|galois|

Connectivity at the Poll
Internet Service Provider Coverage

RFI #EO-200616-1
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. We do not intend to submit to a following RFP on this topic, unless such an RFP permits the submission of proposals for complementary support work on matters relating to independent third party distributed systems' correctness and security analysis. We'd be happy to discuss with the CEO our capabilities and services in these areas, some of which we provide to government agencies and intelligence services around the world.

Free & Fair is an elections technology vendor, as described in more detail below. As such, we are also responding to the complementary electronic poll book (hereafter, EPB) RFI, and therein we provide more details about our products relevant to these RFIs.

In this response, however, we take a step back and reflect upon the tradeoffs that directly impact the trustworthiness of an election and its election infrastructure. As such, we hope that this information is useful regardless of which vendor eventually provides technology to Ontario. Tradeoffs involve balancing technology features against usability, utility, cost, and security.

Often the introduction of a new feature impacts not just the functional properties of a system (it can now do X!), but also adversely impacts non-functional properties, such as the system's security, deployability, and cost. We broadly map out the problem space and solution space of networked election technology such as EPBs, highlighting the tradeoffs inherent in different regions of those spaces. We hope that this analysis has utility to the CEO in framing future RFPs for connectivity and EPBs.

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 plurality and non-traditional election schemes, such as 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 a dozen 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 Galois's 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 and *Free & Fair* development approaches, beginning with the specification of a system's domain model, requirements, and software and network 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

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

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

We recognize 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 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.

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

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. One of *Free & Fair*'s business principles is that the amortized costs of digital elections infrastructure should trend to zero over time.

Reflections on Ontario's Questions

We reflect upon this RFI's questions both in the context of the design of our EPB product (detailed only in our response to the sister RFI) as well as in the larger context of networked digital election systems, and large scale EPB deployments in particular.

The problem space of such technology is summarized in Table 1. We use *feature modeling* to characterize the *problem space*. As such, each dimension of the model is characterized by a *feature*, whose realization in the *solution space* falls somewhere between minimal and maximal *configurations*. To learn more about feature modeling and its utility, see Apel et al.,[4] Clements and Northrop,[5] or van der Linden et al.[6]

We characterize the *problem* and *solution spaces* for EPBs below, and then provide some specific recommendations given what we know of the context of Ontario's elections.

Problem Space

We first characterize the problem space using six key features of the EPB problem space. Most products in the market hit one particular configuration of this feature space. Others, like *Free & Fair*'s EPB product, are flexible and cover many different configurations, depending upon the client's needs.

The fundamental reason we map out the full problem and solution spaces for you is that the recommendations we make below with regards to this Connectivity RFI are intimately tied to the networking and security architecture of your solution, as well as to other critical non-behavioral properties of your elections (i.e., historical incidence of fraud)

[4] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. Feature-Oriented Software Product Lines: Concepts and Implementation. Springer, 2013.
[5] Paul Clements (Author), Linda Northrop. Software Product Lines: Practices and Patterns. Addison-Wesley Professional, 2001.
[6] Frank J. van der Linden, Klaus Schmid, Eelco Rommes. Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer, 2007.

Feature	Minimal	Maximal	Notes
Connectivity	EPBs are not networked	Fully connected, always live	EPBs with no connectivity require significant amounts of resources for pre-election provision and post-election audit. Fully connected EPBs mandate complex and expensive networking infrastructure.
LAN Protocol and Infrastructure	Ad hoc hotspot	High-speed Ethernet	Well-designed distributed algorithms require low bandwidth, moderate latency updates and can use simple, automatically configured ad hoc RF networks. Badly designed distributed algorithms necessitate high bandwidth, low latency networking infrastructure such as a physical network switch (or two, for redundancy) plus Ethernet network cabling per polling place. This configuration has high deployment and operational complexity and cost. Likewise, a cryptographic protocol with unknown security properties (i.e., lack of scientific evidence of correctness) requires physical controls to mitigate some security concerns, such as a mandatory (and expensive) Ethernet deployment.
Network Architecture	Client-server	Ad hoc peer-to-peer	Traditional client-server architectures are simpler to design and build, but do not scale well without significant server-side complexity and cost. Peer-to-peer models are more complex to design and build, but scale very well and necessitate little to no server support and cost.
Network Infrastructure	Piggyback on election officials' phones	Dedicated network equipment and lines with VPN	The network infrastructure to accommodate WAN protocols (described below) can range from piggybacking on election officials' devices (e.g., using an election official's iPhone as a hotspot, which under most telecom plans would effectively be free and reliable) to deploying a dedicated VPN over a physically deployed infrastructure (ranging from reusing public Internet services, such as extant cable modem deployments, to short-term dedicated DSL deployments for a given election). Obviously the deployment of dedicated hardware and network infrastructure has very significant complexity and cost considerations.
Server Infrastructure	Serverless	Dedicated hosted servers	A serverless infrastructure is possible using novel distributed algorithms; consequently there is no design, deployment or cost associated with a data center or a cloud service provider. The most complex and expensive option for this feature is to deploy one's own hardware in a colocation facility for a traditional client-server network architecture.
WAN Protocol	SMS	High bandwidth, low latency IP	Well-designed distributed algorithms require low bandwidth, moderate latency updates and can use fairly "old school", inexpensive (sometimes free) unreliable communications technology, such as SMS or modems. Badly designed distributed algorithms necessitate high bandwidth, low latency networking infrastructure that has high deployment and operational complexity and cost.

Table 1: Tabular Feature Model of EPB Problem Space

Interpreting Feature Models

To interpret the features of the problem space of EPBs, as summarized in Table 1, think of designing an EPB for your jurisdiction as a "Choose your own adventure" game. We can depict these features graphically in what is known as a *feature model*, a portion of which is seen in Figure 1 below.

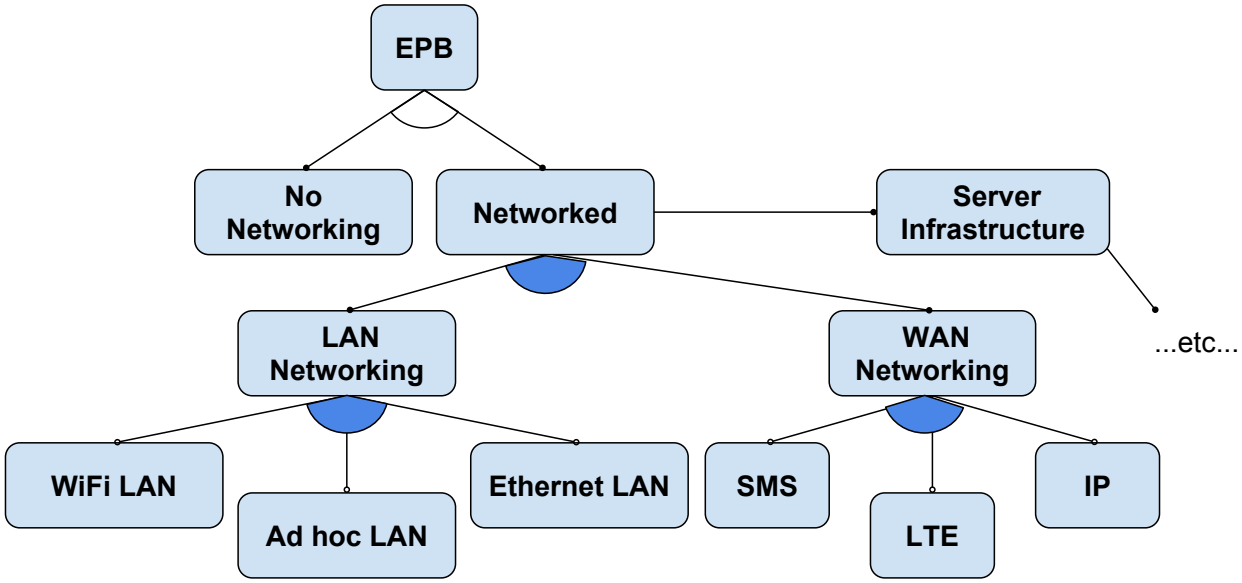

Figure 1: Feature Model of EPB Problem Space

Start at the top of this diagram in the **EPB** feature. Since it is the topmost node, you clearly want an EPB, but to choose one you need to make a number of choices as you move down the graph. An open wedge, like the one just below **EPB**, means that you must choose either the left feature, **No Networking**, or the right feature, **Networked**. If you choose the right feature, then you need to choose either or both features below that, as denoted by the filled in wedge. The solid circles at the ends of the line segments mean that the feature touching the circle is mandatory; an open circle (like the one touching the lowermost networking features) means that the feature is optional. Once you have made all your choice you have fully configured your feature model, and you now understand your problem space and can look for a solution/product that fulfills those requirements. Some development tools provide direct support for this kind of interactive configuration, going so far as to write code automatically for you to fulfill your configuration choice.

Forcing Functions

The forcing functions that help you choose a particular solution in this problem space are characterized in Table 2, which is a tabular version of a feature model like that seen in the above figure. We identify twelve features in the model.

Feature	Minimal	Maximal	Notes
Cost	pennies/vote	dollars/vote	The cost of an election, or more properly, the budget assigned to the electoral authority, is often the main factor that politicians, elections administrators, and at least indirectly, the general taxpaying public pay attention to, at least when an election runs without a hitch. At the low-end, paying pennies per vote for a given election service (e.g., an EPB) is reasonable. At the high-end, several dollars per vote is not unheard of.
COTS	Purportedly bespoke hardware from a single vendor	A system capable of running on any modern off the shelf hardware	Digital elections systems that can run on COTS hardware open up a bevy of possibilities. Hardware is less expensive (and independent of development vendor) in procurement. Jurisdictions can upgrade the elections infrastructure as new, more capable and less expensive hardware comes to market. New technologies, particularly those that assist the disabled and provide greater security, can be taken advantage of as they become available, rather than a decade or more after the fact as we see today in the elections marketplace.
Deployability	No field configuration	Hardware and software deployment and configuration necessary	The deployability of a technology has as much to do with the complexity of deployment (hardware and software pieces, configuration, etc.) as it has to do with pre-election training of IT staff and elections officials. In general, the less deployment overhead, the better. Needing little to no field configuration is at the minimal end of the feature scale; requiring dedicated hardware and per-election or per-polling place configuration is at the upper end of complexity and cost.
General Openness	A proprietary, closed system	A component based, open data and API-standards based system	At the minimal end of the openness scale is a proprietary, closed system that uses undocumented binary file types, cannot be composed with other systems, and can only be maintained and upgraded by a single vendor (i.e., the current state of the industry). At the maximal end of the spectrum is a component based system that operates only on open data, has and uses open APIs, and is built for composition and transparency. This latter class of system is what the new VVSG is working towards and is what *Free & Fair*'s product offers.
Hardware Reuse	Hardware that is only used once per	Hardware that can be continuously	Hardware that can only be used in each election and otherwise sits in a secure warehouse for months at a time is inefficient at best. The opportunity cost lost by such a

	election (a few times per year) for a few years	used in addition to periodically running elections	deployment, coupled with the impact on the environment, is tremendous. For example, rather than buying tens of thousands of iPads to only use them for a few days of time over their five year lifespan and then end up in a landfill, imagine the utility in seeing those same devices deployed for other purposes, day in and day out, outside of the critical time frame around one's elections.
Intellectual Property	Owned by the client	Owned by the vendor and licensed temporarily to the client	There has never been a question of ownership of a digital election systems until very recently. Existing vendors virtually always license technology for use over a fixed time frame. New vendors, such as *Free & Fair*, offer to sell to jurisdictions bespoke versions of open source technology fit for their purposes for a one time cost. The jurisdiction is then free to do with that system as they will.
Legal Constraints	Measure incidence of double-voting	Catch violators in real time	Legal constraints, especially about election operation and citizen and election data handling, are the yin to the threat model/security policy yang when it comes to forcing one toward a particular product. At the minimal end, one might wish to simply detect after-the-fact the incidence of potential double-voting across a region, only reporting on such a figure. At the maximal end, one might wish to catch potential perpetrators of such (accidental or malicious) actions in real time, with the intention to prosecute.
Open Source	Open source under a liberal license	Closed source under a very restricted license	Like the conversation about digital election systems' intellectual property options, open source election systems have never been commercially available until the past few years. We need not recount the obvious benefits of open source systems, especially in matters relating to mission-critical systems like elections, as the literature on such is voluminous.
Procurement	Any legitimate entity versed in the state of the art can compete	A single vendor can submit	The preconditions on procurement dictate how much competition you will witness for your RFP. The vast majority of the time, RFPs in the area of elections systems look like they were written by the current vendors; they set unrealistic demands on proposers, such as "must have run elections in our jurisdiction for the past five years". RFPs that see fierce competition are those that view the development of elections systems as any other IT product, and only place reasonable demands of skill in the art and business practices on bidders.
Security Policy	The minimal policy mandated by law in the domain	A context-free overly aggressive security policy	The security policy mandated by law and operational practice is further framed by the policy promised and realized by a product. One must take care to not define a context-free (i.e., unaware of application domain or legal framing), overly aggressive security policy, which is the high end of this feature, otherwise EPB products will be over-engineered and overly complex. Complementarily, a minimal policy that only conforms to the letter of the law, but not its actual intent, is

			asking for trouble in the long run.
Threat Model	No adversaries	Determined APTs and insider threats	The threat model under which your election operates will be a dominant forcing function for determining which products meet your needs. The minimal threat model—which is completely unrealistic, from both a legal and operational points of view—is that there are no adversaries (purposeful or accidental, human or machine) who wish to compromise election operations in any fashion. The maximal model presumes that state-level actors (such as North Korea or China) wish to compromise your election and you must operate under the assumption that your systems and network *are already compromised today*. Moreover, you must also presume that some election officials with full administrator access to your digital elections infrastructure (your servers, network hardware, devices in the field, etc.) are bad actors and will attempt to compromise the election.
Training	Little or no training necessary	Hours of training per official	Complementing the training inherent in deployment mentioned above, election administrators, officials, and volunteers may need training for a given EPB product. Of course, needing no training whatsoever is the optimal goal; needing hours of training per person is on the high side. The complexity and cost of the infrastructure (both physical and digital) and personnel for such training can be surprisingly expensive.

Table 2: Tabular Feature Model of EPB Constraints

We recommend that you draw on a whiteboard this solution space feature model, as summarized in Table 2, adding in any requirements and features that you have already defined which we have omitted or missed, and reflect upon the implications of each configuration choice.

Dominant Forcing Features

With regards to such configurations, we note that the threat model of your election infrastructure is usually the most important constraint that drives you toward a particular solution. Any legitimate vendor must be able to provide for you a precisely characterized threat model and security policy for their product. If they cannot, we recommend you avoid them assiduously.

Likewise, your positions on open source, IP ownership, and reusable COTS hardware will quickly dominate any other feature selections you make. It is important to note that open source products are much more easy to customize and evolve for your specific challenges and opportunities (e.g., integration with new infrastructure or other products), as you can hire anyone you like to do that development and you are not bound to a particular, closed vendor.

General Recommendations

After reflecting upon Ontario's particular electoral situation, we make the following recommendations with respect to connectivity (and hence, EPB product architecture and requirements, and problem and solution space configuration). The main factors that influence our recommendations are cost, electoral history, geography, and operational deployment plans (i.e., total numbers of voting locations and advance poll locations).

Free & Fair recommendations:

- Leverage existing infrastructure and avoid deploying networking and support hardware used only during an election (of any kind, including switches, LTE modems and SIM cards, printers, scanners, etc.).
- Think about novel means by which to connect voting locations if denying double voting is a requirement of paramount importance (i.e., piggyback on election officials' cell phones, use SMS for communication in areas that are not well served by cellular Internet).
- Aim for distributed solutions that do not require large-scale physical or digital infrastructure. A solution that runs in a public cloud is superior to one that demands a server farm; a solution that needs no server deployment at all is optimal.
- Convince local and national telecom providers to offer equipment, support, and service for the good of the country during elections.
- Reflect upon your historic evidence of election fraud that will be mitigated by a fully networked EPB. We suggest that such a configuration is overkill and the cost/benefit analysis is not in its favor. For example, consider networking only advance poll locations, and do so in a lightweight fashion (e.g., a single LTE modem on a single EPB should be sufficient for all cross-poll location synchronization with an appropriately designed and implemented EPB protocol).

Obviously, our general recommendations in the problem space of EPBs have driven the design of our product. As such, you should take them with a grain of salt and confirm our assertions with other independent technology experts.