



# Secure Select Use Procedures and Technical Specification

VERSION 2.0

PREPARED FOR CALIFORNIA SECRETARY OF STATE

# Table of Contents

---

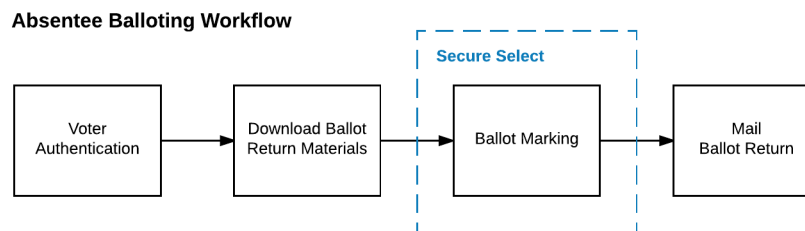
<b>Table of Contents .....</b>	<b>2</b>
<b>Secure Select Use Procedures .....</b>	<b>3</b>
<b>1. Introduction .....</b>	<b>3</b>
1.1. Terms and Definitions .....	4
1.2. System description and components .....	4
<b>2. Ballot Definition .....</b>	<b>7</b>
2.1. Paper and printing specifications .....	7
2.2. Printed Selection Specification .....	7
2.3. Printed Barcode Specification .....	7
<b>3. Election Set-up and Definition .....</b>	<b>7</b>
3.1. Programming and configuration of election management system/software .....	7
3.2. Programming and configuration of vote recording/tabulation devices .....	8
3.3. System diagnostic testing procedures .....	8
3.4. Logic and accuracy testing .....	9
<b>4. System Installation and Configuration .....</b>	<b>10</b>
4.1. Hardware requirements and specifications .....	10
4.2. Hardware and network set-up and configuration .....	10
4.3. Software installation and configuration .....	10
4.4. Acceptance Testing .....	10
4.5. Software and firmware upgrades .....	13
<b>5. Polling Place Procedures .....</b>	<b>13</b>
<b>6. Absentee/Mail Ballot Procedures (Central Tabulation) .....</b>	<b>13</b>
<b>7. Official Canvass and Post-Election Procedures .....</b>	<b>13</b>
7.1. Post-election logic and accuracy testing .....	13
7.2. Back-up and Retention of election material .....	13
<b>8. Security .....</b>	<b>14</b>
8.1. Physical security of system and components .....	14
8.2. User-level security .....	14
8.3. Procedures for verifying, checking, and installing essential updates and changes .....	14
8.4. Ballot Audit trail .....	15
<b>Appendix A: WCAG 2.0 Conformance .....</b>	<b>16</b>

<b>Appendix B: Ballot Data Specification .....</b>	<b>18</b>
<b>Appendix C: QR Code Specification .....</b>	<b>20</b>
<b>Appendix D: Secure Select Technical Details .....</b>	<b>21</b>
<b>1. Architecture and Codebase.....</b>	<b>21</b>
1.1. Secure Hosting.....	22
1.2. Scalable Architecture.....	23
1.3. Flexible Architecture .....	24
1.4. Application Review and Certification .....	24
<b>2. Source Code Verification .....</b>	<b>25</b>
2.1. Storing the Secure Select Hash Code.....	25
2.2. How to use Hash Code Verification .....	25
<b>Appendix E: Acceptance Testing Tables .....</b>	<b>26</b>

# Secure Select Use Procedures

## 1. Introduction

Absentee balloting is composed of four main components: 1) Voter authentication, 2) Obtaining ballot return materials, 3) Marking ballot selections, and 4) Returning marked ballot to the local elections office. Secure Select is a cloud based application focused solely on *ballot marking*. Separating ballot marking as a micro service introduces flexibility to counties and several benefits to voters.



Secure Select was designed from the ground up to meet the highest levels of accessibility. It satisfies all WCAG 2.0 guidelines including screen reader compatibility, full keyboard access, and color, contrast and font sizing requirements (*see Appendix A for details*). Secure Select is compatible with macOS and Windows screen readers including, but not limited to, the following:

Operating System	Web Browser	Screen Reader
Windows 10	Internet Explorer 11, Edge 14	Narrator
Windows 10	Firefox	NVDA
macOS 10.12	Safari 10.1	VoiceOver

Per the California State Elections code for ballot marking, Secure Select does not require, nor allow interaction with a remote server during the ballot marking process. Once the Secure Select application is loaded from the cloud, no further connection to the server, or Internet is required.

### 1.1. Terms and Definitions

**Ballot Definition File** – A file containing all data needed to display a specific ballot style (headers, contest, measures, candidates, candidate order, etc.). Ballot Definition Files are stored on a remote server and are downloaded and parsed by Secure Select to ballot styles to voters.

**Box** – When used in the context of a ballot, represents any content on a ballot such as contests, measures or propositions (which are typically enclosed in a box).

**Option** – When used in the context of a ballot, represents any markable content on a ballot such as candidates, measure responses, or write-ins.

**Micro service** – An application or service with an isolated set of functionality meant to be used as part of a larger application or workflow.

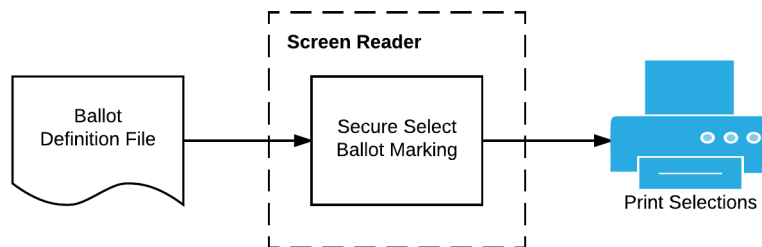
**URL** – A location on the internet accessible by typing it into a web browser

**QR Code** – A machine-readable code consisting of an array of black and white squares, typically used for storing information for reading by the camera on a smartphone.

### 1.2. System description and components

Secure Select is composed of three main components. A Ballot Definition File is created and passed into Secure Select. Secure Select parses the Ballot Definition File and presents a ballot style to the voter. The voter can optionally use a Screen Reader to navigate through the ballot. After marking their ballot and reviewing their selections, the voter can print their selections.

**Secure Select Components**



#### 1.2.1. Ballot Definition Files

Ballot Definition Files must meet the Ballot Data Specification defined in *Appendix E* and be hosted at a publicly accessible URL. Once the Ballot Definition File has been uploaded, it can be passed into Secure Select using the following format.

**Example URL:** `https://ss.liveballot.com?data=DEFINITION_URL&lang=LANG_CODE`

- **data** – An absolute url (including https://) to a Ballot Definition File
- **lang** – A language code specifying which language to display to the voter. Allowed language codes are en (English), es (Spanish), zh-hans (Simplified Chinese), and zh-hant (Traditional Chinese).

### Exercise: Understand how to pass a ballot definition file into Secure Select

**Step 1:** Secure Select has a built in data file that can be used to help understand the process. This file is *publicly accessible* by entering <https://ss.liveballot.com/app/assets/multilingual.json> in a web browser. Enter this URL into a browser to view the data file.

**Step 2:** Pass this URL into Secure Select using the data parameter. Use the example URL provided above and replace DEFINITION\_URL with "<https://ss.liveballot.com/app/assets/multilingual.json>". Replace LANG\_CODE with "en" to use English. The final URL will look like this:

<https://ss.liveballot.com?data=https://ss.liveballot.com/app/assets/multilingual.json&lang=en>

**Step 3:** Finally, replace "en" with "es" in url above to show a Spanish ballot. The final URL will look like this:

<https://ss.liveballot.com?data=https://ss.liveballot.com/app/assets/multilingual.json&lang=es>

#### 1.2.2. Ballot Marking (the Secure Select Application)

Secure Select is an HTML5 Single Page Application (SPA) that runs inside web browser. During page load, Secure Select downloads and stores everything it needs to run. After page load, the application logic is completely isolated to the browser window. The voter is taken through the following pages *without any server communication*:

**Instructions** - Clear instructions are presented to the voter detailing how to navigate Secure Select and what steps they will be taken through. The voter clicks **Continue** to progress to the **Ballot Marking** screen.

## Welcome to Secure Select






Your universal remote absentee balloting tool.


### About this Application

This application will allow you to mark and print your ballot selections for the **Demo Election**. You will complete the following steps:

1. Mark selections for each contest or question on your ballot
2. Review your ballot selections
3. Print your selections

### Keyboard Controls

- Use the up  and down  arrow keys to navigate through content.
- Use the  space bar to toggle a checkbox or click on a link.
- Use the plus  or minus  keys to adjust the font size.

Continue

**Step 1: Ballot Marking** – The voter can mark their ballot using their keyboard, mouse, or any assistive technology. Voters are prevented from over-voting contests with a clear warning.

### Step 1 of 3: Ballot Marking

Your full ballot is presented below. Click on a checkbox to mark your selection. Click any checked box a second time to remove that selection.

To enter a write in candidate on the ballot, first check the write in checkbox then click in the field provided next to the checkbox and enter the write in candidate's name.

National	
<b>US President</b> <small>Four Year Term Vote for One</small>	
<input checked="" type="checkbox"/>	<b>Waylon Dalton</b> <small>Democrat Justine Henderson for Vice President</small>
<input type="checkbox"/>	<b>Marcus Cruz</b> <small>Republican Thalia Cobb for Vice President</small>
<input type="checkbox"/>	<b>Write-In</b>

If a write-in candidate is selected, a text field is provided to enter a candidate name.

<input type="checkbox"/>	<b>Marcus Cruz</b> <small>Republican Thalia Cobb for Vice President</small>
<input checked="" type="checkbox"/>	<b>Write-In</b> <input type="text" value="Thomas Jefferson"/>

After marking selections, the voter clicks **Continue** to progress to the **Selection Review** screen.

**Step 2: Selection Review** – The voter is presented with a summary of their selections. They are notified if they are missing any selections for any contests. Clicking **Change** next to any contest will take the voter directly to that contest on the **Ballot Marking** screen. After reviewing selections, the voter clicks **Continue** to progress to the **Print Selections** screen.

### Step 2 of 3: Selection Review

Your ballot selections are shown below. You may change any selection by clicking the Change link next to your selections.

National		
<b>US President</b>	Waylon Dalton	<a href="#">Change</a>
Propositions		
<b>Proposition 101</b>	⚠ No Selections	<a href="#">Change</a>
<a href="#">Go Back</a>	<a href="#">Continue</a>	

**Step 3: Ballot Printing** – The voter prints their ballot which contains a QR code representing their selections. After the selections have been printed, the voter clicks **End Session** to progress to the **Complete** screen.

### Step 3 of 3: Print Selections

Click on the button below to print your ballot selections. After your selections finish printing, click End Session to close your session.

Print Selections

⚠ Warning! Closing your session will clear all your ballot selections. This cannot be undone.

Go Back End Session

**Complete** – The voter selections are cleared from memory and the voter is presented with a thank you message.

### Thank you for using Secure Select

Your session has been closed and your ballot selections have been cleared. Please follow the instructions provided in your return packet to send your selections to your local elections office.

## 2. Ballot Definition

### 2.1. Paper and printing specifications

The printed output from Secure Select is designed to print from a typical home computer on US Letter (8.5x11) paper.

### 2.2. Printed Selection Specification

The printed output from Secure Select includes the *options* marked by the voter for every *box* on the ballot. The printed output is intended to be a representation of the voter's selections, not of the entire ballot. If the voter did not mark any selections for a *box*, the text "No Selections" is included to clearly identify where no selections have been made.

### 2.3. Printed Barcode Specification

The printed output from Secure Select includes a QR Code representing the voter's selections. The QR Code does not include any voter information and can be scanned using any modern smartphone or 2d barcode reader. The QR Code is included to allow for integration with 3<sup>rd</sup> party solutions such as auto duplication software. The QR Code data specification can be found in *Appendix C*.

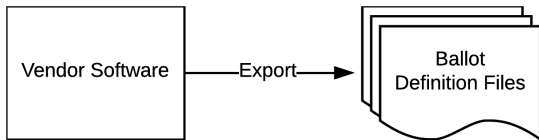
## 3. Election Set-up and Definition

### 3.1. Programming and configuration of election management system/software

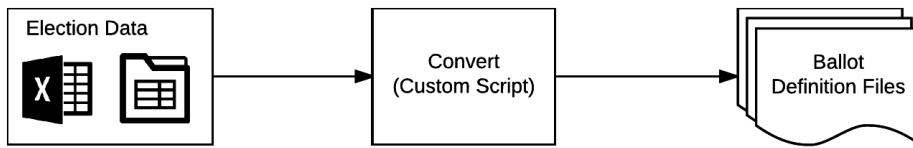
Secure Select is a *micro service* focused on accessible ballot marking. There is no election creation or ballot configuration in Secure Select. These processes happen outside of Secure Select in an Election Management System (EMS), a third party ballot building software, Excel, etc. Once election data has been prepared it should be exported or converted to

JSON files conforming to the Ballot Data Specification defined in *Appendix B*. The following list outlines three popular options for generating Ballot Definition Files:

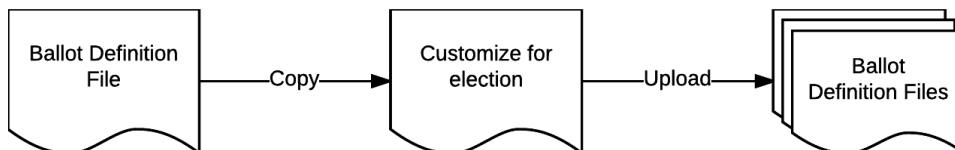
- 1) **Export data from existing software** – If your county has election data already loaded into another software, ask your vendor if they can export the data or generate a report to meet the Ballot Data Specification. For example, current and future solutions provided by Democracy Live include data exports that meets the Ballot Data Specification.



- 2) **Provide Excel or CSV files for conversion** – It is common for counties to use Excel files to organize election data before an election. These files typically include a structured way of associating contests, candidates, ballot styles, precincts, and even rotation. A developer (either internal IT, a contract developer, or a vendor) can write a script to convert Excel (or CSV) files into Ballot Definition Files. It is important for the county and the developer to agree on a template to ensure a streamlined process in future elections. Democracy Live technical support representatives can work with counties to create a custom script for ballot data file generation. Democracy Live can also work with internal IT staff to help setup a script to be used internally.



- 3) **Manually create ballot styles** – This approach does require knowledge of how to write JSON data. This method is great for small elections with limited number of ballot styles and content. The best way to use this method is to copy an existing Ballot Definition File and then modify the content. There are several online resources to help write and validate JSON data such as: <https://jsonformatter.org>



### 3.2. Programming and configuration of vote recording/tabulation devices

Secure Select is only a ballot marking solution and does not record or tabulate voter data.

### 3.3. System diagnostic testing procedures

Secure Select must be online at all times for voters to mark and print their ballot selections. Secure Select includes a ping URL which can be accessed at any time to verify the system is online. Accessing the URL will return a 200 response header and text if the application is available and working.

**Ping URL:** <https://ss.liveballot.com/ping>

The Secure Select application is hosted on two or more parallel servers at all times. Democracy Live monitors this endpoint on each server 24/7 to detect any service interruptions. If a server does not return a 200 response, it is flagged



as unhealthy and is decommissioned. A new Secure Select server is created and added to the load balancer ensuring there is always two healthy servers available. Additional information regarding Secure Select's server configuration is available in *Appendix D*.

### 3.4. Logic and accuracy testing

Ballot Definition Files will be generated for each ballot style in an election. Elections officials are encouraged to test each Ballot Definition File to verify Secure Select displays ballot content correctly.

#### 3.4.1. Pre-conditions for performance of tests

To conduct testing in Secure Select, the following steps must be followed:

1. Store Ballot Definition Files on server with a publicly accessible URL.
2. Create an Excel document with three columns: Name, URL, Status. If you work with a vendor to generate Ballot Definition Files, request a file in the following format:

Name	URL	Status
Style 1 – En	<a href="https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=en">https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=en</a>	
Style 1 – Es	<a href="https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=es">https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=es</a>	
Style 2 – En	<a href="https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=en">https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=en</a>	
Style 2 – Es	<a href="https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=es">https://ss.liveballot.com?data=https://definitionurl.com/style1.json&amp;lang=es</a>	

**URL Format:** [https://ss.liveballot.com?data=DEFINITION\\_URL&lang=LANG\\_CODE](https://ss.liveballot.com?data=DEFINITION_URL&lang=LANG_CODE)

#### 3.4.2. Accuracy Test procedures

For each URL defined in the file generated in 3.4.1, perform the following tasks:

1. Visit the URL in a web browser.
2. Verify the ballot content is correctly displayed.
3. If the ballot style is correct, type "Approved" in the Status column.
4. If the ballot style is incorrect, enter a reason for the error. If you are working with a vendor to generate Ballot Definition Files, the notes provided in the status column will help with error correction.

#### 3.4.3. Logic Test procedures

Load a Ballot Definition File from the file generated in 3.4.1 and test the following:

1. **Over-vote Protection** – Ensure voters are not able to over-vote for a contest.
2. **Correct Review Page** – Confirm selections and write-ins are correctly shown on the review page.
3. **No Selection Warning** – Confirm a warning is shown on the review page if no selections are made.
4. **Under Vote Protection** – Confirm an under-vote warning is shown if not all selections are made for a contest with more than one selection available.
5. **Print Selections** – Confirm selections and write-ins are correctly printed.
6. **QR Code** – Scan the QR code with a smart phone and confirm the selection data represents the printed output.

#### 3.4.4. Retention of Test materials

The paper ballots generated from this testing should be saved under the county's normal elections document saving protocols and requirements.

## 4. System Installation and Configuration

### 4.1. Hardware requirements and specifications

Secure Select is a cloud based solution. There is no software installation or configuration required. There are no hardware requirements to use Secure Select outside of what is required to run an internet browser.

### 4.2. Hardware and network set-up and configuration

Secure Select is delivered to voters over the internet using SSL encryption. Users must have an internet connection and a web browser capable of accessing a website using SSL encryption.

### 4.3. Software installation and configuration

Secure Select is a cloud based solution. There is no software installation required. Voters can use the default web browser that comes with their computer to access Secure Select.

#### 4.3.1. Custom Installations

Secure Select can be installed on any Linux, FreeBSD, or Windows servers. Democracy Live technical support representatives can assist IT administrators with custom installations upon request.

### 4.4. Acceptance Testing

Secure Select has a narrow scope of functionality limited to the accessible display, marking, and printing of ballot selections. The purpose of this design is to provide a modular application capable of integrating with new and existing software. As such, there are four key points of testing required for Secure Select:

1. **General Functionality** – Does the application allow voters to view, mark, and print their selections accurately?
2. **Screen Reader Accessibility** – Is the application fully functional by using a screen reader?
3. **Keyboard Accessibility** – Is the application fully functional by using only keyboard controls?
4. **Voter Privacy** – Does the application work without transmitting any voter data to a remote server?

The Acceptance Testing Tables in **Appendix E** can be printed to keep track of test items and their status.

#### 4.4.1. Testing General Functionality

The following steps can be taken to test the general functionality of Secure Select.

##### 4.4.1.1. Setup

1. Open a Secure Select URL from the file generated in 3.4.1. You may also use <https://ss.liveballot.com?data=demo> to load a demonstration election for testing purposes.

##### 4.4.1.2. Test Items

1. Read the on-screen instructions and click Continue.
2. Read the instructions at the top of the page.
3. Click on candidates to mark a selection. Click on a candidate again to deselect.
4. Click on the checkbox next to a candidate to verify it toggles selections as well.
5. Try to over-vote for a contest. Verify an over vote warning is displayed.

6. Verify a text field is presented to enter a candidate name when checking a write-in candidate. Fill in a write-in candidate.
7. Leave at least one contest without any selections (to be used later).
8. Click Continue
9. Confirm the selections on the Review Page are accurately displayed
10. Click change next to a selection. Verify it takes you to the specific contest on the Ballot Marking Page.
11. Change the selection. Verify there is a shortcut link to go back to the Review Page.
12. Go back to the Review Page and confirm changes have been made.
13. Confirm write-in values are accurately presented on the Review Page.
14. Click Continue to continue to the Print Selections page.
15. Click the Print Selection button. Confirm a print dialog is triggered.
16. Print the selections and confirm they are accurately printed.
17. Go back to Secure Select and click End Session.
18. Return to the testing URL, click continue, and verify your selections are no longer visible.

#### 4.4.2. Screen Reader Accessibility

For Screen Reader testing, verify all Test Items under *4.4.1 Testing General Functionality* are accessible using screen reader specific key commands (these are different than the instructions shown on the instructions page).

##### 4.4.2.1. Setup

1. Open a Secure Select URL from the file generated in 3.4.1. You may also use <https://ss.liveballot.com?data=demo> to load a demonstration election for testing purposes.
2. Turn on the screen reader using the commands below. When the screen reader is activated, it is important to focus only on what you hear from the screen reader. It can be helpful to close your eyes while testing to avoid being distracted by the screen reader's focus element moving on the page.
3. The web browser should have focus while using the screen reader. If the focus is changed outside of the web browser, use the mouse to click back into Secure Select. Refresh Secure Select to allow the screen reader to reinterpret the application.
4. Use the screen reader's specific keyboard commands (not the keyboard commands displayed on screen for sighted voters) to navigate the application
  - a. **macOS – Voice Over**
    - i. Press Command-F5 to start Voice Over
    - ii. Press Control-Option-Right Arrow and Control-Option-Left Arrow to navigate between content
    - iii. Press Control-Option-Spacebar to activate an option
  - b. **Windows – Narrator**
    - i. Press Windows Key + Enter to open windows narrator
    - ii. Press the Caps Lock Key + Space to turn on scan mode. Scan mode is an easy way to navigate through a page. Use I to move between items, H to move between headers, and press the Spacebar to activate an item. Hold down Shift + I and/or Shift + H to reverse the direction of the previous commands.
    - iii. Narrator will exit scan mode if the application changes. If the screen reader begins reading the letter of each key when pressed, press Caps Lock + Space again to re-enter scan mode.

- iv. When entering a write-in, Narrator will ask you to press Space to enter edit mode. When you are done entering text, you must press Caps Lock + Space again to go back to scan mode to continue.
- v. **Advanced Usage:** Holding down the Caps Lock Key, use the Up and Down arrow keys to change the reading mode. In a specific reading mode, hold the Caps Lock key and press the Left and Right arrows to navigate. Different reading modes are suitable for different scenarios and can be used in conjunction with Scan mode. For more information about reading using Narrator, visit this help article: <https://support.microsoft.com/en-us/help/22809>.

#### 4.4.2.2. Test Items

1. Verify the on-screen instructions on page one are not read by the screen reader.
2. Continue to Ballot Marking
3. Verify you can mark selections using screen reader's specific keyboard commands and unmark selections
4. Verify over vote warnings are read when attempting to over vote
5. Verify you can write in candidates
6. Verify selections are clearly read when navigating up and down the page

#### 4.4.3. Keyboard Accessibility

A large component of the WCAG 2.0 accessibility guidelines includes keyboard controls. Verify all Test Items under 4.4.1 *Testing General Functionality* are accessible using only your keyboard.

##### 4.4.3.1. Setup

1. Open a Secure Select URL from the file generated in 3.4.1. You may also use <https://ss.liveballot.com?data=demo> to load a demonstration election for testing purposes.
2. Disconnect your mouse or place out of reach to ensure the mouse is not used for any functionality during testing.

##### 4.4.3.2. Test Items

1. Verify the keyboard controls presented in the on-screen instructions operate as expected. Specifically, test the up, down, left, right arrow keys, the space bar, and the + and - keys.
2. Verify the text can be zoomed to 200% of the original size
3. Verify keyboard focus is clearly presented when moving around the screen (a visual indication should show you where you are at all times).

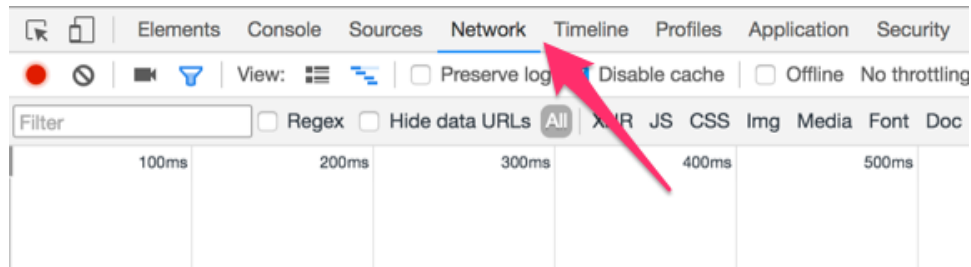
#### 4.4.4. Voter Privacy

Voter privacy is protected in Secure Select by eliminating all network communication with remote servers and by clearing voter selections at the end of their session. Once Secure Select has loaded, all actions the voter takes happen on their local machine.

##### 4.4.4.1. Setup

1. Open a Secure Select URL from the file generated in 3.4.1. You may also use <https://ss.liveballot.com?data=demo> to load a demonstration election for testing purposes.
2. Use developer tools to open the network inspector in your browser. The network inspector will show you all communication sent to local or remote servers in real time.
  - a. In Chrome: Open View > Developer > Developer Tools. Then click on the Network tab

- b. In Internet Explorer and Edge: Open Developer Tools and click on the Network Tab



3. If there is any network activity, click the clear button to clear it out
4. (Optional) Disconnect from the internet

#### 4.4.4.2. Test Items

1. With the Network tab open under Developer Tools, complete all items in section 4.4.1 *Testing General Functionality* above. After each action (selecting or deselecting a candidate, entering a write in, navigating between pages, and printing your selections) verify no network activity is shown.

### 4.5. Software and firmware upgrades

Democracy Live maintains application servers with regular security and software updates. Only approved updates to Secure Select will be deployed during an approved update window. The California Secretary of State can confirm no unapproved software updates have been deployed by verifying the application source code hash (see *Appendix D* for details).

## 5. Polling Place Procedures

Secure Select is not intended for polling place use.

## 6. Absentee/Mail Ballot Procedures (Central Tabulation)

The selections made by the voter using Secure Select are printed and submitted back to the County per the State and County requirements. The County will then duplicate or transcribe the voter's intent onto tabulatable ballots, per the Counties standard duplication procedures.

## 7. Official Canvass and Post-Election Procedures

### 7.1. Post-election logic and accuracy testing

It is recommended the County conduct a post-election test of Secure Select, showing ballot selections were printed as intended. County does this by printing a test set of ballots via Secure Select.

### 7.2. Back-up and Retention of election material

Ballots returned from Secure Select users should be retained per county document retention requirements.

## 8. Security

### 8.1. Physical security of system and components

Democracy Live utilizes a proven, cloud based platform to securely host Secure Select. Our hosting provider's data centers are state of the art, utilizing innovative architectural and engineering approaches. The data centers are housed in nondescript facilities. Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video surveillance, intrusion detection systems, and other electronic means. Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors and contractors are required to present identification and are signed in and continually escorted by authorized staff.

Our hosting provider only provides data center access and information to employees and contractors who have a legitimate business need for such privileges. When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee. All physical access to data centers by employees is logged and audited routinely.

For more information on hosting security, please refer to *Appendix D*.

### 8.2. User-level security

Democracy Live employs multiple levels of user security throughout the Secure Select development lifecycle. Access to the Secure Select hosting environment is restricted to approved server administrators. Server administrators must use two-factor authentication to access and manage the server environments. Additionally, access control lists (ACL) prevent any connections to Secure Select servers without prior approval.

The Secure Select codebase is stored in a secure code repository. Access is limited to developers and requires an SSH connection via approved SSH keys. All code changes applied to the repository are auditable and include the developer, changes made, and a reason for the changes.

### 8.3. Procedures for verifying, checking, and installing essential updates and changes

Secure Select is hosted in a secure, cloud based server environment. Secure Select servers are installed on clustered nodes capable of scaling to meet higher loads due to spikes in network traffic. Critical security patches are applied immediately by implementing automatic updates for critical security patches. Minor updates are performed during low traffic times outside of active elections. Server administrators perform updates with zero down time by using the following update workflow:

1. The server administrator provisions a new node running a Secure Select server.
2. All updates and patches are applied to the new node.
3. The new node is tested to verify Secure Select is running correctly.
4. The new node is then added to the load balancer. User traffic is now directed to the new Secure Select node.
5. After the new node is added to the load balancer, an existing node (needing updates) is removed from the load balancer and is decommissioned.
6. This process is repeated until all nodes in the node cluster are running updated software.

#### 8.4. **Ballot Audit trail**

County administrator should ensure the number of ballots returned, match the number of ballots duplicated and submitted for tabulation.

# Appendix A: WCAG 2.0 Conformance

Guideline	Pass	Technique
<b>Principle 1 – Perceivable</b>	<b>AAA</b>	
<b>Guideline 1.1 – Text Alternatives</b>		
1.1.1 Non-text Content – Level A	Yes	Limited use of graphic content. Text alternatives provided for graphics and icons when necessary.
<b>Guideline 1.2 – Time-based Media</b>	<b>n/a</b>	
1.2.1 Audio-only and Video-only (Prerecorded) – Level A	n/a	
1.2.2 Captions (Prerecorded) – Level A	n/a	
1.2.3 Audio Description or Media Alternative (Prerecorded) – Level A	n/a	
1.2.4 Captions (Live) – Level AA	n/a	
1.2.5 Audio Description (Prerecorded) – Level AA	n/a	
1.2.6 Sign Language (Prerecorded) – Level AAA	n/a	
1.2.7 Extended Audio Description (Prerecorded) – Level AAA	n/a	
1.2.8 Media Alternative (Prerecorded) – Level AAA	n/a	
1.2.9 Audio-only (Live) – Level AAA	n/a	
<b>Guideline 1.3 – Adaptable</b>	<b>Yes</b>	
1.3.1 Info and Relationships – Level A	Yes	Use of landmarks, roles, labels, headings, semantic markup, and structured HTML. Use of CSS to control visual display
1.3.2 Meaningful Sequence – Level A	Yes	Content ordered from top to bottom. DOM order matches visual order.
1.3.3 Sensory Characteristics – Level A	Yes	Warning icons are accompanied by warning text.
<b>Guideline 1.4 – Distinguishable</b>	<b>Yes</b>	
1.4.1 Use of Color – Level A	Yes	Warning text is accompanied by a graphic icon, bold typeface, and the word warning. CSS is used to change visual representation of items with focus.
1.4.2 Audio Control – Level A	n/a	
1.4.3 Contrast (Minimum) – Level AA	Yes	All text and background text meet a 4.5:1 contrast ratio. Warning text is also bold and 16pt for readability.
1.4.4 Resize text – Level AA	Yes	Text can be resized to 200% using the + and - keys
1.4.5 Images of Text – Level AA	n/a	
1.4.6 Contrast (Enhanced) – Level AAA	Yes	All regular text is a 7:1 contrast. All large text is at least a 4.5:1 contrast.
1.4.7 Low or No Background Audio – Level AAA	Yes	No background audio used.
1.4.8 Visual Presentation – Level AAA	Yes	Headers specify text and background colors in CSS. Borders are used to separate content. Main text does not use text or background color attributes.
1.4.9 Images of Text (No Exception) – Level AAA	Yes	No images of text are used.
<b>Principle 2 – Operable</b>		
<b>Guideline 2.1 – Keyboard Accessible</b>	<b>Yes</b>	
2.1.1 Keyboard – Level A	Yes	All elements and functionality are accessible via keyboard using tab and arrow keys.
2.1.2 No Keyboard Trap – Level A	Yes	No elements trap keyboard focus.
2.1.3 Keyboard (No Exception) – Level AAA	Yes	All elements and functionality are accessible via keyboard using tab and arrow keys.
<b>Guideline 2.2 – Enough Time</b>	<b>Yes</b>	
2.2.1 Timing Adjustable – Level A	Yes	No time limits are imposed on users.
2.2.2 Pause, Stop, Hide – Level A	Yes	No moving, blinking, scrolling, or auto updating information.
2.2.3 No Timing – Level AAA	Yes	No time limits are imposed on users.
2.2.4 Interruptions – Level AAA	Yes	No interruptions are presented to users.
2.2.5 Re-authenticating – Level AAA	Yes	Users do not have expiring sessions.
<b>Guideline 2.3 – Seizures</b>	<b>Yes</b>	
2.3.1 Three Flashes or Below Threshold – Level A	Yes	No flashing
2.3.2 Three Flashes – Level AAA	Yes	No flashing
<b>Guideline 2.4 – Navigable</b>	<b>Yes</b>	
2.4.1 Bypass Blocks – Level A	Yes	Using headings, landmarks, and semantic HTML. Also do not use repeated blocks.



2.4.2 Page Titled – Level A	Yes	All pages have an H1 title tag.
2.4.3 Focus Order – Level A	Yes	Yes, all items are focusable using tab or arrow keys.
2.4.4 Link Purpose (In Context) – Level A	Yes	All links use text that describes what the link does.
2.4.5 Multiple Ways – Level AA	Yes	The application is a step by step process with forward and backward navigation.
2.4.6 Headings and Labels – Level AA	Yes	Structured headings are used on every page. All input elements are properly labeled.
2.4.7 Focus Visible – Level AA	Yes	A clear focus indicator highlights the focus of all active elements.
2.4.8 Location – Level AAA	Yes	Page steps are clearly identified using x of y format.
2.4.9 Link Purpose (Link Only) – Level AAA	Yes	All links use text that describes what the link does.
2.4.10 Section Headings – Level AAA	Yes	All page content is separated by hierarchal use of headings.
<b>Principle 3 – Understandable</b>		
<b>Guideline 3.1 – Readable</b>	<b>Yes</b>	
3.1.1 Language of Page – Level A	Yes	Lang attribute is applied to html element
3.1.2 Language of Parts – Level AA	Yes	Full page content is translated including ballot content.
3.1.3 Unusual Words – Level AAA	Yes	Simple, common language is used throughout the application.
3.1.4 Abbreviations – Level AAA	Yes	No abbreviations are used.
3.1.5 Reading Level – Level AAA	Yes	Simple, common language is used throughout the application.
3.1.6 Pronunciation – Level AAA	Yes	Simple, common language is used throughout the application.
<b>Guideline 3.2 – Predictable</b>	<b>Yes</b>	
3.2.1 On Focus – Level A	Yes	Focus is shown, but does not change context or content.
3.2.2 On Input – Level A	Yes	Changing any input value does not change focus or context.
3.2.3 Consistent Navigation – Level AA	Yes	Navigation is the same on every page, in the same place, using a navigation role.
3.2.4 Consistent Identification – Level AA	Yes	Labelling and styling are consistent through the application.
3.2.5 Change on Request – Level AAA	Yes	Automatic updates or changes in context are not made.
<b>Guideline 3.3 – Input Assistance</b>	<b>Yes</b>	
3.3.1 Error Identification – Level A	Yes	Errors are clearly identified using an icon and are presented in descriptive text.
3.3.2 Labels or Instructions – Level A	Yes	Ballot instructions are provided before ballot marking.
3.3.3 Error Suggestion – Level AA	Yes	Overvote errors describe why the error occurred, and how to resolve the error.
3.3.4 Error Prevention (Legal, Financial, Data) – Level AA	n/a	
3.3.5 Help – Level AAA	Yes	Each page includes instructions for the voter.
3.3.6 Error Prevention (All) – Level AAA	Yes	Users are presented with a review page. They can change any selection before printing.
<b>Principle 4 – Robust</b>		
<b>Guideline 4.1 – Compatible</b>	<b>Yes</b>	
4.1.1 Parsing – Level A	Yes	Application has valid HTML including unique IDs and hierarchal structure.
4.1.2 Name, Role, Value – Level A	Yes	All elements use semantic markup, or define aria-label, aria-labelledby, and role attributes.

## Appendix B: Ballot Data Specification

Secure Select loads ballot data definition from a remote source defined by a query parameter. The ballot data source must be a JSON document meeting the following specification. This data can be created manually or by using a product such as LiveBallot. The JSON data should then be uploaded to a server and made publicly available (or at least available from the Secure Select server).

### Example passing data parameter to Secure Select:

[https://ss.liveballot.com?data=URL\\_TO\\_DATA](https://ss.liveballot.com?data=URL_TO_DATA)

#### 1.1.1 Base Data Structure

Property	Type	Description
<b>ballot</b>	Ballot	Ballot data definition.
<b>ballotId</b>	string	Optional ballot id to include with barcode.
<b>election</b>	Election	Election definition.
<b>precinct</b>	Precinct	Precinct definition.

#### 1.1.2 Ballot

Property	Type	Description
<b>code</b>	string	Ballot style code
<b>name</b>	string	Ballot style name
<b>boxes</b>	[]Box	Array of boxes on the ballot (default, header, text)

#### 1.1.3 Box

Property	Type	Description
<b>id</b>	integer	Unique identifier
<b>type</b>	string	Type of ballot content. Allowed Values: default, header, text
<b>titles</b>	[]Text	Array of title text. Used in default and header boxes.
<b>text</b>	[]Text	Array of text content to show. Used in default and text boxes.
<b>text_after</b>	[]Text	Array of text to show after options. Used in default boxes.
<b>sequence</b>	integer	Box order
<b>num_selections</b>	integer	Number of selections that can be made
<b>options</b>	[]Option	Array of ballot options (candidates, yes, no, etc.)

#### 1.1.4 Option

Property	Type	Description
<b>id</b>	integer	
<b>titles</b>	[]Text	Array of title text for the option.
<b>type</b>	string	Type of option. Allowed values: default, writein, text
<b>sequence</b>	Integer	Option order

#### 1.1.5 Text

Property	Type	Description
<b>value</b>	string	Value to display.
<b>format</b>	string	Type of text to display. Allowed values: style, text, html
<b>style</b>	string	Style to use if type is set to style. Allowed values: default, subtitle (for box and option titles)
<b>translations</b>	map[string]string	A map of key/value pairs that represent langCode and translation values

#### 1.1.6 Election

Property	Type	Description
<b>title</b>	Text	Election title. Displayed to voters on first page.

#### 1.1.7 Precinct

Property	Type	Description
<b>id</b>	string	Precinct ID
<b>name</b>	string	Precinct Name

## Appendix C: QR Code Specification

The QR code presented on the Secure Select printout includes JSON data representing the voter's ballot selections. The QR code does not include any information about the voter. The QR code contains header data containing a version number, ballot style, precinct identifier, and a unique ballot id. The unique ballot id cannot be related to the voter in any way. It's only purpose is to identify distinct ballots printed from Secure Select.

The selections in the QR code are stored as numbers relative to the beginning of the ballot and contest respectively. For example, if the voter selected the third candidate of the first contest, skipped the second contest, and marked the second candidate of the third contest on the ballot, the QR code data would be 1:3 and 3:2 to represent those selections.

The following data specifies what information is defined in the QR code and how it is generated.

Line	Key	Value	Notes
1	v	1.2	QR Code format version
2	bs	string	Ballot style code – from Ballot Definition File
3	pid	string	Precinct identifier – from Ballot Definition File
4	id	string	Unique ballot identifier generated by Secure Select
5+	Contest number	Selection number	Contest number starts with 1 for the first contest on the ballot.  Selection number starts with 1 for the first option in the contest.  Multiple selections are joined by a comma (,).  A write-in is represented by the option number, followed by a dash (-) followed by the write in value enclosed in quotes. If a quote is part of the write-in value, it must be escaped with a backslash (\).

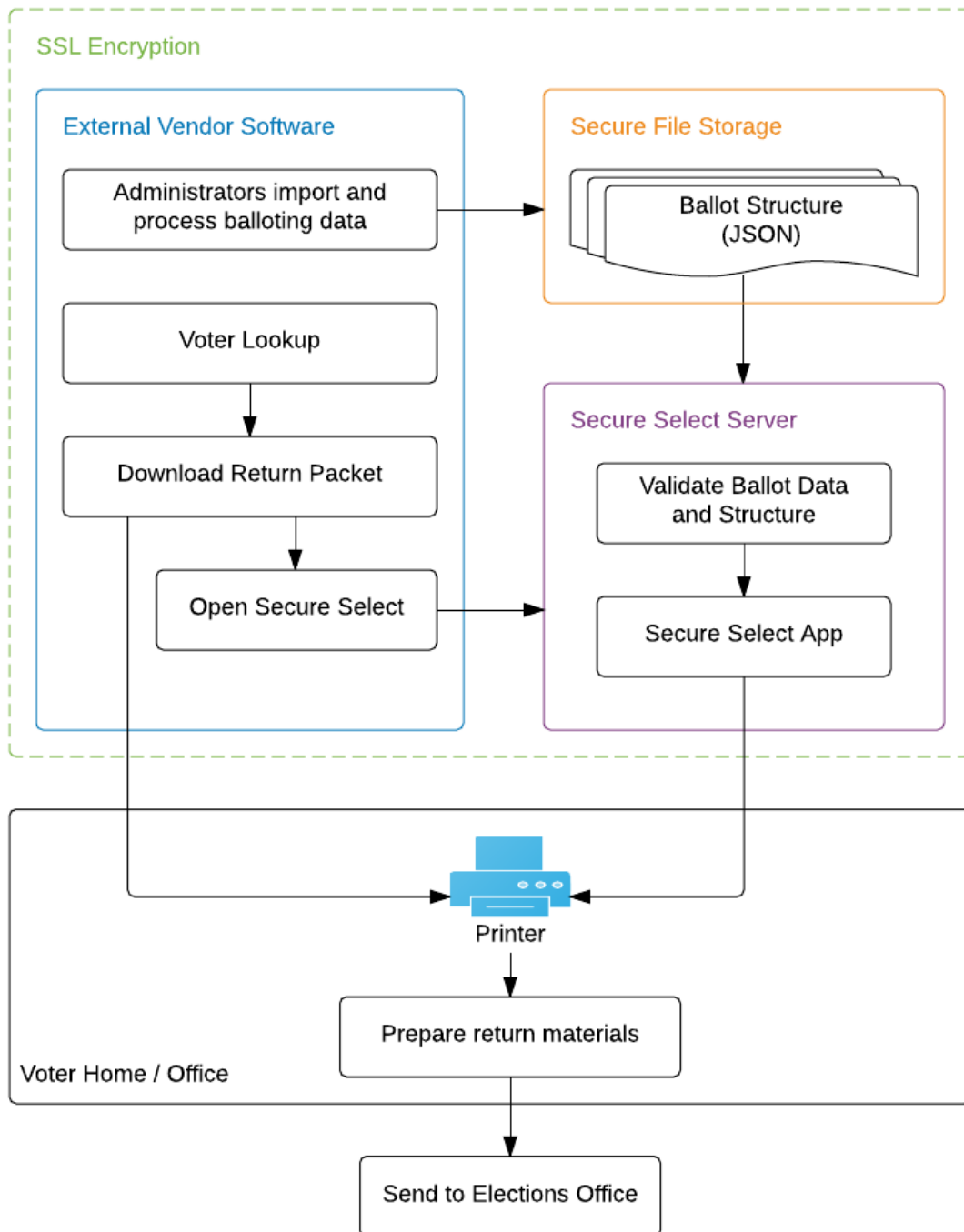
Examples of selection data in the QR code

Value	Notes
1:2	First contest, second candidate marked.
2:3,4	Second contest, candidates 3 and 4 marked
5:1,2-"Thomas Jefferson"	Fifth contest, first candidate marked. Second candidate marked (a write in) with the value Thomas Jefferson entered.
6:3-"Jim \"Jimmy\" Smith"	Sixth context, third candidate (a write in) selected with Jim "Jimmy" Smith entered.

# Appendix D: Secure Select Technical Details

## 1. Architecture and Codebase

Secure Select is designed to have a flexible architecture. Below is the recommended architecture using a third party vendor for voter identification and Democracy Live to host Secure Select as Software as a Service (SAAS).



### 1.1. Secure Hosting

Democracy Live utilizes a proven, cloud based platform to securely host Secure Select. Our hosting provider's computing environments are continuously audited, with certifications from accreditation bodies across geographies and verticals, including ISO 27001, FedRAMP, DoD CSM, and PCI DSS.

By operating in an accredited environment, Democracy Live reduces the scope and cost of audits needed, allowing us to focus on our area of expertise. Our hosting provider continuously undergoes assessments of its underlying infrastructure—including the physical and environmental security of its hardware and data centers—so customers can take advantage of those certifications and simply inherit those controls.

In a traditional data center, common compliance activities are often manual, periodic activities. These activities include verifying asset configurations and reporting on administrative activities. Moreover, the resulting reports are out of date before they are even published. Operating in an accredited environment allows Democracy Live to take advantage of embedded, automated tools for validating compliance. These tools reduce the effort needed to perform audits, since these tasks become routine, ongoing, and automated.

#### 1.1.1. Physical Security

Our hosting provider's data centers are state of the art, utilizing innovative architectural and engineering approaches. The data centers are housed in nondescript facilities. Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video surveillance, intrusion detection systems, and other electronic means. Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors and contractors are required to present identification and are signed in and continually escorted by authorized staff.

Our hosting provider only provides data center access and information to employees and contractors who have a legitimate business need for such privileges. When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee. All physical access to data centers by employees is logged and audited routinely.

#### 1.1.2. Network Security

Democracy Live utilizes several security capabilities and services to increase privacy and control network access. These include:

- Built-in firewalls that allow creation of private networks, and control network access to instances and subnets
- Encryption in transit with TLS across all services
- Connectivity options that enable private, or dedicated, connections from Democracy Live offices or on-premises environments
- DDoS mitigation technologies as part of our auto-scaling strategy

#### 1.1.3. Inventory and Configuration Management

Democracy Live server administrators deploy and monitor Secure Select servers using a series of tools including:

- Deployment tools to manage the creation and decommissioning of resources

- Inventory and configuration management tools to identify resources and then track and manage changes to those resources over time
- Template definition and management tools to create standard, preconfigured, hardened virtual machines
- Containerized environments based on secure images ensuring quick scaling and reproducible environments

#### 1.1.4. Access Control

Democracy Live server administrators define, enforce, and manage user access policies across services. These include:

- Identity and access management capabilities to define individual user accounts with permissions across resources
- Multifactor authentication for privileged accounts
- Integration, and federation, with corporate active directory

#### 1.1.5. Monitoring and Logging

Democracy Live server administrators utilize tools to monitor our server environment. These include:

- Deep visibility into API calls, including who, what, when, and from where calls were made
- Log aggregation and options, streamlining investigations and compliance reporting
- Alert notifications when specific events occur or thresholds are exceeded

### 1.2. Scalable Architecture

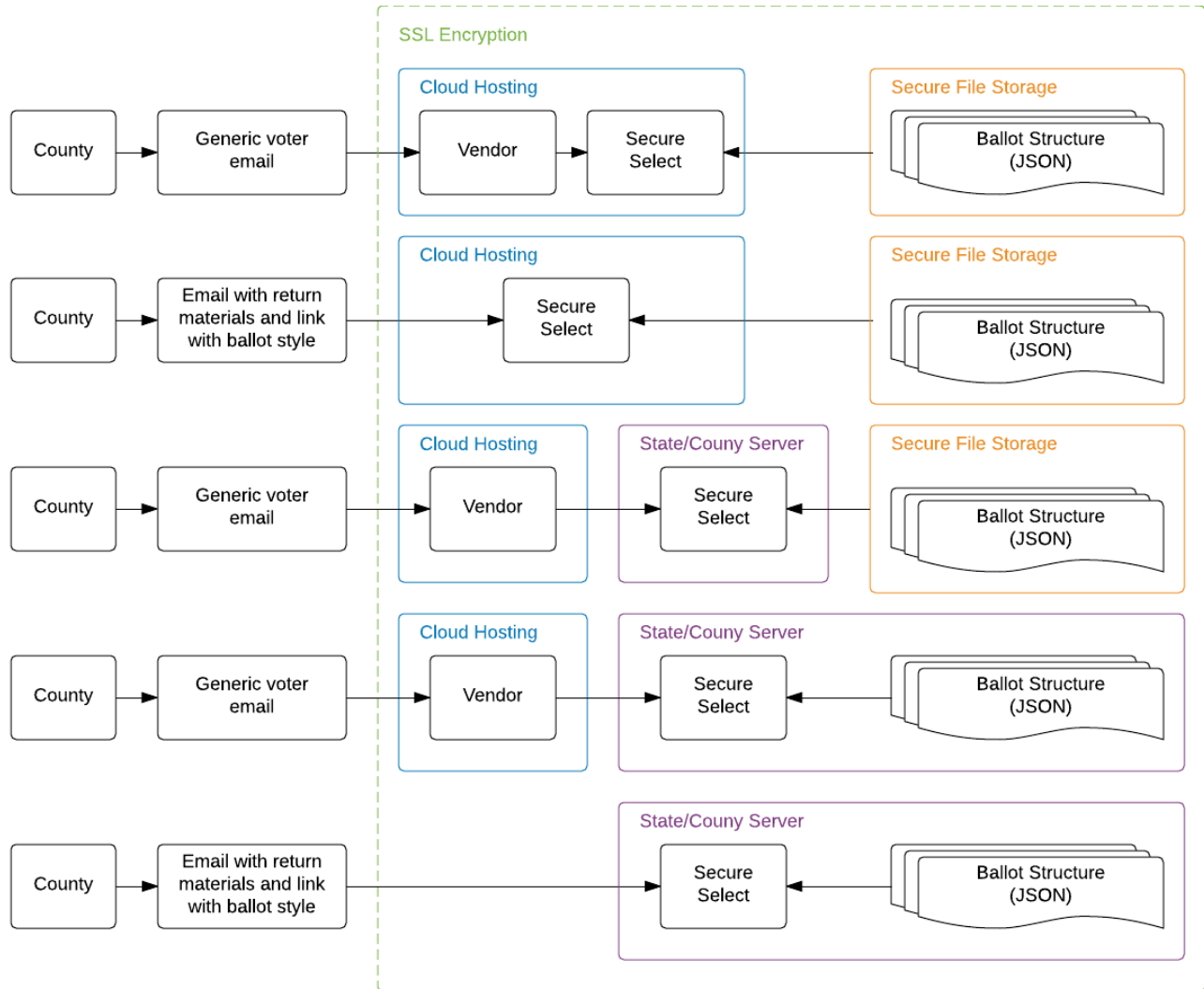
Using automatic scaling, Secure Select stays online and responsive to voters even during traffic spikes around critical election dates. Secure Select is built using a stateless server architecture making it possible to dynamically provision new server nodes without manual interaction from a server administrator. CPU and memory utilization on Secure Select servers are monitored 24/7. If the CPU or memory usage of a server surpasses a threshold, automatic scaling invokes the following steps:

1. A new Secure Select server (node) is provisioned.
2. Once the new node reaches a steady state (it has started up), a health check is performed on the node
  - a. If the node is healthy, it is added to the load balancer.
  - b. If the new node is unhealthy, it is deprovisioned and the process repeats at Step 1.
3. Traffic is now distributed evenly across all nodes including the new node.

This process will repeat until CPU and memory usage on all servers is at an acceptable level.

### 1.3. Flexible Architecture

Secure Select can be utilized in a variety of different configurations to meet the needs of any state or county. The following diagram shows several possible configurations including options for county or state hosting of the Secure Select application.



### 1.4. Application Review and Certification

Secure Select is composed of two main components with a complete codebase under 2,500 lines of code. This makes a full codebase review possible in just a matter of hours. The entire application is just under 20MB.

#### 1.4.1. HTML5 Application

The HTML5 application is written using the AngularJS framework using HTML and TypeScript. The entire application has been written in under 1,600 lines of TypeScript code and under 300 lines of HTML with an average of less than 100 lines of code per file.



#### 1.4.2. Web Server

The web server is responsible for hosting the HTML5 application, along with downloading, sanitizing, and preparing ballot data. The web server can be run on any Virtual Machine with no server requirements. The web server port is configurable such that it can run in parallel with existing web servers (Apache, NGINX, etc). This provides administrators with complete flexibility using new or existing infrastructure.

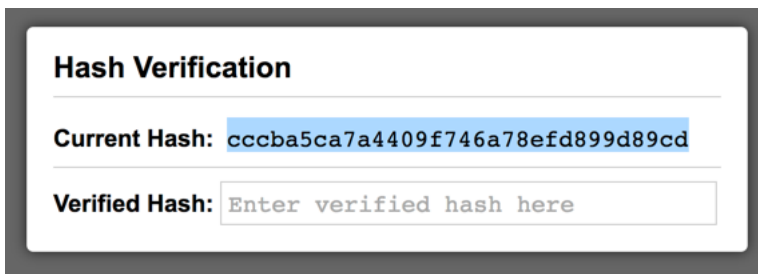
The web server codebase is written in Go 1.7 and is under 250 lines of code.

## 2. Source Code Verification

A hashcode is a unique character string created by a one-way encryption of any data. Secure Select provides a verification page which displays a hashcode generated from the text of every file and executable in the application. This hashcode can be stored for comparison after pre-election verification. To assure that no changes have been made to the codebase, the verification page can be used to compare hashcodes at any time to verify the codebase has not been changed.

### 2.1. Storing the Secure Select Hash Code

After a version of Secure Select is approved, anyone can access the Secure Select verification page (<https://ss.liveballot.com/verify>) to view the *Current Hash*. The *Current Hash* is a unique hash code generated from every file in Secure Select. This code will change if any line of code in the application changes. The California Secretary of State can record this code for future reference.



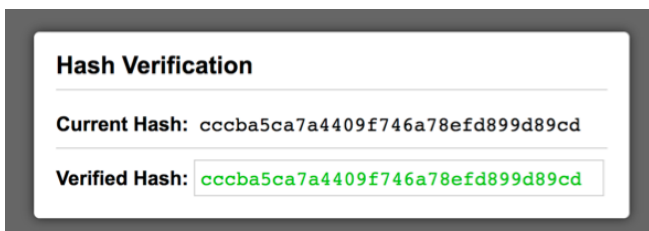
**Hash Verification**

**Current Hash:** cccba5ca7a4409f746a78efd899d89cd

**Verified Hash:**

### 2.2. How to use Hash Code Verification

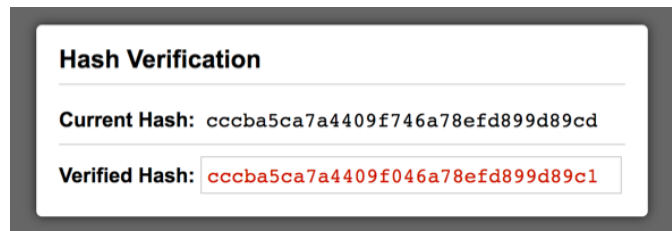
At any point in the future, the Secure Select Verification page can be accessed. Simply enter the hash code saved on record to verify no changes have been made. If any changes to the source code have been made, the Secure Select hash code will not match the hash code saved after approval. Different hash codes are clearly displayed to the administrator verifying the system.



**Hash Verification**

**Current Hash:** cccba5ca7a4409f746a78efd899d89cd

**Verified Hash:** cccba5ca7a4409f746a78efd899d89cd



**Hash Verification**

**Current Hash:** cccba5ca7a4409f746a78efd899d89cd

**Verified Hash:** cccba5ca7a4409f046a78efd899d89c1

# Appendix E: Acceptance Testing Tables

1.0 General Functionality		Status
1.1	Candidates can be selected and deselected by clicking on name.	
1.2	Candidates can be selected and deselected by clicking on the checkbox.	
1.3	Over-voting is not allowed.	
1.3.1	A warning is presented when an over-vote is attempted.	
1.4	A textfield appears when selecting a write-in.	
1.4.1	A candidate name can be typed into a write-in field.	
1.4.2	Deselecting a write-in checkbox clears the candidate name entered.	
1.5	Selections represented on the Review Page represent selections made on the Ballot Marking Page.	
1.5.1	A “No Selections” warning is shown for any contests missing selections.	
1.5.2	Under-votes are clearly identified on the Review Page.	
1.5.3	Write-in candidates are displayed on the Review Page.	
1.6	Voters can change their selections.	
1.7	The printed ballot accurately displays selections on the Review Page.	
1.7.1	Only selections made are presented on the printed ballot (not all candidates).	
1.7.2	Write-ins are shown on the printed ballot.	
1.8	After ending the user session and returning to the application, selections are no longer visible.	

2.0 Screen Reader Accessibility		Status
2.1	All functionality in 1.0 is accessible using screen reader keyboard commands.	
2.2	Verify the on-screen instructions on page one are not read by the screen reader.	
2.3	Over-vote warnings are clearly read by screen reader when attempting to over-vote.	
2.4	Screen reader clearly identifies selected and unselected candidates when navigating the ballot.	

3.0 Keyboard Accessibility		Status
3.1	All functionality in 1.0 is accessible using only the keyboard.	
3.2	Keyboard controls presented in the on-screen instructions operate as expected.	
3.2.1	The up and down arrow keys move keyboard focus up and down.	
3.2.2	The + key zooms text up to 200% of the original size.	
3.2.3	The – key shrinks text down to the original size.	
3.2.4	The space bar can be used to activate an item.	
3.3	Keyboard focus is visually identified on screen.	

4.0 Accurate Ballot Display		Status
4.1	Contests titles, subtitles, and text display correctly.	
4.1.1	Contest order is correct.	
4.1.2	Contests have the correct header.	
4.2	Candidate titles and sub text display correctly.	
4.2.1	Candidates display in the correct order.	
4.3	Write-ins display correctly.	
4.3.1	The correct number of write-ins display.	

5.0 Voter Privacy		Status
5.1	No network communication is made while performing all steps in 1.0	
5.1.1	No network activity occurs when marking a selection.	
5.1.2	No network activity occurs when printing selections.	