Request for Quotation

Victorian Electoral Commission

QUOTE REQUESTED FROM:		QUOTE REQUESTED BY:
Joseph Kiniry *kiniry@galois.com* *Galois, Inc.* *421 SW 6th Ave., Suite 300* *Portland OR 97204* *USA*		Request for Quote No: vvote6 Date issued: 1/11/2013 Requested by: Craig Burton, Manager E-voting and Special Projects Telephone No: +61 3 8620 1178 Facsimile No: +61 3 9629 8632 **Please submit via email with subject line:** "Request for Quote for Pre-election e-voting Expert Analysis" To: *craig.burton@vec.vic.gov.au* By COB on: 30/11/2013

Details of Goods / Services sought:

Please provide a quote for goods and / or services as detailed in the attached specification (**Pre-election e-voting expert analysis FINAL**).

The proposed form of contract to apply is also attached.

Please note that this is **not** an instruction to proceed with the supply of any goods and / or services (unless and until the Department signs and returns the Acceptance section below).

Details of Quote:

Please complete your response to this Request for Quotation for goods and / or services in the spaces provided below.

Breakdown of quoted rates/fees or cap (on a GST inclusive and exclusive basis):

We propose a capped cost for the full body of initial audit work described below at $184,837 AUD (all costs quoted are GST-exclusive). The capped total cost for a follow-up audit is one quarter of the initial audit, thus 152 hours at $46,209 AUD (GST-exclusive). A two day kick-off meeting for the project is budgeted at $8,256 (GST-exclusive). Thus the total cost for a full audit, kick-off meeting, and re-analysis is $239,303 AUD (GST-exclusive). The cost of various experts on the project ranges considerably, thus if the VEC chooses a subset of the work to be performed based upon the breakdown of the work, including prioritization and cost, that follows, the computed fractional cost of said hours (as indicated by the below effort table) is only a rough approximation of the actual costs, as Galois will have to identify which experts' time are necessary to perform the chosen subtasks.

The time required to complete the contract from the date of notification of Acceptance:

Initial review and audit needs 608 man-hours of work to be completed in 4 weeks of calendar time. A follow-up audit after the system's developers perform updates/corrections based upon the original audit is expected to take 1/4 the time and effort of the original audit.

The numbers on audit types in the table below conform to the enumeration provided in the RFQ. Audit types are ordered from most critical to least critical from top to bottom and left to right. Thus, critical subsystem testing of the MBB is the most important audit to perform, while analyzing the build system of the VPS is the least important. A set of deliverables, including inter-dependences, is included below as well. If the VEC wish to select a subset of the work to be completed, both these suggested priorities and their associated dependencies must be taken into account.

The Protocol Audit type is described below and is not highlighted in the RFQ. After reading the relevant public related materials we recommend that a Protocol Audit be performed to ensure that the published protocol has, in fact, the correctness and security properties desired by the VEC. Consequently, we have added a Protocol Audit row to the chart and labeled it audit type #10.

VEC audit effort estimates

	fractional contribution	**MBB**	**MIX**	**EVM**	**VPS**	**totals**
fractional contribution		**30%**	**30%**	**30%**	**10%**	**100%**
9. Critical Subsystem Audit	25%	45.6	45.6	45.6	15.2	152
10. Protocol Audit	20%	36.48	36.48	36.48	12.16	121.6
2. AppSec	20%	36.48	36.48	36.48	12.16	121.6
4 & 6. Existing Test Quality Audit	5%	9.12	9.12	9.12	3.04	30.4
7 & 8. Functionality Matching	15%	27.36	27.36	27.36	9.12	91.2
1 & 3. Source/Doc Quality	10%	18.24	18.24	18.24	6.08	60.8
5. Build System	5%	9.12	9.12	9.12	3.04	30.4
totals	**100%**	**182.4**	**182.4**	**182.4**	**60.8**	**608**

Performing audits conforming to the validation types enumerated in the RFQ #1, 3, 4, 5, 6, and 10 will follow best-in-practice techniques and use the most advanced research and industrial tools of the trade in rigorous software engineering with Java for the MBB, MIX, and VPS Java-based component. Performing these validation types on the EVM and subsystems of the VPS that are HTML and Javascript-based rely much more upon peer-review, as advanced tool support in this domain is poor. As described in the RFQ, penetration testing and capacity/stress testing are not within the scope of this proposal.

Audit types #2 (AppSec), #7 and #8 (functionality matching), #9 (critical subsystem audit), and #10 (protocol audit) require much more specialized expertise and tools and techniques, including formal specification writing about protocols and source code and runtime and static verification using logic-based verification techniques and model-checking which depend upon said specifications.

In general, our approach pursues three lines: (1) a focus upon the properties of the voting system protocol and the question of whether the implementation conforms to that protocol, (2) the question of whether the implementation of the system is of high quality, is correct, and is secure, and (3) an evaluation of the quality and character of non-source artifacts. These three lines are orthogonal, as (2) can be pursued with the assumption that the protocol is correct and secure, which is a property that can be verified by pursuing line (1). Furthermore, (3) can be pursued independent of the other inquiries, as evaluating the quality of software engineering artifacts (including documentation, analysis, architecture, etc.) can be accomplished independent of a source audit.

With regards to inter-audit type dependencies, audit types #9 and #2 rely upon the results of audit type #1 and #3, and audit types #7 and #8 depend upon the results of #9 and #10 of the relevant subsystems. Thus, for example, to perform audit types #7 or #8, and #9 or #2, on subsystem MBB, audit types #7, #8, #1, and #3 are mandatory.

Based upon our experience with past audits and system construction of both election systems and cryptographic systems and libraries, we estimate that 30% of the total effort must be devoted to each of the MBB, MIX, and EVM systems, and the remaining 10% to the VPS system. Likewise, we estimate that the fractional effort needed to devote to the various audit types is summarized in the above table, ranging from 25% of effort for critical subsystem audit to 5% for auditing build systems and existing test quality audit.

The primary technologies we intend to use are the Java Modeling Language and its supporting tools (primarily OpenJML and JMLunitNG) and a combination of Haskell, protocol model checkers, and F* (for specifying the system's protocols for protocol conformance validation and verification). We also use a host of static and dynamic verification tools for the JVM platform.

Of course, the specific choice of applicable tools can happen only after the team has received the system's code and can triage techniques according to its size, complexity, language use, and architecture.

To perform mechanically-supported validation and verification we intend to write JML contracts for critical subsystems and formally specify the system's protocols in Haskell and F* in order to perform the appropriate analysis of various subsystems.

Using these techniques it is common to find numerous problems with both the specification of high-assurance systems (where specifications are both formal and informal, and the problems are both correctness and security issues) as well as the implementation of said specifications (i.e., there are functionality mismatches).

The kinds of properties that we will focus upon are primarily those that either cause the relevant systems to perform erroneously under certain conditions (e.g., compute incorrectly, leak information inappropriately, crash, etc.) or permit the relevant systems to be manipulated by external or internal parties (i.e., election malfescence by outsiders or insiders before, during, or after and election). Specific properties commonly found include overflow/underflow errors, null pointer dereferences and other Java- and Javascript-specific crash flaws, information leakage, improper input/output validation, logic errors in computations, improper assumptions about operating environments, and the OWASP top ten vulnerabilities including injection, cross-site scripting, broken authentication and session management, security misconfigurations, etc.

We will not be performing a full mechanical verification of the systems under audit. That would take significantly more resources than we have budgeted for here. We will make best effort to verify critical subsystems and the system's protocols, if those tasks are chosen in the bid response.

Names of key staff that will complete the contract (where appropriate):

The consortium that we have constructed to fulfill this audit represent represents five organizations and includes some of the top experts in the world in the areas of this RFQ, especially in terms of election systems, cryptography, application security, and rigorous software engineering in Java. They are as follows:

1. Joseph Kiniry (Galois)

2. David Cok (GrammaTech)
3. Daniel Zimmerman (Harvey Mudd)
4. Douglas Wikström (KTH)
5. Martijn Oostdijk (Novay)
6. Aaron Tomb (Galois)
7. Joe Hendrix (Galois)

The key staff performing the audit will be led by Joe Kiniry, a recognized world-expert in constructing and auditing election systems, application security, and software engineering for high-assurance systems. He has led audits of several national electronic voting systems over the past decade including the The Netherlands' KOA remote voting system, Norway's internet voting system, the Estonian internet voting system, and the Scantegrity II voting system. He will contribute to all technical aspects of the project.

David Cok is the author of OpenJML (a typechecker, runtime assertion checker, and extended static checker for JML-annotated Java), and Dan Zimmerman is the author of JMLunitNG (a unit test generator for JML-annotated Java), and both, along with Joe, are recognized world-experts in the specification and construction of rigorously engineered Java systems. Martijn Oostdijk has experience in both election systems construction and audits as well as application security audits. Douglas Wikström is the author of the Verificatum crypto system for Java and a recognized world-expert in application security and applied and pure cryptography and the application of cryptography to election systems. Finally, Galois staff Aaron Tomb and Joe Hendrix are experts in the rigorous construction and V&V of high-assurance systems, including the use and analysis of Haskell and C, the V&V of cryptographic algorithms and protocols, and the application of mechanical static and dynamic verification techniques for application correctness and security.

Response to any other relevant matters referred to in the Specification:

After reading the relevant public related materials we recommend that a Protocol Audit be performed to ensure that the published protocol has, in fact, the correctness and security properties desired by the VEC. Consequently, we have added a Protocol Audit row to the above chart.

The schedule for this audit is very aggressive. The specified staff are available during the proposed time interval, but their availability is subject to change if the schedule is changed significantly.

Will a potential conflict of interest occur?

If Yes, or the possibility exists, provide details:

No

Proposed amendments to the contract (if any):

Galois' bid is contingent upon the following:

1. Modification of the warranties section of the terms and conditions associated with this effort to read as follows:

The Supplier warrants to the Department that:

(a) (Purpose) where the Department has, either expressly or by implication, made known to the Supplier any particular purpose for which the Services are required, the Supplier will utilize best efforts to ensure that the Services will be performed in a way as to achieve that result;

(b) (Conflict) it and its employees, agents and contractors do not hold any office or possess any property, are not engaged in any business or activity and do not have any obligations whereby duties or interests are or might be created in conflict with or might appear to be created in conflict with its obligations under this Agreement; and

(c) (IP) to the best of Supplier's knowledge, it is entitled to use and deal with any Intellectual Property which may be used by it in connection with the Services;

2. The work performed (both technical work and the associated report) is released under an open-source license at the same time that the full vVote system is released under an open-source license;

3. The Supplier (including the Supplier's consultants and contractors) is permitted to publish the results of their work in a peer-reviewed forum upon the earlier of (i) the date the system is released under an open-source license, or (ii) two quarters after completion of the technical work; and

4. The Supplier (including the Supplier's consultants and subcontractors) retains ownership of all the tools and techniques used in performing the technical work.

Offer:

The Supplier offers to supply the goods and / or services detailed in the Specification (i) at the fees and charges offered (ii) within the period offered and (iii) on the terms of the attached proposed contract and any amendments which have been offered.

Signed for and on behalf of the Supplier by (who represents that they have the authority to bind the Supplier)**:**

Name and position Joseph Kiniry, P.I.

Signature JRK

Dated this…………30th…………….day of ……November…...2013……..

Acceptance:

The Department accepts the offer of the Supplier to supply the goods and / or services as set out in the Offer section above.

Signed for and on behalf of the Department by:

Name and position……………………………………………………………………
Signature………………………………………………………………………………

Dated this……………………….day of …………………………………...20……..