Open Source Election
Technology Foundation
530 Lytton Avenue
2nd Floor
Palo Alto, CA 94301 USA

OSET FOUNDATION

TRUST THE VOTE PROJECT

20 April, 2015

John Arntz, Director of Elections
City and County of San Francisco
1 Dr. Carlton B. Goodlett Place
City Hall, Room 48
San Francisco, CA 94102

VIA EMAIL: sandro.burgos@sfgov.org

RE: Response to RFI# REG2015-01

Greetings John and Shane:

On behalf of the Board of Director/Trustees and all of us at the OSET Foundation, we greatly appreciate the opportunity to respond to your RFI as provided hereunder.

The Open Source Election Technology Foundation (www.osetfoundation.org) (hereinafter, "OSET") is a tax-exempt 501(c)(3) non-profit California public benefit corporation chartered with research, development and education on election technology innovation. Our flagship effort is the **TrustTheVote Project** (www.trustthevote.org), whose objective is to develop freely available, more verifiable, accurate, secure and transparent election technology. For more background, please see our web sites.

We have been in existence since November 2006 and are a team of 48 part and full time volunteers and paid staff with technology R&D centers in San Francisco, CA, Palo Alto, CA, Portland, OR, Boston, MA, Washington D.C., and Edinburgh Scotland, UK. We are funded by philanthropic gifts and grants including grant-making Foundations such as the Knight Foundation.

In the past 8-years and with assistance from over 200 elections professionals and officials, we have amassed a considerable amount of domain expertise in elections administration, processes, and technology.

Our technical and business teams come from the Silicon Valley and other technology centers around the U.S. and abroad and bring decades of commercial technology product development and life cycle experience. They come from companies including Apple, Facebook, Netscape, Google, Oracle, Sun Microsystems, and elsewhere—bringing a wealth of consumer digital innovation experience with them.

The OSET Foundation maintains relationships with dozens of elections experts and organizations. The technology work of the TrustTheVote Foundation is patent-pending intended for public ownership and is freely available under an OSI-compliant open source license. Much of the open source Election Technology Framework you will learn about in this response document is under design and development now, slated for incremental release over the next three-years.

Respectfully Submitted,

Gregory Miller
Co-Founder, Chief Development Officer

Part A: Summary Statements of Proposed System and References

1. Provide the organization's or firm's legal name and address.

Open Source Election Technology Foundation, **Inc**.
530 Lytton Avenue, 2nd Floor
Palo Alto, California 94301 USA

2. Provide the name, title, address, telephone number, and email address of the person(s) who will serve as the contact(s).

Gregory A. Miller
Chief Development Officer
Open Source Election Technology Foundation, Inc.
530 Lytton Avenue, 2nd Floor
Palo Alto, California 94301 USA

650.600.1450 Main | 503.703.5150 Direct | gmiller@osetfoundation.org

3. Provide a letter of introduction

Our letter is attached as cover to this RFI Response Document.

4. Provide a summary of the products and services offered, including annual license fees, annual support fees, and/or annual subscription fees. Include third party applications that are being recommended. List prices are acceptable.

The Open Source Election Technology ("OSET") Foundation is a non-profit, tax-exempt California public benefit corporation providing research, development, and education in the innovation of election technology for the administration and conduct of public elections. The flagship effort of the OSET Foundation is the TrustTheVote™ Project, which is developing a framework of election and voting technology that is:

- Comprised of open source software, subject to an OSI-compliant open source license (*with said license currently under review by the OSI for approval and endorsement at this writing*);
- Freely available for any jurisdiction to adopt, adapt, and deploy in whole or any part;
- Available without licensing fees, annual support fees, or annual subscription fees;
- For which there is no recommendation or requirement for any 3rd party application at this writing;
- For which the software (*only*) components to comprise a complete voting system, based on a paper ballot of record and supporting ranked choice voting are *not yet* finished in development, and have not yet been sufficiently tested to be ready for certification under state or Federal processes, but;
- For which said voting system is forecast under best estimates to be complete and ready for such certification in 28 months as of this writing (*i.e., the beginning of calendar year 2018*); and
- Where said development schedule will be accelerated should the balance of required philanthropic fund raising be completed sooner (*i.e., such accelerated schedule could enable completion as soon as the end of Q-2 2017*).

- Refer to the Appendix for a summary description of the open source Election Technology Framework ("ETF"). An accompanying visual walk-through of the framework is available at http://bit.ly/OSETosetf.

Hardware Integration

It must be emphasized that the TrustTheVote Project is delivering only the software components as described in the Appendix. A complete system for ballot casting and counting will require software and hardware integration. The Foundation's mission is to supply the software component for a system comprised of commodity off the shelf ("COTS") hardware. It is intended that commercial organizations (*"systems integration" vendors*) will conduct and manage the process of integration—in many cases including certain "adaptation" work to deliver a finished system that meets or exceeds the local regulations and requirements of the acquiring jurisdiction (*i.e., the City and County of San Francisco*).

Using the TrustTheVote Project's open source Election Technology Framework ("ETF") software, a jurisdiction (*county, city, etc.*) will either perform this work internally using its own I.T. resources, or contract with an outside services firm to do so. In short, this amounts to a "software-based solution" using commonly available hardware with some minor modifications and integration required, plus any adaptation work to tailor the basic software to the local needs of the jurisdiction. Importantly to this, neither the OSET Foundation nor any of its initiatives or projects, is not a commercial delivery organization. It is only a resource of freely available election administration and voting software technology.

Of equal importance is the recognition that the Foundation's mission includes rejuvenating the flagging industry for voting technology by lowering the barriers to entry of new delivery organizations and services providers, basing finished systems on this open source software technology. Such approach will lower costs and improve quality of technology as has been demonstrated in other segments of technology such as the commercial Internet.

5. Describe any election-related services that the organization or firm offers, including, without limitation, integration assistance, training, and ongoing support. Provide a rate structure or other costing information (i.e. hourly rate or pricing methodology) for the professional services offering. List prices are acceptable.

See generally, the discussion above in response to question #4. The OSET Foundation does provide a 3rd level of technical support for the software produced by the TrustTheVote Project (*wherein the 1st level of support is internal to the I.T. support team of the adopting jurisdiction, and the 2nd level of support is the systems integration vendor that delivers the finished system*). In this capacity, the Foundation serves to answer any technical question or resolve any technical issue that may arise in the course of adaptation, deployment, integration, training, and support. We believe that as adoption spreads, more software subject matter experts will emerge, producing an ever-growing community of support capability. However, the Foundation intends to maintain a Core Team of specialists closest to master source code.

6. Describe the different implementation approaches (i.e. big bang vs phased roll out) that the organization or firm can offer to the City to fully implement a particular solution. Include the benefits and/or risks of each.

A delivery organization should be able to provide either a phased roll out or single system "forklift upgrade." The ETF will support either (*but see discussion below*). Let us back-up and describe the software technology system approach the Project is taking in order to discuss a couple of key points we believe San Francisco must bear in mind as it proceeds.

The TrustTheVote Project systems architecture contemplates that portions of the ETF can be delivered as purely software—cloud hosted and served, without requirement for a hardware footprint. The components of the ETF to be made available as such comprise what we describe as "book ends" to the total system and include:

Pre-processing of Election Administration

- Voter registration and related services
- Ballot design, layout, generation, and distribution
- Election preparation and management

Post Processing of Election Administration

- Tabulation
- Participation, performance, and results reporting
- System analytics
- Election audit

The processes of ballot casting and counting are between those two bookends and require hardware footprints (*i.e., integrated hardware and the ETF open source software*). With that context, we turn to a couple of key points for consideration in the totality of this RFI response.

1. Adaptation work is condition precedent

First, we believe it is highly unlikely that any open source solution that becomes available and suitable for San Francisco will be sufficient on acquisition and without need for significant adaptation (*customization and tailoring*). We believe the focus of that work will be spread across two different aspects of a total election administration system: those elements that address administration, and those elements that address the actual process of casting and counting ballots in an election. With that in mind, four (4) key considerations:

1. The TrustTheVote Project ("Project") has either completed or is in the process of software development for the bookends (*components of pre- and post-processing listed above*)—many of which can be adopted, adapted, and deployed to meet or exceed specific requirements.
2. The Project is designing and developing a complete, fully functioning reference system which includes design assumptions and consensus requirements from our Stakeholder Community about elements of ballot layout (*digital and print*), usability aspects, multiple language support, and RCV support for typical state and federal election formats. However, these are important elements that will likely require specific adaptation work to meet or exceed the requirements of San Francisco city

and county. It is possible the TrustTheVote Project could direct its reference implementation to meet or exceed San Francisco's requirements. Said requirements could expedite (*or potentially, but unlikely hinder*) the development schedule.

3. The OSET Foundation already considers San Francisco elections officials to be among the "Stakeholders" in the work of the TrustTheVote Project, and as such, we would embrace, encourage, and welcome pro-active comment, contribution, and participation in the integration of San Francisco requirements in the work of the TrustTheVote Project as it is occurring. After all, this non-profit public benefit project makes its home in the Silicon Valley—backyard to the city and county of San Francisco, wherein the entire region is considered the technology innovation center of the world. It makes sense for San Francisco and the TrustTheVote Project to collaborate, regardless of the ultimate course or direction San Francisco ultimately chooses.
4. Regarding roll out as mentioned at the outset to this answer, not only is a phased roll out possible, so is a per component phased rollout. A big advantage of open source is that San Francisco can adopt on a "*try before we invest in solution*" basis, in order to test out individual components with low or no cost prior to production-grade deployments for scale, service level agreement, etc. This can even be done outside of the standard RFP process, because the RFP is for procuring the services for production-grade deployments. Exercising a "free trial system" is not just part of pre-evaluation, but can also provide input back to the open source development in progress. In fact, this is precisely how the TrustTheVote Project Stakeholder Community process works, as discussed earlier.

This segues to the 2nd and perhaps even more important consideration.

2. User-centered design is essential, but lacking

To date no commercial vendor as far as the Foundation's research can determine has every conducted the necessary user experience and user interface design work to ensure maximum usability for both administrators and voters alike. This includes usability testing, accessibility test for both paper and screen ballots that combine multi-language, and RCV alongside other ballot choice methods. The OSET Foundation considers this the single most important aspect in design next to ensuring the mandates of accuracy, verification, security, and transparency (*in process*). In fact, only Los Angeles City and County, CA and to a lesser extent, Travis County, TX has taken steps to ensure design for usability.

To be sure, we are fairly certain that not only are there no open source voting system alternatives finished and ready or will be ready within the next twelve months, we are very confident that no system alternative will have mandated and implemented design for usability, and here is why: no one system can serve all. This reinforces the first consideration of mandatory adaptation work. But it also discloses a huge opportunity for San Francisco.

The TrustTheVote Project is blessed to have a core interactive design team comprised of ex-Apple and Netscape designers—the same talent that responsible for many well known and wildly popular consumer interactive products and services. They are designing in a user-centric manner the reference system for the TrustTheVote Project including, but not limited to, all election administration services, ballot marking devices, ballot design and layout tools, poll books, and the user experience of counting and tabulation devices.

However, even in this work, absent pro-active engagement with San Francisco, the results will nevertheless only be referential. The results might be very close to what San Francisco wants to deliver, but adaptation will still likely be required. (*Incidentally, we're designing to make adaptation as easy as possible through configuration wizards and preference settings, but some highly localized tailoring may nevertheless be required.*) With that in mind, we strongly make two recommendations to San Francisco regardless of who or what is ultimately chosen for solution.

1. San Francisco should not simply decree by vendor contract that the solution provider (*commercial vendor*) make a "high quality balloting user experience," or "accessibility design," or any user interface attribute qualities as a condition of delivery. It has been demonstrated repeatedly in the sector of election technology that such is not the core competence of most software development organizations absent a robust interactive design team leadership. History has repeatedly shown that when faced with the high bar of requirements as contained in the San Francisco RFI, a "*best efforts to ensure contract value margin*" or a "*take it or leave it*" product presentation is the result. This means that usability is an after-thought of implementation, and not forethought of design. The practical reality is usually a product of time and cost to do so. (*Which, incidentally, is one of the compelling reasons for the cause and mission of the OSET Foundation and TrustTheVote Project.*) San Francisco can, as has Los Angeles, avoid this, do better, and set a new standard in design for usability, which is our next point.

2. We recommend that in advance of any procurement and perhaps during the RFP phase, San Francisco find ways to perform exploratory user experience design, and prototype testing, similar to what LA County has done. If San Francisco can end up producing a demonstrably usable, accessible, multi-language, RCV inclusive (*not restrictive*) ballot format specification as a result, then the technical side of implementing such will be extraordinarily straightforward and simple.[1] We provide some more discussion of this in response below.

7. Provide a brief description of the overall software and architectural design of applicable products.

A summary defining each component in the ETF is provided in the Appendix to this Response. The overall architecture is based on a well defined open data standards layer that is currently working its way through the approved standards making process with the **U.S. EAC** (*Elections Assistance Commission; www.eac.gov*); the **IEEE** (the Institute of Electrical and Electronics Engineers) and **NIST** (the National Institute of Standards and Technology). This architecture addresses the entire election ecosystem and is designed for adoption in whole or part; to be interoperable with existing systems;

[1] For the more technically conversant reader we offer this consideration in detail: While agile software development processes abound, especially in the open source world, and there are continuing discussions of the potential for so-called "agile design" our design and engineering teams both remain dubious at best. A formal design cycle utilizing a process of story development and coding sprints to rapidly prototype is ripe for runaway cost overruns and schedules. If, however, story development is informed by prototypes that have emerged from sound user-centered design cycles, the implementation of those stories is not a scrum or a sprint, but a straightforward implementation.

to be highly flexible and malleable (*i.e., "adaptable"*). It is the result of collaboration with over 200 election officials nationwide over a 6-year period and is now patent pending with a portfolio of domestic and foreign patents in prosecution—to be assigned to the public, in order to ensure public ownership in perpetuity. The ETF is currently under development, with portions complete, but much more to complete.

8. Describe the recommended operating environment(s) required to install and use any relevant systems and the minimum system requirements necessary to run such systems. Include any suggested production, development/test, and disaster recovery environments.

Referring back to main points in previous answers, about software-only components (i*.e., wherein no hardware is required to implement the particular capability or service*), these are designed for any of [**A**] in-house, [**B**] cloud, or [**C**] private/managed cloud deployments on complete standardized open source software "stacks" (*i.e., a compilation of software required to create a complete operating environment*). A robust environment will be comprised of separate production, and test/evaluation environments and systems, including a disaster recovery infrastructure via data restore to either fail-over or separate production systems. This is a well-tested operating environment commonly the architecture of many large-scale cloud-hosting facilities both commercial (*e.g., Amazon Web Services*) and governmental. We note here that a corporate sponsor of the Foundation, Amazon Web Services (*aka "AWS"*) has a well-developed government services division providing precisely this type of operating environment already to various state and federal agencies. The reader can learn more by visiting https://aws.amazon.com/ on the Web.

For hardware + software integrated systems (*i.e., those used in polling places for the casting and counting of ballots*), specific hardware system requirements will be determined by the desired system solution using a (to be developed) hardware configurator as part of the TrustTheVote Project, but importantly specific requirements can be dictated by San Francisco to the chosen commercial delivery organization.

9. Describe how the organization or firm envisions its software and hardware solutions changing over the next five to ten years.

The ETF is designed to enable innovations to occur at any subsystem level, down to the individual components. The intent is to ensure that any hardware advances deemed to be useful for the purposes of the ETF could be immediately leveraged on a sub-system basis without requiring regeneration of an entire monolithic system. Therefore, the ETF is intended to be an organic framework of technology continuously evolving and keeping pace with hardware technology advances.

How we envision that happening involves a growing community of individual, government, and corporate contributors to the code-base. The TrustTheVote Core Team will be responsible for incorporating submissions into the reference implementation and cataloging all contributions. The OSET Foundation envisions itself or a like-kind organization or well established and qualified academic institution sustaining the repository of open source election technology.

Accordingly, the OSET Foundation itself is envisioned to evolve from a highly focused technology research and development non-profit to a non-profit having a primary focus on enhancing, maintaining and sustaining the base repository with a more minor, but continuing effort in research and development of next generation technology. We believe commercial interests will emerge and rise up to contribute to that sustainment by value-adding software of their own to the Framework as well as growing a business of professional enterprise-grade technical services and support.

For an analogous worked example one only need to look to the phenomenon of the open source Linux operating system which runs 90% of the world's web servers today as part of the open source "stack" called LAMP. That open source project, emanating from the LINUX Foundation is served, supported, and sustained by participants like IBM Global Services and RedHat Inc.

10. If applicable, submit at least two (2) references of federal, state or local governments equal in size or larger than the City and County of San Francisco that have implemented the proposed system, or, a similar system, within the last five (5) years. Alternatively, for any organization or firm that is unable to provide references associated with the actual implementation and use of a voting system, indicate any pilot programs, testing, etc. that sufficiently support a proposed system being successfully implemented and operated.

While we cannot cite implementations at this juncture because as explained above we have not completed the system yet to a level sufficient to begin testing, we can refer San Francisco to several nationally respected technology thought-leaders, including a former United States Chief Technology Officer, **Aneesh Chopra**, who can opine on the approach, architecture, engineering, and technical efficacy.

Similarly we are capable of providing several state and local election officials who are familiar with the ETF for advice, comment. and input.

If desired, we can provide references for components of election administration that we already have in production, such as the Voter Services Portal in the **Commonwealth of Virginia**.

But to be sure, our response here is to profile an existing major design and development effort underway, philanthropically funded to be available at a future point. We do so confident that San Francisco's decision and determination of a path toward a next generation open source based voting system (*not to mention election administration services*) must be based on technology not yet in production anywhere because this is a leading-edge to address a growing national problem of expiring voting technology.

At this writing we are aware of only three major projects sufficiently down the path of design, engineering and development, and they are inter-related in that they share "architectural DNA" and each offers important elements of interest to San Francisco. However, two of these projects (*Travis County, TX and LA County, CA*) are municipal undertakings for which it is unclear and too early to understand how the results of their work would or could be widely distributed. There are several reasons for this:

A. A state (*let alone local*) government generally does not have "technology transfer offices" or infrastructure to facilitate such distribution.

B. State and local governments do not have the resources to provide (*technically or operationally*) the inevitable support necessary for an adopting jurisdiction.

C. The systems contemplated and/or under development in those two counties are designed to specifically fit within certain requirements and specifications for their local implementation and are not of a framework nature meaning that substantial modification would likely be required to implement the solution in another jurisdiction (*this is less so the case with Travis County's STARvote system*).

D. It is unlikely any thought has been given to a software licensing and distribution schema. While this is appurtenant to "**B**" *supra*, the distinction worth calling to San Francisco's attention here is the recognition that with these sources of technology (*and any source for that matter, including the TrustTheVote Project*) procurement requirements may compel examination of the terms and conditions of the open source license with which the technology is freely acquired. Contrary to some opinion n the open source community, this unique type of government technology requires certain terms and conditions to be addressed within the language of the license, which on its face suggests that an open source license like the GPL may not work for your procurement regulations. This is what gave rise to the preeminent open source software licensing attorney, **Heather Meeker**, working with the OSET Foundation to develop a license, the OSET Public Licenses (OPL), a derivative of the Mozilla Public License, uniquely tailored to these procurement issues. You can learn more by visiting www.osetfoundation.org/public-license.

   All of that said, there is one point of important distinction necessary here for San Francisco. The licensing issue is raised because [**A**] it will likely present an issue for either LA County or Travis County TX attempting to serve as a technology transfer agent for their work product, and this is one of the many reasons it is not a likely result or sound strategy to do so; and [**B**] it may present an issue for San Francisco procurement in any open source acquisition. However, here is where open source software subject to the OPL has an interesting "end around" option: the license offers the option for either [**A**] a non-governmental or academic entity not subject to the procurement issues contemplated by the OPL or [**B**] any government agency whose procurement is free of said issues to accept any source code developed in the TrustTheVote Project under the **GPL** license scheme. So, for San Francisco (*although early research suggests this is not the case*) if procurement regulations do not raise the objections giving rise to the OPL, all source code can then simply be accepted under the terms and conditions of the GPL. To learn about those issues we encourage the reader to visit the URL provided *supra*.

We note in closing on this 10th question and adding in relevant part to our answer for Question #9 above, that the OSET Foundation anticipates its open source repository of election technology will eventually include and incorporate software technology from LA County and Travis County, TX, as well as contributions from several other academic and commercial efforts to produce open source election technology. As such, the Foundation will be able to provide that distribution and support infrastructure for all of this technology—managing it as a public asset for critical democracy infrastructure.

Part B: Specific Criteria for New Voting System

Format Note: In the following sections for readability sake, the TrustTheVote project open source Election Technology Framework is abbreviated to "ETF." For the sake of brevity and no disrespect, the City and County of San Francisco are abbreviated to "SF." Likewise, the OSET Foundation, Inc. is abbreviated to "OSET."

1. Functionality

A. Approved by the Secretary of State for use in California before the City obtains the new system.

As ETF voting systems components are completed, OSET will submit them for test and certification in accordance with then current certification programs both Federally and in CA.

B. Designed for votes to be cast and tabulated using paper ballots.

Paper ballots are a fundamental element of the ETF, whether the ballots are marked by hand or by an accessible ballot-marking device ("*BMD*"), and whether marked in person or marked remotely and physically transported to election officials.

C. Designed so that all or part of the system's software operates using open source software.

All ETF components are open source software. All ETF components conform to emerging U.S. standard common data formats—and often serve as initial reference implementations of those standards. Should any ETF adopter wish to "mix and match" with commercial products (*such as SF's current Election Management System from DFM Associates*), interoperability can be achieved by data interchange based on common data formats. For existing products that do not support common data formats, ETF includes data conversion utilities from frequently used commercial formats to common data formats.

D. Assigns the least restrictive software license so that third parties may also utilize the code.

Any organization can use the source code of ETF components, so long as the organization is able to abide by the terms of the OPL open source license. ETF components are available under the OPL, which was specifically designed for ease of adoption by government organizations, without the restrictions of other open source licenses (*e.g. no venue for disputes*). Adopters may if they choose, convert their usage from the OPL to the widely used GPL. For more information, see the response to Part A, Question 10 (list item D).

E. Incorporates ranked-choice voting and allows for the formatting and tabulation of ballots that list the same number of selections as there are candidates, including qualified write-in candidates.

All relevant ETF components will support RCV with n-of-n selections: the Election Management System ("*EMS*") supporting specification of contests as RCV/n-of-n contests, and similarly supporting ballot composition and proofing of ballot specifications containing such contests; the ballot design studio supporting layout of ballots that include such contests; BMD displaying such contests, gathering voter choices for them, and producing a scan-ready paper ballot with those choices; ballot scanners supporting recording of voter selections for such contests as indicated on marked paper ballots; master tabulator aggregating tallies from ballot scanners, including election result computation using RCV algorithm, as well as externalizing tally data in a common data format, to facilitate independent RCV computation.

F. Accommodates the formatting of multiple-language ballots and is designed to integrate additional languages with minimal preparation of and modification to the overall system.

Flexible ballot format preparation is the purpose of the ETF Ballot Design Studio ("*BDS*"). BDS currently supports AIGA standard ballot layouts, and will be extended to include a template management feature in which distinct ballot templates can be created as re-usable variations of base templates. Multiple-language ballot layouts will be defined as extensions of AIGA ballot designs. Support for RCV/n-of-n contests will also be added.

However, we note that there has been very little if any rigorous usability/accessibility testing of RCV/n-of-n contest layout in ballots with multiple voting methods; and no testing that we are aware of that focuses on multiple-language ballots with RCV/n-of-n contests. We strongly recommend that SF, perhaps in collaboration with other elections organizations, conduct ballot layout usability and accessibility testing to create model ballot layouts for multiple-language ballots with RCV/n-of-n contests and contests with other voting methods.[2]

G. Requires the staging of one piece of equipment per precinct for each polling place and supports all voters.

In the architecture of the ETF, the accessible BMD is a separate component from ballot counting components. For equal protection of all voters, and the ability of risk limiting ballot audits to draw equally from of pool ballots from all voters, it is essential that accessible "voting machines" produce paper ballots that are processed thenceforth in the same manner as hand marked ballots.

It is essential that the BMD—which a voter interacts with privately and out of sight of elections staff—be a physically separate device from ballot counting devices that must remain under election staff control, and likewise separate from digital poll book or similar voter check-in devices.

Like all ETF components, the accessible BMD is capable of operating in a single-precinct election-day mode, or an "*all ballot styles supported*" mode.

SF can decide for itself: whether to use one or more BMDs per polling place of any kind; how many precinct count optical scan devices to use per polling place of any kind, including in a central-count model; whether and how many digital poll book

[2] **Note**: The OSET Foundation has started exploring the possibility of a grant-funded RCV ballot layout usability study. Support for such model ballot layouts are not a difficult matter technically, using DBS facilities for defining new basic templates. However, the definition of these model ballot layouts is no small task, and not one that is essentially a technical matter, but rather one of design and scientific testing of both general population usability, as well as accessibility by a variety of types of voters with needs for enhanced access.

units to use per polling place of any kind. That stated, ETF does *not* support a single piece of equipment that combines all the functions of digital poll book, accessible BMD, and optical scan ballot counter—for the reasons noted above.

H. Utilizes high-speed scanners to tabulate vote-by-mail ballots.

ETF includes a central-count configuration of the basic optical scan ballot counting software, plus central-count-specific functionality for alerting operators to ballots that require human interpretation or intervention. The software will be packaged to run in commodity computing hardware that supports integration with commodity high-speed scanners that produce digital images in common data formats. There is no need for tight software integration with specific scanner products; such integration deprives adopters of choice and could artificially raise hardware costs.

I. Creates a digital image of all (paper) ballots cast and facilitates the posting of the images on the Department's website while allowing for quick referencing between the paper ballot and its digital image.

All configurations of ETF ballot scanning software will create, for each ballot, an individual cast-vote-record of the ballot, including the digital images of the pages of the ballot. These records are essential for supporting risk-limiting ballot audits with the least constraint on the audit process. The digital images, with or without the cast-vote-record, can certainly be aggregated and published, where the aggregation occurs as part of the process of aggregating data from each counting device.

J. Meets or exceeds the most recent security standards set as minimum requirements for voting systems by the Election Assistance Commission and the California Secretary of State.

The Federal certification process will demonstrate the ETF meeting or exceeding these standards. Although we can only conjecture at the evolution of these standards over time (*notwithstanding our significant investment in participation*), we believe that exceeding security standards will be notably achieved by the use of the ETF "*Device Manager*" component, which prepares read-only boot media of immutable system images, for each ballot casting/counting device, for each election.

K. Allows for automated formatting of ballots with minimal manual manipulation of content by importing candidate information from the Department's existing election management system.

Such automated formatting is the primary purpose of the ETF's Ballot Design Studio, which is designed to be data driven (*from standard common data formats*) with the amount of human input controlled by selections of the BDS user. If the SF's existing EMS does not support the standard common data formats, the Department can use the data conversion features of the ETF election management components.

L. Includes auxiliary battery power to run polling place equipment for at least two hours of continuous use.

Any final system integrated with ETF software should provide this, of course. We refer the reader to our answer in Part A question 4 for more discussion on hardware.

M. Designed with minimal moving parts to reduce maintenance and associated costs of any mechanical operations.

Any final system integrated with ETF software should provide this, of course. We refer the reader to our answer in Part A question 4 for more discussion on hardware.

N. Includes clearly written documentation available before implementation for both hardware and software functions and provides instruction and reference materials for all system-related processes.

Documentation deliverables are an essential item for the bill-of-materials of every ETF component. Documentation is also not a closed process – as an open collaborative effort, the ETF work includes feedback from adopters of all kinds on quality factors of all kinds, including quality of documentation.

O. Permits the auditing of ballot cards at multiple points in the tabulation process and with minimal disturbance of operations to reduce the reliance on post-Election Day audits and to affirm the system is operating successfully.

Because each relevant ETF component provides maximum support for ballot audits (*see section "I" supra*), any ballot batch can be audited by comparison with cast-vote-records, at any time where a counting device's data acquisition can be performed. For example, an individual counting device's data for one day of early voting could be the basis for selection of a ballot batch for a ballot audit of the individual device. Similarly, when data is aggregated from any selection of devices (*any number of them large or small*), that data can be the basis for selecting a ballot batch that spans the full set of selected devices.

P. Produces easily customizable reports containing any audit data or other information collected by the system.

Reports based solely on ballot casting and counting operations are focused on generating results reports that are: at any level of granularity supported by the U.S. national standard common data format for election results reporting; selected in focus to any subset of a jurisdiction's reporting units and/or electoral jurisdictions.

Rendering of data in common formats allows for data to be unconstrained for access for reporting, data analytics, and data mining. Where the ETF reporting components do not provide some future needs of reporting, despite the parameterized nature of these reports, the Department can choose to extend the capabilities of the open-source software, or produce additional reports using a reporting engine of their choice, based on the common data formats.

Q. Logs all normal and abnormal events and ensures that event logging cannot be disabled or altered.

For the ETF components to cast and count ballots, event logging is an essential feature, *not* an "option." All device logs will be externalized in the U.S. standard common data format that is currently in process of development. Use of the common data format will allow log data to be unconstrained for access and data analytics and data mining.

R. Seamlessly supports risk limiting auditing of results by generating random samples, reconstructing electronic records for comparison, and handling statistics.

The ETF ballot counting components inherently support single-ballot audit, enabling SF to create risk-limiting audit ballot batches using a random sampling method of the Department's choice, not limited to some particular software's random sampling function. Externalizing cast-vote-records in common data formats enables the use of any method of randomly generating a ballot batch based on the total set of aggregated cast-vote-records, or any subset (*see section "O" supra*).

S. Facilitates the review of voted ballots or contests by election personnel using digital images to resolve issues when possible using a digital interface, and subsequently facilitates the posting of such actions on the Department's website.

This is inherently supported; see sections "I," "O," and "R" *supra*.

T. Allows for reporting results in near real time in such manner that does not require elections personnel to manually prepare and post results-related information.

The ETF aggregated tabulations component will produce reporting results (*including partial results during election night*) as often as SF elections staff choose to produce these results, yielding a reports dataset in the U.S. national standard common data format, with as much granularity as supported by the standard and selected by the Department. Each such dataset can be physically transported from the air-gapped tabulation component to any publicly-accessible server or site of the Department's choice, to comprise an up-to-the-moment data feed for use by any organization or system that is standard's compliant.

Should SF wish to provide the general public with web-based data visualizations and results reporting, the Department could choose to engage an I.T. services organization (*in-house or trusted outside service provider*) to operate an election-night instance of the ETF "VoteStream™" subsystem. For more information on the Knight Foundation funded VoteStream, please visit the URL: votestream.trustthevote.org.

U. Designed so that the Department can transport equipment using minimal resources and requires a small footprint inside delivery vehicles.

All ETF ballot casting and counting devices are open source software designed to run on commodity hardware. While SF could choose to use a default certified hardware configuration, SF could also choose smaller-size hardware components, subject to CA's then-current re-certification processes.

V. Allows elections personnel to set voting patterns when preparing logic and accuracy testing.

Generation of sample ballot datasets, with user-specified parameters, is a function of the ETF election management subsystem, which can be used in concert with SF's existing EMS, via the data integration described in section "W" *infra*. Conversion of sample ballot datasets to sample ballot PDFs is a function of the ETF Ballot Design Studio.

W. Operates in a manner that is compatible with the Department's existing election management system from DFM Associates.

The ETF election management subsystem may have desirable features that are not present in SF's DFM EMS. The TrustTheVote Project ETF EMS can be used as a parallel supplementary EMS, *without* abandoning DFM use. At any point during pre-election processes, an election definition can be exported from DFM in any of its supported data formats (*e.g., Dominion, ES&S*). ETF EMS includes data conversion components that transform these legacy formats into standard common data formats that are the basis for ETF EMS. In a simple linear process, DMF EMS would be used for basic election definition, followed by export conversion and import into TTV EMS components for ballot specification generation, ballot specification proofing, and export to the ETF Ballot Design Studio.

Even if no part of the ETF EMS is used, the data conversion components, can be used to bridge election definition data from DFM to the ETF Ballot Design Studio in order to format multiple-language ballots contained RCV/n-of-n contests.

X. Allows elections personnel to meet the pre-election testing requirements for automated reporting established by the California Secretary of State in such a manner that does not require manual results generation.

The ETF EMS subsystem does not have any specific pre-election reporting requirements. However, EMS election definition data can be shared with the ETF reporting components, which can be extended to support a state-specific pre-election report.

2. Usability/Transparency

A. Accessible to all voters to cast ballots in an independent and confidential manner.

This is the main function of the ETF accessible ballot marking device, when deployed in a polling place with physical privacy measures.

B. Provides fully accessible and intuitive features for all voters and includes connections and ports to fit all currently known types of assistive devices.

See item (1)(U) *supra* regarding the Department's control over physical hardware choices.

C. Promotes intuitive setup and operation of equipment in the polling places so that poll workers do not require specialized training on the equipment.

This will be addressed with usability testing of the human interfaces for device setup.

D. Indicates how the system tallied each vote on every ballot card and indicates if any votes were unreadable while ensuring the confidentiality of each voter's ballot.

See item (1)(S) *supra* regarding ballot-level cast-vote-records. Rather than recording any mark as unreadable, CVRs include meta-data about the confidence level of a mark or non-mark recording. In the case of central-count ballots, solicitation of operator input (*interpreting voter intent*) is a configurable feature.

E. Indicates any action taken for every ballot card or contest that elections personnel reviewed and generates a digital audit log for posting on the Department's website that records such actions.

The ETF counting components' action logs and CVRs address this very important requirement for accountability. Audit logs can be exported in a U.S. standard common data format, for publication as raw data. Reporting components will also be able to post-process the raw data.

F. Issues all result reports, ballot tally files, audit logs, in open data formats (machine-readable) and human- readable formats to increase the scope of election transparency.

All TTV ETF components conform to U.S. national standard common data formats; see item (1)(C) *supra*.

G. Creates and facilitates the posting of ballot image files on the Department's website so that members of the public can tabulate the same vote information that the Department uses when tallying the official results.

This is inherently supported; see item (1)(I) *supra*.

H. Collects and then converts the election information in a manner that facilitates the Department's ability to provide reports in data formats and styles requested from other agencies, the media, and members of the public.

The ETF tabulation functions support the U.S. national standard for election results reporting, and the VoteStream component can be used to publish the data in a variety of formats, including via a web services API.

3. Results Reports

A. Produces rapid, versatile, and easily customizable reports, including in real-time, when issuing results reports on Election Night.

See item (1)(F) *supra*, regarding election night reporting.

B. Provides easily customizable reports for a wide variety of purposes, including the reporting of partial election returns throughout Election Night, final unofficial election returns, and canvass reports.

See item (1)(F) above, regarding election result reporting.

C. Organizes and exports data in a variety of formats including but not limited to TXT (delimiter-separated), CSV, XLSX, PDF, and XML/EML that the Department can upload to its website and provide to the Secretary of State, the media, etc. with minimal intervention.

Results data export in standard common data formats is supported by the ETF subsystems and components for counting and tabulating. Post-processing into alternative data formats is a feature of separate reporting modules.

4. Adaptability

A. Anticipates the City modifying its use of the system or the system's components in response to changes in law such as the possible implementation of Senate Bill 450 that would allow the City to conduct mail-ballot elections with voting centers staged at multiple locations in the City.

All relevant ETF subsystems and components are designed to operate in any of traditional precinct mode, multi-precinct mode, or all-precinct mode. Mail-in ballots are inherently supported.

B. Implemented in the City under a possible final agreement that institutes a purchase, lease, lease-to-own, or any other mechanism that best suits the City's interests in obtaining a new system.

Any final system integrated with ETF software should provide this, of course. We refer the reader to our answer in Part A question 4 for more discussion on hardware.

Actually, purchase and lease are irrelevant to open source software. Integration with SF's selected hardware is a separate function with costs/terms defined entirely independently of the software. The TrustTheVote Project is one organization to provide the open source software, that's what our RFI response here addresses. SF would choose a vendor or vendors for HW integration, support, etc. base don an RFP. Said RFP might call out as a specification for solution the TrustTheVote Project ETF open source software.

C. Allows the City to obtain the new system and its components and also provides the City with the flexibility throughout the term of the agreement to upgrade components, including software, when improvements to the new system become available, including an option to fully replace the new system.

Any final system integrated with ETF software should provide this, of course. We refer the reader to our answer in Part A question 4 for more discussion on hardware.

Software upgrades of open source software do not by themselves effect cost, though the hardware integration vendor or support vendor(s) may have costs for performing upgrades. Upgrades to non-hardware-integrated software (e.g., EMS, BDS, etc.) might or might not have upgrade costs depending terms and conditions of services of support vendors. We would expect any cloud-hosted/provided services should benefit from dynamic upgrades depending on the subscription and service level agreement.

D. Allows the Department to continue to select how all voting-related services are obtained such as for ballot printing and translations without restrictions from the design of the new system.

The ETF subsystems for election management and ballot layout make no assumptions about ballot printing or translations, other than that printable ballots are printed from PDFs, and that translation data are provided during the election definition process. In the OSET Foundation's related work on voter registration, we have amply demonstrated the technical framework for translators to provide their input during the process of defining paper forms and online experience.

APPENDIX—ETF OVERVIEW

The TTV ETF consists of 16 separate components, each of which can be used separately, can interact with other ETF components via standard CDFs, and can interoperate with elements of other election administration systems or voting systems that also support the standard CDFs.

The entire ETF architecture -- components, interaction via CDFs, and workflow -- is documented in a large comprehensive architecture diagram with a more detail an annotated guide document that is available at bit.ly/OSETosetf. Please use that visual illustration to accompany this description below.

Please note that the data layer segment of the architecture has not yet been updated to reflect the most recent CDF standards activities at NIST and EAC.

In this section of our RFI response, we provide the brief summary requested by RFI items A(4) and A(7). This summary will provide some context for the point by point responses to the RFI's Section B.

1. **Election Data Manager** is an EMS component for data management pertinent to election definition, with a workflow that spans the creation of a new election through proofing of ballot specifications that will then be the focus for later ballot layout. For the Department, this component may be useful alongside the existing commercial EMS (more details in responses to section B), but even if none of its features are necessary, one of its modules can be used to convert election definition and ballot definition data into a standard EDS for later ballot layout.
2. **Ballot Design Studio** is the ETF component for layout ballot, based on a dataset in a CDF that provides a proofed set of ballot specifications for each ballot style in an election. As a proven multi-language and CRV/n-of-n ballot format is developed (see response to B.f below), the BDS's template set will be extended to support that format so that ballot layout is largely automatic.
3. **Device Manager** is the ETF component that prepares ballot casting and counting devices for a particular election, consuming data from election definition and ballot layout, and creating a boot image for each casting/counting device. This approch of using boot images is a critical part of the system integrity architecture: each device's software is completely encapsulated, able to be validated at device start time, but not present (and available for tampering) when the device is not in use.
4. **Accessible Ballot Marker** is ETF's ballot marking device, intended for compliance with HAVA requirements for independent voting by all voters. It receives ballot definitions in a CDF, either TTV's EMS or any other standards compliant EMS. The Department can use data translation functions to convert DFM ballot definition data to a standard CDF. The ABM produces a paper ballot of record for the voter to cast; these are in a "selected choices only" format, for compactness, fewer sheets, and ease of audio playback. In the case of RCV contests, every candidate would by included with that candidate's rank.
5. The **Precinct Ballot Counter** is the ETF's PCOS device, with the ability to scan both ABM produced ballots and hand marked pre-printed ballots. Ballot definition data are part of the boot image created by the Device Manager. Output is tally

datasets for vote totals; cast-vote-records for each ballot specifying which votes were collected; a ballot image for each cast-vote-record. For RCV contests, the tallies are tallies of binary votes, but tallies for each candidate for each rank.

6. The **Central Ballot Counter** is the ETF's CCOS device. Ballot definition data are part of the boot image created by the Device Manager. It differs from PCOS in that the ballot images are deliver to the software by integration with a high speed scanner; and that there is a human operator to be notified of ambiguous marks, where the human operator can usea human interface to resolve ambiguities. This "interpretation of voter intent" activity is an essential part of the event logging of this component. Besides this type of additional log data, the output data central count is essentially the same as precinct count.

7. **Tabulator** is the ETF component that aggregates tally datasets from ballot counting devices, consolidates them, totals them. Output is: complete consolidated input data (nothing is omitted), complete consolidated CVRs, vote totals (including rank totals for RCV contests), and -- where applicable -- contest results (in the case of contests entirely within the elecoral jurisdiction). First-past-post contest results are simple totals. RCV contest results result from computing RCV algorithms. We note that the current standard election tally CDFs could do better to support RCV. As the standards evolve, Tabulator implementation will be updated.

8. **Analytics** is an ETF component that treats election results, participatioon data, and other data as read-only inputs, and generates reports. Because all the data is in one or more CDFs, Analytics has a fixed data schema, and an extensible report definition approach where new reports can be specified in terms of the underlying data schema. In some cases, the Department's needs could be met by existing parameterized reports. In other cases, additional reports will need to be defined.

These 8 components can be adopted separately or in toto. If in toto, these 8 should suffice to meet most or all of the requirements described in the RFI; details are provided below.

However, the ETF also contains other components for registration, several voter services, voter check in (digital poll book), voting place check in troubleshooting, and more. While some of these may be useful to the Department in a new voting system deployment, they do not directly address RFI requirement. Hence, in the interest of the resulting brevity, we refer to the annotated guide document for more information.