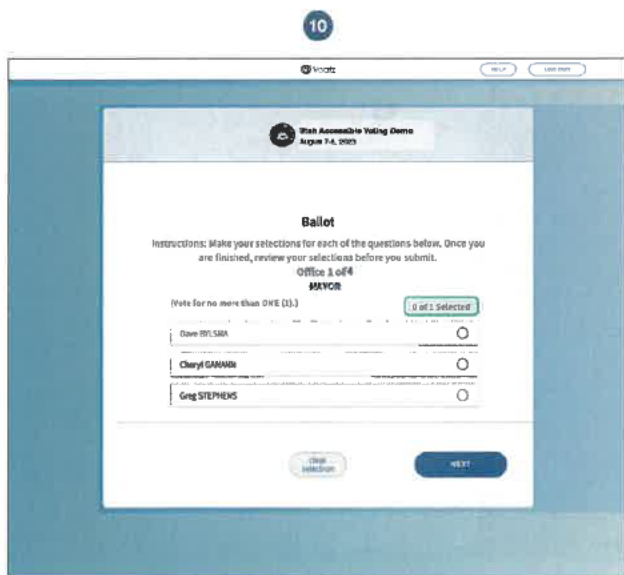
The system will confirm your identity; this process again may take several minutes. Do not refresh your browser or navigate away from the page, or you will have to start the upload process again.



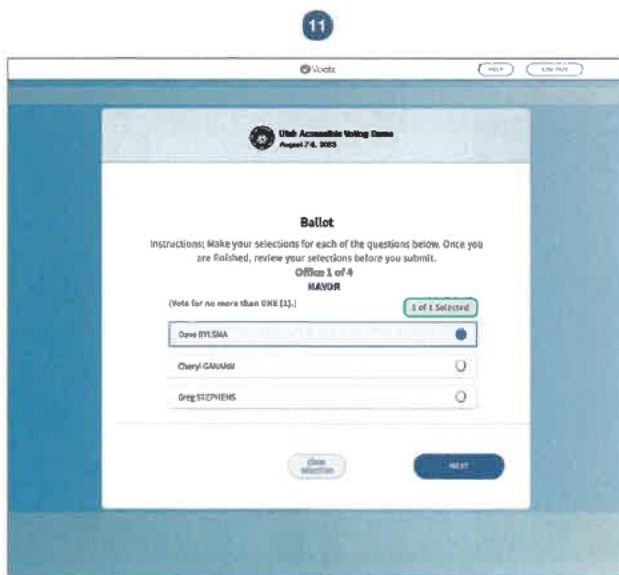
Upon successful upload and verification, you will automatically proceed to your ballot.



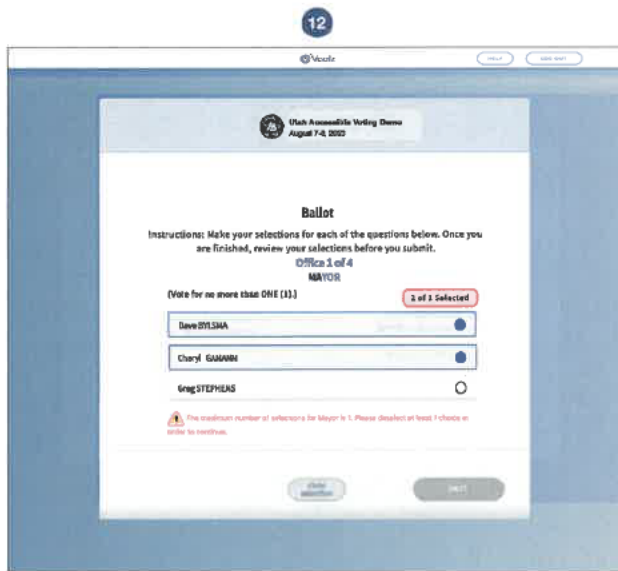
By the state law, UOCAVA Voters living overseas may choose to submit their ballot electronically. Otherwise, you can choose to mark, print and mail your ballot.



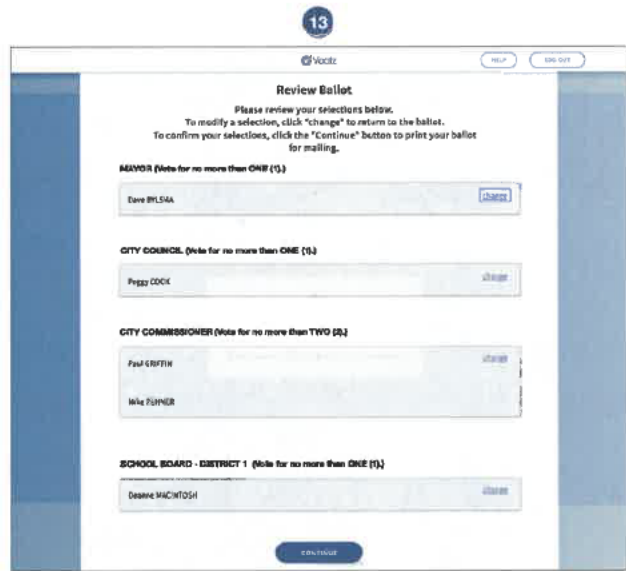
Make your selection for each question. Be sure to review your selections before you proceed.



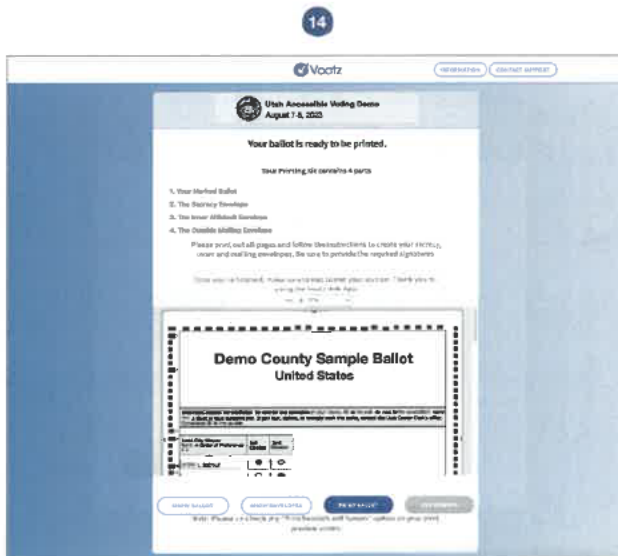
When you are finished, click NEXT.



Overvotes are alerted and the **NEXT** button is disabled.



Your selections for each contest will be visible on the next screen. You can choose to return to the ballot if you need to make changes, or proceed.



Print your ballot by tapping the **PRINT BALLOT** button. Make sure to select **NO MARGINS** on the printing model.



Tap the **SHOW ENVELOPES** button and print your envelopes by tapping the **PRINT ENVELOPES** button. Make sure to select **NO MARGINS** on the printing model.

Voatz

INFORMED CON. CONTACT SUPPORT

25

U77th Accessible Voting Demo
August 14, 2020

Your session has ended.

[Return to the poll page.](#)

The end session screen will confirm that your session has been terminated.

12. Attachment M: About Voatz

Voatz (*pronounced “Votes”*) is an award-winning election management platform, providing cutting-edge solutions to the needs of each election phase: Pre-Election, Election Day, and Post-Election. Voatz has created the first truly accessible online and mobile voting platform that enables eligible citizens to vote – regardless of their circumstances – from virtually anywhere in the world. Voters use their smartphone to authenticate themselves, receive, mark, and submit their secret ballot using an immutable blockchain infrastructure. Election managers easily consolidate remote results and in-person results and take advantage of Voatz’ unique audit features to facilitate independent auditing of the entire electoral process. To date, over 2.3 million voters have used the Voatz platform in more than 130 elections for national, state and local governments, unions, and major political parties, as well as national referendums.

Voatz originated when the founders won the ‘Hack to the Future’ hackathon at the 2014 South by Southwest conference in Austin, Texas. The founders were intrigued by the possibilities of combining the latest security features in modern smartphones, with emerging techniques capable of real-time identity verification, with biometrics capable of assuring the registered voter is present, all with the irrefutability offered by blockchain and the concepts of a distributed ledger. Their winning prototype – a mobile application designed to assure the privacy of the voter, protect the integrity of the election, and reduce the chances of voter coercion – became the foundation for the current Voatz platform.

Following its first private election in 2016, Voatz entered the public elections space in 2018, partnering with the state of West Virginia to record the first mobile vote in United States Federal Election history. In 2019, Voatz ran the first mobile vote in United States Municipal Election history with the City & County of Denver in Colorado. And in 2020, Voatz provided the first smartphone and blockchain based election system to be used for voting in a United States Presidential Election. The Voatz platform has been used in 32 counties across 5 United States states for public elections. In parallel to these public elections, Voatz has run political party conventions, elections, and committee votes in 7 different states for both major political parties in the U.S.

Milestones

- In 2018, Voatz ran the first accessible mobile vote in U.S. Federal Election history using identity access management (IAM) technology with the state of West Virginia.
- In 2019, Voatz ran the first accessible mobile vote in U.S. Municipal Election history with the City of Denver, Colorado.
- In 2020, Voatz provided the first mobile and accessible blockchain-based election system to be used for voting in the U.S. Presidential Elections with Utah County, Utah.
- In 2020, Voatz was selected to facilitate Venezuela’s Consulta Popular, a national referendum organized by the members of Venezuelan opposition and the Asamblea Nacional (“National Assembly”) de Venezuela. Voatz was engaged by the organizers to

create a digital voting platform for remote and in person citizen consultation. The voting platform was accessible through web browsers, mobile applications, and tablet applications as well as compatible with third party services (such as a Telegram bot application). During the referendum, over 1.7 million votes were processed via the Voatz platform.

- In 2022, Voatz was selected by 15 cities in Canada for the 2022 Ontario Municipal Elections wherein nearly 400,000 voters were eligible to utilize the Voatz platform to return their accessible ballots electronically.

Key Differentiators

Since the company's inception, Voatz has always emphasized cybersecurity and extreme testing. The Voatz founding team has extensive background in cybersecurity and the Company has emphasized those roots working with government agencies (such as the Department of Homeland Security which performed an infrastructure audit) and independent security partners (such as Digital Boundary Group, Consult Hyperion, LedgerOps and Synack) to conduct comprehensive security testing, penetration testing, and provide feedback which Voatz incorporates into our platform.

We also note that Voatz became the first RABDMR (Remote Accessible Ballot Delivery, Marking and Return) system in the United States to successfully complete comprehensive VSTL (Voting Systems Test Laboratory) testing with ProV&V in 2020, which determined that the Voatz system meets the applicable **subset** of requirements set forth for voting systems in the U.S. Election Assistance Commission (EAC) 2015 Voluntary Voting System Guidelines (VVSG), Version 1.1 for Usability, Accessibility, Functionality, Security and Accuracy. To our knowledge, Voatz is the only online based solution to conduct such a comprehensive review within the United States by a federally accredited testing lab.

Many of the elections conducted by Voatz have included public, citizen-led post-election audits, where, for the first time in US election history, anyone can participate in the post-election audit process. Independent third parties such as The National Cybersecurity Center have been responsible for managing the audit process, which involves a comparison between all ballot receipts, paper ballots, and the blockchain data to assure that the results are accurate, represent the will of the voters, and that the election holds integrity.

Members of the Voatz team are active participants in the development of standards and guidelines for Accessible Ballot Delivery & Return and have made draft submissions to the EAC (in partnership with the Global Blockchain Association for inclusion of a draft supplement to VVSG 2.x.). Voatz also participates in similar regulatory efforts in Canada with the Online Voting Standards Project (supported by the CIO Strategy Council - a national non-profit accredited by the Standards Council of Canada to develop National Standards of Canada):

Awards

Voatz is the winner of several technical and civic innovation awards including the MassChallenge 2017 Gold Award Winner, Microsoft Civic Innovation Award 2017, Election Center's Democracy Award (Denver County) 2019, Innovative Entrepreneurship in Blockchain Award (Public Sector Services) 2019, and was a finalist at the GSMA Mobile World Congress 2020 Awards for Best Mobile Innovation for Accessibility and Inclusion. Voatz was recently recognized as an Honoree at the 2021 Webby Awards in the Apps and Software: Public Service & Activism Category, a Finalist and Honoree at the Fast Company World Changing Ideas Awards 2021 in three categories and a Bronze Winner at the 2022 Anthem Awards. In September 2023, Voatz was recognized as a **Trusted Blockchain Solution for Elections** after undergoing a rigorous assessment by the Government Blockchain Association using their Blockchain Maturity Model.



Voatz, Inc.
50 Milk St Fl 16
Boston MA 02109 USA
Phone: +16176696366
Email: ns@voatz.com

Proposed solution



Voatz Web App
(print blank, mark/print/mail)



Voatz Admin Portal



Voatz Mobile Apps
(electronic mark/return)



Voatz Audit Portal



Military Grade Security

Cutting-Edge data encryption and voting security standards in use to ensure security, privacy and auditability



Multiple Voting Channels

Voatz conveniently supports all ballot and voting formats, whether electronic return or print and mail returns.



Increased Accessibility

Increase participation with multiple methods of communication and voting, and with advanced accessibility features



Save Time

Optimized mobile/tablet voting experience; administrators benefit by not having to hand duplicate ballots



Audit Trail

Conduct multiple, rigorous post-election audits with two audit trails — one for the voter, one for the Election Administrators