# Revitalizing Elections with Verifiable Internet Voting for the United States (REVIVUS)

Free & Fair

2 October 2020

Joseph Kiniry, Daniel Zimmerman, Joey Dodds

FREE & FAIR

# Executive Summary

Free & Fair proposes to lead a team of world-renowned, internationally recognized and respected experts in the first constructive phase of attempting to create an end-to-end verifiable Internet voting (E2E-VIV) platform. Phase 1 focuses on defining the precise, measurable, independently verifiable requirements for such a system. This project is called *Revitalizing Elections with Verifiable Internet Voting for the United States (REVIVUS)*. The estimated period of performance for Phase 1 is approximately 7 calendar months with a budget of $471,000.

We also describe our vision for the subsequent phases of this project: what the design of an E2E-VIV system might look like, taking into account recent R&D in elections technologies, software development, system and network security, and secure hardware.

In 2015, Free & Fair principals led the team that crafted the high-level requirements for such a system. The result of that project, a report called *"The Future of Voting: End-to-End Verifiable Internet Voting — Specification and Feasibility"* (henceforth, the *FoV Report*), is the de facto reference against which all Internet voting systems are measured.

Now, around five years later, it is time to take the next step in this democratic R&D journey. It is time to refine and polish the specification found in the *FoV Report* and plan for the critical R&D necessary to create a 21st century Internet voting system. While the *FoV Report* is incomplete in some respects (for example, it omits a detailed discussion about blockchains and any discussion about the criticality of hardware security), it is a strong foundation for this project.

This work must be framed and informed by the excellent work of our colleagues, which is found in a handful of sister reports such as those published by the National Academies of Science, Engineering, and Medicine (NASEM), the Brennan Center for Justice, and the European Commission (see the Bibliography). Likewise, we must seriously reflect upon the lessons learned both from other nations' (e.g., Estonia, Switzerland, Norway, Australia) attempts at deploying Internet voting systems and from attempts to deploy commercial Internet voting products here in the U.S.A. (e.g., Voatz and Democracy Live). Finally, we must recognize the politics involved in the deployment of a remote voting system of any kind—Internet or absentee.

Since this project is widely recognized in the R&D community as a *Grand Challenge Project*, we must define precise and measurable project milestones so that we know where we are heading and how to determine whether we have reached our destination. We must precisely describe both desired and undesired outcomes. These outcomes are especially critical, as we must be able to compare and contrast this new kind of voting with traditional voting, whether it be vote-at-home or voting in-person at a polling place.

The creation of an actionable, computable risk and threat model is especially critical to these comparisons, as it is a fundamental input into system design and implementation decisions. Such a model must be based upon peer-reviewed from-the-field data and be frank about the threats to our democracy and voting processes posed by adversaries, foreign and domestic.

We must be clear-eyed in the challenges and opportunities afforded by the vision of having radically increased citizen engagement and voter participation via a convenient, usable, accessible, and secure voting system.

# The Team

**Free & Fair.** Free & Fair is a public benefit corporation (B-Corp) whose main office is in Portland, OR. Free & Fair and its principals, Josiah Dodds, Joseph Kiniry, and Daniel Zimmerman, are widely recognized as the top R&D firm in the world in high-assurance elections technology and election cybersecurity. Some of our past projects include creating the reference specification and implementation of Microsoft's ElectionGuard end-to-end verifiable software developer's kit, Colorado's Risk-Limiting Audit system used for statewide audits of state and federal elections, and the U.S. Vote Foundation's Future of Voting Report, whose entire focus is on defining the requirements for any end-to-end verifiable Internet voting system. We also support federal, state, and local authorities in the U.S.A. and Canada on matters relating to the trustworthiness and security of elections and elections technologies. Free & Fair strongly believes in open technologies, its principals have been prominent members of the open source community since the late '80s, and every project that Free & Fair has ever executed results in open source or open hardware technologies. As an employee-owned company we care about not only the technologies we develop but also the path we take to create them, and we inherit the ethos of our parent company, Galois (discussed below).

**Center for Civic Design.** The Center for Civic Design and its principal, Whitney Quesenbery, focuses on design principles for election systems and the standards and principles for the machinery of democracy. They are leading authorities on usability and accessibility for ballot design, ballot standards, and voting systems, having been deeply involved in developing the Voluntary Voting System Guidelines since the very first draft, and continuing to today. Their research looks for new ways to engage voters and run elections, including ballot marking systems for vote-by-mail and vote centers, electronic poll books, and voting methods like ranked choice voting.

**Galois.** Galois specializes in the research and development of new technologies that solve the most difficult problems in computer science. They are passionate about the trustworthiness of critical systems, and work to ensure that the systems clients depend on work as intended, and only as intended. Galois's clients, including DARPA, DHS, the DoD, the IC, the DoE, and NASA, derive value working with Galois as a trusted advisor and hold Galois to high standards for the production of algorithms and code that embody joint work. Galois is a privately held U.S.-owned and -operated company established in 1999 in Portland, Oregon, founded on the core principles of deriving joy from work, being authentic and trustworthy in all interactions, and cherishing the opportunity to develop and steward new technology.

**KAIA.** KAIA [kī-a] delivers Design, Engineering, & Communications for the world's most innovative organizations. Over two decades, they have crafted hundreds of projects for Fortune 100 firms like Nike, Sony, and Disney, and institutions like NASA, the FAA, and MIT Media Lab. Jon Levy, KAIA's principal, is an Emmy Award-Winning Design Director and served as Studio Director of the NASA Advanced Concepts Laboratory, a digital innovation studio that gives form

to vehicles, missions and technologies through conceptual design and visualization. KAIA believes the best work deserves recognition, which is why they eliminate confusion and present their clients' concepts so that decision-makers listen.

**nVotes.** nVotes, led by its technical principals Eduardo Robles and David Ruescas, designs, develops and provides open source secure electronic voting software. nVotes has over a decade of experience in the development of citizen participation systems and secure electronic voting and more than 2 million votes counted with their online voting platform. They have experience at the technical level in designing and development cryptographically secure voting systems in the context of digital democracy as well as managing elections and their protocols. Their real world experience includes both private as well as institutional actors, ranging from small single contest elections up to multi-ballot electoral processes.

**Oxide Design Co.** Oxide Design Co. is a branding and design firm established in 2001 and led by its principal Drew Davies. Their work spans the consumer/retail, business-to-business, and public/civic design spaces, and they work regularly with states, counties, and federal government agencies. Oxide served as part of the core design and research team that developed the U.S. Election Assistance Commission's national ballot design standards, and subsequently helped develop the Field Guides to Ensuring Voter Intent—pocket-sized guides containing field-researched, critical election design techniques that help ensure that every vote is cast as voters intend. In addition, Oxide designed the Anywhere Ballot online ballot marking interface: a thoroughly tested, highly usable tool allowing U.S. military, overseas citizens, and persons with physical or cognitive disabilities to vote more easily. Building on that expertise, they also designed the UI/UX for Microsoft's ElectionGuard ballot marking device.

**Computer Security Lab, Rice University.** The Rice University Computer Security Lab, led by Professor Dan Wallach, started in 1998 when Dan joined the Rice University Department of Computer Science. Their work has considered a variety of topics in computer security: web and smartphones, peer-to-peer networking, electronic voting systems, and they are also actively engaged in the policy side of security issues. They collaborate widely, as computer security issues have broad impact well beyond traditional computer science. Dan is the IEEE Representative to the EAC's TGDC (Election Assistance Commission, Technical Guidelines Development Committee), served on the Air Force Science Advisory Board from 2011–2015, and worked on the Python implementation of Microsoft's ElectionGuard in 2020.

**SpiderOak.** Founded in 2006, SpiderOak's mission is to protect the world's data. They have invented leading-edge security and privacy technology and are unashamed to take chances and be creative where others don't. With tens of thousands of customers worldwide depending on their No Knowledge approach to security, SpiderOak offers customers all the advantages of cloud computing without the risks trusting other peoples' computers. All of SpiderOak's products are built to be secure from the ground up; unsafe modes of operation are not offered. SpiderOak's principal supporting this project is their CTO, Jonathan Moore.

**Verificatum.** Verificatum, led by its founder and principal Prof. Douglas Wikström, specializes in designing, analyzing, and implementing advanced cryptographic protocols as well as advanced consulting in cryptography. Verificatum is based in Stockholm, Sweden. Verificatum is a spin-off company of the School of Electrical Engineering and Computer Science (EECS) at the KTH Royal Institute of Technology, one of the top universities in Sweden. Douglas is responsible for the creation of the Open Verificatum Project, which is a collection of free, open source, and correct implementations of provably secure cryptographic primitives and protocols that are easy to deploy and use to implement a wide range of electronic voting systems.

## Other Industrial Partners

We believe that major corporations like Amazon, Apple, Google, and CloudFlare should be directly involved with this project, especially during Phase 1, for three reasons:

1. **Public Good.** This technology has the potential to have a massive positive political impact in the world. If in fact the invention of an E2E-VIV system trusted by the public dramatically increases citizen participation in elections, , politicians become more accountable to their constituents, and trust in the democratic process grows, the resulting public good will reflect well upon those companies that supported the work.
2. **The System is the Infrastructure.** Any Internet voting system is composed of every device used by every voter, election official, and observer that participates in the election. Infrastructure provided by some or all of these companies will inevitably be part of any realization of an E2E-VIV system. Existing Internet voting technologies like those created by Voatz and Democracy Live use servers hosted by cloud providers and include clients that run on mobile devices created by Apple, Google, and Android licensees. Because these companies provide services and sell devices, they will be a part of any IV system whether they like it or not. Thus, they should be involved early in the definition of the requirements and threat models of the system, if only to protect their own brand reputations.
3. **Grand Challenges Create New Technologies and Services.** Constructing an E2E-VIV system is a Grand Challenge of computing. The requirements are seemingly inconsistent, with expertise required in several highly technical areas including cryptography, hardware security, software verification, usability and accessibility, network security, and law. Solving even some of the subproblems required to realize this system will likely lead to the creation of new technologies and services that may open up new business opportunities, in addition to being useful in their own right.

Our early outreach to Amazon, Apple, CloudFlare, and Google has suggested that their involvement be of the following form:

1. very lightweight involvement (no more than 8 hrs/month) from the appropriate expert(s) in key technologies relevant to the project within the organization;
2. peer review of key requirements and properties, as direct technical input “from the horse’s mouth” is critical to both (a) ensuring that assumptions, claims, and requirements and realistic and achievable, and (b) building trust in the outcome of the project due to

the direct involvement of the companies responsible for the technologies on which it depends; and
3. the commitment to be transparent about the technical work, ensuring that peer review is completely grounded in science and technology, not business or political motivations.

It is expected that the organizations or individuals that are involved from these companies will be compensated for their time as subcontractors to Free & Fair. We have only rough estimates based upon past collaborations with these entities as to the cost of their experts' involvement.

During execution (Phases 2 and 3), this kind of project would also require significant resources from device and OS providers (Apple and Google), cloud providers with HSM support (Amazon, Google, and Microsoft), and a DDoS mitigation provider like CloudFlare. In addition, several key individuals from the research community will have to switch gears in their careers and dedicate their time to the project.

## Election Officials

It is critical that current and former election officials are directly involved in this project from its earliest days. We have worked extensively with election officials on prior projects, but have not yet approached any to be on the team. Election officials nationwide are extremely busy preparing for November's general election, and we believe it will be more effective to involve them after Tusk Philanthropies decides to move forward with our proposal.

# Technical Proposal

This project focuses on the grand challenge problem of defining an end-to-end verifiable Internet voting (E2E-VIV) system for mobile devices that could someday be used in public federal elections in the U.S.A.

We have six main objectives. They are to *define the problem*, set the *requirements* on the system, define a *threat model*, perform a *risk analysis*, *identify the science and technologies* that are likely relevant to realizing the system, and to define *objective, independently verifiable properties and measures* that justify progress.

Today's Internet voting systems are either built for or by nation-states (e.g., those used or experimented with in Australia, Estonia, Norway, and Switzerland [Halderman and Teague, Springall et al., Markussen et al., Franke]) or built by private companies (such as Democracy Live, nVotes, Smartmatic, and Voatz, and the now-defunct Scytl, Everyone Counts, Votem, and numerous others that no longer exist). Some of the nation-state systems were built with the help of other companies, such as Scytl and Verificatum.

None of those systems have been engineered using anything approaching rigorous systems engineering. None of them have been designed and assured as a nationally-critical system. None of them have used formal methods at all. In fact, very few of them even have a threat model that is viewed by the elections research community as rising to the level of mediocre. This is *especially* true of commercial technologies like those of Democracy Live, Smartmatic, and Voatz [Specter and Halderman, Springall et al., Specter et al., Trail of Bits].

This state of affairs is unacceptable. No voting system should be used for public democratic elections in the U.S.A. that is not vetted as a nationally-critical system, which is to say, that the continuation of our democracy depends on our voting systems, and our voting systems must be robust in the face of foreign nation-state adversaries who can and will attempt to disrupt it. Any Internet voting system that uses mobile devices brings our voting systems much closer to the reach of these adversaries, who are willing to bring enormous computational and fiscal resources and brilliant hackers to bear on the challenge.

Our approach is fundamentally different from all other voting systems ever developed worldwide. The *only* way to build a system in the face of this adversarial frame is to adopt the very same assumptions made by every comparable (mission-critical, nationally-critical, national security) system, known as Shannon's maxim: 1) assume your adversary knows and understands your entire system, and 2) assume your adversary (foreign or domestic) has compromised the system and has administrative access to the system's functions and controls.

Even when working from these assumptions, it is often possible to define and create a system that can both operate correctly in the face of that adversity and facilitate detection and mitigation if any fundamental tenet of the system is compromised by the adversary.

We believe that our approach will be successful because we are following in our and others' well-worn footsteps. Many critical systems have been created in this fashion by others. Many have been created by us.

Additionally, we have both the reputation and the public evidence to demonstrate we are the perfect team for the task. Free & Fair is widely recognized as the top R&D firm in

high-assurance election systems in the world. The company from which Free & Fair was birthed, Galois, is widely viewed as the top applied formal methods company in the world. Finally, we three principals at Free & Fair and Galois (Drs. Dodds, Kiniry, and Zimmerman) are widely viewed as international experts in all of the domains relevant to this project.

Creating a high-assurance, secure, usable, accessible end-to-end-verifiable Internet voting system would revolutionize the means by which citizens vote for representatives in public elections. Additionally, *the same infrastructure can be used to permit citizens to engage with their government in a trustworthy manner for a variety of other purposes*. Other services enabled by this infrastructure could include voter, candidate, and proposition registration, citizen polling by local government and representatives, and private elections.

This project has substantial risks, insofar as it is widely viewed that solving the problem of Internet voting—even to make it comparable in risk and utility to vote-at-home today—is a high hurdle. *There is no research-free straight engineering path from where we sit today to a working system.* Not only are there scientific and technological risks, but *there is a non-zero risk, which we cannot quantify, that governments, election officials, or citizenry will reject any Internet voting system due to social, psychological, or political factors.* Our expertise is research and development, not politics. We hypothesize that one way to help mitigate some of these risks is through transparency, humility, and communication.

We have built a world-class team to tackle this challenge. It is led by experts at Free & Fair and Galois, and includes a internationally renowned cryptographer whose specialization is cryptography for elections, a company that has developed an Internet voting system that has been used in local elections for years, the top nationally-recognized experts in user interfaces for voting that are usable and accessible, an Emmy-winning conceptual and industrial design firm, and more. Additionally, we have already had preliminary project discussions with the major Fortune 100 companies on which any Internet voting technology will run, including Apple, Google, and Amazon.

We currently estimate that this first phase of the project will require approximately a six month period of performance to form the full working group, craft requirements, identify the most promising R&D to solve the problem, and lay out an R&D project plan that includes milestones, effort analysis, and risk analysis. Our current estimate for the first phase project budget is $471,000. We have found on numerous projects that time, effort, and cost spent on project planning has paid dividends during project execution, saving along all three dimensions in the long run.

Previous conversations in the community around the cost of the full R&D endeavor necessary to solve the grand challenge of Internet voting have often estimated the timeline and effort as years of work and dozens of FTE of effort. Investigations into starting an R&D program at DARPA with this goal resulted in a comparable estimate of timeline and funding. Of course, until a precise project specification exists—the outcome of our proposed phase one—we cannot accurately estimate the time and effort required to complete the subsequent phases of the project. It is foreseeable that the systems engineering required will take a reasonable-sized team of world-class experts and engineers several years to complete at a cost of tens of millions of dollars.

In order to ensure that this project is on the right track, we insist that this work should be done in the public eye, peer-reviewed by invited adversarial experts and the public. We have defined a project plan with milestones that are independently publishable and peer-reviewable. This gives us, Tusk Philanthropies, and the community observing and critiquing the project measurable outcomes that, if accomplished with quality, transparency, and diligence, will provide a means to slowly, carefully, with honesty and integrity, build trust and confidence in the project's outcomes.

All phases of this project must reuse existing high-quality peer-reviewed artifacts in order to leverage the effort and expertise of others and build on the reputation and quality of existing work. For example, the European Commission's report "Recommendation /Rec(2017)51 of the Committee of Ministers to member States on standards for e-voting", the NIST report NIST-IR 7551 "A Threat Analysis on UOCAVA Voting Systems", and reports published by the Swiss Confederacy contain extensive threat and risk models for Internet voting. Additionally, there are some underlying principles of threat models for critical infrastructure published by the DHS and the DoD that are relevant to this work.

Likewise, the excellent work from NIST on threat models for mobile devices and infrastructure (NIST-IR 8144), security considerations for remote UOCAVA voting (NIST-IR 7770), security best practices for the electronic transmission of election materials for UOCAVA voters (NIST-IR 7711), and information security best practices for UOCAVA-supporting systems (NIST-IR 7682) must all be reused and leveraged for this phase of the project.

Reuse leverages both constructive existing work as well as anti-patterns the team should learn from. For example, Voatz's threat model is an example of a deficient threat model.

In the following sections we discuss each phase of this project. Early phases are described in greater detail than later, more speculative, phases.

Once again, *we propose that at this moment only phase 1 should be funded*. The outcome of phase 1 is critical not only to Free & Fair as a potential performer for phase 2 and beyond, but to any organization that is responsible for any phase of the project, and any third-party peer-reviewer, evaluator, or certification authority.

After phase 1 is complete and its outputs are peer-reviewed, a decision should be made about further work. One possible outcome is a decision that the project should not progress. Perhaps the problem is deemed too difficult or expensive. Perhaps the risk analysis shows that, even if the technology is created, its risk to democracy and trustworthy elections is too great.

Another possible outcome is that Tusk Philanthropies decides that the project should continue, and either opens up another CFS to find the best organization for the next phase, or negotiates with Free & Fair to continue the project in alignment with the high-level overview found in this document.

## Phase 1: Requirements

In the first phase of the REVIVUS project, we focus on **Requirements**. In order to successfully tackle a project as complex as this one, we must carefully *define the problem*, set the *requirements* on the system (starting from the requirements and goals stipulated in the Mobile Voting Project CFS), define a *threat model*, perform a *risk analysis*, *identify the science and technologies* that are likely relevant to the project, and define objective, independently verifiable *properties and measures* that justify progress.

Complementing this highly technical work and set of deliverables, we also strongly recommend an early focus on conceptual and industrial design and communication about the project and the technology. Election officials, government officials, certification authorities, and most importantly voters need to understand the nature of the system from its earliest days. Because they are not, in general, a technical audience, they will *not* understand specifications, source code, or cryptography. They *will* understand conceptual renderings of mobile device user interfaces, pictures of voters using those devices on their couch and in coffee shops, and visual descriptions of how election officials, elected officials, and citizens can verify an election's outcome.

For each task in this first phase we will use industrial best practices for the high-level analysis of nationally-critical infrastructure and relevant technologies. Requirements will be written in a style comparable to those mandated in NIST federal requirements documents for existing elections technologies (e.g., VVSG 2.0) and federal IT systems (as seen in NIST Special Publication 800-63C). The threat model will be developed on top of the threat models already published in the FoV Report, by the European Commission, and by the Swiss Chancellery, using industrial best practices for security-centric products [Shostack]. Our science and technology assessment will take input from advanced product and R&D groups at Amazon, Apple, Google, and CloudFlare, as well as smaller technology companies developing potentially relevant hardware, firmware, software, and services, such as identity verification, augmented model device security, and novel cryptographic frameworks.

The most technical and challenging tasks of this first phase of work are **Risk Analysis** and **Properties & Measures**. These two tasks have a mild research component that we believe critical to examine before any significant resources are spent on designing or developing an end-to-end verifiable Internet voting product.

The **Risk Analysis** task focuses on a broader remit than one would normally perform during the development of a product. In that work we must develop a dynamic risk model that can permit the design team and election officials to assess the risk to an election's outcome and trustworthiness under different threat scenarios. We believe that this model must take into account supervised voting and vote-by-mail voting as well [National Vote at Home Institute], since virtually all elections will include multiple modalities of voting and a shift of voters from one voting mechanism to another will have a notable impact on the risks associated with that election and the focus of adversaries. We imagine that the final result will be a dynamical model that can be used to perform stochastic/Monte Carlo simulation on a great many election scenarios; its UI will include a simple web dashboard that is usable and useful to election

officials to help make objective, evidence-based decisions about the adoption of Internet voting technologies. Such a model could be a powerful means by which to make tradeoff arguments balancing security and voter engagement, which is critical to adoption in jurisdictions that have already gone on the record about their doubts in any future Internet voting technologies [San Francisco Elections Commission, Richards, Hoke].

The other challenging task is **Properties & Measures**. In this task we must refine high-level, semi-formal requirements written in English into more precise *properties* (predicates that are checkable by some, preferably computational, means) or *measures* (value functions that are measurable, in the mathematical sense, and are preferably computational as well). The point of defining this concrete set of properties and measures is that it establishes a quality bar that any end-to-end verifiable voting product must exceed. Since we will ensure that as many of these properties and measures as possible are computational, they will enable the assessment of a product to largely be performed neutrally, automatically, by a computer, preferably *during* system design (phase 2) and engineering (phase 3). Those properties and measures that are not computational must be defined precisely enough for any expert to use them as an independent assessment checklist, so that two experts, working independently on an assessment of a product, will arrive at the same conclusion as to its quality.

Finally, as we expect that the final outcome of this phase of work will be widely shared, deeply examined, and must be understood by parties both technical and political, we intend to put a reasonable amount of effort toward polishing the report, ensuring that it is visually appealing, compelling, evocative, and inspirational, rather than just a wall of technical text. This is the focus of the **Final Report** task, which we will carry out in collaboration with our conceptual and visual design teams.

Each of these tasks is summarized on the following page in the *Statement of Work* table, along with their respective deliverables and estimated budgets.

| Task Name | Task Description | Deliverable | Estimated Budget |
|---|---|---|---|
| **Problem Definition** | Write a high-level characterization of the problem space to guide the development of a full set of detailed requirements, a threat model, and a risk analysis. | Chapter of final report. | $80,000 |
| **Requirements** | Develop the full set of requirements, both technical and non-technical, to address the problem defined in the previous task. Every requirement or goal stipulated in the CFS must be accommodated for. | Chapter of final report. | $80,000 |
| **Threat Model** | Develop the threat model for the resulting system, in accordance with best practices exemplified in existing threat models for critical infrastructure. | Chapter of final report. | $64,000 |
| **Risk Analysis** | Perform a risk analysis for the system, and develop an executable analytical model thereof, taking into account the requirements and threat model developed in prior tasks. | Chapter of final report and a baseline executable analytical model. | $56,000 |
| **Science & Technologies** | Identify the existing science and technologies that are relevant, and new science and technologies that must be researched and developed, to realize a deployable implementation of the system. | Chapter of final report. | $62,000 |
| **Properties & Measures** | Define an objective set of properties and measures that can be used to evaluate progress toward meeting the system requirements at acceptable levels of risk. | Chapter of final report and an example set of development artifacts to support developers. | $80,000 |
| **Final Report** | Write a report presenting the results of all Phase 1 investigations, including the output of all the previous tasks and an estimate of the resources and time required to move forward with a production implementation of a system. | Final polished report. | $49,000 |
| **Total** | | | **$471,000** |

## Phase 2: Design

After phase 1 has concluded, the *REVIVUS Report* is published, and a period of public peer review and dialogue is completed, Tusk Philanthropies may decide to move to the second phase of the project: **Design**.

The *FoV Report* insists that a rigorous systems engineering methodology based upon peer-reviewed best practices in critical infrastructure development is necessary for the REVIVUS project.

As such, in the Design phase—as required by the underlying *FoV Report* and the expected output of the *Requirements* phase—several design artifacts must be created. These include the following critical components, which are widely recognized as foundational to the creation of any rigorously engineered system.

- **Domain Model**

  A *domain model* is a precise definition of the concepts in a specific domain, such as elections. While a glossary of terms and an example workflow for elections has been created as a part of the VVSG, neither is detailed enough to use as a foundation for a precise specification of a system. We previously produced a partial domain model for elections as part of the FoV Report (see its appendices). This task would extend that model to include new concepts and their interrelationships relevant to this project.

  Domain models are typically written in a combination of semi-formal English and a general-purpose formal language, typically based in logic. We have created formal domain models using many technologies, including Alloy, Coq, Event-B, PVS, RAISE, VDM, and Z.

- **System Architecture**

  A *system architecture* precisely describes the coarse-grained shape of a system, including its hardware, firmware, software, and network architectures. It explains how components relate to each other, where data flows between components, where security principles (ownership, access control, keys, etc.) are created, managed, shared, transmitted, and destroyed, and more.

  System architectures are typically written in a combination of semi-formal English and a domain-specific set of formal languages used to represent the different-but-interrelated facets of the architecture. We have created system architectures using many technologies, including BON, F*, Lando, SysML, and UML.

- **Product Line and Feature Model**

  A *product line* describes a family of products in a succinct fashion, thereby capturing the various design and productization decisions relevant to different customers' needs. Each configurable aspect of the product line is called a *feature*, and the model that relates all features to each other to explain the product line is called a *feature model*.

The domain of elections—especially elections technology—is a perfect place to use product line engineering to define and build products. Each requirement defined by a different election official customer, whether driven by statute, rule, practice, or existing technology, is an example of a feature. Defining and building a product line against a feature model can solve the problems of all potential customers simultaneously, while generating the product fit-for-purpose for each specific customer.

Product lines and feature models are typically written in a combination of semi-formal English and a formal language that has first-order parametrization and logic in its underpinnings. We have created product lines and feature models using many technologies, including Alloy, BON, Clafer, Coq, Event-B, Lando, PVS, VDM, and Z.

- **Executable Model**

  An executable model of a system is a precise description of the system that can be computationally executed, observed, and (often) reasoned about. It is not the same as an implementation because it contains abstract models representing more complicated components. For example, a TCP network connection can be modeled as a queue, or a cryptographic hash as a one-way function.

  High-quality executable models serve several purposes. They are executable specifications that developers or peer reviewers of any kind, from certification authorities to red teamers, can use to better understand and experiment with the ideas of a system before the real system is built. We also use them to automatically generate unit, subsystem, integration, and system tests for the eventual full implementation, either via testbench generation or model-based oracle runtime verification. Finally, we use them as "refinement checkers" for runtime verification—traces of a concrete implementation's execution must conform to possible traces of the abstract model's behavior.

  Executable models are typically written in a combination of semi-formal English and a formal language that has an executable component. We have created executable models using Alloy, Coq, Cryptol, Ivy, Maude, PVS, and VDM, in addition to implementing them in more traditional programming languages like Java and Python.

- **Cryptographic Protocol**

  One or more cryptographic protocols are at the core of any verifiable voting system. A cryptographic protocol is a composition of cryptographic primitives. Composition is accomplished by passing information between primitives and compositions of primitives.

  A cryptographic protocol is created to fulfil a set of requirements, both *behavioral*—what the protocol must accomplish, such as guaranteeing the transmission of a vote from a device to a server—and *non-behavioral*—how much and what kinds of resources must it limit itself to and what security properties it must have, such as guaranteeing that the vote cannot be observed by an adversary while in transit.

  A cryptographic protocol specification is typically written in a combination of semi-formal English (as one would see in an academic paper, book, or standards document) and a formal language that preferably has both an executable component

and a reasoning component. We have created cryptographic protocol specifications using many technologies, including Coq, Cryptol, CryptoVerif, EasyCrypt, F*, ProVerif, and Verifpal.

Some of these technologies also support automatic generation of "correct by construction" implementations directly from specifications; these are often used as reference implementations, and sometimes used as production implementations.

- **Behavioral Specification**

  A *behavioral specification* denotes *what* a system must do in order to be deemed correct and fulfil its intended purpose. It does not specify *how* the system should implement that behavior.

  Behavioral specifications are the core *contract* between the design and the implementation. In more rigorously designed systems using wise technology choices, these specifications can be used in multiple ways to ensure that a product's behavior is exactly what was intended—no more, no less.

  This evidence comes in several forms, using many complementary technologies. Behavioral specifications are the highest-quality documentation of a system's behavior. They are also commonly used for runtime verification in two ways: (1) runtime assertion checking, using specifications as oracles; and (2) automatic runtime verification test suite generation, using specifications as test generators. Using this testing approach obviates the need for hand-writing and hand-maintaining tests—they are generated and kept up-to-date automatically. Behavioral specifications can also be used to reason about an implementation using powerful automatic theorem provers, proving that the implementation does not have well-understood bad properties (such as crashing in some potential future) and does have well-characterized good properties (those in the behavioral specification).

  Tools that support behavioral specifications are our historical academic focus. These kinds of specifications are written in a combination of semi-formal English and either a Behavioral Interface Specification Language (BISL), such as the Java Modeling Language (JML) or the ANSI/ISO C Specification Language (ACSL), or a general-purpose specification language with tool support for formal reasoning and test generation, such as Galois's Cryptol.

- **Continuous Integration and Verification Infrastructure**

  When a set of servers automatically obtain a copy of all the development artifacts for a product, build the product, and evaluate the built product in some fashion, This is called continuous integration if the build crosses subsystems, and verification if the evaluation includes testing or formal verification. Together, continuous integration and verification (CI&V) serves as the computational judge that holds a team to a quality standard.

  We commonly set up CI&V prior to the start of systems engineering, typically in the first weeks of a project. The vetting that the CI&V service provides evolves over the lifespan of the project, increasing in its comprehensiveness and thoroughness, and

refining the kinds of properties that are checked as the project moves from early core datatype and algorithm development, to subsystem development, to UI/UX development, to integration and deployment testing.

The reports generated by a CI&V system can also facilitate public observation of development. As bugs are identified and fixed and enhancement requests are fulfilled, the evidence associated with both kinds of development become a part of the permanent evidence record of the system through regression testing and verification.

Most of our projects include CI&V based upon Jenkins servers attached to either public GitHub projects or private (self-hosted) GitLab projects.

- **Validation and Verification Harness**

  In the field of rigorous systems engineering, testing is used to *validate* that a system implements the properties that it is expected to (as defined in phases 1 and 2), and *verification* is used to mathematically *prove* that the system implements exactly certain properties (those provable using modern verification tools). Verification is *the* key technology used in the development of mission-critical, nationally-critical, and safety-critical systems worldwide in a wide range of domains including transportation, aeronautics, avionics, spacecraft, biomedical, financial, national security, and, in the case of our past work, elections.

  Comprehensive validation through testing is expensive because the (sub)system under test has to be executed (very) many times—it is not uncommon for test suites for medium-sized systems to include hundreds of thousands of tests and take hours to run. Moreover, testing can be used to show the presence of bugs, but can *never* show their absence. This is especially true of non-behavioral properties like security. It is impossible to test a system, either as a developer or via red teaming, and claim that it is “secure”; it is only possible to claim that the testing did not expose any security flaws. Despite these drawbacks, testing is critical as it exercises a system on real hardware, networks, and infrastructure. Finally, most non-technical clients and certification authorities rely on testing as their primary form of evidence about a system’s correctness.

  Verification complements testing; it takes correctness not to the next level, but to the ultimate extreme. For example, it is commonly the case that developers would test an optimized addition function with a handful or two of values to make sure that it behaves as they expect. Unfortunately, doing so only exercises a microscopic fraction of the state space of the function. This kind of testing makes developers “feel good”, but actually shows very little about the behavior of even a trivial system. Formal verification proves that a system behaves correctly for all permitted inputs, not just for the handful that are tested. By using powerful formal methods technologies, one can prove not only that the optimized function is correct for all inputs, but also that it is exactly equivalent to an unoptimized version of the code or a formal algebraic description of the function. Moreover, the computation time necessary to verify these properties is often substantially less than the time required to run a representative set of tests.

  A validation and verification *harness* is the concretization of the correctness and security properties defined in earlier phases of the project into one or more verification

technologies specific to the problem at hand. It is created in the design phase, and is embedded into the CI&V system mentioned above. It is used “on-demand” by developers and Q/A personnel during systems engineering, and by the CI&V system continuously on behalf of developers and evaluators.

Verification is one of our core strengths. We use it in many of our projects, including many election systems. The technologies used to verify properties of protocols, models, and systems are too numerous to list, but the tools we use for verifying cryptographic protocols and algorithms and election systems include many that were already mentioned above, as well as tools like CBMC, Frama-C, and Galois’s Crux and SAW.

Woven through the process of creating these design artifacts is a continuous exploration of what existing technologies—software, firmware, hardware, concepts, or theory—can or should be reused in the project.

For example, a few dozen end-to-end verifiable cryptographic protocols have been published and peer-reviewed. A full review and analysis of all of them is warranted, since creating and verifying a new protocol from the ground up is commonly a calendar year of work. Perhaps one of them is very close to what is necessary for the REVIVUS system and can be evolved slightly to fit the bill.

Once these artifacts are created and publicly peer-reviewed, the project will be ready to move to its next phase, **Implementation**.

## Phase 3: Implementation

The *FoV Report* insists that a rigorous systems engineering process based upon peer-reviewed best-practices in critical infrastructure development is necessary for the REVIVUS project. This includes the use of applied formal methods tools and technologies.

Additionally, due to the nature of the expected requirements derived from phases 1 and 2, there may well be more than just *software* to implement. It is entirely possible that some custom *firmware* and *hardware* may be necessary.

Finally, in order to vet the quality, correctness, and security of the implementation, it must be validated and verified (V&V) against the project requirements using the properties and measures defined in phase 1 and concretized in a test and verification harness in phase 2. The resulting V&V evidence is called the system’s *assurance case*.

- **Software Implementation**

  The bulk of the implementation is expected to be written in software. Not all software implementations are created equal. In particular, the choice of software implementation platform, programming language(s), and development methodology often has an enormous impact on the size, complexity, quality, cost, and verifiability of the resulting system. Additionally, using esoteric platforms or programming languages can lead to difficulty, including finding contributors and reviewers with appropriate expertise, ensuring portability across devices, and facilitating certification.

- **Firmware Implementation**

  Some of the implementation is likely to be very low-level software, commonly known as *firmware*, that works directly on top of special-purpose hardware such as secure enclaves. Firmware is the glue that ties together the behavior and security of the hardware with the needs and assumptions of the software. As such, firmware is typically very small and very carefully designed and vetted, and needs to be high-assurance. While traditional firmware is written in assembly language or a low-level programming language like C, there are new, better choices in programming languages (such as Rust) for implementing modern firmware in a secure and correct fashion.

- **Hardware Implementation**

  While we do not expect that any custom hardware will need to be developed during the implementation phase of the project, it is not unlikely that in the long run new hardware may be necessary to solve some of the fundamental challenges of Internet voting for national public elections.

  It is certainly the case that other countries (such as Estonia) have used hardware developed specifically to address matters of national identification and authentication, in order to help bootstrap some of the security problems inherent in elections.

  Additionally, implementing hardware does not necessarily mean fabricating hardware. Some "hardware" is actually virtual hardware that runs on reprogrammable silicon, such as FPGAs. An example of such hardware is the SSITH-funded secure RISC-V Systems on Chip (SoCs) that we and others have implemented for DARPA, which now run on FPGAs hosted in Amazon's AWS cloud. Here, too, there are good and bad choices for the "programming languages" used to create hardware.

  Our team has expertise in nearly all modern alternatives, both mainstream and cutting-edge.

- **Assurance Case**

  The final, most critical, aspect of any critical system's implementation phase is the crafting and vetting of the system's assurance case.

  The assurance case is effectively the argument that the system meets its specification. For this system, that means the implementation produced during phase 3 exactly matches the design stipulated in phase 2, and hence perfectly fulfills the requirements mandated in phase 1. There must be a traceable, comprehensive, understandable argument that the implementation faithfully implements its design and architecture, and hence satisfies the system's requirements. That *relationship* (colloquially) is what is concretized via a *relation* (mathematically) between the *source code*, the *design*, the *architecture*, the *protocols*, and the *requirements* via the properties specified in phases 1 and 2.

  Each property in the assurance case must have independently-reviewable and checkable evidence that the property is satisfied. In some cases this is as simple as seeing that a piece of technology is reused and configured in the correct fashion. In other cases it means having a test suite that is comprehensive and high-quality, both of

which are *measurable* properties whose thresholds would be defined from the measures of phase 1. Finally, but typically only for the most critical of subsystems—those that are foundational to the key requirements and properties of the system—the assurance case would be formal. Examples of formal evidence include mathematical proofs of a cryptographic protocol's security or verification evidence that an implementation behaves correctly. In the case of an election system, these kinds of core, critical properties include "Does the system count the votes properly?" and "Are all of the security properties mandated by the specification fulfilled by the cryptographic protocol, its implementation, and the system on which it depends?"

All of this evidence is meant to be presented to other developers, peer reviewers, certification authorities, customers, users, and even the general public, to convince them that the system meets their needs and our goals. This necessitates a critical element of storytelling, which is partly why we have designers and storytellers on the team to help craft that narrative, maximize understandability, and build trust in the outcome of the work.

Defining and fulfilling an assurance case of a critical system is our core background expertise, as well as that of our ancestor company, Galois.

Only after the system is implemented and has a very strong assurance case published and peer-reviewed should the system move to (pilot) **Deployment**.

## Phase 4: Deployment

The Deployment phase of the REVIVUS project is fairly speculative at this point. Numerous resources and agencies, including the *FoV Report* and the EAC and NIST, recommend a staged deployment of any new critical infrastructure technology like an Internet voting system. For example, the system should be used for several UOCAVA elections, then for small, local elections that have low value to adversaries, etc.

Additionally, it is widely recognized that the use of the system for non-public elections—such as elections for shareholders, professional organizations, etc.—is another excellent way to "shake out" the system, gather feedback from administrators and users, and build familiarity with the voting public.

Also, the *FoV Report* and the *NASEM Report* strongly recommend that end-to-end verifiable Internet voting technology only be deployed at scale in local, state, and federal elections *after* comparable sister end-to-end verifiable voting technologies are deployed for *supervised* elections.

A reasonable timeline for the start of pilot deployment of the REVIVUS system is circa 2023, with potential use of the system for local, low-value public elections in 2024. At this time, we believe it is not reasonable to expect to be able to build, assure, and deploy a system of this criticality in time for the 2024 Presidential election.

# Conclusion

We look forward to continuing a discussion with Tusk Philanthropies about this project and very much hope to work on it with you. We firmly believe that the right first step is this first phase of the project, and no more. If, after this first phase successfully completes, Tusk Philanthropies decides to evaluate some other existing or newly developed Internet voting system, you will have a precise set of properties and measures against which to measure that team and their technology. If, on the other hand, you want to continue to work with us and our team, you will have bountiful evidence of our professionalism, character, nature, and expertise.

# Bibliography

The below bibliography complements the one already included in the FoV Report. We recognize that there have been an additional few dozen papers published on Internet voting after the publication of the FoV Report at venues like EVoteID, FC, and USENIX, and those papers will need to be taken into account during phase 1 of this project.


- The National Academies of Sciences, Engineering, and Medicine. “Securing the Vote: Protecting American Democracy”, 2018.
- The National Institute of Standards and Technology. “Dramatically reducing software vulnerabilities: Report to the White House Office of Science and Technology Policy”, NIST-IR 8151, 2016.
- The National Institute of Standards and Technology. “A threat analysis on UOCAVA voting systems”, NIST-IR 7551, 2008.
- The National Institute of Standards and Technology. “Security Best Practices for the Electronic Transmission of Election Materials for UOCAVA Voters”, NIST-IR 7711, 2011.
- The National Institute of Standards and Technology. “Information System Security Best Practices for UOCAVA- Supporting Systems”, NIST-IR 7682, 2011.
- The National Institute of Standards and Technology. “Security Considerations for Remote Electronic UOCAVA Voting”, NIST-IR 7770, 2011.
- The National Institute of Standards and Technology. “Assessing Threats to Mobile Devices & Infrastructure”, NIST-IR 8144, 2016.
- The Penn Wharton Public Policy Initiative, University of Pennsylvania. “The Business of Voting—Market Structure and Innovation in the Election Technology Industry”, 2017.
- The Brennan Center for Justice. “Dirty Tricks: 9 Falsehoods that Could Undermine the 2020 Election”, 2020.
- The Brennan Center for Justice. “Estimated Cost of Covid-19 Election Resiliency Measures”, 2020.
- The Brennan Center for Justice. “America’s Voting Machines at Risk”, 2014.
- The Brennan Center for Justice. “America’s Voting Machines at Risk - An Update”, 2018.

- The Brennan Center for Justice. “A Procurement Guide for Better Election Cybersecurity”, 2019.
- The Brennan Center for Justice. “Defending Elections: Federal Funding Needs for State Election Security”, 2019.
- The Brennan Center for Justice. “Post-Election Audits: Restoring Trust in Elections”, 2007.
- The Brennan Center for Justice. “A Review of Robust Post-Election Audits”, 2019
- The Brennan Center for Justice. “Pilot Implementation Study of Risk-Limiting Audit Methods in the State of Rhode Island”, 2019.
- The Brennan Center for Justice. “Trump-Russian Investigations: A Guide”, 2017.
- The Brennan Center for Justice. “Voting System Failures: A Database Solution”, 2010.
- The Brennan Center for Justice. “Better Ballots”, 2008.
- The Brennan Center for Justice. “The Machinery of Democracy”, 2006.
- The Brennan Center for Justice. “The Machinery of Democracy: Voting System Usability”, 2006.
- The Brennan Center for Justice. “Recommendations for Improving Reliability of Direct Recording Electronic Voting Systems”, 2004.
- Swiss Confederation, Federal Chancellery FCh, Political Rights Section. “Redesign of Internet Voting Trials in Switzerland 2020”, 2020.
- The Committee of Ministers, Council of Europe. “Recommendation /Rec(2017)51 of the Committee of Ministers to member States on standards for e-voting”, 2017.
- Springall et al. “Security Analysis of the Estonian Internet Voting System”, CCS’14, 2014.
- Hoke. “Internet voting: structural governance principles for election cyber security in democratic nations”, GTIP’10, 2010.
- Markussen et al, “Trust in Internet election observing the Norwegian decryption and counting ceremony”, EVOTE’14, 2014.
- Specter et al. “The Ballot is Busted Before the Blockchain: A Security Analysis of Voatz, the First Internet Voting Application Used in U.S. Federal Elections”, USENIX Security ‘20, 2020.
- Heiberg et al. “Verifiable Internet voting in Estonia”, EVOTE’14, 2014.
- Heiberg et al. “Improving the Verifiability of the Estonian Internet Voting Scheme”, E-Vote-ID’16, 2016.
- Halderman and Teague. “The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election”, Vote-ID’15, 2015.
- Gaudry and Golovnev. “Breaking the Encryption Scheme of the Moscow Internet Voting System”, FC’20, 2020.
- Franke. “Security analysis of the Geneva E-voting system”, Informatik 2013, 2013.
- Richards. “City of Toronto Internet Voting Report: Web Accessibility (WCAG 2.0) Evaluation of Voter-Facing Demonstration Sites”, 2014.
- SAN FRANCISCO ELECTIONS COMMISSION, RESOLUTION ON INTERNET VOTING, (Adopted by the San Francisco Elections Commission (6-0) on April 19, 2017.), “Resolution opposing Internet and email voting in local, state, and federal elections”, 2017.

- Trail of Bits. “Voatz Security Assessment, Volume I of II: Technical Findings”, 2020.
- Trail of Bits. “Voatz Security Assessment, Volume II of II: Threat Modeling Findings”, 2020.
- Specter and Halderman. “Security Analysis of the Democracy Live Online Voting System”, 2020.
- Shostack. “Threat Modeling: Designing for Security”, 2014
- National Vote at Home Institute. “Vote at Home Election Security FAQ”, 2020.