|galois|

Electronic Poll Book Solution

RFI #EO-200616-2
Elections Ontario
July 2016

Dr. Joseph R. Kiniry, Free & Fair, kiniry@freeandfair.us
Dr. Daniel M. Zimmerman, Free & Fair, dmz@freeandfair.us

Responder

Galois, Inc., 421 SW Sixth Avenue, Suite 300, Portland, OR 97204, USA
operating under the assumed name Free & Fair

Point of Contact

Jodee LeRoux, Contracts
Galois, Inc.
421 SW Sixth Avenue, Suite 300
Portland, OR 97204
ph 503.808.7209
contracts@galois.com

Abstract

Galois, operating under the assumed name Free & Fair, welcomes the opportunity to provide expert input to Ontario's Chief Electoral Officer (hereafter, CEO) with regard to this RFI on electronic poll books (hereafter, EPBs). We intend to submit to a following RFP on this topic, and we hope that our input proves useful regardless of which vendor eventually provides technology to Elections Ontario.

We welcome direct contact on any matters relevant to this RFI. For matters relating to our verifiable elections products and services, contact Free & Fair. For matters relating to general consultancy in high assurance systems development, analysis, and auditing, contact Galois. In particular, if the CEO holds concerns about the claims, capabilities, correctness, or security of any elections technologies, or needs constructive work on distributed algorithms for EPB synchronization or secure data storage and transmission, we strongly suggest contacting Galois; we have extensive experience with this kind of research and development.

About Galois and Free & Fair

Galois is a privately held U.S.-owned and -operated company established in 1999. 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. We specialize in the research and development of new technologies that solve the most difficult problems in computer science. Our team works closely with clients to achieve a balance among the privacy/cost/speed challenges involved in making systems more trustworthy.

Galois has 60 employees in 2 offices (Portland, Oregon and Arlington, Virginia), with principal investigators leading research and engineering teams in the areas of cryptography, software correctness, mobile security, cyber physical systems, computer security, machine learning, human machine interaction, and scientific computing. We have won and successfully executed on dozens of multi-year, multi-million dollar R&D projects for numerous federal agencies including the Department of Defense (DOD), the Department of Homeland Security (DHS), Defense Advanced Research Projects Agency (DARPA), Department of Energy (DOE), NASA, and members of the Intelligence Community.

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.

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. A high-assurance system, or trustworthy system, is a system designed from first principles to be free of flaws. These include flaws related to system correctness, security, reliability, assurance, and more. High-assurance systems are historically used in situations where failure can lead to loss of life (e.g., an automated train) or have enormous financial implications (e.g., digital cash in smart cards).

Historically, Galois has not executed on election systems. However, 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.). Our elections team brings together internationally-recognized technical and domain experts who are committed to improving democracy and its realization through e-government systems. 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.

Galois has a flat, peer-to-peer organizational structure. Senior personnel who have national or international experience relevant to the development of elections systems include Dr. Joseph Kiniry (an internationally recognized expert in high-assurance systems, security, and elections) and Dr. Daniel Zimmerman (a former professor at two institutions and an internationally recognized expert in high-assurance systems design and development).

For the past year, we have been developing prototype technologies in this space that include an EPB, a verifiable in-person voting system, and tabulation and auditing techniques that support ranked choice voting. We are in the process of spinning out a class B corporation, Free & Fair, 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. Dr. Kiniry is the Chief Scientist and CEO of Free & Fair, and Dr. Zimmerman is a key member of its management team.

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 EPB 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 (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 EPB 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, Democracy Works, the Overseas Vote Foundation, and U.S. Vote. Dr. Kiniry is a key actor in the newly formed NIST-EAC Public Working Groups.

Dr. Zimmerman, the Technology Lead at Free & Fair, has extensive experience in formal methods, high-assurance software engineering, concurrent and distributed systems, and foundations of computer science. Before coming to Galois, he taught computer science at multiple universities for over a decade. At Galois, he has worked primarily in the areas of rigorous software engineering and verifiable elections technology.

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.

Galois Verifiable Elections R&D

Our design and architecture for election-related systems is highly modular. Each module uses only open data formats for communication, resulting in a system that can be modified and upgraded by anyone who is familiar with the open standards that we use. This modular architecture features an air gap between the software responsible for running the election and the software for designing and reporting on it. A modular architecture assists with compositional validation and verification, experimentation with user experience variants, and phased user acceptance testing. It can also ease customization of the system, allowing new voting methods and ballot styles to be swapped into the system as needed without requiring system-wide changes. Modular design also aligns with what we expect to see in version 2.0 of the U.S. Election Assistance Commission's Voluntary Voting System Guidelines.

Our cryptographic foundations, ranging from authentication to data-at-rest to provenance-preserving logging, are based upon our work on another one of our products, Cryptol,[1] and a host of advanced tools and technologies for ourselves and academic partners. In general, our systems use cryptographically secure authentication and credentials issuance (via technologies like multi-factor authentication), cryptographic databases, cryptographic hardware (including FIPS-certified libraries and hardware), custom formal protocol design and verification, custom formally verified cryptographic libraries, and logging with privacy-preserving cryptographic integrity.

Our systems are all fault tolerant and have sufficient redundancy, both in algorithm design and physical architecture, to ensure that they can survive the simultaneous failure of multiple machines or networks.

Software correctness is an integral part of the Galois development approach, beginning with the specification of a system's domain model, requirements, and software and network

[1] http://www.cryptol.net/

architecture. By formally specifying the initial design, and developing the implementation based on the resulting specification, we guarantee that we are implementing exactly the desired system. We also incorporate the vast majority (typically on the order of 99%) of the software tests within the code itself, rather than developing tests separately, and these tests are, for the most part, generated automatically from formal specifications. This 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 and systematic validation of the code as well as the all evidence-based artifacts and documentation produced.

For the most essential parts of the software we go a step further, performing a machine-checked functional verification of the software. In this process we first design a mathematical model that should be as easily understood as the English language specification. We then provide an implementation that is mathematically proven to meet the specification. This mathematical proof can be automatically checked on any computer, giving unparallelled assurance that the software is correct. These techniques have historically been used for safety-critical systems, where the failure of a system would result in loss of life (e.g., flight control systems at Airbus) or have enormous cost implications (e.g., failure of a mission to Mars).

By combining these approaches, we get a chain of correctness that starts with the high-level system specification and continues all the way down to the smallest implementation details of the most critical parts of the system. At each step in the chain we focus on providing evidence of correctness, generally in multiple forms, including for example refinement proofs from informal to formal specifications, unit test suites, and mathematical proofs of correctness and security. In other words, all the effort we put into ensuring that our system is correct generates tangible artifacts that give external parties the same confidence in our software that we have.

The specific peer-reviewed methodology we use for all of our software is a variant of Design by Contract[2] with some aspects of a Correctness by Construction[3] approach. Our process, method, tools and technologies span several deployment and development platforms, specification and programming languages, and communication and coordination schemes.

Support, Staffing, and Help Desk Services

Galois recognizes that any system implementation not only requires a technically sound and robust product with comprehensive business functionality at its core, but also needs to be supported by professional services throughout the project life cycle. We have expertise in

[2] https://en.wikipedia.org/wiki/Design_by_contract
[3] https://buildsecurityin.us-cert.gov/articles/knowledge/sdlc-process/correctness-by-construction

professional services such as project/program management, software design and development, testing, mentoring and training, implementation and go-live as well as post-implementation operational support services.

At Galois, we typically run a very lean 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 traditional triaged 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 in the system are typically triaged by team members within minutes, responses to issues are immediate, and fixes are prioritized based on conversations between the customer and the development team. We can provide evidence of these claims by simply referring evaluators to our Open Source product repositories.

For field support during deployment and system use, we augment operational support with a front-line team that 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, online video chat support, or any combination of these.

We also provide training offerings related to our products; Open Source technology adoption, legality, and use; certifications; evolving national and international standards in elections technologies; and rigorous software development.

Despite the fact that our products are high-assurance and include a wide range of untraditional artifacts—such as formal specifications, tests, and proofs—to guarantee their correctness, security, usability, and accessibility, they are no more expensive than existing products. In fact, our methodology is intended to significantly decrease the cost and time of certification.

The Free & Fair Electronic Poll Book

The Free & Fair Electronic Poll Book (hereafter, FFEPB) is a open source software-only system that runs on COTS hardware. It is capable of running on a wide variety of hardware, as long as certain minimum requirements (e.g., for memory and available non-volatile storage) are met. It is also capable of running on a wide variety of operating systems, including Android, iOS, Linux, Mac OS X, and Windows. There are no current or pending federal, state, or organizational certifications of the FFEPB software.

With respect to hardware, we recommend the use of iPads or Android tablets as EPBs: they are affordable, easy to provision and administer, and capable of directly accepting voter signatures on their touchscreens and directly scanning identity documents with their cameras (eliminating any need for separate signature pad and scanning devices). Regardless of the choice of EPB hardware, FFEPB works with all standard peripherals (barcode scanners, keyboards, pointing devices, etc.). As discussed further in the Costs section, Free & Fair can facilitate the acquisition of appropriate hardware; Elections Ontario can also obtain hardware directly through other purchasing channels, taking advantage of any discounts offered directly to the government of Ontario by hardware manufacturers or distributors.

FFEPB can accommodate an arbitrary number of EPBs and is designed to be fully fault-tolerant; any EPBs that fail due to hardware malfunction, accidental damage, or other issues can be replaced on-the-spot by poll workers with spares, without requiring technical support intervention and without compromising the security or integrity of election data.

EPBs in the FFEPB system can communicate with each other, both within a single polling place (*local* communication) and among multiple polling places and central offices (*remote* communication). Local communication is required to keep all EPBs in a polling place synchronized, so that they all have current information about voter check-ins, information changes, etc. Network access for local communication can be provided using existing WiFi networks, using dedicated WiFi routers that may be, but need not be, connected to the public Internet (e.g., commodity LTE hotspot devices), or using WiFi hotspot functionality available within the EPB hardware itself (e.g., “Personal Hotspot”); the latter is the most economical choice for polling places with no pre-existing WiFi infrastructure.

Network access for remote communication is not required for correct operation of the FFEPB system, but would be required to fulfill some of Ontario’s detailed requirements in Section 4 related to synchronization and reporting. If remote communication is used, one EPB on the local network at each polling place acts as a “gateway” for the others on its network. It is responsible for all remote communication, which can be carried out in one of several ways: using WiFi and existing broadband Internet access at the location; using LTE or 3G Internet on the cellular network via a poll worker’s cellphone, standalone hardware, or built-in capabilities in the EPB hardware; or using alternate, lower bandwidth methods such as SMS or dial-up Internet connections in areas where broadband access is unavailable or impractical.

We provide more detail about FFEPB’s design, requirements, and capabilities in the following sections.

Reflections on Ontario's Detailed Requirements

Openness and Architecture

As described above, FFEPB is an open source software-only system that runs on COTS hardware. We strongly advocate that any eventual RFP by Elections Ontario include a requirement that the provided solution be open source to the extent possible; all software written by the vendor should be open source, but because of the potential for running EPBs on commodity, closed-source hardware, it is not reasonable for an RFP to require the entirety of the solution to be open source. We also strongly advocate that any eventual RFP take a modular approach, requiring the EPB software, EPB hardware, networking hardware (if any), and any other required components to have well-defined APIs so each can potentially be supplied by and maintained/upgraded by different vendors in a way that best satisfies the needs of Elections Ontario.

FFEPB has a peer-to-peer architecture, and every unit in the system has exactly the same software installed. One unit is provisioned as the "initial" EPB and is supplied with the election data (voter lists, polling place assignments, etc.) via an appropriate mechanism to be determined in conjunction with Elections Ontario (loading from a memory card, connecting to an Elections Ontario system and downloading data files, etc.). Additional EPBs can be provisioned in the same way, or can provision themselves automatically; they need only to be told what polling location they are to be used in, to be connected to the same local area network as *any* already-provisioned EPB (in which case they will discover the FFEPB system automatically) or given the Internet address of *any other EPB in the system*, and to be authenticated as part of the startup process. In practice, one or more EPBs (to be used for election monitoring and gathering of data after the election) would be placed in fixed locations, such as central election offices; the first EPB to be powered on in a broadband-connected polling place would be supplied with the address of one of the fixed EPBs, and subsequent EPBs powered on in the same polling place would configure themselves automatically using the local network. EPBs to be used in polling places without broadband connectivity would need to be provisioned in advance, because any bandwidth available for external communication via SMS or dial-up modem would be insufficient for provisioning.

The peer-to-peer architecture, in conjunction with other aspects of the system's design including the way it handles data replication and consistency, allows FFEPB to scale to an arbitrary number of devices. There is no "central server" that must act as a clearinghouse for all EPB operations, and no single point of failure. EPBs (or groups of EPBs) that are disconnected from the FFEPB network, due to network failures or placement in polling places with no network connectivity, can continue to work and maintain consistent data; they automatically synchronize with the rest of the network when they are connected again.

Software Functions

FFEPB effectively maintains a distributed voter database and can distinguish voters by various criteria, making it straightforward to manage the 122 (or, indeed, any reasonable number) separate lists of voters required by Elections Ontario. It is capable of identifying each voter's correct voting location, updating voter information, registering new voters, and recording voters who have voted. With customization for Elections Ontario's specific requirements, it will also be capable of producing and recording oaths and statutory declarations.

The user interface is designed to be both accessible for all users and "Google simple", with clearly defined operations, minimal choices to be made by poll workers at each step, and online help available for every function. In the unlikely event that technical support is required, poll workers can use a live Internet chat-based interface (where Internet connectivity is available) to work directly with support personnel during the election, in addition to more traditional telephone-based technical support.

With respect to election worker assistance and management functions, we can implement functionality requested by Elections Ontario that is not "core" functionality of an EPB such as sending messages to election workers, surveying them about their experiences after the election, and tracking and reporting voting location tasks and statuses. However, we would encourage any potential RFP to follow a modular approach in requesting such functionality; for example, COTS operating systems such as Android and iOS already have well-tested built-in infrastructure for messaging, and it would be of questionable utility to replicate that functionality as part of a monolithic EPB software package.

With respect to synchronization among EPBs, we described the peer-to-peer FFEPB architecture above. This architecture enables synchronization among EPBs within the same polling place, as well as synchronization among EPBs in different polling places when connectivity is available. However, real-time synchronization of the entire voters list across multiple polling places is impractical in an EPB system as widely and diversely deployed as that proposed by Elections Ontario, because of the considerable overhead of propagating every data update across the network in real time. Synchronization in "real time" is an unsatisfiable constraint in any case because of network constraints; it is clearly unacceptable to make a voter wait for minutes at check-in to be sure that data has been properly synchronized over a low-bandwidth, high-latency connection. Instead, if live synchronization during the election is necessary, relaxing the "real time" requirement to allow for *eventual consistency* of the voter lists would significantly reduce the cost of providing sufficient connectivity to all polling places. In this way, polling places with intermittent or minimal connectivity would still be able to participate in data synchronization by "batching" and transmitting their updates over a longer period of time.

An eventually consistent system would, of necessity, have mechanisms to detect conflicting data updates and flag them for review. For example, if the same voter checks in at two different polling places and it is not detected at the time of the second check-in because of network connectivity or speed issues, it would be detected (and election officials would be alerted) when data updates from the two polling places are reconciled. This sort of automated review process is still a significant improvement over manually detecting such occurrences, which are extremely rare in any event, by examining paper poll books after an election has concluded.

Security

Voter data in the FFEPB system is kept confidential and secure in two main ways: by being protected on the individual EPBs, and by being protected when communicated among EPBs.

Each individual EPB stores voter information in a local data store with three layers of protection: first, access to the data is controlled via a local encrypted channel; second, the data is stored encrypted on the device; third, all sensitive data in the data store is itself encrypted by the FFEPB software. Moreover, computation on sensitive data is performed *without decrypting the data*, so that no sensitive data is ever decrypted into clear text on disk, in memory, or on the network except for display and printing as part of EPB operations. This security architecture means that, even if a malicious party steals an EPB and has administrator rights on its operating system and administrative access to the database software, they can obtain no sensitive information.

All local communication among EPBs, and all Internet-based remote communication among EPBs,[4] uses secure channels (TLS) to ensure communication integrity. Regardless of the communication mechanism, *all* transmitted data is encrypted and digitally signed, ensuring both integrity and authenticity. Finally, our communication protocols are formally specified and verified for both correctness and security.

Each EPB is configured with minimal exposed services, and appropriate firewalls and security measures to minimize the possibility of external intrusion via the public Internet. We envision most EPBs being used in NAT environments, behind other wireless routers or hotspot devices, further minimizing intrusion risk. In the unlikely event of an Internet-based intrusion, standard OS-level security measures in conjunction with the data protection measures described above ensure that the intruder would have no way to manipulate the data on an EPB without being detected.

[4] Remote communication via SMS does not use secure channels for lack of availability. However, all SMS communications are encrypted such that only the intended recipients can decrypt them.

We hope that Elections Ontario would mandate, in any eventual RFP, an implementation platform that does not require antivirus and malware detection; however, any available antivirus and malware detection solutions for the platform in use can be employed without affecting the operation of the FFEPB software. As the FFEPB architecture has no central server, there is no need for central server security.

All activity on each EPB, and in particular every operation that causes a change to the data store, is logged in a cryptographically secure manner. The resulting log is tamper-evident, such that any attempt to manipulate its contents, or to modify the data store without logging the modification, can be detected.

Because it is important to ensure that only authentic EPBs are in use, every EPB must be properly authenticated when it connects as a peer to the FFEPB network. Such authentication is traditionally done using passwords or keys, but in the proposed system is dramatically simplified by using *Tozny*,[5] an Android and iOS app that permits secure authentication to devices and software without the use of passwords or even touching the devices in question. Instead, to provision a new EPB, an official simply points their camera first at an already-provisioned EPB's screen, then at the new EPB's screen; alternatively, if the EPBs have built-in cameras, those can be used for authenticating EPBs that are in physical proximity. Poll workers can use the same method to authenticate themselves to the EPBs and to open and close the polls in polling places with Internet access. This is a completely hands-off, trivial authentication procedure usable by even the computer averse or the disabled, and can be performed using any Android or iOS device with a camera and an Internet connection. In situations where poll workers do not have, or cannot be temporarily issued,[6] mobile devices to use during the election, or in the absence of Internet access, passwords can still be used as an authentication mechanism.

Performance

As previously discussed, we recommend against "real-time" data synchronization. Within a single polling place, peer-to-peer synchronization among the EPBs will be near real time; there should be effectively no perceptible delay in voter processing due to local EPB synchronization, except in cases where the network is recovering from a hardware failure.

FFEPB does not synchronize the entire voter list across all EPBs, but rather ensures that data is available where it is needed and is sufficiently replicated for fault tolerance. Across voting locations where connectivity is available, synchronization speed will vary depending on network

[5] http://www.tozny.com/

[6] It is straightforward to provide affordable Android devices, with no cellular plans or other ongoing costs, to be used for authentication during the election cycle by poll workers who do not have mobile devices of their own. Such devices cost well under $50 each and can be reused for multiple elections.

conditions. We expect changes to be reflected across the peer-to-peer network in seconds to minutes, depending on available bandwidth and on where in the peer-to-peer network the data is being requested. We expect response time for search results within an electoral district to be nearly immediate, and response time for search results province-wide to be a few seconds in most cases.

Setup and configuration for the EPBs was discussed previously. Because of the peer-to-peer nature of the network architecture, no special handling is required for communication among the 25,000 EPBs.

Hosting / Central Data Services Portal

Since the FFEPB architecture has no central server, no dedicated hosting or central data services portal is required. We assume that voter data is already maintained by Elections Ontario in an elections management system (EMS), which is accessible from (at least) some central location; such data will be extracted from the EMS for use on the EPBs, and merged back into the EMS after elections, in a way to be determined in collaboration with Elections Ontario.

Reporting

FFEPB's reporting is completely flexible; it stores all logs and data in databases on which arbitrary queries can be executed. The system can generate any type of report desired by the client, and these can be arbitrarily customized.

The system can be audited by examining its tamper-evident, cryptographically secure logs. The same level of customization applies to audit reports as to regular data reports. In addition, at the closing of the polls, the system provides reconciliation information that allows election officials to check the actual numbers of used ballots of all types and styles against the expected numbers of ballots based on polling place check-ins recorded by the EPBs.

Costs

The following is a high-level description of our pricing model, and estimated pricing, for the FFEPB system. We will provide additional details in any response we make to an eventual RFP.

Pricing Model

Free & Fair's primary goal is to increase the trustworthiness and verifiability of all democratic elections worldwide, while also reducing their costs. With that goal, our business model is unlike those of our competitors, so our pricing model is outside the norm.

Our licensing and support model is also unique in the industry and is based upon our technical capabilities and ability to guarantee defect-free products. We offer a permanent license for a product that you own, with optional annual support contracts for software and hardware. If anyone ever finds a correctness or security flaw in the system, we will fix it for free.

The software license cost is for a license-for-ownership and includes lifetime maintenance, warranty, and support in the form of defect fixes. Our license pricing is based on population and, for FFEPB, is $1.00 per person living in the region served.

The (purely optional) cost of a software support contract (listed below for reference) is fixed and unchanging, at 10% of the original license cost. The support contract includes training, remote and on-site support for your elections, and updates to your software necessitated by changes in requirements or law. You are also welcome to contract with any other firm to provide that support, as you will have all the source code and development documentation for the entire system that you own.

Since we exclusively use COTS hardware, our hardware costs are passed straight through to you with no overheads beyond our actual costs of handling (software installation, configuration, etc.). Our cost estimates do not include potential discounts available to you as a large-scale government purchaser, so they are conservative. Moreover, if you have existing hardware compatible with our software system, you can provision and use it in elections. It is also well within your rights and interest, if you wish, to use the COTS hardware purchased for elections for other purposes outside of the election calendar. Complementing our optional software support contract, we are happy to provide a yearly hardware support contract at 5% of the original hardware cost if you wish to purchase one from us.

Finally, while it is likely that the total cost of ownership for your EPB system over even a short timeframe (e.g., three years) will be lower with our business model than with our competitors', the fact that our software systems effectively have an arbitrarily long shelf life (because they are guaranteed defect-free and run on whatever platform you decide) means that your per-election costs for EPBs will effectively trend to zero in the long term.

Estimated Pricing

The following table shows our estimated pricing for the requested EPB system. We have made no assumptions about the Internet connectivity Elections Ontario intends to provide for the EPBs; we assume that local connectivity within a polling place will be provided by the EPBs themselves with no additional hardware.

As previously mentioned, our cost estimates do not include potential discounts on hardware available to Elections Ontario as a large-scale government purchaser. Also recall that our support contracts are entirely optional; our one-time perpetual software license fee includes lifetime maintenance, warranty, and support of the delivered software in the form of defect fixes. The annual software support cost listed here is for our highest level of service, and includes all training, support, and software modifications (other than defect fixes); lower cost support contracts, with correspondingly lower levels of service, are also available.

Item	Cost
Software	
Perpetual License (described above)	$13,800,000 (based on population)
Annual Support (*optional*)	$1,380,000 per year
Hardware (25,000 units)	
EPB Unit	$500 x 25,000 = $12,500,000
Annual Support (*optional*)	$625,000 per year
Totals	
One-time Costs	$26,300,000
Optional Support Costs	$2,005,000 per year

Additional Considerations

We currently have no insurance coverage with respect to the FFEPB product. However, such coverage would be put in place before we supply the product to Elections Ontario.