NSW Electoral Commission
iVote Core Voting System and associated services
Request for Tender

Attachment B1 - Response Schedule - Phase 1

This attachment contains the Response Schedule required for completion of responses for Phase 1 of the iVote Core Voting System RFT.

The response schedule has been designed to follow the structure of requirements in the RFT document. To assist respondents, references to the requirements as specified in the RFT document have been included against each item in brackets, for example:

[RFT Section 3.2]

Additional supporting material may be attached. Where the respondent intends additional supporting material to be taken into account by NSWEC in evaluating the response to any particular requirement, the relevant content should be clearly identified and specifically cross-referenced in such response, and attachment documents identified in Item 6

It is **mandatory** that tenderers adhere to all contractual and response requirements as defined in Section 3.2. *[RFT Section 3.2]*

Response Schedule - Phase 1

Response cover sheet

Agency	NSW Electoral Commission
RFT Number	RFT-NSWEC-iVote-13/01
RFT title	iVote Core Voting System and Associated Services - Phase 1 Response
Closing date and time	09:30 PM on Tuesday 17 December 2013
Respondent organisation name	Galois, Inc.
Respondent organisation address	421 SW 6th Ave., Suite 300 Portland OR 97204 USA
Trading or business name	Galois, Inc.
ABN	N/A
Name and contact details of officer authorised to represent and legally bind the respondent	Jodee LeRoux jodee@galois.com +1 (503) 626-6616
Name and contact details of person for enquiries regarding this response	Dr. Joseph Kiniry kiniry@galois.com +1 (503) 626-6616
Date of response	9 December 2013

Executive summary

Provide a summary (of no more than 2 pages) to your response to include

- An overview of the organisation's (or organisations', where subcontractors are proposed) experience and capability relevant to the requirements of this RFT
- A brief summary of the proposed iVote Core Voting System Solution, including an outline of the proposed technology, functional modules and custom development, and proposed services, and how they combine to address the requirements of this RFT
- Any information that NSWEC should be aware of before evaluating this response, such as assumptions and limitations.

Galois has extensive experience in high-assurance hardware and software systems, ranging from embedded systems to large-scale secure distributed systems. It is widely considered one of the premier organizations in the world in areas relating to high-assurance system construction, including the use of applied formal methods, cryptography, secure protocol design and verification, and software and hardware verification. Consequently, in terms of general computer science and systems-building expertise, Galois is second-to-none.

Dr. Kiniry, a new Principal Investigator at Galois, has over ten years experience in the design, development, support, and auditing of supervised and internet/remote electronic voting systems while he was a professor at various universities in Europe. He co-led the DemTech research group at the IT University of Copenhagen and has served as an advisor to the Dutch, Irish, and Danish governments in matters relating to electronic voting.

Galois's proposed iVote Core Voting System Solution is a bespoke, fit-to-purpose high-assurance system that, while fulfilling the architecture sketch provided in this RFT, moves beyond the stated requirements in several dimensions. We intend to develop something the world has never seen: a formally verified high-assurance internet voting system that provides end-to-end verifiability, voter-verifiability, and can be completely open source with no risk. We also intend to develop the system in a completely public and transparent fashion from the day the project begins. Finally, we intend to have an public internal review board consisting of internationally recognized experts, both technical and political.

We intend to offer the NSWEC three variants of the Galois Voting System (**GVS**) to choose from, exactly one of which the NSWEC can select: (1) a *standard* hosted system variant running on commodity operating systems, (2) a *minimalist* variant which runs without an operating system or LAMP stack, (3) a *cloud* variant that runs on a public cloud platform. While each variant has pros and cons from the point of view of some of the primary goals of the RFT (transparency & trust; public confidence), all fulfill the technical requirements of the RFT, including the cast-as-count vote verification property.

The technological underpinnings that Galois uses on high-assurance projects includes pure functional programming languages like Haskell, formally verified architectures and protocols, formally verified procedural and object-oriented programming languages like C, Java, and C#, and formally verified cryptographic schemes and protocols.

For the *standard* variant of the Galois Voting System a COTS open source software stack would be reused (e.g., OpenBSD and Linux for operating system, minimalist web

servers like nginx and lighttpd, several high-quality crypto libraries like NaCl, and a cryptographic database like CryptDB). The *minimalist* variant uses no operating system or database, as Haskell and Java code is compiled directly down to a Lightweight/Hardware Virtual Machine. The *cloud* variant is uses the same kinds of technologies as the standard variant, but its protocols differ in significant ways, and it runs on a public cloud platform, like those provided by Amazon and Google.

The volume of bespoke vs. reused code varies considerably between these variants, thus the expected budget of each variant will vary as well. With any choice, Galois focuses upon fit-for-purpose systems that do exactly what requirements dictate—no more, no less.

The Galois Voting System's protocols will be formally specified and documented in such a way that members of the electorate, election officials, or party representatives can develop their own versions of various **GVS** subsystems. The purpose of these independent implementations is to perform public auditing of the election, both live, during the election as well as after the fact by performing a trace-based re-execution of the election based upon secure logs and system traces. Such public auditing capability goes beyond that which the RFT mandates (essentially only a tally audit and a non-transparent, delegated trust election auditor), vigorously pursuing the goals of electronic transparency and electorate trust.

Galois's assumptions with regards to this RFT are that:

A. the NSWEC is open to the possibility of a fully transparent open source election system, including direct public view of the source code repository during system design and development,

B. the NSWEC is comfortable with the use of non-mainstream tools and technologies to pursue its lofty goal of a high-assurance, verifiable internet voting system,

C. the NSWEC is open to modifications to the proposed architecture and protocols to ensure that their internet voting system does, in fact, have the correctness and security properties mandated by law and stipulated by the RFT, and

D. the NSWEC is comfortable with the idea that a premier vendor in high-assurance systems, whose clients include the likes of the Department of Defense and the National Security Agency, is well-capable of developing a first-in-class verified, voter-verifiable electronic voting system on time and on budget, while partnering with a professional services firm to provide on-site and remote support of the voting systems before, during, and after elections.

Contractual and Response Requirements – [RFT Section 3.2.1]

1.1 Response Submission

[RFT section 3.2.1 (a)]

Confirm response submission is in accordance with Section 3.6

Galois confirms that this response submission is in accordance with RFT Section 3.6.

1.2 Response Completeness and Format

[RFT section 3.2.1 (b)]

Confirm responses are provided to all sections of this response schedule.

Galois confirms that this response includes responses to all sections of this response schedule.

1.3 Supply using Procure IT Customer Contract

[RFT section 3.2.1 (c)]

Confirm agreement to provide solution and services under a contract based on Procure IT Version 3.1 Customer Contract

Galois agrees to provide solution and services under a contract based on Procure IT Version 3.1 Customer Contract.

Capability and Experience – [RFT Section 3.2.2]

1.1 Organisation capability

[RFT section 3.2.2 (a)]

Provide a profile of the respondent organisation including areas outlined that demonstrates substantial capability and experience relevant to the requirements of this RFT

Company

Galois' mission is to create trustworthiness in critical systems. We take innovative ideas and turn them into real-world technology solutions through a combination of applied research and engineering.

We work with clients in government and industry to develop solutions that have significant impact on assuring safety, security, and privacy.

Facilities

Galois' office in Portland, OR contains workstations for the engineers with standard office and IT infrastructure. Research is done on equipment and using software owned by the applicant, Galois, Inc. No additional equipment will need to be purchased for this effort.

Year Founded: 1999

Industry / Customer Base

Software assurance / research services

Company Structure & Ownership

Privately held C corporation, incorporated in the State of Oregon

Staff

45 employees (35 engineers, 10 support)

CEO - Dr. Robert Wiltbank

Chief Scientist - Dr. John Launchbury

Representative Clients

- US Department of Defense
- US Department of Energy
- US Intelligence Community
- US Defense Advanced Research Projects Agency (DARPA)
- US National Aeronautics and Space Administration (NASA)

Research & Development

We specialize in the research and development of innovative new approaches and technologies that provide information assurance for challenging systems and software environments.

Our world-class researchers and engineers build upon a solid foundation of mathematics and science to address the most challenging problems today:

- Applied formal methods
- Cryptography
- Language design and compilation techniques
- Security and trusted computing (excluding crypto)
- Systems of systems
- Hardware and cyber-physical systems
- Cross-domain security
- Programming at Internet scale

With our suite of safety and security verification and validation (V&V) tools, you can feel confident that our technologies meet specifications and fulfill their intended purpose:

- Information flow analysis
- Hardware verification
- Driver systems, kernel design
- Type systems and formal mathematical analysis of software
- Automated software analysis for a wide range of properties

- Custom V&V and analysis tools

We believe that the open innovation model for research and development is the most effective means to keep up with disruptive technology advancements. We also continually look to build paths to commercialization through new and existing strategic partnerships.

We care deeply about real-world use of our research and development efforts and work to transition them broadly to existing product/service providers. This supports our desire to spread the benefits of our technologies across a wide variety of government and commercial organizations.

1.4 Experience of Remote Electronic Voting Systems – Supply of Software applications

[RFT section 3.2.2 (b) i]

Detail the organisation's experience in supplying remote electronic voting software applications as described in the Footnote to this RFT section for use in elections of similar scale as proposed for the NSW State General Election in 2015

Galois has not provided electronic voting software applications in the past, but this proposal's lead, Dr. Joseph Kiniry, has over ten years experience in electronic voting systems. Dr. Kiniry recently left a full professorship at the Technical University of Denmark where he was the Head of the Software Engineering Section to join Galois full time as a Principal Investigator. For the past two years he has co-lead the DemTech research group at the IT University of Copenhagen whose entire purpose is research in electronic voting systems. He is also responsible for two security audits of the Dutch KOA internet voting system, he co-authored the high-assurance electronic tally system used in internet European elections for the Dutch government, he has led the development of several other tally systems (e.g., Ireland's PR-STV system, Denmark's list-based scheme, the USA and UK's FPTP scheme), and has led security audits of several internet voting systems including the Dutch and Norwegian internet voting systems, Scantegrity II, and Helios. He is an internationally recognized expert in the design, development, and auditing of electronic voting systems of all kinds.

1.5 Experience of Remote Electronic Voting Systems – Provision of Services

[RFT section 3.2.2 (b) ii]

Detail the organisation's experience in services to develop, implement and support remote electronic voting software applications as described in the Footnote to this RFT section for use in elections of similar scale as proposed for the NSW State General Election in 2015

Galois, by virtue of Dr. Kiniry's move, has extensive experience in the development and implementation of electronic voting systems (supervised and remote). Galois also has extensive expertise in deployment of high-assurance systems in many contexts, ranging from embedded systems with no operating system to secure distributed systems. Consequently, remote electronic voting systems are treated as a new class of secure, distributed high-assurance systems. Galois has no experience in the support of remote voting software, though Dr. Kiniry has supported several open source projects within the domain for over a decade. Galois intends to partner with an appropriate support organization for its phase 2 submission.

Technology and Solution – [RFT Section 3.2.3]

1.1 Proposed Application Architecture

[RFT section 3.2.3 (a) i]

Outline the proposed application architecture and major application components.

The GVS application architecture is, broadly speaking, based upon a strong decomposition of the proposed architecture in the RFT materials into distinct subsystems which have different non-functional properties (e.g., correctness, security, performance, etc.). Consequently, each subsystem may be developed (from formal analysis through verification) using a different technology suite that is fit-for-purpose for the functional and non-functional requirements of said subsystem.

We intend to run each subsystem within a privilege separation architecture, thus compromises in any subsystem cannot impact any other subsystem. We intend for all subsystems to communicate using a mixed-mode cryptographic channel using a message-passing based approach.

We intend to specify the entire architecture using a process calculi like pi-calculus or CSP. By doing so we gain several benefits to system's development, evolution, and infrastructure deployment. For example, we can ensure that the correctness properties of the entire system are verified before system integration. We can also ensure that subsystems can be modified, swapped out, or eliminated altogether when necessary as the NSWEC's requirements evolve in future elections. Finally, deployment of a message-passing architecture like this one scales horizontally and vertically, given the stateless nature of the architecture and the secure network communication substrate.

There are several important non-technical matters to note. First, while we may use different technologies for different subsystems, we do not intend to use a cornucopia of languages and tools, as doing so would make the maintenance and support of the iVote Core Voting System more complex and expensive in the long run. Second, there are excellent public/free and commercial software tools available to reason about the nonfunctional properties of the systems that we are constructing, thus if a third party wishes to double-check our work they need not have access to Galois technology. Finally, while we intend to formally specific all critical algorithms and subsystems, we also intend to document said artifacts in natural language in a manner which is familiar to a standard software engineer. Thus, maintaining and evolving this system is something that others, besides Galois, are capable of.

1.2 Key Technologies

[RFT section 3.2.3 (a) ii]

Outline key technologies on which solution is based

The key technologies used by Galois that differentiate it from any competitor focus on programming languages, formal specification and reasoning, and automated and interactive algorithm and program analysis. A few examples should suffice for this phase 1 submission.

Frequently, critical subsystems or algorithms at Galois are written in Haskell, an expressive, pure, strongly-typed functional programming language. By using an advanced programming language like Haskell, such subsystems have very high assurance guarantees, are much smaller and easier to understand and evolve than if they are written in OO or procedural programming languages like Java or C, and yet are of high performance. Algorithms specified in this fashion are used as reference specifications of behavior. A "consumer" version of the same algorithm is written in a mainstream programming language like Java or C and we use advanced verification technology to compare the behavior of the "consumer" version to its specification, guaranteeing correctness.

Alternatively, Galois also specifies systems like this using an embedded domain specific language (EDSL) in Haskell and synthesizes a program in C or Java. Using a EDSL-based approach we can also synthesize a high-level concurrent specification of the architecture, as discussed in the previous section, in CSP or similar. Doing so helps us keep the architecture and system perfectly synchronized, witnessing no architecture drift or erosion. An example Galois project that followed this methodology is the DARPA HACMS project, which is released under a BSD license, and development work continues within a public GitHub repository[1].

Galois has developed a suite of symbolic simulation and formal analysis tools, called the Software Analysis Workbench (SAW), which provide security analysts and engineers with the ability to generate formal models from C and Java programs, and analyze them using automated verification tools including SAT and SMT solvers such as abc and yices.

SAW has been primarily designed for automated verification of cryptographic implementations. It has been used to prove functional correctness of different AES implementations from libgcrypt, OpenSSL, and Bouncy Castle with minimal help from the user. SAW also supports compositional verification for more difficult implementations. This capability has been used for verifying libgcrypt's implementation of SHA-2, and a high-performance implementation of a digital signature verifier based on Elliptic Curve Cryptography. In fact, Galois's verified implementation of ECDSA is an order of magnitude faster than Bouncy Castle's.

Galois's capabilities in producing formally verified algorithms with outstanding performance is the perfect mix for large-scale cryptographic algorithms, like those used in computer-based elections.

[1] http://smaccmpilot.org/

1.4 Scrutiny and Elector Trust

[RFT section 3.2.3 (a) iv]

Outline how the proposed solution delivers security, privacy, verifiability, reliability, scalability and performance features, that will stand scrutiny and maintain elector trust as an election platform for public government elections for NSW, and provide flexibility to meet future requirements

Galois believes that scrutiny and electoral trust is best supported by a transparent approach to systems engineering coupled with the use of best-in-class rigorous software engineering technologies, particularly those that rely upon automated verification of system code and algorithms.

A transparent approach to systems engineering means that we believe that all artifacts related to the analysis, design, development, validation, and verification of the iVote Core Voting System should be in the public eye, open to scrutiny from citizens and experts alike, from the first day of the project. As such, we would host all development on GitHub and would directly engage with technical input from scrutinizers during development.

We also believe it important to base critical system algorithms, like those used in elections' technology, on peer-reviewed research results. In the context of iVote, this means adopting customized versions of published algorithms like Remotegrity[2] and its brethren to fit the requirements of the NSWEC.

Electoral trust would also be strengthened by involving key technologists, political scientists, and activists early and often in the design of the Core Voting System. We intend to approach high-profile academics, independent consultants, and activists with whom we are acquainted as a public internal review board on the project. Examples of experts we could engage would include technologists like Beneloh and Halderman, consultants like Wikström and Schürmann, political scientists like Alverez and Hall, and activists like Gonggrijp and Harris.

Our technical approach that supports the validation and verifiability of non-functional electoral properties like security, privacy, and verifiability hinge on the formal specification and verification of said properties for the system as a whole, as well as of the implementation of said system. True, only experts can review and comment upon said artifacts, but public statements from numerous experts on the rigor, clarity, and high-assurance engineering that Galois regularly delivers holds enormously more weight than a traditional non-technical review by stakeholders.

Note that a public approach to systems engineering does not conflict with the licensing requirements of the NSWEC nor the intellectual property obligations and mandates of Galois. Open development does not mandate an open source license, though we would argue for the use of such, nor must only a single license apply to the entire system. Different subsystems can be licensed in different fashions, and dual- or multi-license situations are possible. E.g., one might license using GPL for the purpose of academic research, development, & teaching and and non-binding elections, and then use a more traditional commercial license for the purpose of running binding elections.

[2] Zagórski, Filip, et al. "Remotegrity: Design and Use of an End-to-End Verifiable Remote Voting System." IACR Cryptology ePrint Archive 2013 (2013): 214.

Finally, we suggest that clean-room implementations of the iVote verification and audit subsystems is necessary for complete public scrutiny and electoral trust.

1.3 Functional Architecture

[RFT section 3.2.3 (a) iii]

Outline how the required functional modules are addressed by the proposed application architecture

Our answer to question 1.1 above covers these matters sufficiently.

Attachments

Please list below any additional supporting material and attached documents provided as part of this response:

Filename	Document title	Description	Phase 1 Response reference
Binary.pdf	Automated Tools for Binary Analysis & Optimization	A summary of an ONR project whose tools are relevant to this RFT.	1.2 response
Cryptol.pdf	Empowering the Experts: High-Assurance, High-Performance, High-Level Design with Cryptol	A summary of the Cryptol tool and its capabilities.	1.2 response
HaLVM.pdf	Galois releases the Haskell Lightweight Virtual Machine (HaLVM)	A summary of the Haskell lightweight virtual machine.	1.2 response
Kiniry07.pdf	Formally Counting Electronic Votes (But Still Only Trusting Paper)	A peer-reviewed academic paper summarizing work on e-voting in several countries.	on matters of experience with electronic voting
KiniryEtAl06.pdf	The KOA Remote Voting System: A Summary of Work To-Date	A peer-reviewed academic paper summarizing security and engineering work on a Dutch internet voting system.	on matters of experience with electronic voting
KiniryEtAl07.pdf	Verification-Centric Realization of Electronic Vote Counting	A peer-reviewed academic paper summarizing work applying formal methods for election software.	on matters of experience with electronic voting
PROCEED.pdf	Galois to Help DARPA PROCEED to Change the Game	A summary of a DARPA project that uses technologies relevant to this RFT.	1.2 response
SAW.pdf	SAW: The Software Analysis Workbench	A summary of Galois' SAW technology, which is relevant to this RFT.	1.2 response
SMACCMPilot.pdf	SMACCMPilot: Open-Source Autopilot Software for UAVs	A summary of Galois' DARPA project that uses both relevant technologies and a transparent process similar to that which is proposed.	1.2 and 1.4 response