Responses to questions posed by Chris Jerdonek, President of the San Francisco Elections Commission
Joe Kiniry <kiniry@galois.com>
Galois — http://galois.com/
20 October 2015

How long do you think it will be before an open source voting system is certified for use in California and available for use by a jurisdiction like San Francisco?

Based on our estimates for the development of similar systems, we believe that we will develop a system exceeding the requirements stipulated by San Francisco from the ground up in approximately 18 months of development effort. The formal requirements, architecture, and design will be complete in the first four months of this process, at which point we would be able to begin the certification process, showing that our high level design meets the standards for certification.

Our development methodology means that as development continues, we continuously produce *traceable evidence* that the high level design, and therefore the specifications for certification have been met. The evidence we generate far surpasses anything VVSG testing centers have ever seen. Below we reflect on the implications of such a development method; our RFI response characterizes our traceable evidence.

This development process is flexible and generates traceable evidence in many ways. For example, following a traditional testing path, if the certification authority would like a set of test cases to demonstrate a certain behavior, we will provide that in a form that is readily understood and run by a third party.

In excess of current requirements, we also typically produce traceable evidence that not only does our system conform to certification authority's expectations with regards to test runs, but also is mathematically guaranteed to operate correctly in *any* environment, with *any* input, and under *any* adversarial setting. This is the only acceptable form of evidence for the kinds of mission-critical systems we deal with regularly at Galois, as testing does not show the absence of design or implementation errors.

This evidence and flexibility should allow certification to proceed much faster than it has on previously certified systems. As such, the certification process should also be much less expensive than any previous certification effort at the federal or state level.

We have had regular discussions this past two years with employees of the EAC, including all of its Commissioners; employees of NIST, including those responsible for defining federal certification standards; California's certification authority (Ryan Macias in the CA SOS office); and one of the VVSG testing centers (James Long, the VSTL manager at NTS) about our development and internal validation and verification methodology at Galois. All four parties have expressed significant interest in our capabilities.

In particular, at the federal level there is interest in component-based design, development, validation, and verification—something that we have been recognized as world-class in for two decades. And, informally, at the state level and within the testing center there is enormous interest in the strength of our evidence and the rapid means by which any third party can check its veracity. We look forward to being the first ever elections vendor that uses formal methods to create high-assurance systems for public elections.

Why hasn't a system like this been developed and certified yet by your organization or anyone else?

Galois's focus has been on providing R&D services to the federal government and private industry. Only recently has Galois begun working in verifiable elections technology. As such, our main goal moving forward is to develop an open source, verifiable voting system.

Open source software is always an advantage for the customer of the software, which may not always seem ideal for vendors. This is why open source voting systems have not yet been developed. Here are just a few of the ways that open source helps the customer, while (seemingly) making things harder for vendors.

- **Illegitimate perceptions of security:** Based upon numerous audits, software leaks, and FOIA responses, it is clear that existing vendors' voting systems rely on attackers not knowing details about their implementations. If existing systems were made open source, it is extremely likely that many vulnerabilities would be rapidly discovered. While some of these may be simple programming errors that can be fixed, others are security properties that rely on the source remaining secret. In general this allows for lazy design which makes it cheap and easy for vendors to create software quickly. Open source strongly discourages this sort of design.

  If software is built in this way, it has a frightening implication for the customers of the software. They must trust an unknown number of people who work for the vendor to keep the secret and hold it responsibly. Through careful design this fundamental flaw can be avoided, particularly with proper application of cryptography, but for existing systems this could require substantial rewriting and refactoring.

  It is our opinion that, were some existing vendors' software made public and scrutinized by security and correctness professionals, few customers would continue to purchase their products.

  Open software can be as secure as any closed software. Developers of open source software are not tempted to rely false security from secret source, instead opting for much stronger guarantees of security, including mathematical proof. There are many responsible researchers that will inspect open source systems, especially critical ones, reporting bugs to the developer, and often even submitting fixed when they find them. These statements are not theoretical. In fact, for the most part, virtually all of the technology that provides some degree of security in all of our computers, smart phones, and online secure transactions is open source.

- **Pricing models are in favor of traditional vendors:** An open source system that is also free software can be used by anyone. This means that a group that does not need customization or support (e.g., a student council election or a small election jurisdiction) could effectively run an open source election *for free* on computers that they already own. Traditional vendors believe that this costs them revenue, and it violates their intellectual property presumptions. After all, looking at existing vendors' contracts, they believe that they should be able to charge large amounts of money simply for the repeated right to use, not own, their hardware and software systems. The market has rewarded this business model by being largely structured around a supporting approach, giving no incentive for traditional vendors to change. We believe running a successful elections business is not limited to this model. We can provide top quality software to everyone for free, while running our business by supporting and customizing the software for customers with greater needs. Essentially, we intend to be the Red Hat of verifiable elections systems.[1]

- **Competitive and historical advantage:** Historically, the voluminous flow of federal funds into the pockets of past and current closed source vendors—especially in the early days of HAVA when little to no certification standards existed for elections systems—means that the playing field is stacked against any new entrant. Moreover, the vast majority of RFPs that we have reviewed stipulate preconditions on participation that prevent any new vendors from entering the market. As such, there is no opportunity for a new vendor to generate revenue to create open source elections systems.

- **Conservative and restrictive venture capital:** Given our experience with venture capital (VC), it is also clear that raising unrestricted funding for a venture that focuses on open source elections systems is effectively an impossible task. The first problem is that venture capital partners recoil at the prospect of funding a technology company whose primary customer is the government. Second, the strings that are attached to such funding, over and above matters relating to ownership, mean that pursuing a public interest agenda is counter to VCs' capitalist interests. We believe that only a class B

[1] Red Hat Inc. is a publicly traded company whose entire business model is based upon sales and support of Open Source software, primarily the Linux operating system. Red Hat's current market cap is just over $14B. http://www.marketwatch.com/investing/stock/rht

> corporation with no such strings can ethically and morally achieve the goal of technically-assisted, low-cost, publicly verifiable elections.[2]

Many believe that the time is ripe for the creation of an open source voting system for public good. San Francisco is only one large jurisdiction reflecting upon this idea; an idea that will enormously impact the quality, trustworthiness, and cost of future democratic elections worldwide. Galois believes in this vision enough to spin out a company exclusively focusing on this agenda.

What steps do you think need to take place for that to happen? What are some possible ways forward?

One or more large jurisdictions with a strong vision and desire to improve the quality and level of trust in elections throughout the United States must take the lead in funding the development of the first verifiable, open source voting system.

We must also have at least a handful of corporations or non-profit foundations with appropriate publicly-documented expertise and professionalism, longevity, and staff to accomplish the goals of these jurisdictions. Open source software does not write itself, and having legitimate firms to develop and, more importantly, integrate and support such software is critical. Governments are generally not interested in developing or maintaining software systems or services "in house". It is not in their bailiwick, nor do they have sufficient internal capability and political will to pursue such a vision.

Additionally, non-profit foundations or large corporations with significant capital and vision can also significantly impact this vision, as they can provide funding to such a venture for the benefit of the general public. Several similar initiatives exist in adjacent areas today, ranging from public use of open government data to open source, freely software web technologies (e.g., the Mozilla Corporation and its Foundation, the Free Software Foundation, and the Apache Software Foundation).[3]

What open source license or type of open source license do you think should be used and why (e.g. OSI-approved or non-OSI-approved, permissive or copyleft, etc)?

Several licenses are likely candidates for open source elections technology. The technological foundation chosen by a system's implementers may significantly constrain one's choices, since

[2] In the United States, a class B, or benefit, corporation is a type of for-profit corporate entity, authorized by 30 U.S. states and the District of Columbia, that includes positive impact on society and the environment in addition to profit as its legally defined goals. Benefit corporations differ from traditional C corporations in purpose, accountability, and transparency, but not in taxation. See https://en.wikipedia.org/wiki/Benefit_corporation

[3] See https://www.mozilla.org/en-US/, https://www.mozilla.org/en-US/foundation/, http://www.fsf.org/, and http://www.apache.org/.

some licenses such as the GPL are transitive. We commonly develop high-assurance systems and release them to the public using very permissive licenses such as BSD, MIT, and Apache Foundation licenses. On occasion, when warranted, we use other OSI-approved licenses. We have also evaluated the OSET Foundation's public license and find their arguments for its adoption compelling.[4]

We believe that there is also an opportunity for the use of dual licensing, as has been seen in numerous open source products that are available for public, research, or educational use. For example, one of the products we use for high-assurance systems design and development, EiffelStudio, is available under two licenses: one for commercial closed source development and another for open source development.[5] Several other organizations with which we collaborate, such as SRI International and Microsoft Research, have used similar licensing schemes.

If San Francisco were to adopt an open source system, how could San Francisco be assured that the system would continue to be developed and maintained over time?

One of the main contractual guarantees we provide to clients is that we will support our software systems indefinitely. Our core license provides *perpetual* ownership and core technical support. For a relatively small optional recurring fee, we will ensure that the system is maintained and evolved in accordance to the legal constraints and integration needs of our clients.

A key advantage of open source software is that Galois does not need to be the company providing maintenance for a client's software. Any experienced software company should be able to keep the system up and running. In fact, we predict and hope for a future where numerous local firms support and maintain our software for our clients, much like the prolific flowering of Linux systems developers and integrators we have witnessed over the past twenty years. In an optimal future, the bulk of our customers will be using high quality third party software firms whose expertise and business is centered on supporting county and state elections organizations' open source technology needs.

Improvements to the features of the software, for example, in case of large statute changes, must be negotiated and performed on a case-by-case basis. The same advantage to open source holds here though, as anyone can make the improvements. Consequently, the RFPs for such work will be considerably simpler and bidding for such opportunities will be considerably more competitive than the current oligopoly of closed source vendors we witness today.

[4] http://www.osetfoundation.org/public-license/
[5] https://www.eiffel.com/eiffelstudio/licensing/