|galois|

Response to STAR-Vote Request for Information

RFI #1505-003-LC

Dr. Joe Kiniry, Galois, Inc., kiniry@galois.com

1 Introduction

Galois has a fifteen-year proven track record of solving the most complex challenges of the most demanding federal and commercial customers. Our bespoke software products are internationally recognized as being some of the best technology in the world of high-assurance software systems. Consequently, we intend to fundamentally change the nature of elections systems design, development, and support and put the power back in the hands of the voting public. This RFI is an exciting opportunity for us because it asks for a system that does just that.

We are impressed with the quality and detail of the specification in this document. It represents a complete, verifiable system that will be a large step forward in election technology. The system specified in the RFI will be straightforward to implement as specified; we are confident that we have the capability to meet all of the requirements laid out in the document. The modular structure presented has been very useful in the discussion of how such a project will be implemented, and has helped us in giving a meaningful response to this RFI.

2 About Galois

Galois is a privately held U.S.-owned and -operated company established in 1999 in Portland, Oregon; our mission is to provide trustworthiness in critical systems. We were founded on core principles that focus on innovation, authenticity, and deep trust, and we live those principles every day in interactions with clients and among ourselves.

Galois specializes in the research and development of new technologies that solve the most difficult problems in computer science. We are passionate about the trustworthiness of critical systems, and work to ensure that the systems you depend on work as intended, and only as intended. Our team works closely with clients to achieve a balance between the privacy/cost/speed challenges involved in making systems more trustworthy. In general, every one of our projects involves the creation of bespoke high-assurance technology specific to that client's needs on aggressive timelines at reasonable cost.

Galois has won and successfully executed on dozens of multi-year, multi-million dollar R&D projects for numerous federal agencies including the Department of Defense, the Department of Homeland Security, Defense Advanced Research Projects Agency (DARPA), Department of Energy, NASA, and members of the Intelligence Community. We care deeply about real-world use of our R&D efforts and work diligently to transition them into use.

Galois's projects focus on the science and engineering necessary to design, develop, and support high-assurance systems. High-assurance systems are systems that must not fail, because in failing they will cause a loss of life or a severe fiscal or operational loss. Standard high-assurance systems are, for example, avionics, aeronautics, transportation, nuclear power, biomedical, financial, and cyber-physical systems. Galois experts have worked on all of these.

Early in Galois's existence, we recognized that democracy should be treated as a high-assurance system, so we have had a long-term interest in developing technology for elections. With the

arrival of internationally known digital elections expert Dr. Joseph Kiniry in January 2014, we decided to pursue business in the space of elections technology.

Prior to working for Galois, Dr. Kiniry provided commercial and public consultancy services to several governments on matters relating to elections, their technology, security, processes, and verifiability. His experience in the area of elections is both from the perspective of a public employee (as he was a professor of computer science and mathematics at multiple universities for approximately twelve years) and as a scientist-activist. He has worked on election systems for thirteen years; has audited the security, correctness, and reliability of numerous physical and internet-based voting systems; has developed high-assurance prototypes and products of several election technologies (including, but not limited to, tallying, auditing, voting, ballot marking, and electronic poll book systems); and sits on the Board of Advisors of the main verifiable elections nonprofit, Verified Voting.

Dr. Kiniry has formally advised three governments (The Netherlands, the Republic of Ireland, and Denmark) on matters relating to digital elections and has testified before two parliaments. He has also provided informal input and advice to the governments of Norway, Estonia, and the U.S.A.

He co-ran a multi-year research project on digital elections (called the DemTech project) and has supervised numerous BSc, MSc, and PhD theses focusing on elections technologies. His research group has developed several high-assurance peer-reviewed election software systems, including a tally system used in binding European elections for The Netherlands and an electronic poll book system meant to be used in Danish national elections.

Dr. Kiniry also regularly interacts with and provides input to federal agencies related to elections including the EAC, NIST, and FVAP, and several elections-related non-profits including the OSET Foundation, Common Cause, the Overseas Vote Foundation, and U.S. Vote.

Ashish Puri is working with Galois as the lead strategist in the state and local public sector space focused primarily on elections, and shares Dr. Kiniry's drive and passion for improving the elections space. Mr. Puri has worked in U.S. public sector delivery for over fourteen years and has been involved with the U.S. e-government elections vertical for twelve years, holding several key client and delivery roles for Hewlett Packard and its subsidiary companies.

Mr. Puri was the lead elections SME for 9 statewide VREMS projects for HAVA compliance in various roles in analysis, design, product development, project, portfolio and practice management, culminating in his leadership of the Elections practice servicing 13 U.S. states at Saber Corp, a company that was acquired by EDS/HP.

Mr. Puri brings deep understanding of the development and implementation of elections products.In addition to elections, Mr. Puri has managed key projects and practices in the areas of motor vehicles, child support, telecom, insurance, and health and human services.

While Galois has not executed on election systems to date, we have successfully developed many systems that have many of the same challenges (correctness, security, usability, accessibility, etc.) and technologies (operating systems, programming languages, distributed systems, cryptography, etc.). Thus, we are well positioned to bring the assurance one sees in other safety- and mission-critical high-assurance systems to the elections systems and services market, at low cost and with publicly owned open source technology on COTS hardware.

For the past year, we have been developing prototype technologies in this space. We intend to spin out a class B corporation, called *Verifiable Elections*, whose mission is to bring open source, high-assurance, end-to-end verifiable elections to the world. This new company will be a Galois-branded entity and will retain much of the personality, history, technology, and performers of Galois. Consequently, *Verifiable Elections* will be the bidding vehicle for a STAR-Vote RFP.

In general, Galois's work, reputation, and way of doing business—based upon trustworthiness, authenticity, and transparency—means that virtually all our customers become repeat customers. Consequently, we are happy to introduce any potential client to any existing or past client as a referral.

3 Travis County Resource Requirements

In order to ensure project success, the county needs to provide resources and engagement in areas that align with the standard operating practice of the RFP awardee, as well as take into account the IP variant chosen by Travis County (see below). We summarize three variants for resources and project management: **Traditional**, **Foundation**, and **Lean**.

3.1 Traditional

For a typical large e-government IP project, the key areas of resources and engagement required from the client are shown below. These do not replace the project management and delivery responsibility of the vendor, but provide additional oversight for overall project status and receipt of deliverables.

- *Project Governance*: Provide oversight on project management and address escalations.
- *Project Management*: Provide county planning inputs, review and approve overall project plan. Coordinate review and approval of deliverables. Establish project escalation and decision-making procedures, deliverable expectations, review and approval process. Establish vendor-independent quality assurance process, including a red team.
- *End-user Engagement:* Participate in requirements verification and review, functional testing and acceptance, data validation and user training.
- *Technical Oversight:* Define and verify technical/system requirements, review and approve system design and architecture. Lead system acceptance testing.
- *Quality Assurance:* Approve all processes and deliverables as committed in the project plan.
- *Vendor Liaison:* Assist with interactions among the new and existing vendors.
- *Operations Team:* Manage and operate the system after delivery.

3.2 Foundation

If a Foundation (see IP variant #1 below) is running the project, rather than Travis County itself, and if we base a management model on leaner best practices therein, then the areas, roles, and responsibilities are different.

In particular, there is a greater distribution of responsibilities across actors who rarely have a single area or role, and self-organized community input becomes the primary means by which assessments are made about goals, large and small. Coordination happens almost entirely

asynchronously, typically using collaboration technology platforms fit-for-purpose such as GitHub, Slack, SalesForce, etc.

As such, it is not uncommon for large Foundation-driven development initiatives to have project management overhead ratios of 10:1 (one FTE PM-like role per ten developers).

3.3 Lean

At Galois we typically run an even leaner ship when it comes to project management and customer caretaking. We can be lean because our research engineers are all 10x programmers, most of whom have PhDs, and because of our focus on trust and transparency in all business and technology.

For example, we use a model for service guarantees and operational support that is atypical because our systems are high-assurance and formally verified. Instead of a tiered support system, we provide a comprehensive support solution that emphasizes transparency about the product and its capabilities and direct access to the team responsible for the product.

For our traditional projects, customers have direct telephone and email access to the project lead, direct access to the project's ticket system, and direct visibility into the development repository of the project. Support tickets filed into the system are typically triaged by team members within minutes of being filed, responses to issues are immediate, and fixes are prioritized based on conversations between the customer and the development team.

For field support during deployment and system use, we augment operational support with a front-line team who can provide basic support to election officials and volunteers. We plan to provide support of this kind via a toll-free number, an online text chat interface, or both.

As such, our project management overhead ratios are comparable to those of a Foundation development effort (10:1), yet we need fewer resources to accomplish the same technical goals (due to the 10x programmer/scientists).

3.4 The Traditional, Foundation, or Lean Choice

The choice among these management styles will obviously have a direct impact on personnel requirements on Travis County's side of the project, which impacts finances, project velocity, and many other factors. Perhaps more importantly, the choice fundamentally impacts social and psychological aspects of management.

Working within a **Foundation** or **Lean** management method is fundamentally different than a **Traditional** method. Roles and responsibilities are more fluid, there is enormously less turf-fighting, actors have expertise and obligations, but development artifacts are typically egoless insofar as anyone with the requisite skills can contribute, correct, and criticize.

Not every organization is able or willing to adopt such an alternative scheme, despite it being shown to work well in numerous other contexts. As such, we simply state that we are flexible with regard to project management technique, as we have direct experience with all of these variants and more.

4 System and Software Information

Our systems engineering process and methodology is akin to what is used in other organizations that focus on safety- and mission-critical systems. We have a peer-reviewed software development process that results in systems that typically run correctly the first time and have very few bugs. Generally speaking, we use a correctness-by-construction methodology, where each stage of the development process is coupled to the adjacent stages by underlying technologies that guarantee correctness. The artifacts created in this process include, but are not limited to: formal domain models, formal requirements, scenarios of use, system events, formal static and dynamic models of the system, formal contracts of software modules, formal specifications of protocols, hand-written and automatically generated test benches, automatically generated evidence of non-functional properties (such as reliability and scalability), and formal proofs of correctness for protocol and system correctness properties.

Our systems development approach is a combination of adaptability, rapid and frequent prototyping, and continuous evaluation to help mitigate the risk inherent in such efforts. Each team will maintain separate code repositories under a distributed version control system such as Git. In addition to the code base these repositories will contain the full set of development artifacts described above, as well as unit, performance, and integrated functional test suites for each subsystem and for the system as a whole. Servers located at Galois will pull from these repositories and perform automated builds and testing whenever code is updated. In addition to standard functional tests we will place a particular emphasis on performance tests, which allow us to measure the overhead introduced by various diversity, detection, and recovery techniques. This test suite will help ensure that feature changes do not impact performance.

Development will follow a *continuous integration, continuous deployment* approach so that each code change is tested automatically, and as early as can be done, to catch defects as early as possible. Continuous integration is a proven way to reduce development cost and improve productivity. In continuous deployment, code that has passed integration testing is pushed automatically from the main development branch to a deployment staging area that allows all project personnel to access and test the latest working code in a whole-system context.

Development-related team communication will be facilitated by a wiki that is accessible to and editable by all team members. This wiki will also serve as the reporting facility for team metrics, and the home of software documentation during development activities. We will use metrics such as test suite pass rate, defect escape rate, and selected others to monitor code health, adapting our test suites and design review processes to meet goal lines for each metric.

5 Project Management and Communication

5.1 Project Approach and Timeline

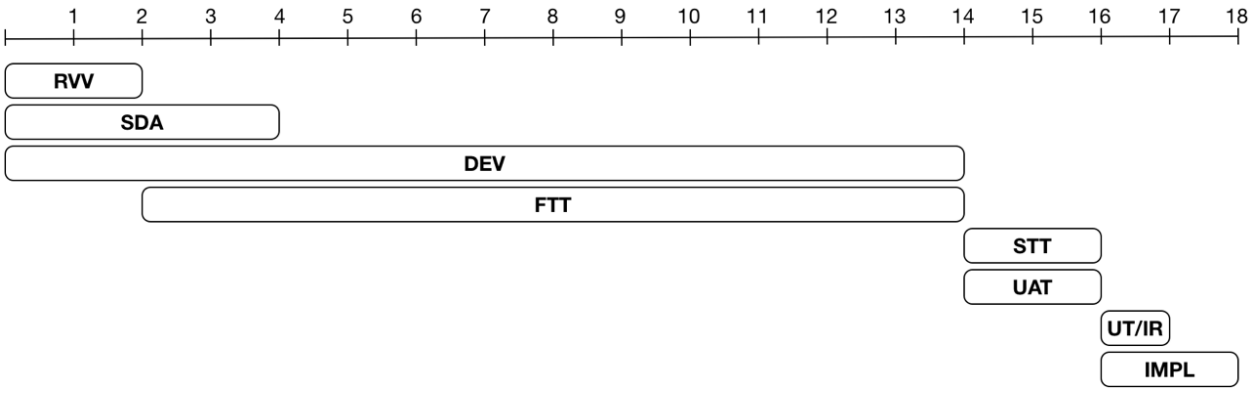
Galois can produce a system that fulfills the functional and technical specifications that appear in this RFI. We have conducted a thorough analysis of all requirements and feel that a 15–18 month time frame is sufficient to build and implement this system. Our financial analysis indicates that the intellectual property (IP) requirements outlined in the RFI will substantially increase the price of the system. With less restrictive IP requirements, the vendor would have the ability to leverage the significant amount of development work done on this project for their benefit and across their other projects/clients. Under those circumstances, we estimate the system could be

implemented at a cost of $7.5–$9.5 MM based on the listed RFI requirements. Without the accommodations to the IP requirements, the cost would need to be much higher to compensate for the additional development costs that would not be recovered with revenue from other clients.

Since we understand the technical requirements of this system fairly well and already have a working prototype based on the original STAR-Vote specification, we can work on the technical design and architecture right from project kick-off in parallel with the requirements verification and validation phase rather than adopting a waterfall approach. That allows us to compress the schedule by at least 3–4 months.

Here is our planned approach to executing this project with the phases and duration:

Phase Name		**Schedule**
RVV:	Requirements Verification & Validation	Months 1–2
SDA:	System Design and Architecture	Months 1–4
DEV:	System Development	Months 1–14
FTT:	Functional and Technical Testing	Months 3–14
SIT:	System and Integration Testing	Months 15–16
UAT:	User Acceptance Testing	Months 15–16
UT/IR:	User Training and Implementation Readiness	Months 17
IMPL:	Implementation	Months 17–18



5.2 Project Management and Tracking

We will follow an iterative approach, generating incremental releases on a periodic basis for both ongoing internal testing of functionality and technical and unit testing of the software. We will also periodically release builds for user testing by the core user group that initially helped in the requirements validation phase and incorporate user feedback in ongoing development.

To ensure that the project stays on track we will use comprehensive PMI-driven project management practices. This encompasses schedule and budget management, nimble resource management, active user involvement during product development, and periodic reporting of project status and the underlying deliverables vis-a-vis the baseline schedule to the project team, red team and the steering committee. Our project governance model ensures that risks are identified proactively and issues escalated for client decisions to minimize negative project impacts in terms of schedule, budget and system delivery.

5.3 Quality Assurance

Quality assurance is an integral part of the Galois development approach. The vast majority of the software tests are written within the code itself, rather than being developed separately. That leads to a software product built with quality inherent in its foundation, rather than with defects to be detected and fixed later. In addition to this pervasive testing, quality assurance is achieved through strict configuration management of the code as well as the systems and documentation produced.

The specific peer-reviewed methodology we use is a variant of Design by Contract with some aspects of a Correctness by Construction approach. Our process, method, tools and technologies span several deployment and development platforms, specification and programming languages, and communication and coordination schemes.

We are committed to providing defect-free, high-assurance solutions to our customers. We provide a lifetime warranty on our software and fix any defects discovered for free and in a timely fashion, rather than limiting such support to a particular time period under a maintenance contract.

5.4 Managing Communication

At Galois, we view proactive communication with stakeholders as the foundation for project success. We achieve this through a mix of periodic status meetings with the project team and the steering committee as well as written status reports. As part of our project management practices, Galois provides standard project status reports on a periodic basis; typically, these are monthly or weekly status reports, per the client's needs. These reports cover the following information of interest to various stakeholders:

- work accomplished in the past reporting period,
- work planned for the upcoming reporting period,
- project budget, spend, backlog, and other financial information,
- status of deliverables and milestones,
- project risks, issues, and action items,
- schedule and resource tracking against the project plan, and
- performance metrics as agreed with the client.

In addition, Galois assigns a *client caretaker* for each project, separate from the project lead and other project developers, whose responsibility is to ensure that any client concerns are addressed in a timely fashion. As previously mentioned, Galois also offers transparency in the development process; this includes full visibility into the project's software repositories, ticket system, and other development artifacts throughout the duration of the project.

5.5 Managing Change

Scope and change management is essential to project success. As part of our project management approach, the following actions are integrated into our project execution methodology:

- Establish initial Statement of Work (SOW) from the RFP with requirements classified as business, functional, technical and system requirements
- Establish a core user/SME group with a comprehensive stakeholder representation at project start that owns requirements and related decisions.
- Using the SOW, conduct a requirements verification and validation with the core user group as the first phase of the project to establish a baseline software requirements specification (SRS).
- Establish a requirements traceability matrix with the final set of verified requirements.
- Develop a system corresponding to the RTM and establish traceability of the software to the requirements and the testing deliverables. Our development methodology inherently provides such traceability through the artifacts it produces.

On an ongoing basis, the project manager receives any requests for requirement changes from the various stakeholders. The items are passed through the change control process as follows:

- New/changed requirements are first verified as being relevant and of priority by the core user group.
- A change request that outlines the impact of the change on resources, time and cost is prepared by the PM.
- This request is provided to the change control body on the project, which makes a decision on whether, when and how the requirement should be met.
- A change order is then passed clearly establishing the path of achieving the change.
- The project plan and other relevant project documents are updated in accordance with the change order.

This process is facilitated by a greater or lesser degree—depending upon the project management style chosen by the client, as discussed above—by technology. Consequently, in a **Traditional** management scheme this workflow may involve email messages, scheduled meetings, and several full-time actors. Typically at Galois in a **Lean** management scheme everything is captured, discussed, traced, decided upon, and executed in an ultra-lightweight style using GitHub and ad hoc stand-up discussions amongst developers and stakeholders.

5.6 Project Governance

As mentioned earlier, key stakeholder communication is central to our project governance practices. The red team is one of the key identified stakeholders, as are the core user team, project sponsor, steering committee and the county project management team. Stakeholders have an added interest in tracking day to day project activities, deliverables and milestones and ensuring project success. During the initial project planning phase, the expectations of the red team will be identified and and a mutually agreeable approach will be established for satisfying these expectations.

We believe that the red team shares a common goal with us in achieving project success, and we will therefore collaborate with them in achieving that goal. In addition to the periodic meetings with the project team and steering committee, we recommend either adding the red team to the steering committee meeting agenda or adding a separate periodic meeting with them. Additionally, they would provide all status reports that are produced.

The Galois development approach emphasizes transparency about the product and its capabilities as well as direct client access to the team responsible for the product. We would extend that access to the red team through demos, code walkthroughs and deliverable reviews. Since quality assurance is integrated into our system development, adding the red team to the process is a natural way to help them achieve their goal of ensuring project progress and success.

5.7 Risk Management

At Galois, we pride ourselves in our risk management expertise. Identifying risks early, estimating risk impact, and implementing risk mitigation measures in time to ensure risks do not become significant issues drive our success on projects. We believe in a partnership approach and therefore operate with complete transparency with our clients when dealing with risks. This allows our project managers to work actively with the client leadership team in effectively dealing with these situations and minimizing the impact of unforeseen events. In situations where the risk inevitably gives rise to one or more issues, we follow a proper change control process to minimize negative impacts as outlined above.

6 Partnering Structures & Intellectual Property

Galois shares the goals detailed in this RFI: protecting the integrity of STAR-Vote; reducing the need for Travis County to solely cover the cost of developing this system; providing a structure that encourages system and software openness; development of new ideas and analyzed improvement; and offering STAR-Vote to other entities at a low cost. However, we believe that the client's full and exclusive ownership of the STAR-Vote system is not only unnecessary to achieve the goals, but will also impact the number and quality of proposals submitted in response to a full RFP.

6.1 Business Partners for STAR-Vote

Galois's in-house expertise aligns with most, but not all, of the capabilities required to build the system described in this RFI. While we are a world leader in applied cryptography, high-assurance engineering, and secure systems, we do not have much in-house expertise in UI design and UX evaluation. Moreover, we often use external partners to facilitate internal red-teaming, in order to avoid group-think.

As such, we expect that we can leverage several outstanding relationships to fulfill these needs, including our relationships with Whitney Quesenbery at the Center for Civic Design, Morgan Miller at Experience Lab, Prof. J. Alex Halderman and his team at the University of Michigan, and a huge range of other scientist-activists and companies in the areas relevant to end-to-end verifiable elections technology.

6.2 Intellectual Property Reflections

The intellectual property (IP) regime Travis County advocates for in a full RFP will be a key factor in determining how many parties will submit a proposal. The requirements needed to realize the lofty end-to-end verifiability and assurance goals of the STAR-Vote system will require rare, highly-skilled technical expertise that is not available in standard IT consultancies, web development shops, and existing election vendors. Firms of this nature rarely engage in contracts where their services are provided as "work-for-hire" with full transfer of IP rights to the client, especially when significant research work is involved in the effort. If Travis County requires such a transfer of rights, many of the qualified firms may opt not to respond.

For those that choose to submit a proposal, we expect that the cost of those proposals will be several times more expensive than if the system in question were to be made available under an alternative IP framing. The size and complexity of the STAR-Vote system described in this RFI is larger than we had anticipated. The core STAR-Vote system described in the academic literature has a half dozen subsystems, each of which has a reasonable set of features, size, and complexity. The full-blown STAR-Vote system of the RFI is several times larger and has a large number of UIs that require significant user-centric usability and accessibility design and evaluation. As such, the cost of the system—even were it bespoke development in arbitrary (and thus very cost-effective) programming languages and deployment platforms, and did not entail IP loss on the part of the firm doing the advanced R&D necessary to create the STAR-Vote system—is high. If the offerors are required to transfer ownership of all IP, the final price of an already costly system may increase significantly.

Travis County could avoid these issues and still meet the goals enumerated in the RFI by advocating for an IP regime that includes joint ownership and long-term remuneration of all parties involved in the creation of the STAR-Vote system. Thus, one of the two IP regimes described in the following paragraphs are more tenable alternatives to the model outlined in this RFI.

Variant #1: Consortium Copyright, Licensed Certification Scheme, Election-centric License, Backend Financing

This variant would be more acceptable to potential vendors and we expect that the development cost of the system will be significantly lower. This position is aligned with best-practices in commercial Open Source product development and maintenance where a product or platform has the potential for long-term, high-impact deployment and use. Examples of such a scheme include the ecosystems around the Linux operating system, the Java platform, the MySQL database, the Apache web server, Google Protocol Buffers, the OpenSSL crypto library, and the Eclipse Integrated Development Environment.

- Create a STAR-Vote 501(c)(3) not-for-profit foundation, akin to the Linux, Apache, or OpenSSL Foundations. Ensure that membership in the Foundation is open to all who share the goals of Travis County and STAR-Vote, including the firm that develops the reference and deployed systems and deployment organizations such as value-added resellers, integrators, consultancies, etc. who build business models around STAR-Vote.
- Use a licensed certification scheme to enforce the STAR-Vote trademark. By providing a reference implementation, subsystem behavioral interface specifications, and an automated means by which a third party can check the conformance of a new

implementation (derivative or otherwise), the design conformance can be guaranteed and thus trademark use can be enforced.

- Release the system under the OSET Foundation Public License, or some other Open Source Initiative (OSI)-approved license that is amenable to public use and adoption. Licenses such as the BSD, MIT, or Apache are could be used as well. Each is broadly accepted and include few encumbrances.
- Provide reasonable royalties to the organization that develops the original reference and deployed systems.
- Prohibit awardees or any company that produces variants of the STAR-Vote system from asserting any patent rights via contract.

Variant #2: Copyright Retained by Awardee, Licensed Certification Scheme, Non-exclusive License

This variant will be more appealing to vendors as well and follows the IP regime used by the Federal Government for research and development efforts.

- The awardee that develops the system retains ownership of the copyright for all of their work. Travis County and its subcontractors (such as the co-inventors of the STAR-Vote system definition itself) retain copyright ownership over all of their work. Work that is performed jointly by Travis County, its subcontractors, and the awardee is jointly owned by the participants.
- As in Variant #1, introduce a licensed certification scheme to enforce design conformance and the STAR-Vote trademark.
- The awardee grants a non-exclusive, non-revocable, perpetual unrestricted license for the STAR-Vote implementation and all associated artifacts developed under the course of the project to Travis County.
- The awardee's contract with Travis County requires the following:
    - awardee will make current and future versions of STAR-Vote available to other jurisdictions under a RAND basis and a simple, clear, straightforward cost scheme that is mutually agreed upon by the awardee and Travis County,
    - awardee will release the system under an OSI-approved license that is mutually agreed upon by the awardee and Travis County, and
    - awardee will not assert any patent rights.

If Travis County decides to pursue the conservative IP position suggested in Section 4.1 of the RFI, we make the following concrete suggestions.

- Ask the vendor to make a full transfer of copyright to Travis County so that copyright protection is in the hands of a single entity. Copyright should be explicitly stated on all development artifacts including source code, specifications, developer and user documentation, etc.
- Issue the system under the GPL version 3 license as it (a) forces derivative works and improvements in deployed product to be provided back to Travis County, and (b) prevents the use of patents from forcing the code to be non-free.

6.3 Data Ownership

Matters relating to the ownership of data are much more straightforward, as we believe that election data produced by and for an election client, no matter where or how that data is housed, generated, or codified, should be owned by the client. No additional costs should be imposed for access to, or trivial manipulation of, that data whatsoever. Also, data should be protected by both copyright and licenses like all other artifacts associated with an open source system such as STAR-Vote. Our favorite licenses for such are Creative Commons licenses.[1]

6.4 Intellectual Property Framing

We suggest that IP protection should be motivated toward framing business models for vendors and the client rather than matters relating to system design integrity.

We presume that design integrity means long-term design conformance. That is, a vendor must be unable to fork incompatible variants of STAR-Vote and call it STAR-Vote. This goal can be fulfilled by appropriate use of copyright and existing license law. A consortium of entities, whether it includes non-government entities or not, can enforce design conformance at least as well as a single entity can, and likely can do better.

6.5 Intellectual Property Design Decisions

We explain in the following our thoughts on design decisions about IP—decisions that must be made about several orthogonal dimensions, including (1) expression and protection of ownership, (2) transfer and restriction of rights of use and modification, (3) avoiding legal encumbrances, and (4) backend finances.

Copyright Transfer

Should copyright be solely or mutually held, and by whom? The obvious choices are (a) full explicit copyright transfer to Travis County or some Consortium, akin to what is done at the Free Software Foundation (FSF), (b) mutual copyright is held between the organizations that develops the STAR-Vote system, akin to what is done with regard to technology development at most universities and some companies in the USA, or (c) copyright is held by the original company that develops the STAR-Vote system and rights to use, relicensing, etc. are all derived not from copyright ownership, but licensing between Travis County and the development organization.

Licence Choice

What kind of license or licenses should be applied to artifacts related to STAR-Vote? With regard to matters of Open Source technology, the Open Source Initiative (OSI) is the main arbiter of license classification and community acceptance. Consequently, using an OSI-approved license, or an elections-specific license that we can expect to be OSI-approved in the future (primarily we are thinking of the OSET Foundation Public License), is a wise choice. Ensuring that non-development artifacts are appropriately licensed is important as well. We prefer Creative Commons licenses.

[1] http://creativecommons.org/

Patent Avoidance

How can Travis County avoid encroaching upon existing patents as well as avoid having other organizations improperly patent aspects of STAR-Vote and thereby make STAR-Vote non-free? The simplest means by which to achieve this goal is to adopt the GPL version 3 license for the system, as it contains clauses specifically addressing this concern. Alternatively, Travis County can stipulate, via license and contract, that no patent rights can be asserted by either the winners of the RFP or any company that produces variants of STAR-Vote.

Finances

Obviously finances will have a major effect with regards to the IP scheme chosen by Travis County. Consequently, such choices will expand or constrict the number of proposers for any RFP.

As mentioned previously, Foundation schemes are not uncommon in the commercial Open Source space. Major operating systems, development platforms, and system components have been successfully developed and supported for decades using such structures.

We believe that both variants we summarize above hits slightly different sweet-spot with regards to addressing the key concerns of Travis County and ensuring that RFP responses are less expensive than outright replacing aging equipment with traditional vendor technology.

7 Making STAR-Vote Better

On the technical level, the goal of our comments below is to help insure that the eventual STAR-Vote implementation is as correct, secure, and verifiable as possible.

One of our main concerns is a sentence at the bottom of page 33 that reads, “At this time, Travis County is not planning to offer the decrypted content of the spoiled/challenged ballots on the Bulletin Board”. Unless it is impossible for legal reasons, Travis County should strongly consider posting the decrypted content of the spoiled ballots to the bulletin board. This is one of two ways that any voter can participate in the audit of the system, and it can be the strongest evidence that votes are actually recorded correctly. At the very least, more than the administrator (for example the audit team) should be given a chance to inspect that the PVR matches the EVR for challenged or spoiled ballots. This is a weaker audit, however, because only the voter will know their original intent.

The connection between cast votes and votes recorded is also established by comparison risk limiting audits, but if a voter feels that their interests are not represented by the audit team, this may not be satisfying to them. Furthermore, by distributing the effort of checking that votes are being recorded correctly, posting spoiled ballots can give a higher degree of public confidence than could ever be achieved by a reasonable risk-limiting audit.

We are confused about the distinction that seems to be drawn between challenged and spoiled ballots. We suspect that challenging and spoiling a ballot are exactly the same process described using two different names; if this is the case it should be stated explicitly on page 43, which currently reads “The procedures for creating a challenge ballot are the same as those for a spoiled ballot.” If it is not the case, the difference between a challenged ballot and a spoiled ballot should be explicitly described. It does seem like a good idea to introduce the terminology of a

"challenged" ballot, since "spoiled" (as used in current election terminology) isn't as indicative of the voter's intent when spoiling a ballot for verification purposes.

In the specification provided, it is possible that an implementation might hide or obscure the transfer from the scanner into the ballot box. Being sure that the paper ballots are stored is a crucial part of having end-to-end confidence in an election—if the software counting process fails for some reason, a voter should have confidence that their ballot will be included in a manual count. For this reason, it should be clear to a voter that the ballot box/scanner is actually depositing a ballot into the box rather than discarding it.

While the system tests themselves seem to be well-defined in this document, the testing procedure (in particular, when the tests are to be run and on what computers they are to be run) is not as well specified. For example, page 27 describes a scenario where the EDF is loaded onto a selection of computers to be used for testing. If testing succeeds, the EDF is then loaded onto the remaining computers. This is convenient, but may not provide the highest assurance. Along with this testing, there should be some required automatic testing (initiated by software) when each system boots or starts the software. Local testing should be run to completion on each local network once it reaches the configuration that will be used in the election. In general, any time there are changes made to the system that might introduce problems, it is worthwhile to do at least some testing.

Requirement 7.2.4 is very broad, and its interpretation depends critically on the definitions of "problem" and "detailed data". It is difficult to imagine a system that meets this requirement in 100% of cases for reasonable definitions of those terms, especially when malicious activities are included. As written, it is likely that the requirement will mostly be ignored, with a few minor features added to address notification and logging for the obvious types of problems that can occur with such networked systems.

Section 10.9.2.6, which states that the Trustee Software must not have any domain-specific knowledge, seems to contradict other parts of the specification. Perhaps part of the trustee system can meet this requirement, but certain other requirements in 10.9 seem to require domain specific knowledge. For example, being able to talk to the shared network layer might be considered domain specific knowledge. To clarify what this means, rather than referring to "decrypting the aggregated vote tallies", the specification could refer to "an encrypted value (for example the aggregated vote tallies)". It is also not clear that data export can take place in a convenient manner without some domain specific knowledge about the data being exported.

Requirement 11.2 says that the web voter ballot style lookup and viewer system must show "precisely" what will be seen/heard on election day. We are concerned that this will be difficult, since what will be seen may depend on the hardware that a voter ends up using at their polling place. Things like pagination and page organization could easily vary between voting machines. Instead of saying "precisely", we suggest that the web view either show something "similar" to what will be seen/heard on election day, or allow voters to select a specific screen size, possibly providing information about what screen sizes might appear at different polling locations.

Requirement 11.4.2.4 specifies that the RFI requires A1 paper. The PVR printer requirements list only sizes ranging from about 3" x 8" to standard legal size, and we suspect it is unlikely that paper much larger than A4 will be used. Perhaps A6 (about 4" x 6") or A7 (about 3" by 4") paper was meant instead of A1 (about 23" x 33").

We are unsure of the purpose of the Polling Location Network Traffic Inspector, described in 14.2.1. This is a reference implementation that can communicate with and interpret the messages sent over the shared network layer, and generate and present a human readable representation of the data captured. We don't know what the value of providing such a system to the public is, since the public will not be allowed to attach it to an air-gapped network during a real election, and are unclear about what information the implementation should convey as part of the human-readable representation. A clarification of the expected use of this software would likely answer both of these questions.

Section 17.5.2 mentions that there should be a debug compile configuration to enable extensive logging facilities with respect to performance and user interface measurement. We suggest that *all* debug information that is not collected during normal election situations should be enabled only by such a compile configuration. It can be very difficult to determine what additional debug information might compromise voter privacy, and it is not worth the risk of having such compiled code on a production machine where it might be enabled by some attack.

Furthermore, whenever a debug mode is being used, it should be obvious to the voter or official using the system. This could be achieved by a watermark saying "DEBUG MODE: NOT FOR USE IN ELECTIONS" that overlays any user interface. It is important to ensure that debug code can not be intentionally, or maliciously, installed on any machine that is going to be used for an election.

Requirement 17.7.3 specifies that all NIZK ballots received over the network should be verified unless it is too computationally intensive to do so. If it is computationally feasible, it might make sense to build NIZK verification functionality into the shared logging and network layer rather than implementing it independently across the various communicating modules that need to verify proofs. It might also be useful to specify some minimal level of NIZK verification that must be performed if verifying every NIZK is deemed to be too computationally intensive. For example, the networking layer could require that some threshold number (or percentage) of machines agrees that a given NIZK proof is correct.

The specification of Air-Gapped system communication in 17.8, and in particular 17.8.2, seems like an unnecessary weakening of security. In the absence of some specific reason why the shared network and logging layer should not be used universally within the system, there is no reason to allow for network use outside of the shared layer.

Section 17.9.4 is written in a way that makes it difficult to understand. The exact requirements there are difficult to interpret; for example, 17.9.4.1 seems to actually be two requirements, one stating that a file of plaintext votes must be produced and the other stating that an "Audit Plaintext Commitment File" (further described in 17.9.4.3 and 17.9.4.4) should be produced. In addition, it is not clear what is meant by an "Audit Plaintext Reference Key"; that term only appears once in the document (in 17.9.4.3.3), though its capitalization indicates it is a definition that should be understood.

The poll book/check in station is not a part of this RFI, but it needs to communicate with the BCS (through tokens) and the system that validates the hashes and transmits them to the receiving stations at the end of each day. These interfaces are not very clearly specified, and will need clarification if a system is going to be successfully implemented.

There are a few locations where we feel that the RFI over-specifies the solution to a problem. These might be acceptable ways to solve the problems, but they are worded in such a specific manner that they might eliminate other reasonable solutions. These requirements are:

- 9.1.29: Provide an on-the-page keyboard with all options for entering and modifying non-English items. The need here is to provide an input method that can be used for the various languages supported by the system, but there may be other reasonable ways to accomplish this aside from a single on-the-page keyboard.
- 9.2.4.3.5 The Ballot User Interface screen as defined may not be the best solution for displaying the needed information. For example, it may not need to be a separate screen, it could instead be a separate mode. We feel that the requirements of the Ballot User Interface screen could simply be stated as requirements of the system as a whole, and their accessibility can be left up to a team with UX/UI experience.
- 11.1.2.2.1 The export must use the Microsoft Office Open XML document format for Word. This might be a difficult format to export to, where a simpler format might be used that could still be imported into Microsoft Office for editing.
- 17.10.2.1: Google Protocol Buffers must be used. We understand the need for a standardized message format to facilitate the modular design, but there are other reasonable choices that might be made.

The algorithms provided as homomorphic encryption algorithms are only options that might be used. These options happen to have a degree of randomness built in; that is, two different encryptions of "0" look completely different from each other, as do two different encryptions of "1". This randomness is essential in order to preserve privacy within the system and, as such, should be mentioned in numerous places where homomorphic encryption is mentioned, and certainly in 18.4. The discussion of commitment consistency and non-interactive zero knowledge proofs is also a bit confusing. Much of the confusion could be avoided by a brief discussion of how the two relate (i.e., a sentence reading "commitment consistency can be achieved by a NIZK proof or commitment associated with each encrypted object, as well as by the primitive itself being commitment consistent, as is the case for PPAT).

In Appendix I, the citations have appear in the form [1]. However, the bibliography doesn't have any related numbering. One of these formats should change to make it easier for readers to find the appropriate sources.

8 Conclusion

Galois is committed to outstanding delivery and execution of products and services and strives to bring together not only internationally recognized technical and domain experts on its team, but also people who stand united in their commitment to improve democracy and its realization through e-Government systems. Our project teams have experts in a broad collection of fields who have proven success towards that public service objective, and we present the same commitment to Travis County through our team on this proposed project.

With the STAR-Vote system described in this RFI, Travis County has laid out a bold new vision for the future of the American democratic process. Galois's experience and expertise makes us the ideal partner to realize that vision and establish a new era of verifiable elections.