



Los Angeles County Registrar-Recorder/County Clerk

DEAN C. LOGAN  
Registrar-Recorder/County Clerk

# VSAP Open Source

---

## Preliminary, Conditional Implementation Plan

Dated: October 2021  
Version: 5.0

**DOCUMENT INFORMATION**

This document describes a high-level plan to make custom code, developed as part of VSAP, open source. Refer definition of open source at <https://opensource.org/osd>. This document describes tentative timeline, responsibilities, and mechanisms to be established in order to meet the objective of enabling VSAP code as open source.

**REVISION HISTORY**

VERSION #	DATE	DESCRIPTION
1.0	July 2021	Initial draft
2.0	Aug 2021	Final draft – added sections for governance and code management
3.0	8/23/2021	Edited document and added content to each section – Final draft
4.0	8/31/2021	Made updates based on feedback from ESC presentation
5.0	9/30/21	Final, published version

**REFERENCED DOCUMENTS**

TITLE	DATE	PROVIDED BY
VSAP 2.1 Conditional Approval	10/01/2020	CA SOS

**TABLE OF CONTENTS**

**1.VSAP OPEN SOURCE INITIATIVE – AN OVERVIEW .....4**

**2.OPEN SOURCE GOVERNANCE .....5**

    2.1 GOVERNANCE MODEL - EVALUATION..... 5

    2.2 RECOMMENDED GOVERNANCE MODEL..... 7

**3.INFRASTRUCTURE AND PUBLIC-FACING PORTAL .....7**

    3.1 VSAP OPEN SOURCE MANAGEMENT TEAM..... 7

    3.2 INFRASTRUCTURE PLATFORM..... 8

    3.3 OUTREACH AND COMMUNICATIONS ..... 9

**4.CODE MANAGEMENT.....10**

    4.1 CODE ACCESS..... 10

    4.2 CODE CONTRIBUTION ..... 11

    4.3 CODE MANAGEMENT & VERSIONING..... 11

    4.4 SECURITY PATCHES AND VULNERABILITIES..... 12

        4.4.1 *Bug/Vulnerability Discovery & Reporting* ..... 13

        4.4.2 *Patch Management* ..... 13

    4.5 TESTING ..... 13

    4.6 DIGITAL SIGNATURES AND HASH VERIFICATION ..... 14

**5.LICENSING .....15**

    5.1 THIRD-PARTY COMPONENTS..... 15

    5.2 ANALYSIS OF THIRD-PARTY COMPONENTS ..... 16

**6.CERTIFICATION AND REGULATORY FRAMEWORK.....17**

**7.RECIPROCITY.....18**

**8.TIMELINE .....19**

**9.RECOMMENDED NEXT STEPS.....20**

**10. APPENDIX A – OPEN SOURCE WORKGROUP ROSTER.....21**

**11. APPENDIX B – WORKGROUP MEETINGS.....21**

**12. APPENDIX C – TOPICS OF DISCUSSION .....22**

    12.1 FEBRUARY 2021 ..... 22

    12.2 APRIL 2021..... 23

    12.3 MAY 2021 ..... 24

    12.4 JUNE 2021..... 25

    12.5 SEPTEMBER 2021..... 25

## 1. VSAP Open Source Initiative – An Overview

---

In March 2020, the Voting Solutions for All People (VSAP) system was implemented to replace the County of Los Angeles Registrar-Recorder/County Clerk's (RR/CC) antiquated voting system. The model was implemented with the goal to offer the solution in an open source forum for other local and state governments to enhance their jurisdiction's voting experience and, in general, to enhance transparency and market expansion for voting solutions.

Additionally, as part of the VSAP 2.1 certification from California Secretary of State (SOS), condition number 28 specified that a plan to make VSAP source code available and open should be developed in partnership with the SOS. It was specified that this plan should include how the County intends to inventory and track components and vulnerabilities, provide timely updates and patches of components that will be made, and ensure ongoing active involvement from the open source community. The exact certification condition is given below:

-- Excerpt from <https://votingsystems.cdn.sos.ca.gov/vendors/LAC/vsap2-1/vsap21-cert.pdf>--

28. By October 1, 2021, Los Angeles County in partnership with the Secretary of State shall develop a plan to make the source code of the Voting Solutions for All People voting system available and open, which appropriately addresses concerns related to the security of making the source code available and open. The plan shall include, at a minimum, how the County will inventory and track components and vulnerabilities, how timely updates and patches of components will be made, and how the County will ensure ongoing active involvement from the open source community.

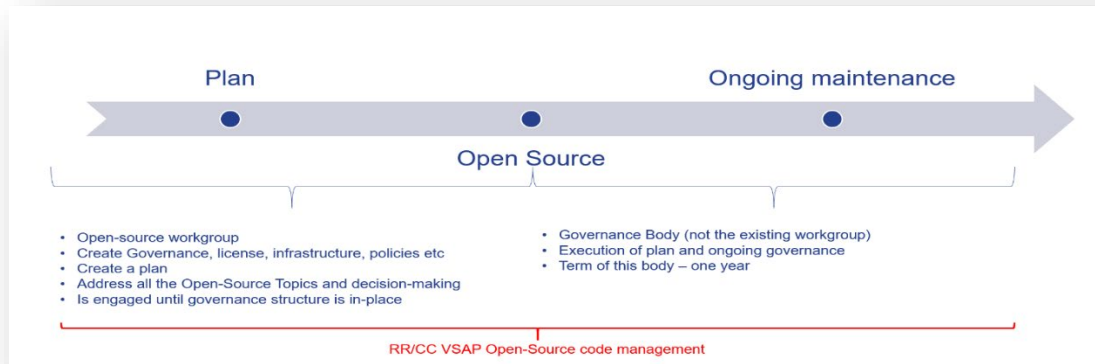
-- end excerpt--

To aid in the creation of this plan and the overall structure of the VSAP Open Source Initiative, the VSAP Open Source Workgroup was established, comprised of key stakeholders, open source advocates, and industry leaders. The purpose of this workgroup was to aid RR/CC in considering a governance model and the creation of this Open Source plan.

Based on the discussions within the workgroup, this high-level plan was established. The goal of this plan is to define the required steps to establish a governance team, determine licensing models, define infrastructure and policies, and determine the ongoing lifecycle and management of VSAP Open Source.

This plan is solely a recommended approach based on research and input from industry experts. Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.

A representation of the sequence of events is given below. This sequence is based on the information available to date and is therefore, an estimate. A more detailed plan will be established once required information is received from an appropriate governing body and once a dedicated Open Source Project Manager is onboarded.



The pillar of the VSAP Open Source Initiative is governance. The governance body will be responsible for managing the selected application platform, issuing code, and providing ongoing maintenance and enhancement processes. It is recommended that an, objective and experienced open source governance team be established.

## 2.1 Governance Model - Evaluation

To identify the appropriate governance model for VSAP Open Source, VSAP Open Source Workgroup evaluated six (6) established governance models. Although a model for the initial launch of a governance team is recommended here, it is further recommended that the decision be fluid and reevaluated within two (2) years of the formation of the governance team.

The following governance models were evaluated:

- Do-ocracy - Whoever invests most time, energy, attention has the most influence in those areas and governs de-facto. This model has been used for Node.js and Rust.
- Founder-Leader - Individual/entity who started the project also administers, establishes vision, and makes decisions. This model has been used for Python.
- Self-appointing Council/Board - Leadership groups to govern various aspects of a project: Technical Advisory, Code Advisory, security advisory, etc. This model has been utilized for Data.gov and Code.gov.
- Electoral - Conduct governance through elections. Multiple qualified/ interested parties to play same roles. Precise documentation is required when utilizing this model. This model has been used for Open Web Application Security Project (OWASP).
- Corporate-Backed - Run by industry consortia. This may create mismatched expectations among adopters. This has been used for Golang.
- Foundation-Backed - Run by non-profit entity. This is a more inclusive model. Funding/ legal requirements generally limit to larger projects. This model is used for Apache.

Although each model offers a unique approach to governance, the pros and cons were assessed to identify any potential model pitfalls and possible negative implications to applying the model for VSAP Open Source. The pros and cons of these various governance models are given below.

MODEL	PROs	CONs	DIFFICULTY TO IMPLEMENT
<b>Do-ocracy</b>	Flexible model, ideal for small projects, peer-review is faster	No Governance – depend on very frequent communication with community, individuals exert control	Low
<b>Founder-Leader</b>	Clear vision, Lines of power and authority are typically clear, Ideal for start-up projects	Limitations appear as a project grows to a certain size; The initial entity becomes a bottleneck; non-founders may feel they cannot influence the roadmap	Low
<b>Self-appointing Council/Board</b>	A number of leadership groups to govern various aspects of a project: Technical Advisory, Code Advisory, security advisory etc., useful when project does not have a sponsoring foundation and establishing electoral mechanisms is difficult.	May hinder community participation in leadership activities; member-selection processes spawn self-reinforcing leadership cultures	Medium
<b>Electoral</b>	Conduct governance through Elections, communities establish and document electoral procedures to which they agree, then enact those procedures as a regular matter of decision-making; ideal for large projects	Need for precise documentation, Elections may become contentious, distracting, and time-consuming.	Medium
<b>Corporate-Backed</b>	Backed by corporates; accelerates adoption, spurs development activity atop a software platform, supports a plugin ecosystem, avoids the overhead required for cultivating an external developer community	Funding needed. Objections to this model arise if a project claims to support an open community but is in fact wholly controlled by a company or consortium. This can create mismatched expectations among adopters.	High

MODEL	PROs	CONs	DIFFICULTY TO IMPLEMENT
Foundation-Backed	Backed by foundation; accelerates adoption, spurs development activity atop a software platform, supports a plugin ecosystem, avoid the overhead required for cultivating an external developer community	Funding needed. Objections to this model arise if a project claims to support an open community but is in fact wholly controlled by a company or consortium. This can create mismatched expectations among adopters.	High

## 2.2 Recommended Governance Model

After evaluating the six (6) proposed models, the pros and cons, the difficulty to implement/maintain, and their fit to the open source application, it is recommended that the RR/CC move forward with the Self-Appointing Council Model. This model ensures that the initial governance team is comprised of interested parties with knowledge and expertise to establish this plan. Additionally, this model offers the necessary oversight that is lacking in the Do-ocracy and Founder-Leader model without the complications found in the Electoral, Corporate, or Foundation-Backed model. This model should be re-evaluated within two (2) years of formation and every two (2) years thereafter.

Furthermore, the SOS as the legally designated regulatory authority for the testing and approval of voting systems in California will ultimately need to recognize the governance structure and provide applicable regulatory framework for making voting system related source code open and available. The SOS will remain the regulatory authority of this initiative until a determination is made that that authority is or should be legally distributed elsewhere.

## 3. Infrastructure and public-facing portal

The establishment of an open source forum would require a dedicated platform, accessible by local and state governments, and managed by a dedicated team.

### 3.1 VSAP Open Source Management Team

The RR/CC recommends the formulation of a team of IT specialists to manage the source code and the repository. This teams should be referred to as the **VSAP Open Source Management Team**. This team should administratively be part of RR/CC Information Technology Bureau (ITB) organization. This team should be solely dedicated to VSAP Open Source management and any initiatives and independent of Department IT operations.

Various models of this management team were evaluated by the Workgroup, including establishing this team within RR/CC ITB, outsourcing to shared services (ISD), outsourcing to an external commercial entity, or outsourcing to a non-profit organization.

It is recommended that VSAP Open Source should begin with this team in-house with the RR/CC, but as the program evolves, this would be revisited. The benefits of utilizing an in-house versus outsourcing or shared services are:

- Initial control of the processes and procedures
- Gaining time in learning about how the audience reacts to VSAP Open source
- Establishing processes that can easily be transferred to another model, if decided, later-on

In order to support the recommended in-house model, the RR/CC would need to obtain hiring authority for the following positions to support ongoing development and management of the platform. These recommended positions include:

ROLE	DUTIES
<b>Open Source Manager</b>	Oversight and management of VSAP Open Source technology and resources
<b>Infrastructure Developer</b>	Ongoing development and maintenance of the repository, code management and CRM platforms
<b>Quality Assurance/Testing</b>	Ongoing testing of newly submitted or developed code/updates to repository or CRM platforms

These recommended positions would need to be acquired by the Department and filled within one (1) year of the finalization of this plan to maintain the high-level timeline provided in Section 8.

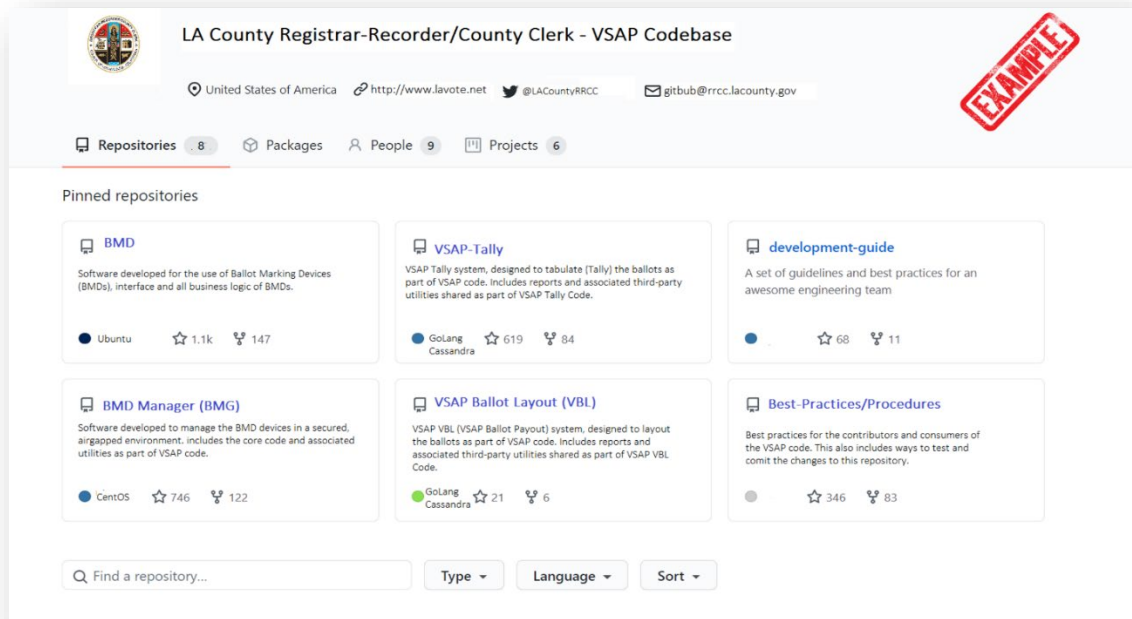
This recommended approach is based on research and the input from industry experts. Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.

### 3.2 Infrastructure Platform

The Open Source Workgroup recommends the RR/CC host all the VSAP code (custom code) in a public-facing GitHub repository. Research concluded that the use case for this platform aligned to the open source initiative. In addition, several government agencies utilize the platform for similar cases, including the Cybersecurity and Infrastructure Security Agency (CISA).

It is recommended that the repository be managed and owned by RR/CC and should have logical divisions of multiple applications or products (e.g., BMD, BMG, ISB, etc.) for the source-code. A sample representation of such a GitHub repository is given below.





The code repository should be established within six (6) months of adoption of this plan. The release of the repository should coincide with proactive outreach efforts planned for Outreach and Communications (Section 3.3) to notify interested parties of the platform’s availability.

This recommended approach is based on research and the input from industry experts. Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.

### 3.3 Outreach and Communications

Once the Governance team and infrastructure is established, the RR/CC should create an outreach and communications plan to raise awareness for local and state governments who may be interested in obtaining access to VSAP Open Source code.

The communication from RR/CC on VSAP Open Source (for public) should include the following:

- A VSAP Open Source Management Team. The team will work under the Assistant Registrar-Recorder/County Clerk, ITB.
- A dedicated email address should be established for outgoing communication from the VSAP Open Source Management Team.
- A dedicated web page should be established for interested parties to locate pertinent information.
- Social media accounts will be created for outgoing communication. For example, twitter handle such as @vsap\_opensource. Similarly, other social media accounts will be utilized for outgoing communication regarding VSAP Open Source.

The complete communication channels and an outreach plan should be established within six (6) months of adoption of this plan.

This recommended approach is based on research and the input from industry experts. The RR/CC is awaiting direction from the SOS as to who should be responsible for hosting and maintaining these social media accounts. Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.

## 4. Code Management

---

After the establishment of the infrastructure and regulatory framework, the RR/CC should determine the process for code access, how to manage code contributions, how to track versioning, and subsequent testing and licensing for new modified or introduced code.

### 4.1 Code Access

Members of the public should have access to VSAP Open Source code through GitHub. There are two “types” of users anticipated using the VSAP Open Source code. These are:

- **Code Consumers:** These are members of public (or entities) that want to read or review the source code of VSAP. This may include commercial entities, security experts, researchers (for educational purposes) and anyone who wants to review the code. Eventually, the “consumers” may become “contributors” if they want to offer enhancements/fixes to the source code.
- **Code Contributors:** Code contributors are individuals (or entities) who want to contribute to the enhancement of VSAP source code or use the code for development purposes. Code contributors will need to follow the guidelines and procedures to “commit” to the codebase. Any code change will need to go through a review and vetting process, before getting accepted into the codebase.

It is anticipated that RR/CC will track who has accessed the code. For that, a simple online application (form) would be submitted by the “consumer/contributor” of the code. The simple application would have fields like name, organization being represented, purpose of code access, modules to be access (web-based ISB, BMD, BMG, Tally, VBL) etc. This application then goes to the VSAP Code Management Team, which then verifies the recipient and provides digital signatures to verify the authenticity of downloaded codebase.

The information of the recipient would be kept in a simple CRM (customer relationship management) system. The RR/CC should implement a cloud-based CRM (Salesforce) for the record-keeping purposes as part of the plan.

The CRM and the process to obtain the source-code should be established within one (1) year of adoption of this plan. It is anticipated that code distribution should be in multiple phases. ISB/RAVBM, which are the web-based components should open first, followed by BMG, Tally, VBL and finally, BMD.

As part of the code preparation, RR/CC would work with its partners to ensure that code is clean and free of any libraries that may contain restricted licenses. This effort would be completed in conjunction with license and third-party libraries (Section 5).

A rollout plan of these components should be published within one (1) year of adoption of this plan. The rollout plan will describe the timelines of this phased source code and the associated code-preparation.

## 4.2 Code Contribution

### Code Contribution Guidelines

As local and state governments use the existing open source code, modifications or additions to the code would be submitted to meet each jurisdiction's unique needs. As a result, guidelines need to be defined to ensure new contributions only enhance the existing VSAP code. It is recommended that these guidelines include the following:

**Quality:** Code contribution to VSAP Open Source should meet guidelines used in the VSAP coding with sufficient test cases, descriptive commit messages, evidence that the contribution does not break any compatibility commitments or cause adverse feature interactions, and evidence of high-quality peer review.

- **Size:** VSAP Open Source project's culture should be one of small pull requests (PRs) submitted regularly. The larger a PR, the more likely it is required to be resubmitted as a series of self-contained and individually reviewable smaller PRs.
- **Scope:** To ensure VSAP Open Source remains coherent and focused, contributors need to ensure that they adhere to the feature's scope.
- **Maintainability:** If the feature requires ongoing maintenance (for example, support for a particular brand of database), local and state jurisdictions should be asked to accept responsibility for maintaining it.
- **Non-duplicative:** If the contribution duplicates feature that already exist or are in progress, local and state jurisdictions should be asked to work with the project maintainers to reconcile.

In addition, there are several additional requirements that may apply to larger contributions, which would need to be addressed by the Governance Team. A code management plan will need to be created within one year (1) of the adoption of this document.

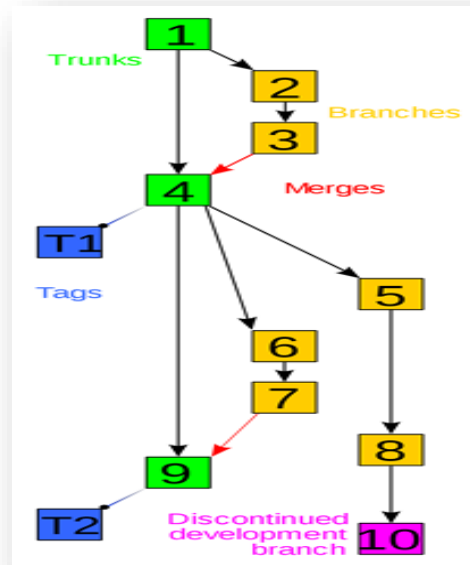
This recommended approach is based on research and the input from industry experts. Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.

## 4.3 Code Management & Versioning

It is recommended that version control be tracked as new iterations and versions of the code are updated and released. Version control is the recording of changes to a file or set of files over time so that you can retrieve specific versions later and manage a collection of files for use. Version Control can be in the form of any three of the following, Local, Centralized, or Distributed.

- **Local**
  - Local Version Control uses a simple database to keep all the changes to files under revision control.
- **Centralized Version Control**
  - Centralized Version Control has a single repository that contains all the versioned files, and several users that check out files from that central place.
- **Distributed Version Control**
  - In a distributed Version Control, users don't just check out the latest snapshot of the files: they fully mirror the repository.

Distributed Version Control concepts/methodologies are recommended as the source code will be housed in a public-facing GitHub repository.



RR/CC will await direction from the SOS as to which of the three (3) methods above would be adopted and authorized. Based on the adopted type, there exists a variety of version systems that would be employed to achieve the intended results. The RR/CC will make a recommendation for the versioning control system to be used after the SOS determines the version method that should be adopted.

A policy on versioning will be created by RR/CC in conjunction with an independent subject matter expert group within one (1) year of adoption of this plan and determination of pending items from SOS. The policy will describe the best-practices and mechanism to manage the versions of VSAP source-code.

#### 4.4 Security Patches and Vulnerabilities

One of the promoted features of open source software is the wide availability of the source code and the potentially large number of critical eyes examining the source code. The results of this scrutiny are more robust and more secure software and applications. In light of these commonly-held beliefs, there is a growing perception that open source software, for example the various instantiations of the Linux operating system and various software applications, is inherently more secure, due to the freely available source code and greater levels of critical scrutiny and transparency.

VSAP Open Source prioritizes identification and focus on security related bugs and vulnerabilities. Based on the feedback received from the public and security researchers, if there are vulnerabilities found in the source code, a list of such issues would be maintained by the RR/CC Open Source Team.

#### 4.4.1 Bug/Vulnerability Discovery & Reporting

It is recommended that a public-facing bug and incident reporting and tracking system (based on JIRA software) is established and kept up to date. Using this system, members of public and security experts can report any bugs or vulnerabilities found in the source code. This list of bugs is reviewed and prioritized on an ongoing basis. A regular and recurring update to the priorities would be reported to the governance team. Based on the priority of these bugs (as a product backlog), the next version of the affected module will be planned.

Proactive communications would be established to keep the community updated on the status of versions and how bugs/vulnerabilities are being mitigated, addressed, and resolved.

#### 4.4.2 Patch Management

Patch management would be an inherent task of the VSAP Code Management Team. The patches would be planned as part of the versions of the software. It will be necessary for the SOS to provide the regulatory framework for testing and approving minor and major versions (including the fixes being applied to the code).

The patch management schedule as well as the vulnerability discovery/reporting processes would be determined as part of the code management plan within one (1) year of the adoption of the document and the determination of and relevant pending guidance from the SOS.

### 4.5 Testing

Updates to VSAP Open Source code, would require ongoing testing prior to release of updates or patches. Software Testing Governance is a test discipline for software systems that governs the test management process by the means of applying suitable test strategy, test process improvements, test optimizations and test performance to measure and improve the quality of deliverables. The aim of the Software Testing Governance is to provide the transparency of software system measurement and performance assessment to enable further improvements. Software Testing Governance operates through a set of well-defined test organization strategies, strict test management policies, principles and test tools to control and assess the quality and progress of the software system.

Testing Governance implementation needs significant management of people, processes and technology with transparent communication and information.

**It is recommended that the software testing guidelines implemented for Open Source should include the following:**

- **Test Goal** – Test goals must be identified prior to beginning testing.
- **Test Strategy** – Test strategy must be documented and detailed prior to beginning testing.
- **Test Management** – Testing should be effectively managed through the entire operation through tracking, documentation, and necessary course corrections as needed.

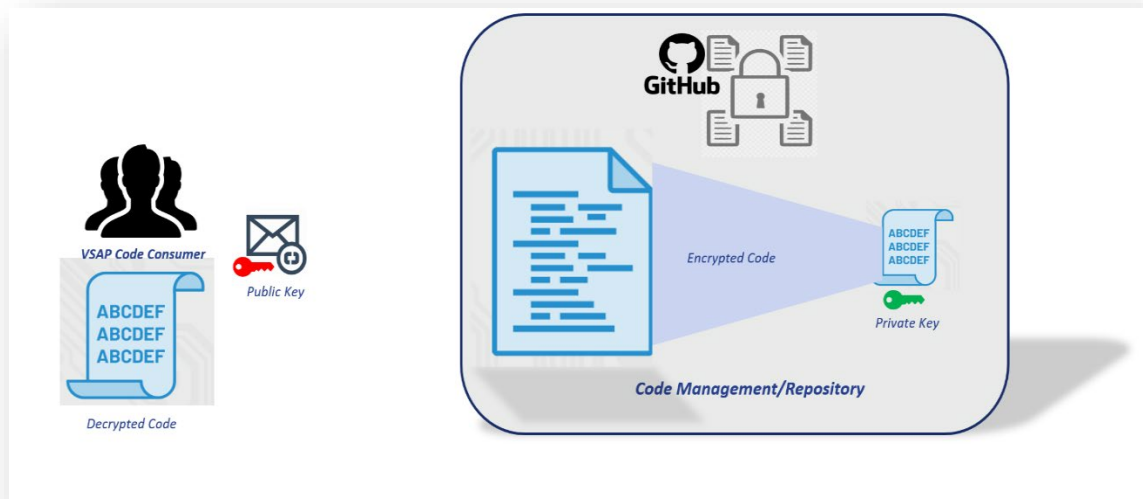
- **Test Measurement** – All testing should be measured, and the incidents associated with the test operation recorded.
- **Test Evaluation** – All incidents recorded and discovered should be evaluated and root cause should be determined for the risks and incidents identified.
- **Test Improvement** – After each iteration of testing, a retrospective should be performed to suggest test improvements to minimize further risks.

The **VSAP Open Source Management Team** identified in Section 3.1 should manage the open source code testing for VSAP Open Source. As mentioned earlier in this document, it was recommended that the VSAP Open Source Management Team should begin with the in-house model. The benefits of utilizing an in-house model versus outsourcing or shared services are:

- Initial control of the testing processes and procedures
- Establishing processes that could easily be transferred to another model, if decided, later-on

A VSAP Open Source testing plan should be provided within six (6) months of adoption of this plan. Additionally, the recommended positions to fill these roles would need to be authorized by the RR/CC and filled within one (1) year of adoption of this plan to maintain the high-level timeline provided in Section 8.

#### 4.6 Digital Signatures and Hash verification



As part of the process of delivering the source code to the consumer, it is recommended that a hash value of released source-code be provided and methods to obtain the hash values of the downloaded code. The consumer can check the hash values to confirm if the source code released by RR/CC is the same that has been downloaded. This is the only verification that will be provided with the source code.

RR/CC would rely on the digital signature functionality provided by GitHub; with hash verification being the only mechanism to ensure that correct code is being used by the consumers. A Bill of Materials (BoM) will be published as part of the VSAP code package being delivered. Details of this BoM will be listed in the documentation related to the code repository setup.

## 5. Licensing

---

The RR/CC will work with Los Angeles County Counsel (County Counsel), with input from the community and relevant stakeholders (e.g. Open Source Workgroup and SOS), to review and determine the appropriate licensing structure for VSAP going forward. The licensing model may be a widely known open source license type [whether permissive (e.g. BSD, MIT, Apache, etc.) or restrictive/copyleft (e.g., GNU Public License 2.0 or 3.0)] or a proprietary license specific to VSAP (while still ensuring the VSAP goals of openness and transparency are met). Multiple considerations will drive the determination of the final license type, including:

- The analysis of third-party components under Section 5.2, how those components are used in the code (e.g., are those components distributed separately from the VSAP source code), and their individual licenses will need to be considered to determine if they restrict available licensing options (for example, certain licensing types are incompatible with each other);
- Whether reciprocity is to be required under the license under Section 7.0; and
- Unique security, regulatory, and certification concerns for a voting system under Section 6.0.

### 5.1 Third-party Components

VSAP applications use many third-party libraries released with a range of licenses: Apache, MIT, BSD, LGPL, and CDDL, etc. Each of these licenses has its own “paperwork” requirements. For example, the Apache License, v2.0 says (excerpt):

*If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file.*

The MIT license contains a copyright notice and says (excerpt):

*The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.*

The BSD license also contains a copyright notice and says (excerpt):

*Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.*

LGPL v.3 says (excerpt):

*(You should) give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.*

LGPL and CDDL licenses may also require supplying the source code along with a binary form of a library, so the information on the way in which the source code can be obtained should be provided somewhere.

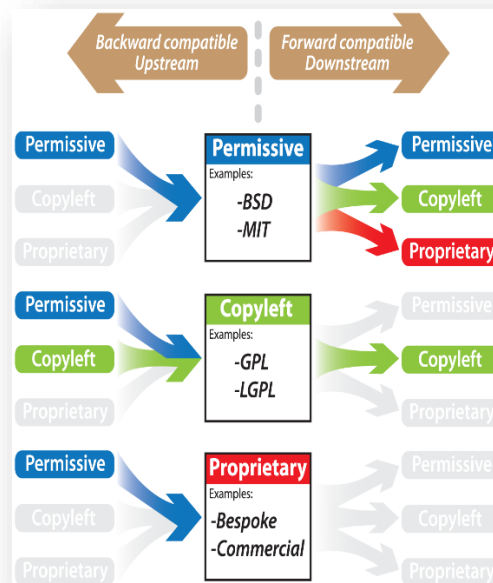
## 5.2 Analysis of third-party components

An analysis on all the third-party components being used within VSAP Open Source needs to be completed, along with the respective licenses. The purpose of this analysis is to:

- Quantify exactly the third-party libraries being used;
- Understand the licenses that are copy-right or copy-left and restrictions of those licenses;
- Understand how the code will be used and if components with permissive licenses can be distributed along with the base source code; and
- Assist the RR/CC (LA County) with the licensing decisions.

It is recommended that RR/CC work with its partners to document the third-party licenses and not distribute these third-party licensed components. The build/compile documentation should be created for the users to get the third-party libraries independently. Documentation would also include how the user is able to compile the VSAP source code, along with the third-party libraries to obtain a functional system.

RR/CC will work with County Counsel to perform analysis on the existing third-party components utilized in VSAP source code. The output of this analysis should provide an accurate picture of which third-party components are to be distributed as part of the open code and which ones need to be referred in the documentation.





RR/CC will work with the development partners to ensure:

- ReadMe files are kept updated, along with the distribution packages.
- Appropriate licenses (for third parties) are kept separate in appropriate text files (for example, LICENSE-3RD-PARTY.txt).
- Links to the 3rd party libraries are provided in the documentation (online as well as distributed with the code).
- Appropriate credits are maintained in the documentation (where applicable).
- If a third-party component explicitly demands that distributions keep the file layout identical, re-arrange the code appropriately.

The timeline of this analysis is given below:

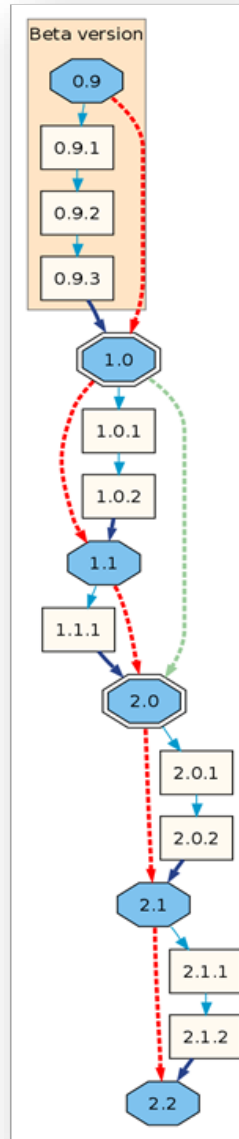
Timeline	10/21	2/22	12/22	2/23	4/23
<b>Deliberation on Preliminary, Conditional Open-Source Plan</b>					
Perform VSAP third-party components scan					
Review VSAP third-party component, usage, licensing compatibility, distribution options					
Governance Committee's final determination on reciprocity, permissiveness, etc.					
<b>Finalization of license</b>					

---

## 6. Certification and Regulatory Framework

---

RR/CC will work with SOS team to describe how the California Voting Systems Standards, testing and approval process will be impacted as a result of VSAP being open source and getting used by other jurisdictions. The certification and regulatory framework will need to be in place before any code is made open source (refer to the timeline in subsequent sections). Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.



## 7. Reciprocity

There are two (2) ways to encourage reciprocity from the code-contributors. The RR/CC will need to ensure that reciprocity is addressed in:

- Licensing
- Outreach to the code consumers (and contributors), educating them on the importance of enhancing the VSAP system for public good

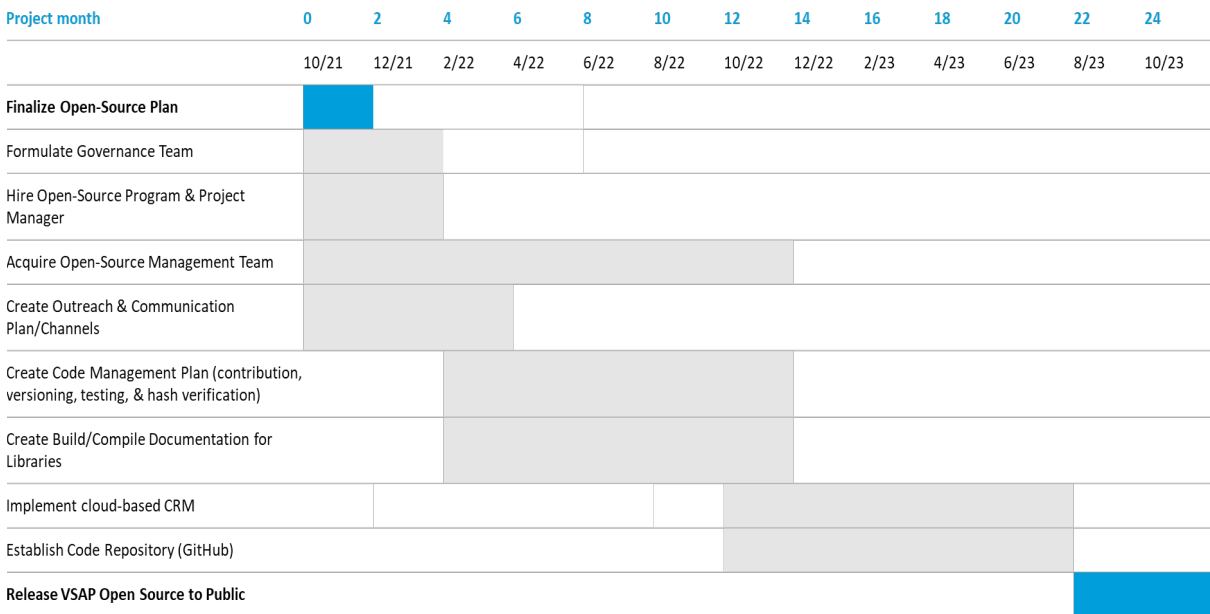
The RR/CC will work with County Counsel to explore options that can be incorporated as part of the licensing. Once licensing framework is established, appropriate parties will be notified of the plan to encourage reciprocity from the contributors.

As part of the Communication and Outreach Plan (as described in section 3.3 of this document), RR/CC will ensure that periodic, proactive communication is established with code contributors informing them of the importance of reciprocity. It is anticipated that with various proactive outreach and communications, the code contributors will be able to add their enhancements to the base source code.

## 8. Timeline

A representation of sequence of events and the order in which they need to occur is given below. This timeline is based on the information that is available to date and is therefore, an estimate. A more detailed, accurate timeline will be established once required information is received from governance authorities and once a dedicated Program/Project Manager is onboarded.

Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.



## 9. Recommended Next Steps

---

This plan is solely a recommended approach based on research and the input from industry experts. Adoption in whole or in part of any recommendations and approaches is conditioned on applicable regulatory guidance and authority by the SOS or other relevant governing body.

Furthermore, VSAP is currently approved for use by the County of Los Angeles only. Prior to the implementation of any of the recommendations found in this document, the SOS would need to determine the following:

- If the conditions of the VSAP testing and approval are transferrable to other jurisdictions;
- If jurisdictions are responsible for seeking their own approval, testing, and certification for any updates or modifications that may be made to the code;
- If the County of Los Angeles is liable for any issues that may arise from making the VSAP code available in an open source forum; and
- Any other pending items detailed within this document that would prevent the RR/CC from moving forward with the implementation of this plan.

## 10. Appendix A – Open Source Workgroup Roster

Following are the members of VSAP Open Source workgroup

Member	Email	Organization/Affiliation
Ben Adida	<a href="mailto:ben@voting.works">ben@voting.works</a>	VotingWorks
Brandii Grace	<a href="mailto:brandii.grace@live.com">brandii.grace@live.com</a>	Open Source Advocate
Chris Smith	<a href="mailto:cbsmith@gmail.com">cbsmith@gmail.com</a>	Open Source Industry Expert
Dave McGivney	<a href="mailto:mcgivney@gmail.com">mcgivney@gmail.com</a>	Social Sens.AI, Inc
Ginny Badanes	<a href="mailto:ginnyb@microsoft.com">ginnyb@microsoft.com</a>	Microsoft Corporation
Henry Balta	<a href="mailto:hbalta@cio.lacounty.gov">hbalta@cio.lacounty.gov</a>	LA County CIO Senior Associate
James Long	<a href="mailto:james.long@smartmatic.us">james.long@smartmatic.us</a>	Smartmatic Inc
James Vasile	<a href="mailto:james@opentechstrategies.com">james@opentechstrategies.com</a>	Open Tech Strategies
Jared Marcotte	<a href="mailto:jared@turnout.rocks">jared@turnout.rocks</a>	The Turnout
Mark Roden	<a href="mailto:mmroden@gmail.com">mmroden@gmail.com</a>	Tetra Bio Distributed
Maurice Turner	<a href="mailto:Maurice@MauriceTurner.com">Maurice@MauriceTurner.com</a>	Center for Democracy & Technology
Michael Owens	<a href="mailto:mowens@counsel.lacounty.gov">mowens@counsel.lacounty.gov</a>	County Counsel
Monica Childers	<a href="mailto:monica@voting.works">monica@voting.works</a>	VotingWorks
Sean Roberts	<a href="mailto:seanroberts66@gmail.com">seanroberts66@gmail.com</a>	Google
Steve Vo	<a href="mailto:vsvo@digitalfoundry.com">vsvo@digitalfoundry.com</a>	Digital Foundry Inc
Susan Lapsley	<a href="mailto:slapsley@sos.ca.gov">slapsley@sos.ca.gov</a>	Deputy Secretary of State
Robin Rowe	<a href="mailto:robinsrowe@gmail.com">robinsrowe@gmail.com</a>	Chair at OpenVoter
NaKeshia Robinson	<a href="mailto:Nakesha.robinson@sos.ca.gov">Nakesha.robinson@sos.ca.gov</a>	Secretary of State Senior Election Technology Analyst

## 11. Appendix B – Workgroup Meetings

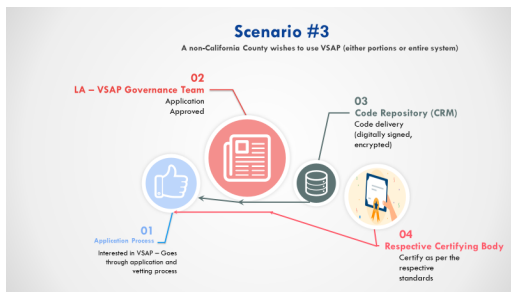
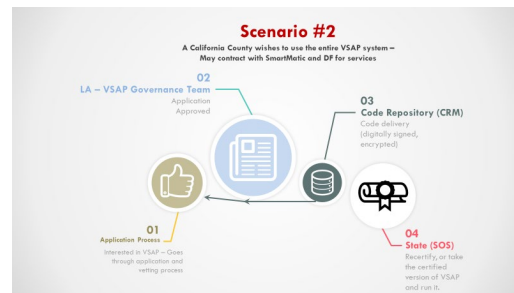
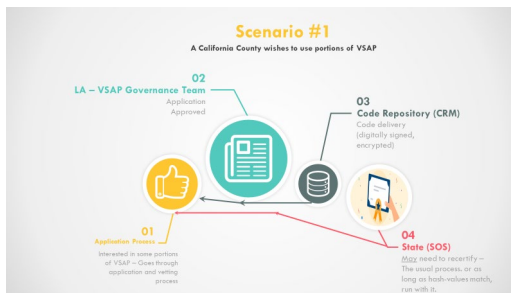
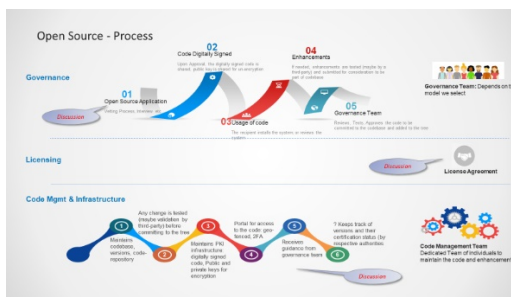
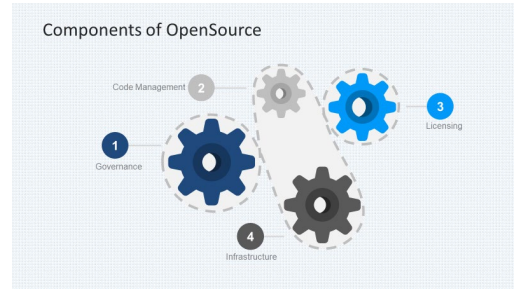
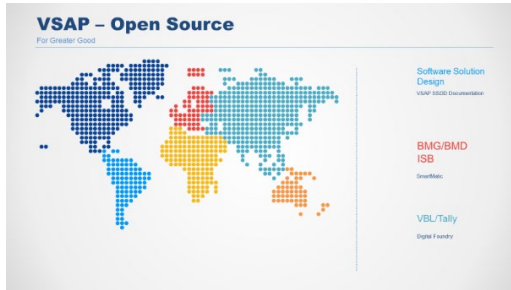
VSAP Open Source Workgroup meetings were conducted as per the following schedule.


- 2/26/21
- 4/29/21
- 5/25/21
- 6/29/21
- 9/16/21

# 12. Appendix C – Topics of Discussion

Following were the topics of discussion in the meetings with VSAP Open Source Workgroup.

## 12.1 February 2021





## VSAP Open Source

---

**Governance Model**  
**Regulatory Framework (State's involvement)**  
**Access to Codebase and Infrastructure**  
**Third-party Libraries**

LAVote.net

## Governance Model - Feedback

The governance model can be changed at any time. We should go ahead with whatever suits LA County needs and run with it. We can always change it at a later stage.

If blended, the models that best fit VSAP are Self-Appointing Council/Board and Foundation-Backed. Los Angeles County should have a permanent seat on the board while the other board members can be made up of a mixture of community/interest/advocacy groups and foundations. Foundations' inclusion on the board should come with a funding pledge as the board seat would allow them to have a direct influence over the project, but not overarching control given that it will/should be a diverse board. How the remaining members of the board are selected is a more difficult task.

LAVote.net

## Governance Model

---

- Start with Self-Appointing council
- Ensure that codebase is made open (as per the plan)
- Revisit after one year and adjust accordingly

LAVote.net

## Regulatory Framework - Feedback

Support from SOS Office (Advisory Committees, Modernization Board, Office of Cybersecurity) could help get ahead of potential regulatory issues regardless of the governance model selected

The state would ideally also have a seat on the board. If for some statutory/policy reason this is impossible, figuring out some way for the state to stay involved and help guide the process is key as it will ensure the board won't make decisions that may jeopardize any part of the certification process.

LAVote.net

## Regulatory Framework

---

- In addition to LA County, following should have a seat on the board (1 year term):
  - State (SOS)
  - Cybersecurity Experts
  - Members from this forum
- This will be finalized within two months (as part of the charter)

LAVote.net

## Access to Codebase - Feedback

Unrestricted. It's not truly Open Source otherwise.

Unrestricted. Agreed that it needs to be in order to be truly Open Source. Additionally, it would be administratively complex to limit access to a subset, not to mention most election jurisdictions are going to need outside help to execute anyways.

*Open Questions: If an active open-source development community starts to coalesce around the code base, my open questions are:*

1. How would licensing be handled (i.e., if someone committed to the repository, would they be formally be required to transfer rights to Los Angeles County, declare the development was done in the public domain, or some other scenario)?
2. Who would review the changes to the code if the change was specifically for a Los Angeles-based use case?
  - a. Conversely, if the change was for a use case not specific to Los Angeles, who would review the change?
3. Would changes to the code by unaffiliated third-party developers cause any issues with certification?

LAVote.net

## Access to Codebase

---

- Will be made public, unrestricted (TBD)
- To be hosted on infrastructure open to public (like GitHub)
- Discussion: do we need to know who is accessing the code?

LAVote.net

### Third-party Libraries

- Huge list of utilities that are used within the code that come with their own licensing (MIT, GPL, BSD, MPL etc)
- Replacing the libraries with a more favorable licenses is not practical at this stage
- Recommend that we make the VSAP codebase open and refer to the third-party libraries in the documentation; at least for now. To be revisited at a later stage

LAVote.net

### Licensing Model - Feedback

MIT Open-Source License

MIT for the license. You may also consider whether you want to trademark "VSAP" and whether you want to pursue branding guidelines - but those are unrelated to the license itself.

There are two dominant Open-Source licensing models: MIT and GPL. The MIT license lets anyone do anything with the code and disclaims any warranty. It releases the copyright, GPL (copyleft) blocks derivative commercial works. This license creates a legal headache for commercial companies incorporating Open-Source code into their codebase, because it makes everything it touches GPL. This is intended, that the GPL is trying to block commercialization by third parties. However, the cloud has made the point somewhat moot, as software is no longer sold but used within a cloud service. There are many other Open-Source licenses than the basic two. When these get selected it is because an organization has an unusual legal constraint to meet. See <https://opensource.org/licenses>

LAVote.net

### VSAP Open Source

- Third-party Libraries
- Licensing model
- Code Management
- Testing of Enhancements
- Digital Signing (Chain-of-Custody/Authentication)
- Reciprocity

LAVote.net

### Third-party Libraries - Feedback

Any libraries that have restrictive (undesirable) licenses form LAs perspective can (or should) be outside the source code being shared. That way, we only share the code with licenses that are aligned with what LA county desires.

If omitting third-party libraries renders VSAP inoperable and incomplete, then the mandate to be Open Source is not met. If the third-party libraries provide optional features (such as interfacing with proprietary hardware) and VSAP will operate without it, it may not be an issue to omit it. If third parties must be persuaded to Open Source their code for inclusion, they may prefer the GPL to block their code being used commercially elsewhere.

While it would be nice to supply a "batteries-included" solution to prospective developers, if there are applications or libraries that don't require bundling Governance Model, compiling with the source code, it makes sense to list those applications as external requirements—e.g., X version of PostgreSQL—rather than worry about them affecting the licensing process.

LAVote.net

### Licensing Model

- (SOS and LA County) Counsel may work on a favorable licensing model
- Based on an existing license

LAVote.net

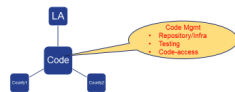
### Code Management- Feedback

Taking over an Open-Source project by a standards body or NGO is unusual because it is a cost that is difficult for a non-profit to shoulder. Examples are x.org, mozilla.org, linuxfoundation.org (I'm a committee member), aswf.io (I'm a committee member), kernel.org, blender.org and libreoffice.org. Standards bodies are ANSI in the USA and ISO internationally (I'm a committee member of ANSI/ISO). Neither manages code. There are no government regulatory agencies for software except for safety-critical systems such as transportation (DOT) and medical devices (FDA) software. EAC and DHS are government agencies involved in election security. The SOS is the state regulator.

LAVote.net

### Code Management

- Costs (resources + infrastructure + processes) are involved
- Need to decide who takes on this responsibility



LAVote.net

### Testing of Enhancements - Feedback

I feel that testing of new components should be the responsibility of the jurisdiction that is going to use the resulting system. That way, we keep things simple during the code management.

Everyone should test, contributor, VSAP team and indie Open-Source developers. The software should have unit tests included, so a lot of testing is automated.

Partnering with a university may be a way to outsource testing. I'm thinking that schools like MIT and Michigan are very likely to be interested in testing anyway. A CA school like USC would be preferable but not necessary.

LAVote.net

### Testing of Enhancements

- Every time code enhancement is committed to the codebase, it should be independently tested (costs?)
- Any jurisdiction using the VSAP code should get it certified (and tested as part of their certification)

LAVote.net

### Discussion

- Digital Signing (Chain-of-Custody/Authentication)
  - Is it necessary (or desirable)?
  - SOS and LA County to decide on this
- Reciprocity

LAVote.net



**VSAP Open Source**

Reciprocity  
Process of Sharing codebase  
Code Management

### Reciprocity

- Discussion

### Code Encryption (1)

- Developer requests certificate from a Certificate Authority (CA)
- Developer uses Code Signing Certificate for digital signing
- Once the digital signature is attached, a hash mark is made; hash value generated that can't be duplicated (irreversible method of cryptography)
- The content is published on GitHub
- Consumer downloads the software and uses a public key for decryption of the signature

### Code Mgmt (Version-control & Testing)

- Separate team, independent of RRCC IT operations
- Version-control
  - Who, What, When, Why
  - Tools – Github
  - Governance, Risk, Compliance or such regulatory requirements? Policy/regulatory framework
  - Procedures & code practices
- Testing
  - Automated builds
  - Code reviews
  - Test cases

LA County Registrar-Recorder/County Clerk - VSAP Codebase

Repositories

- VSAP Tally
- VSAP Ballot Layout (VRL)
- VSAP Ballot Procedures

### Sequence of Events

- Open-source workshop
- Create Governance, license, infrastructure, policies etc
- Create a plan
- Address all the Open-Source Topics and decision-making
  - It engaged until governance structure is in place
- Governance Body (not the existing workshop)
- Execution of plan and ongoing governance
- Term of this body – one year

RRCC VSAP Open-Source code management

### Immediate Next Steps

- LA County to formalize governance team/model
- Work with Counsel and start review of third-party libraries and associated licenses (eventually will form basis of licensing)
- Formulate a code-management team and work on the infrastructure (GitHub)
- LA County and SOS to start discussions on policy/governance
- Continue to work on the Open-source plan (completion by October 2021)

---END OF DOCUMENT---