The E2EVIV Report

Many People

April 14, 2015

Contents

Note: Names following chapter titles are the currently-assigned writers; percentages following writer names are very rough estimates of the approximate percentage of completion. Some material factored into the percentages may not yet appear in the generated report because it needs to be brought in from external sources.

List of To Do Items

Chapter 1

Executive Summary (Joe K./Susan) (0%)

Chapter 2

Introduction (Joe K./Susan) (25%)

2.1 The E2E VIV Project

2.2 Goals

2.3 People

2.4 Methodology

2.5 Outcome

2.6 Next Steps

Chapter 3

Remote Voting (Philip) (45%)

3.1 Rationale

For each subsection::

- Who is in this group?
- How many people are in this group?

3.1.1 Geographic Dispersion

3.1.2 Accessibility

Studies issued by the International Center for Disability Information and the National Institute on Disability and Rehabilitation Research indicate that 20% of Americans live with disabilities.

The Help America Vote Act (HAVA) of 2002 requires that all polling places in elections for federal office, anywhere in the United States have at least one voting system . . .

3.1.3 UOCAVA

In 1986, Congress enacted the Uniformed and Overseas Citizens Absentee Voting Act, stating citizens that are part of the uniformed services, merchant marines, and their families or citizens residing overseas are allowed to register and vote absentee for federal office.

- approximate count
- Table 2.1 Convenience Voting and Technology

State	**Overseas Voting Eligible Population** (McDonald 2009)	**Overseas military and federal civilian employees** (US Census Bureau 2010)
Texas		
California		
Florida		
New York		
Pennsylvania		
Illinois		
Ohio		
Michigan		
Georgia		
Washington		
North Carolina		
Tennessee		
Virginia		
Total		

3.1.4 Early Voting

3.1.5 Expectations

- American Political Science Association study (1952)
  - Ability to vote without registering in person
  - Ability to vote without unreasonable requirements and costs (federal postcard)
  - Insure enough time is permitted for ballot transit
- HAVA, ADA
  - Multilingual
  - Accessible for individuals with disabilities
- Privacy of Vote
- Integrity of Vote (VVPAT)

3.2 History

Political attitudes and legislation on absentee voting has been a slow moving effort, due to dominant partisan attitudes changing, and regulations being enforced at the state level.

Before the civil war US citizens primarily voted in their places of residence, and many states legally barred the casting of votes from outside state borders. There was little effort from any state to accommodate absentee voting. However, in 1864 with the American Civil War displacing soldiers from their residences, Lincoln's re-election was at risk. With much lobbying on behalf of the republican party (and opposition from the democratic party), nineteen of the union's states adopted absentee voting procedures for military voters on federal elections in time for the election. Unfortunately since the motivation to passing these laws was securing Lincoln's re-election, rather than persistent enfranchisement, many absentee military voter laws were treated as temporary and repealed after the war.

In 1918 America's War Department decided that it was not ready to support the military vote. World War I had displaced such a large number of voting eligible persons and military units were rarely composed of same state citizens. Not even states in support of military vote were allowed the soldier vote, even on matters at the state level.

As in the Civil War, World War II inspired another push for the military vote in hopes of supporting the re-election of the presidential incumbent. This introduced the Soldier Voting Act (1942) which, although passed too late for the presidential election, mandated military personnel rights to absentee vote on federal elections during times of war without subjugation to voting tax or postage costs. From this point forth all overseas voting would be regulated at the federal level and implemented at the state level. However, by 1944, the state mandate to support military absentee voting was amended to a recommendation.

Progress with absentee civilian vote was a further behind. In 1896, states began introducing civilian absentee voting legislation. By 1924 only three states in the union had no absentee voting legislation, but all states had different laws and restrictions. Major progress on this front wasn't made until federal voting laws were passed that combined for the handling of civilians and military votes. The Voting Assistance Act of 1955 was the first to federally combine voting policy recommendations for overseas civilian government employees with military. In 1986 Voting Assistance Act was amended to include individuals temporarily living outside the United States. With lobbying from sympathetic groups and the quickly growing population of overseas civilians, in 1974 Overseas Citizens Voting Rights Act passed extending the recognized vote to citizens regardless of their intentions to return to the United States.

By 1986, combatant attitudes towards overseas votes had finally settled, and the Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA) was passed replacing/combining Overseas Citizens Voting Rights Act and the Federal Voting Assistance Act, and finally made supporting the overseas absentee ballot a requirement.

Voting Rights Act (VRA) of 1965 was the first legistlation to enfranchise voters with disabilities. The VRA granted voters who require assistance to vote by reason of blindness disability or inibility to read or write assistance by a person of the voters choice. This also introduced some of the earlier legistlation defining a disabled citizen.

The Voting Accessibility for the Elderly and Handicapped Act of 1984 (VAEHA) was passed to improve access for handicapped and elderly individuals. However, states were left to set their own standards of *access*, and limited the disabled voters group to those with *physical dissabilities*. The VAEHA did, however, mandate 'no notarization of medical certification shal be required of a voter with a disability with respect to an absentee ballot or application for such ballot.'

The 1990 American Disabilities Act (ADA), although not specific to voting rights, required that people with disabilities have access to basic public services, including the right to vote, however does not strictly require that polling locations are accessible. ADA did however extend the definition of disability to

> "a person who has a physical or mental impairment that substantially limits one or more major life activities, a person who has a history or record of such an impairment, or a person who is perceived by others as having such an impairment."

The majority of federal laws passed to protect disabled voting have struggled to clearly define an representative range of dissabilities, and are often focused on access at physical polling locations; which is often expensive for states to implement. Additionally, state defined polocies often ignore rights to voting privacy, and exclude persons with multiple disabilities sighting that aiding technologies are not yet availible.

- integrate history of disabled voters rights
- ⋆ TODO as necessary: ADA, HAVA, MOVE

3.2.1 Integration with Local Elections

Every state has their own requirements, deadlines, and transmission restrictions which the FVAP documents in a 'Voting Assistance Guide' distributed to potential UOCAVA voters.

3.3 Shortcomings of Current Practice

- In the 2008 Post-Election UOCAVA Survey Report and Analysis 52% of attempted votes were not counted because the ballots were late or never arrived.
- 20% of Americans with disabilities have said that they were unable to vote in presidential or congressional election due to barriers at or getting to the polls. (as of 2007)
- "ten states explicitly require a privacy waver if a voter uses fax or e-mail to return a voted ballot"
- FVAP's *Voting Assistance Guide* is complicated and results in many failed registration attempts.
- Entire process for UOCAVA voters generally can take up between 2 weeks and 2.5 months
- Persons with manual dextarity impairments can prevent them from marking paper ballots.
- Most practices for disabled voters forfit independent private voting places.

3.3.1 Use of Communication/Internet

The major motivations for use of internet and communication technologies for UOCAVA has been to address ballot transit time, and simplify voter registration.

- OVF's streamlined website for FPCA in states that allow online registration

3.3.2 Accessibility and Usability

3.3.3 Auditing

Current Practice

Digital vs. Physical

- VVPAT provides voters with paper statements

Risk-Limiting Audits

Risk limiting audits use a public random auditing process to make an argument about the statistical confidence of a particular election result.

Chapter 4

E2E VIV Explained (Philip/Daniel/Adam) (45%)

1: signposting; some content is here, but we need an intro and transitions explaining what content is about to happen

Typical Internet voting election processes have six phases:

Setup During the setup phase, the election officials gather the information needed to run an election. This includes gathering registration information for all voters, identifying the issues and races that will be voted on, designing and specializing ballots, sending instructions and other information about the election to voters, and so on.

Distribution Once the election has been set up, election officials must distribute ballots to the voters. Different voting system architectures use different mechanisms, including postal mail, email, or by having voters interact with a website.[1]

Voting Voters then fill out their ballots, often with the help of software installed on their own computers.

Casting Filled out ballots are then returned to the election officials; as with distribution, different architectures use different mechanisms.

Tallying The tallying phase includes the remainder of the election finalization tasks: counting votes and announcing the election outcome are common to almost every process, though some include other miscellaneous tasks like publishing certain information needed for audits.

Auditing Some elections will inevitably be disputed; in such cases, there is a final phase in which interested parties look for evidence that the election outcome is correct (or not!).

One major concern for Internet voting involves ballot integrity during the distribution, voting, and casting phases. For the election outcome to be correct, it is important that the ballot that is received by and displayed to the voter match the ballot that was created and sent by the election officials; that the computer used to fill out the ballot faithfully reports the intention of the voter; and that the filled out ballot be received by the election officials exactly as it was when it was sent by the voter. Typical Internet communications involve not just the computers owned by the two parties communicating, but also many intermediary computers controlled by neither party. A good election system needs to account for this, making it impossible for these intermediates to intercept ballots for viewing or modification during transit. Another concern is that voters computers are rarely administered by experts, and as a result many of them are compromised by outside forces. One consequence of this is that the voting phase itself may become corrupted: even if the ballot arrives unchanged at the voter, malware on the voter's computer may change the way the ballot is

[1] We distinguish between sending instructions to voters and distributing ballots; there is no hard and fast rule for the distinction, but a rule of thumb is that instructions are applicable to many voters, whereas anything that has been specialized for a single voter is part of the ballot and falls under the distribution phase.

displayed or the way the vote is recorded before casting the vote. It can be difficult to design a system that is resistant to this kind of attack without seriously sacrificing the usability of the system. Some systems use alternative distribution mechanisms as cross-checks; for example, sending something to the voter by postal mail which can be used to check that the ballot displayed by their computer is correct.

To the extent that it is possible, it is desirable for Internet voting to be private and anonymous. Voters should feel comfortable voting the way they like (and not feeling pressured to vote for a particular candidate or to vote a particular way on some issue); the fewer people who know or can find out the way a given voter voted, the more comfortable they can feel. On the other hand, election officials only want to record votes from people who are registered to vote, and even then want to record only one vote from each voter. Thus there is a tension during the vote casting phase between retaining the anonymity of votes and ensuring that a vote is coming from somebody who ought to be able to vote.

One popular approach to this problem in existing systems is to initially require each vote to be tied to the voter who cast it long enough to decide whether to include the vote in the later tally or not; then to keep the vote but delete the information about who cast the vote. This approach can work; however, audits of systems that take this approach shows that it is all too easy to accidentally retain the connection between votes and voters longer than intended, and make this information much more widely visible than intended. From the privacy side of the tradeoff, it would be better if the voter could be confident that there was no connection stored because the information they send to the election officials during the casting phase does not include any personally identifying material.

2: ask Joe for a reference to the audit that showed that voting terminal logs retained exactly who voted for what

There is a subtle distinction being made here. We certainly want our Internet voting systems to be correct, private, secure, and so forth. It is important for the people developing these systems to verify that they are correct and take an active role in seeking out and eliminating defects in the system. But the goal of verifiable Internet voting is to go even farther: not just correct, but *visibly* correct. That is, it must be possible for the parties using the system to be able to *check* that the system is behaving correctly, without trusting in the abilities of the people who created the system to avoid bugs or trusting in the inability of third parties to influence the behavior of the system. As applied to anonymity: since it is not easy to prove to somebody else that you have deleted some information, one must simply avoid giving them that information in the first place. This theme—of being not just correct, but verifiable—is one of the central ones of verifiable Internet voting, and is a critical part of the defense against the software bugs, security vulnerabilities, and sophisticated cybercrimes that history tells us are sure to crop up.

The verifiability theme also rears its head in vote tallying, where voters may be interested in verifying that their vote has been recorded correctly and included in the tally.

3: flesh out the next two paragraphs

The goal of these individual concerns is to add up to a top-level property: end-to-end verifiability. Cast as intended; recorded as cast; counted as recorded.

4.1 Shortcomings and Expectations of E2EVIV

4.1.1 Access to Communication/Internet

4.1.2 Accessibility

4.1.3 Usability

4.2 E2E VIV in Practice

A number of practical voting systems have been developed based on the principles of E2E VIV. This section describes several systems that have been used in a real election or in a pilot.

4: Sort these somehow; make a timeline?

4.2.1 RIES [17]

RIES, the Rijnland Internet Election System, was first used in 2004 to support elections to the Rijnland water management board, supplementing the system of postal voting used by the water board. A subsequent version was used to allow expatriate voters to participate in the Dutch parliamentary elections [15].

Election Procedure

- Before the election, credentials are mailed to every voter in the form of a very long number. The same mailing also includes instructions for the voter.
- During the election, voters log into an election web site that includes a client-side voting application written in JavaScript. The client-side application encrypts the vote by passing the voter authorization code and the public ID of the candidate through a one-way function to create the encrypted vote. The encrypted vote is then placed on a public bulletin board that serves as a ballot box.
- At the close of the polls, the election authority releases the final vote tallies along with a codebook containing the encryptions of all valid credentials with all candidate IDs.

Verifiability

The algorithms and protocols used RIES are public, and each voter, having access to all of the inputs and outputs, may (in principle) check the computations. This is weaker than the desired individual verifiability, but nonetheless, far stronger than conventional voting systems.

5: how/why?

Analysis

The Organization for Security and Co-operation in Europe (OSCE) sent an election assessment team to observe the use of RIES in 2006. Their report contains observations of critical security features of the system that could not be observed [19]. Further weaknesses were revealed by the Eindhoven Institute for the Protection of Systems and Information (EiPSI) in 2008 [18], notably that:

- the procedure of voter self-check is quite complicated,
- the two-channel (mail and Internet) voting makes system less transparent,
- too much power is given to the election administrator and the system's Internet host,
- issues arise when modifying the codebook due to a revoked ballot, and
- there are realistic ways to forge votes via cryptographic hash collisions.

One of the more important lessons learned through RIES is that when voter authorizations are distributed long in advance of the election, a mechanism must be provided allowing voters to obtain replacement credentials and invalidate lost credentials. These mechanisms add significant complexity to system, and is a source of some of the problems reported in the OSCE and EiPSI reports.

Another feature of RIES rife with tradeoffs is the ability to perform testing during the election: pre-invalidated test ballots are deliberately added to the bulletin board in order to test the network path from selected Internet clients to the server. While such testing in principle can increase confidence in the election integrity, in practice it opens the system to spoofing and denial of service attacks. Furthermore in the RIES implementation the system is aware of the fact that it is processing a testing ballot, and all of the test ballots were voted identically from the same computer, limiting the confidence added at the expense of these vulnerabilities.

In the wake of these critical reports, plans to use RIES in the 2008 Dutch parliamentary elections were scrapped, and Internet voting as a whole was banned in the Netherlands.

4.2.2 Prêt à Voter [11]

The state of Victoria in Australia held a governmental election in November 2014, using a version of the Prêt à Voter system [5]. An attempt was also made to use Prêt à Voter in a student election at the University of Surrey in February 2007 [3]. The failure of this attempt illustrates many of the pitfalls of adapting a research system to an actual election, such as a short timetable, a lack of clear requirements, and the need for rigorous implementation practices.

Election Procedure

Prêt à Voter uses two-part paper ballots with the candidate names on one part and the voting targets plus a ballot ID number or barcode on the other part.

- Before the election, the paper ballots are printed. Typically, the two parts are printed as a single sheet with a perforation to divide the sheet after voting.
- From the voter's perspective, the order of the candidate names on the ballot appears to be random. The voter marks her choice next to the candidate name of her choice, separates the two parts of the ballot, and destroys the candidate names. She may take a copy of the voted part home for later verification.
- For tabulation, there is a cryptographically secure mapping from the ballot ID numbers to the apparent random order of the candidate voting positions. Multiple custodians using a mixnet or similar technique use this mapping to decode cast ballots into anonymized plain-text ballots which are then posted to a bulletin board.

Verifiability

Unvoted ballots may be audited before, during and after the election to ensure that the decoding of cast ballots is being correctly performed. Randomly selected stages in the decoding can be challenged to prove the integrity of the count, and the plain-text decoded ballots are easily counted for verification by any interested party.

An individual voter may also search for their voted ballot ID on the bulletin board. This reveals the positions that were marked on that ballot, but crucially, it does not show the corresponding candidate names. The voter may therefore verify that the positions marked at the polling place were correctly recorded by the election officials, but because the voter no longer has the part of the ballot linking candidate names to ballot positions, the voter cannot prove to anyone else how the ballot was voted.

Analysis

Since there is no pre-election bulletin board posting of the valid ballot IDs, there is potential for ballot stuffing by insiders. This can be detected by cross-referencing polling place data with the bulletin board after the fact, but that requires additional trust in the poll workers.

4.2.3 Punchscan [20, 21]

Punchscan was used for the graduate student association elections of the University of Ottawa in 2007 [13]. It is likely the first E2E voting system with ballot privacy used in a binding election.

Election Procedure

The election experience for a Punchscan voter is very similar to that of Prêt à Voter. The system uses a two-part paper ballot where the top part has candidate names and candidate numbers (or letters) and the bottom part has numbered (or lettered) voting targets. Holes punched in the top part expose the voting targets below. The order of the voting targets for each race appears random to the voter. Both halves of the ballot bear an identical serial number.

The voter casts their vote by marking her choice with a bingo dauber, and the two halves are separated. Either side can be scanned (since the bingo dauber marked both through the hole and around it) as the cast ballot. The other side is destroyed, and a copy of the cast side may be retained by the voter.

Verifiability

A curious voter may inspect the public record of any cast ballot exactly as with Prêt à Voter. It does not matter which half of the ballot the voter retained, because there is no public display of the numbers that link candidate names to voting positions; only the position that was marked is displayed. Again, individual ballots may be audited, and the key to tabulating the votes is that there is a cryptographically secure mapping from the ballot serial numbers to the apparent random order of the candidate voting positions.

Analysis

Punchscan elections rely on procedure to maintain many of its desirable properties, much like Prêt à Voter. For example, during the University of Ottawa elections, more ballots were cast than voters recorded in the pollbook, showing that ballot stuffing can be caught after the fact by poll workers, but is not an inherently verifiable property of the system.

4.2.4 Scantegrity II [8, 9]

Scantegrity II (Invisible Ink) was used in the Takoma Park, Maryland municipal elections in 2009 [6]. In 2011, it was used for in-person voting with Remotegrity (4.2.5) used for absentee voting. The 2009 Takoma Park election was the first use of an E2E system with ballot privacy in binding governmental elections.

Election Procedure

- Before the election, officials generate the seed to a pseudorandom number generator using a secret sharing scheme. Three-letter alphanumeric codes are created for each choice on each printed ballot using this seed, and additional tables are created so that interested parties can later confirm that the tally was computed correctly.
- During the election, the voter experience is nearly identical to that of conventional optical-scan paper ballots. When the voter marks their choice, the ink in the pen reacts with invisible ink on the paper to disclose the three-letter code in the marked voting target. The ballot ID number and the displayed code are posted to a public bulletin board.
- After the election, public verification of the final tally proceeds with the public bulletin board in a manner similar to that of Punchscan and Prêt à Voter.

Verifiability

In addition to the public verification, an individual voter who takes note of their ballot ID number and the code revealed from invisible ink may use the public bulletin board to check that their ballot was indeed tabulated, though this information is not sufficient to prove that they voted a particular way.

Analysis

6: Roll this in with other analysis in 4.3

4.2.5 Remotegrity [26]

Remotegrity is a remote coded voting system that was used for absentee voting alongside Scantegrity (4.2.4) for in-person voting for the 2011 Takoma Park, Maryland municipal elections.

Remotegrity voters receive a coded voting ballot and an authentication card in the mail. The codes on the ballot can be covered by a lottery-style scratch-off field. The authentication card contains several authentication codes under scratch-off, a lock-in code under scratch-off, and an acknowledgment code. Both cards have serial numbers. The voter can be sent two ballots so that she can use one for auditing purposes.

To vote, the voter enters both serial numbers, the codes corresponding to her choices, and an authentication code obtained after scratching-off a surface chosen at random.

She returns to the election website a few hours later to check if her codes are correctly represented, and to see if the election authority has posted her acknowledgment code next to the codes. This indicates to her that the election officials received valid codes for her ballot.

She scratches off the lock-in code and posts it on the website. This affirms to the election officials, observers and other voters that her vote is correctly represented on the website.

Among all of the systems discussed here, this is the first one that asks the voter to take positive action to confirm that the vote was correctly posted.

As with RIES, if we assume that there is no communication between the computer used to print the credentials and the computer used to collect the votes, the latter computer does not know the mapping from codes to candidates, so the vote is not revealed to the computer. Further, because the computer does not know a valid code corresponding to another candidate on the ballot, it cannot change the vote. Finally, and uniquely, because the computer does not know the acknowledgment code, its presence on the election website assures the voter that the election officials received a valid code for her ballot.

The tally is computed from the codes in a verifiable manner that corresponds to the coded voting system used.

If a jurisdiction is nervous about using the Internet for remote voting, Remotegrity ballots can be mailed in, and voters can check for their codes on the election website to be assured that their vote correctly reached election officials.

4.2.6 Helios [1, 2]

Helios is a system developed for web-based Internet voting. It was used for the election of a Belgian university president in March 2009 and by numerous universities and associations since then, including the Association for Computing Machinery and the International Association for Cryptologic Research.

Election Procedure

- Before the election, officials input the email addresses of the voters who will be participating. The system emails the voters their randomly-generated login information and the link to the election website.
- During the election, the voter enters their choices on the website. After entering her choices, the voter has an option to spoil their ballot in order to verify that it was recorded correctly. Upon completing a non-spoiled ballot, the system sends an email confirming the receipt of their vote, though not their choices. At any time before the close of the election, the voter can repeat these steps and the new vote will replace the old vote.
- After the election, Helios uses homomorphic vote tallying with the optional addition of mixers and mixnets in some derivatives [4, 25].

Verifiability

Voter authentication is not required until after the voter decides to cast the ballot, so any interested party may prepare and audit ballots. All cast ballots are posted in encrypted form on a public bulletin board so that voters may check that their ballots have been correctly recorded. Similarly, after the polls close, the decryption and vote tally may be checked.

Analysis

Because officials can enter voters by email address, Helios provides limited protection against insider ballot stuffing. Due to the complexity of publicly auditing the election results, this would be difficult to detect [24].

4.2.7 Norwegian System [14]

Between 2011 and 2014, the Norwegian government ran an Internet remote voting trial using a cryptographic protocol designed by Scytl, a commercial voting system vendor. Scytl and the Norwegian government assert that this is an E2E system, which if accurate is the first effort by commercial voting system vendors to enable E2E elections.

Election Procedure

The Norwegian system uses a three-channel model involving postal mail, the Internet, and SMS text messaging.

- Before the election, the voter receives authorization codes to cast a ballot via postal mail.
- During the election, the voter uses a computer to cast an encrypted ballot. The voter can cast multiple ballots; only the last ballot cast is counted, and if a voter votes both on paper at a polling place and by Internet, the paper ballot overrides the Internet ballot. After casting a ballot, the voter receives a confirmation code offering a partial end-to-end proof via an SMS message.

Verifiability & Analysis

Available descriptions of the Norwegian system are incomplete, so it is not possible to analyze the system in depth. However the system's claims to protect voter privacy are weak: "If the voter's computer and the return code generator are both honest, the content of the voter's ballot remains private." In addition, the receipt delivered to the voter proves only that the encrypted ballot was received as cast, not that it was counted as cast or that the encrypted vote matches the voter's intent.

The system evolved significantly between its first use in 2011 and 2013, with added complexity to attempt to assure voters that their ballots were stored as cast. In 2013, the Carter Center mounted a serious effort to observe the Norwegian system in action. Their report on the operation of the system and the problems they had observing it offers useful insight into the administration of E2E systems in general as well as the particulars of the Norwegian system [7].

4.2.8 Wombat [16]

The Wombat voting system has been used for multiple pilot elections in Israel. It is an in-person voting system where the voter votes on a touch-screen and obtains a printout of her vote with an encryption of it. The voter can choose to cast or audit the encrypted vote. If she chooses to audit the vote, she may check if the vote was correctly encrypted. If she chooses to cast it, the ciphertext is posted online, and she casts the unencrypted vote in the ballot box (this may be manually counted) and takes the ciphertext home. The votes are tallied using a verifiable mixnet.

4.2.9 DEMOS [12]

DEMOS is a coded vote system where the voter is given a two-part coded ballot; she audits one part and uses the other to vote. Associated with each choice on the ballot is

- a vote code—the encryption of the vote, which is entered in the voting machine by the voter, and
- a receipt code which the voter does not enter, but which is posted online next to the vote code.

The voter can check the receipt to ensure her vote reached the election authorities. The ballot also has a QR code containing all the information on the ballot which can be scanned by the voter if she prefers not to manually enter the vote code. Once the ballot is entirely represented on the computer, the voter can then make her choices. Note that if the voter scans the QR code, the scanning computer knows how she voted. The vote codes represent homomorphic encryptions of the votes and the verifiable tally is obtained in a standard manner.

A pilot study of DEMOS was carried out in 2014.

4.3 Limitations of Existing Systems

7: This section should be filled in with material from history.tex and comparative_analysis.te and then history.tex and comparative_analysis.te should be "retired" and their resources moved into e2e_viv_explai

E2E systems inherit many of the limitations of traditional voting systems. Reliability of equipment, reliance on procedure, trust in insiders, and accessibility are all problems with traditional in-person voting systems. For remote systems, the integrity of postal systems, turnaround time for mailed materials, access to Internet or fax technology, and reliability of Internet servers are all well-documented obstacles to voting.

Existing E2E systems mitigate some of these limitations. For example, code voting limits the ability for attacks against postal mail systems to change the candidates marked on voted ballots. However if an attacker simply intercepts and destroys the voted ballot, a replacement might not arrive in time for that voter to participate in the election. To mitigate this, election officials might choose to instead accept voted ballots via fax, email, or website, but such expedient measures often trade off the verifiability that makes an E2E system desirable in the first place.

In this section, we examine the limitations of E2E systems with a particular focus on the limitations that are unique to or exacerbated by E2E characteristics.

4.3.1 Voter Secrecy

8: reference other sections about this

Systems like Prêt à Voter (4.2.2) and Punchscan (4.2.3) rely on a randomized candidate order or a code on printed ballots to ensure voter secrecy. Voted ballots must appear on a public bulletin board in order to verify the election results, and so to protect secrecy only the selected position or code is visible on the final ballot along with a ballot ID.

If an insider is able to review the printed ballots before the election, they can record how the candidate positions are arranged for each ballot ID and therefore identify which candidate is marked on the voted ballots, thus violating secrecy [5].

Recent writing on Prêt à Voter recommends printing ballots on demand at polling places in order to limit this possibility [23]. Printing on demand introduces additional problems and expense compared to centralized printing. More printing equipment is required at each polling place, that equipment can break or be difficult to operate, and the printing equipment must have some way of communicating with the rest of the election infrastructure to ensure it has, for example, the correct cryptographic seeds for generating new ballots.

Scantegrity II (4.2.4) uses invisible ink to hide the vote codes on unvoted ballots, and Remotegrity (4.2.5) can use scratch-off fields to hide vote codes and other information required to cast a ballot. These techniques limit the opportunity for insiders to learn secrecy-compromising information without being detected through the presence of a marked or damaged ballot.

Even with techniques to mitigate insider foreknowledge of the ballots, secrecy still can depend on voters and poll workers correctly following procedures. A voter can leave the polling place with a complete Prêt à Voter ballot, for example, failing to shred the half with the candidate order. With both halves of their ballot, they can prove how they voted, losing receipt-freedom.

RIES makes a deliberate secrecy tradeoff by weakening the receipt-freeness requirement in exchange for providing universal verifiability and a degree of individual verifiability. The results of an entire election can be independently audited with only the information publicly available after the election. However if a voter discloses her credential or her encrypted vote, the same public information may be used to violate ballot secrecy. The developers of RIES judged this violation to be no more severe than the threats to ballot secrecy inherent in postal voting, and therefore worth accepting for the benefit to verifiability.

4.3.2 Ballot Stuffing

9: Flesh out these points

Insiders can print or generate more ballots than there are voters.

Defense in most systems is to crossreference with polling place counts. This depends on procedure being correctly followed rather than something verifiable.

Publishing pre-election tables can help so that all votes in the final result have an (anonymized) provenance. However this can make it harder to replace lost, stolen, or spoiled ballots. Also same-day registration increases turnout and requiring rolls ahead of time precludes it.

4.3.3 Infrastructure & Equipment

10: Flesh out these points

Election equipment fails in practice. A true E2E system must be resilient to failures while not giving up E2E properties. For example, if a remote voting website fails, email should not be the backup mechanism.

Verifiable election systems can require more sophisticated equipment than traditional systems. For in-person voting, a verifiable system might require ballots to be printed on demand, a high-quality shredder for two-part ballots, and more sophisticated assistive devices.

Systems that post encrypted ballots during an election to a public bulletin board are distributed systems. Depending on the networking scheme this can open equipment to distributed denial of service (DDoS) attacks, network partitions, inconsistency, and other problems inherent to distributed systems.

Internet systems add to the difficulties of distributed systems by requiring the systems to be accessible via the public Internet, increasing the possibilities for DDoS and other malicious attacks. Furthermore, many systems allow voters to use their own computers to vote, leading to many potential problems: malware on the voter's computer might undermine security, incompatibilites might arise due to operating systems or web browser versions, and the network infrastructure between the voter and the central election system might be compromised with a man-in-the-middle attack.

4.3.4 Support & Responsiveness

4.3.5 Usability

11: Flesh out these points

Traditional election systems struggle with usability. Verifiable systems add more steps and complexity, making usability even more difficult. The mechanics of marking a ballot become more complex with code voting as in Remotegrity, and position or shape matching as in Prêt à Voter and Punchscan. Individual verification, not even possible in traditional systems, is an entirely new process that voters must master to take full advantage of E2E guarantees.

Usability of equipment by poll workers is also crucial. Traditional systems' accessibility features, for example, are in practice often unusable by poll workers and require specialized expertise [22].

Usability translates into confidence.

4.3.6 Accessibility

There are ability requirements for many E2E systems in various stages of the voting process. For example, a sighted voter is able to see the correspondence between candidate position and marking position on a Punchscan ballot, but a non-sighted voter cannot without assistance. In addition to obstacles to marking a ballot, some schemes with individual verification lack provisions for disabled voters to participate in individual verification without assistance. Information required for verification is frequently delivered through a paper receipt, an invisible ink code, or requires writing down receipt data.

Accessible verification protocols have been proposed that take care to protect voter secrecy and allow participation in individual verification [10]. However, these protocols require using accessibility equipment with an audio, sip-puff, or switch interface to read and mark the unencrypted ballot. The device must therefore be trusted not to record the votes, which would violate voter secrecy. The device must also represent the ballot faithfully to the voter so that votes are recorded as intended.

Requiring trust in assistive devices is not unique to E2E systems [22]. In non-E2E systems, though, trust is already widely distributed. In the context of having to trust the chain of custody of ballots, the integrity of poll workers, and the outcomes of any audits, having to trust an assistive device is a relatively small concession to make in an already-flawed system.

On the other hand, a well-designed E2E system requires a much smaller base of trust for voters to have confidence in the results of an election. By requiring an expanded base of trust in order to be accessible, the existing E2E systems undermine their E2E properties.

4.3.7 Social & Political

12: Flesh out these points

The public is reluctant to adopt E2E approaches when systems with E2E properties fail during an election or are revealed to have substantial integrity issues.

Novel systems face a difficult bootstrapping problem: in order to be adopted in large-scale elections, they must have a successful track record. However in order to build up that track record, systems must be successful despite the limited resources available during small-scale pilot programs. With limited resources, corners are cut in the implementation of the election system leading to a greater chance of problems with equipment, software, and support.

Broader computer security concerns are becoming topics of household conversation with vulnerabilities like Heartbleed and droves of personal data compromises making the headlines. These concerns rightly make the public wary of any system with a computerized component, even if the Internet is not involved.

It is difficult for the guarantees provided by an E2E system's underlying cryptographic design to be understood and accepted by voters without technological expertise.

Even when a voter wants to participate in individual verification, the steps to do so are frequently too difficult or obscure.

Chapter 5

Required Properties of E2E Systems (Dan) (100%)

We now describe the required properties that E2E VIV systems must have in order to be considered for use in real elections. These requirements can be broadly divided into two groups: *technical requirements* and *non-functional requirements*. Technical requirements are those that can be directly addressed by the design and implementation of the system, such as authentication requirements for voters and election officials. Non-functional requirements are those that are imposed on the system by external entities or where the system depends on external behaviors outside its control, such as specific election certification guidelines and operational procedures. Each of these groups is itself divided into several categories, and Figure 5.1 gives a high-level overview of these.

The following is a high-level description of the categories and many of the requirements within each; Appendix A contains a complete listing of all E2E VIV system requirements expressed in the Business Object Notation.

5.1 Technical Requirements

There are ten categories of technical requirements for E2E VIV systems: functional, accessibility, usability, security, authentication, auditing, system operational, reliability, interoperability, and certification.

5.1.1 Functional

The functional requirements of an E2E VIV system deal primarily with the casting and recording of ballots and associated voter records. One important requirement is that there must be a correspondence between the recorded ballots and the voters that are listed as having voted; a ballot cannot be recorded without a voter casting it, and a voter cannot be listed as having voted without casting a ballot. Similarly, if a voter is informed by the system that her ballot has been successfully cast, the system must correctly retain the record of her having voted and her cast ballot information even in the event of server failures.

Another functional requirement is the property of *receipt freedom*: it must be impossible for a voter to prove to anybody any information regarding how she voted her ballot, beyond what can be mathematically deduced from the final distribution of votes. For example, if a referendum passes with 100% of the vote, there is no way to hide the fact that every voter approved of the referendum; however, if the result is mixed, it must be impossible for any individual voter to prove how she voted. This must be the case even when the voter can create digital evidence of her actions by, for example, video recording the ballot casting process or photographing a completed ballot.

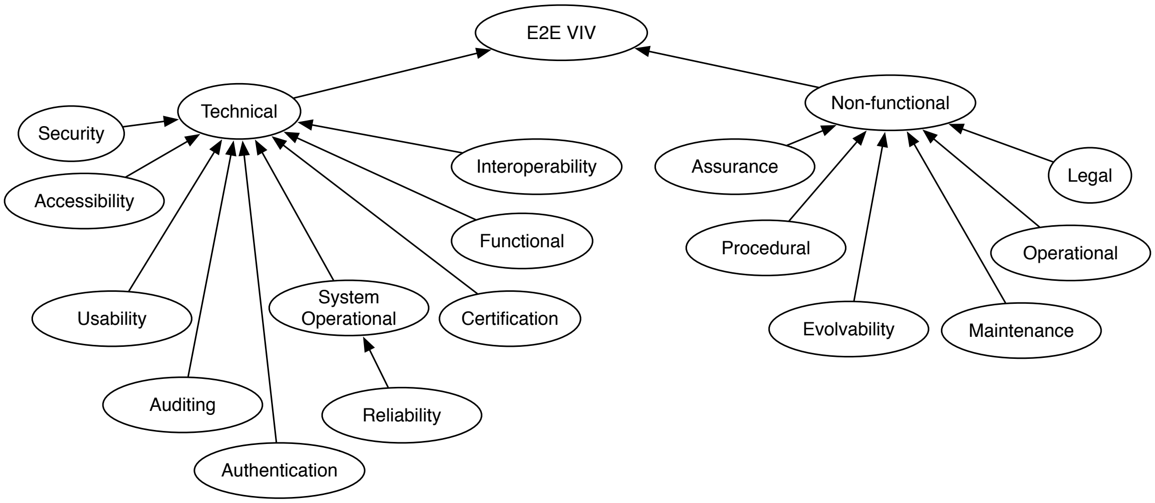

Figure 5.1: The hierarchy of requirements for E2E VIV systems.

In some elections voters are allowed to cast multiple ballots with only the last cast ballot counting toward the final election tally, while in others voters are prohibited from casting multiple ballots. The system must accommodate both of these election formats, ensuring that only the last cast ballot is counted for each voter when multiple ballots are allowed and ensuring that each voter casts at most one ballot otherwise.

Maintaining voter anonymity is critical, so it must be impossible after the election to reconstruct a link between a cast ballot and any identifying information about the voter who cast it. However, in systems that support the casting of multiple ballots, it is important to maintain links between voters and their ballots *during* the election to ensure that later ballots replace the correct earlier ballots. To balance these concerns, any link between a ballot and the voter who cast it must be irrevocably broken once it is conclusively determined that the ballot will be counted toward the final tally.

Finally, because the voter should be able to focus on the voting process without undue distractions or external influences, the voting system must not display or permit the display of any advertising or commercial logos during a voting session; the exception to this rule is that an election jurisdiction may display its own logo to the voter during the voting process. Along the same lines, the voting system must not display any links to other Internet sites outside of the voting system, except to provide help with the actual mechanics of voting.

5.1.2 Usability

The usability of an E2E VIV system is critical to its successful adoption and use. Since the user experience is so important, many of the requirements of the system have some relation to usability even though they may be categorized under other headings. There are, however, two requirements that are exclusively related to the usability of the system with respect to vote casting and one general usability requirement that applies to the system as a whole.

The first vote casting requirement is that, if a voter receives a final vote confirmation (e.g., “Thank you for voting!” or a similar notice) from the system, she can be certain that her ballot was recorded correctly. This is the usability counterpart to the functional requirement that ballot records and voter records must be maintained correctly even in the event of server failures.

The second vote casting requirement is that, if a voter is uncertain whether or not her ballot was recorded (e.g., she clicked a “submit” button but never got a response from the system), she must be free to attempt to vote again.

Finally, usability testing must be performed on any E2E VIV system before it is deployed. The reports of the usability testing must be made public, and the system must achieve satisfactory test results before being deployed in a real election.

5.1.3 Accessibility

Accessibility—the property of being usable by and useful to the disabled—is one of the main goals of an E2E VIV system. It is closely related to usability, but there are several requirements associated specifically with accessibility that go beyond typical usability requirements.

Users must be involved in the design of the system to identify accessibility constraints at each stage of the development process. Consideration must be given to the system's compatibility with existing technologies designed to help disabled individuals; for example, the system should be developed in a way that allows assistive input devices such as switches, eye trackers and screen readers to be used in addition to keyboards, mice and touchscreens. Similarly, the system's presentation of voting options should be optimized to voters' needs by providing alternative display fonts, audio representations, braille representations, and other representations as appropriate.

All possible measures must be taken to ensure that the system can be used by all voters and, if that is not possible in all circumstances, to provide access to alternative methods of voting for those voters who cannot use the system.

Finally, accessibility testing must be performed in addition to the previously-mentioned mandatory usability testing. The reports of the accessibility testing must be made public, and the system must achieve satisfactory test results before being deployed in a real election.

5.1.4 Security and Authentication

Security and authentication are closely related and together represent the broadest set of technical requirements, consisting of both requirements on the E2E VIV system itself (data storage, communications, etc.) and requirements on the voting and counting processes enabled by the system (voter authorization, voter privacy, tally accuracy, etc.).

It is crucial that data integrity be ensured throughout the system. Therefore, measures must be taken to ensure that no data can be permanently lost in the event of a breakdown or fault affecting the system; that the system maintains the integrity of the voters' register, lists of candidates, ballot information, cast ballots, and other critical information, in addition to authenticating the original source(s) of that information and tracking provenance where appropriate; that all data communications within the system have associated integrity checks; that system equipment under the control of the electoral authority is protected against influences that could modify the election results; and that the integrity of the election results does not depend in any way upon the security of system equipment not under control of the electoral authority. The system must perform regular "health checks" to ensure that data integrity has been maintained, that all its components are operating in accordance with their specifications, and that all system services are available.

Accurate timing information is critical to security, both in terms of providing evidence of compliance with applicable regulations and in terms of detecting attacks on and potential breaches of the system. The system must therefore maintain reliable synchronized time sources, with sufficient accuracy to maintain timing data for audit trails, election observation data, and time limits for various aspects of the election process. It must be possible to determine, using the timing information stored by the system, whether nominations (and, if required, acceptance thereof by the candidate or electoral authority), voter registration, and vote casting have occurred within the prescribed time limits for those actions.

Authentication and authorization are also important aspects of security. The system must ensure that each individual can be identified uniquely, so that there is no possibility of mistaking one individual for another. The system must also maintain the privacy of individuals, by ensuring that all personally identifiable data is kept confidential as far as is allowed by the legal requirements of the electoral jurisdiction. The system must allow access to each of its services only to authorized users; for example, only individuals who represent the electoral authority may be allowed to load ballot information into the system.

The authentication mechanisms used to gain access to the system must, as far as possible, protect authentication secrets (passwords, one-time access codes, biometrics, etc.) so that unauthorized entities cannot acquire them. Authentication to the system may not be carried out through third parties; that is, existing online accounts such as those at Facebook, Google and Twitter may not be used as authentication mechanisms. The security of the authentication mechanism must not be affected by any potential breach of any public or commercial database (e.g., a credit card database, the Social

Security database), and it should not be possible for an attacker to impersonate a voter even if the entire database used for authentication in the system is compromised. Individual authentication secrets themselves must be changeable or revokable at any time, at the behest of either the individual or election officials, and must be changed for all individuals at least once in every election cycle.

With respect to the actual voting process, only eligible voters may be allowed to cast ballots and the system must ensure that only the appropriate number of ballots is cast by each voter. It must be possible for a voter to verify that the system has presented her with an authentic ballot and, in the case of remote voting, that she has a secure connection to an official server.

The privacy of the vote must be preserved end-to-end to the maximum extent possible, and individual voters may not waive the privacy of their votes. In the case of remote voting, vote privacy must be preserved even in the presence of arbitrary malicious code on the voter's computer (corrupted client software, key logging software or devices, etc.). Any client software used in remote voting must not send data to any Internet host except those associated with the E2E VIV system or provide any information to third parties (e.g., Facebook, Twitter, etc.) regarding the act of voting. Any residual information that could be used to discover a voter's choices must be destroyed after a ballot has been cast; if a voter uses a computer outside the control of the electoral authority to cast her vote, she must be provided with instructions for destroying any such information on that computer.

With respect to vote counting, the system must accurately count the votes and the counting process must be reproducible. The system must also maintain the availability and integrity of all information used to generate the final tally and all information regarding the counting process itself for as long as required. Vote tabulation must be *software independent*; it must be possible to reconstruct a correct tally from some record even if the election system software is compromised.

Finally, it is expected that a deployed E2E VIV system will be an attractive target for highly-capable adversaries that wish to influence election results or to disrupt election processes. With this in mind, the system must be designed and tested assuming that an adversary has a budget of US$10 per voter per election that can be applied toward any critical subset of votes or voters of their choosing; thus, an E2E VIV system for use in a U.S. presidential election would need to be designed and tested assuming that an adversary has a budget of approximately US$1,300,000,000.

The electoral authority shall have overall responsibility for compliance with these security requirements, and such compliance shall be assessed by independent bodies as appropriate.

5.1.5 Auditing

The ability to perform comprehensive audits of system activity is one of the important distinguishing aspects of an E2E VIV system as compared to other voting systems; as a result, there are several system requirements related specifically to auditing, in addition to those security requirements (such as the tracking of accurate timing information) that touch on auditing.

First, the audit system must be designed and implemented as part of the E2E VIV system from the beginning; it cannot be added as an afterthought to an existing system. Audit and monitoring facilities must be integrated into all levels of the system, from low-level communications among individual computers to high-level interactions with election officials. The system must keep audit logs of all activity relevant to the conduct and outcome of the election, and these logs must be unmodifiable once they are written and as complete as possible without violating voter privacy.

The audit system must actively report on potential issues and threats, rather than merely serving as a passive repository of system logs. It must record at least the following events and actions with accurate timing information: all voting-related information, including the number of eligible voters and votes cast, the number of invalid votes, count and recount results, etc.; any detected attacks on the operation of the system or its communication infrastructure; and any system failures, malfunctions, or other detected threats to proper system operation. It must provide sufficient information to election observers in real time, and after the election's conclusion, to verify that the election is carried out in accordance with applicable law.

The audit system must also be able to cross-check and verify the correct operation of the voting system and the accuracy of the election results, to detect voter fraud, and to prove that all counted votes are legitimate and that all ballots have been counted. In situations where the system cannot verify the legitimacy of all the votes, it must be capable of giving an upper bound on the number of affected ballots. If a tradeoff must be made between maintaining voter privacy and identifying the perpetrators of fraud, the system must resolve that tradeoff in favor of voter privacy.

In order for an E2E VIV system to be trusted, its auditability must extend to its own source code as well as the activities it performs during an election. Therefore, the E2E VIV system software, including any official monitoring and auditing applications, must be published in source form along with documentation, instructions for building and running, and a digital signature as a proof of authenticity.

5.1.6 System Operational

System operational requirements ensure that the system is configured, updated, and run in a transparent, accountable way that allows for the other requirements to be fulfilled. One important such requirement is that there must be official published manifests of the system used to run any election, indicating details of the software and versions used, dates of installation, and brief descriptions of their functionality. Both public and private manifests must be maintained; these should be identical, except that details about software used solely to protect the system against attacks may be omitted from the public manifest for security reasons. Well-defined procedures must exist for both updating the manifests to reflect changes to the installed software and checking the installed software against the manifests to detect tampering.

Before every election period, all equipment (including all software) must be checked and approved in accordance with procedures devised by the electoral authority. This check must include a check of the software against the manifests, as well as any necessary tests to establish that the system complies with its technical specification.

During an election period, key equipment must be located in a guarded, secure area at all times. There must be a contingency plan for system failures including provisions for backup and failover systems, which must conform to the same standards and requirements as the systems they replace. In addition, sufficient arrangements for data backup must be in place, continuously monitored, and always available during the election; election staff must be ready to intervene rapidly, according to a procedure established by the electoral authority, in the event of incidents during an election. Individuals responsible for the voting equipment must follow established procedures to ensure that the equipment and its use satisfy requirements.

To ensure accountability on the part of the electoral authority and election system vendors, a report containing every software manifest change and every violation of data security, system security, physical security or control procedures must be prepared and made public by the electoral authority within a reasonable amount of time after every election.

5.1.7 Reliability

In order to be successfully used to conduct elections, an E2E VIV system must satisfy strict reliability requirements with respect to both its behavior under normal conditions and its behavior while under attack.

In general, the back-end (i.e., non-voter-facing) components of the system must have a proven mean time before failure (MTBF) of at least one week under constant peak expected load; that is, it must have been shown in multiple actual tests of mock elections to run continuously for at least a week at the highest expected voter participation rate. The one week MTBF requirement applies only during normal operation, not while the system is under attack.

In addition to the MTBF requirement, the system must also exhibit 99.9% uptime during the election period, and must be able to recover from any failure other than a regional natural disaster or malicious attack in less than 10 minutes. This must be demonstrated by inducing failures in actual mock election situations, e.g., by unexpectedly unplugging servers or disconnecting storage devices. Redundant failover components must be in place for all critical components of the system in order to ensure the 10 minute maximum recovery time.

An E2E VIV system is likely to be a tempting target for distributed denial of service (DDoS) attacks; it must be able to continue correct operation during a sustained DDoS attack at a specified level on any combination of its back-end components with no more than a specified acceptable degradation of response time to voters during the attack. The specified attack level and acceptable degradation of response time will vary among election types; for example, a system running a national election must be able to resist a significantly higher level of attack than a system running a county election. Our initial suggestions for the thresholds for a national election are that the system must continue operating correctly under a DDoS attack at a level of 100 gigabits per second, with no more than a 15 second degradation of response time.

The ability of the system to survive DDoS attacks and continue operation while fulfilling the response time requirements must be demonstrated in the actual network configuration to be used during the election, and the required thresholds for these values should be re-evaluated every election cycle to keep pace with advancement in attack technology.

5.1.8 Interoperability

E2E VIV systems must use open, rather than proprietary, data and communication standards for interoperability among their various components and services. Whenever possible, the Election Markup Language (EML) or a similar standard ratified by an international standards body should be used for data interchange and configuration within the system. The standards used within the system should allow for localization of election data in situations where such localization is required.

The log data for the system, and documentation describing its meaning and format, must be available for public download so that anybody can download, inspect, and publish concerns based on the system logs.

5.1.9 Certification

In order to provide sufficient evidence for certification of an E2E VIV system, each functional requirement must have an associated set of automated tests that demonstrate its fulfillment. These tests must be runnable on demand, and their results should be unambiguous and easily understandable.

In addition, the election protocol implemented by the system (communication, cryptographic, etc.) must have associated formal proofs of correctness and security.

5.2 Non-functional Requirements

There are five categories of non-functional requirements for E2E VIV systems: operational, procedural, legal, assurance, and maintenance/evolvability.

5.2.1 Operational

The operational requirements on E2E VIV systems deal with several distinct issues including election and registration timing, voter registration, candidate nominations and lists, receipt freedom, voter assistance, and the handling of hardware and software platform issues and election integrity violations.

Voters must be informed, in clear and simple language, of how electronic voting will be organized and what steps a voter will need to take in order to participate and vote electronically. Support and guidance with respect to voting procedures must be available to all voters. In the case of remote voting, such support and guidance must be available through a different, widely-available communication channel (such as a dedicated phone number) in addition to being available via the Internet. Voters must receive clear guidance about exactly what client configurations (i.e., hardware platforms, operating systems, browsers, browser plugins, other applications, and versions thereof) are required by or supported by the E2E VIV system, and what common components, plugins, or other software (e.g., pop-up blockers, script blockers) may interfere with voting. In addition, voters must receive clear guidance about configuration choices they can make to more strongly protect their privacy; for example, disabling cookies and browser history logging, running privacy-protecting browser plugins, voting from temporary virtual machines, logging out of social networks, disabling non-election-related Internet communications, etc.

In any election carried out using an E2E VIV system, the relevant jurisdiction's legal provisions must provide for clear timetables concerning all stages of the election. The period during which a vote may be cast electronically must not begin before the public is notified of the election; in particular, with respect to jurisdictions that allow remote electronic voting, the voting period must be defined and made known to the public well in advance of its start. In jurisdictions where remote voting takes place concurrently with voting at supervised polling stations, the time periods for remote and supervised voting need not be identical; however, remote voting should not be allowed after the period for supervised voting has ended.

An E2E VIV system must have a publicly accessible voters' register that is regularly updated. Each voter must be able to check, at a minimum, that her information as recorded on the register is accurate, and must be able to request corrections of any inaccurate information. In jurisdictions where remote electronic voting takes place concurrently with voting at supervised polling stations, the system must be designed in a way such that it prevents any voter from voting more than once.

On any electronic ballot, all voting options must be presented equally; that is, there must be no distinguishing fonts, sizes, styles, or other embellishments that could cause one or more of the voting options to be perceived by a voter as "preferred". The ballot must be free of any information about the voting options—biographical information about candidates, interpretations of and statements about ballot initiatives, etc.—other than information strictly required for casting the vote or required by law to be on the ballot (for example, candidate party affiliation is often required to appear). The system must also avoid displaying any messages that may influence voters' choices. Additional information about voting options might be made available from an electronic voting site as part of an E2E VIV system, separate from the actual electronic ballot; if so, such information must be presented without bias.

E2E VIV systems are likely to be made available for testing by voters and election officials, both before and during elections. They must therefore indicate clearly, before the final casting of any ballot, whether the ballot is being cast in a real election or as part of a test. In the case of a test that occurs simultaneously with a real election, individuals casting test ballots should subsequently be directed to the appropriate voting channel for casting real ballots.

E2E VIV systems must exhibit receipt freedom (mentioned previously in the technical requirements); that is, they must not enable the voter to possess a proof of the choices they have made in a cast vote. In a supervised environment, voting information should disappear from the display (visual, audio or tactile, depending on accessibility requirements) used by the voter to cast the vote as soon as the vote has been cast. When a paper proof of an electronic vote is provided to the voter at a polling station, the voter must not be allowed to show it to any other person or to remove it from the polling station.

With respect to counting the votes, an E2E VIV system must not allow the disclosure of any vote counts until after the system has stopped accepting electronic ballots. Tally information must not be disclosed to the public until after the end of the voting period (including all polling station voting). Any decoding required for the counting of the votes shall be carried out as soon as practicable after the end of the voting period; representatives of the electoral authority must be able to participate in, and observers must be able to observe, the counting process. A record of the counting process must be kept, including timing information and identifying information for all persons involved in the counting process. In the event of any irregularity affecting the integrity of votes, it must be recorded that the affected votes had their integrity violated; the effect of such integrity violations on the election results will vary based on the legal provisions of the involved jurisdictions.

Finally, any deployed E2E VIV system must function correctly as an open system, where large parts (specifically, any remote client hardware and software) are unknown, unsecured, uncertified, and completely out of the control of election officials. The system must be auditable to the extent possible given this requirement, and the conclusions drawn from the audit process should be applied in future elections.

5.2.2 Procedural

Successful deployment of E2E VIV systems requires certain procedures to be followed with respect to their provisioning, certification, maintenance, availability, and use. Because such systems are critical pieces of public infrastructure, information about their functioning must be publicly available and information about the specific components of a system must be disclosed, at least to the relevant electoral authority, as required for verification and certification purposes. Before any such system is introduced, at appropriate intervals after its introduction, and in particular when any changes are made to the system, an independent body appointed by the electoral authority must verify that the system is working correctly and that all necessary security measures have been taken.

After introducing a system, the electoral authority must take steps to ensure that voters undesrtand its use and have confidence in the system; these may include outreach, practice elections, and any other measures the electoral authority sees fit. In particular, voters must be given an opportunity to practice any new electronic ballot casting method before, and separately from, the casting of an electronic ballot during a real election.

The electoral authority must take steps to ensure the reliability and security of the E2E VIV system; for example, guarding equipment, providing suitable reliable power supplies, etc. All possible steps should be taken to avoid the possibility of fraud or unauthorized intervention during the voting process, and the electoral authority must satisfy itself that the E2E VIV system is genuine and operates correctly before using it to conduct a real election.

Only individuals appointed by the electoral authority should have access to the central infrastructure, the servers, and the election data, and clear rules should be established for such appointments. Critical technical activities must be carried out by teams of at least two people, and the composition of such teams must be regularly changed. As far as possible, critical technical activities should take place outside of election periods.

Observers must be allowed to be present, to the extent permitted by law, to observe and comment on the conduct and establishment of the results of any election conducted using an E2E VIV system. During an election period, any authorized intervention affecting the system must be carried out by a team of at least two people, be the subject of a written report, and be monitored by representatives of the election authority and election observers.

The system must maintain the availability, integrity, and confidentiality of the votes. It must also keep the votes sealed until the counting process begins. Any votes stored or communicated outside controlled environments must be encrypted. Recounts must be possible, and any features of the system that may influence the correctness of the result must be verifiable. The system must also support partial or complete re-runs of elections.

Finally, there must be clear technical and legal procedures to be followed in the event that voters can prove that their votes were not received accurately or counted, or in the event that the official election verification application does not verify that the results of the Internet portion of the election are correct.

5.2.3 Legal

Legal requirements arise primarily from the application of existing law to E2E VIV systems. These include requirements on accessibility and availability; on the counting of votes, number of votes per voter, and anonymity of votes; and on restrictions with respect to reverse engineering or testing of E2E VIV systems.

To comply with accessibility and availability requirements, the voting interface of an E2E VIV system must be understandable and easily usable, and registration requirements for electronic voting must not pose an impediment to voter participation. E2E VIV systems should be designed, as far as is practicable, to maximize the opportunities they provide for the disabled. Unless remote electronic voting channels are universally accessible, they must be used only as an additional and optional means of voting beyond polling places or more traditional remote voting methods.

The E2E VIV system must insure that at most one electronic vote from each voter is included in the final tally, that every vote cast electronically is counted, and that each vote cast electronically is counted only once. In jurisdictions where electronic and traditional voting channels are used in the same election, there must be a secure and reliable method to aggregate all votes, prevent multiple votes by the same voter from being counted, and calculate correct results.

The way in which voters are guided through the process of electronic voting should be designed to prevent their voting precipitately or without reflection. Voters must be able to alter their choices at any point during an electronic voting process before casting their vote, or to stop the voting process, without their previous choices being recorded or made available to any other person under any circumstances. The electronic voting system must not permit any manipulative influence to be exercised over the voter during the voting process, must provide the voter with a means of participating in the election without exercising a preference (e.g., by casting a blank ballot), must indicate clearly to the voter when the voting procedure has been completed, and must preserve voter anonymity.

There must be no legal impediments to interested parties who want to study the E2E VIV system. In particular, no nondisclosure agreement or contract of any kind may be required for such download and study, or for building, testing and publishing test results for the E2E VIV system.

5.2.4 Assurance

There are several assurance requirements with respect to the implementation, documentation, and licensing of E2E VIV systems. First, client side software—that is, any software that is expected to be used on a system serving as a voting terminal, whether a supervised machine at a polling place or an unsupervised machine belonging to a voter—must be free of known bugs on a wide range of platform and software stack combinations. As previously discussed in Section 5.2.1, the specific supported platform and software stack combinations for the software must be clearly conveyed to voters. The system must exhibit strong security with respect to voter authentication, such that there is no way to automate forging or invalidation of voter authentication credentials without compromising the cryptographic protocols or secrets used in the system.

All aspects of the design, architecture, algorithms and documentation for the entire Internet voting system (not just the E2EV core) should be published and available for free download by anyone. As the system changes, all associated documentation must be kept up to date, and no new version of an E2E VIV system should be certified until it has up-to-date documentation.

The source code, build scripts, issue tracking system, security features, and related development information for the entire Internet voting system—all versions, for all supported platforms—should be made publicly available for free download and inspection, under a license that permits anyone to download, build, instrument, and test the system.

5.2.5 Maintenance and Evolvability

Maintenance and evolvability requirements are closely related, and essentially stipulate that an electoral authority, or any entity engaged by an electoral authority, must be able to change an E2E VIV system in response to changes in the legal or technical environment in which it operates.

The electoral authority must have the right and the ability to update the election system to conform to changes in applicable law, available technology, or threats to system integrity independent of the original vendors of the system. The electoral authority must also have the right and ability to patch election systems to correct flaws discovered in the algorithms, implementation, or deployment, subject to the documentation update requirement described above and the procedural requirement that the system must be re-verified for correct operation before being used to conduct a real election.

Chapter 6

Crypto Specification (Joe K./Dan) (15%)

6.1 Ideal Functionality of an E2E System—$\mathcal{F}_{\textsf{e2e}}$

The basic protocol followed by any system implementing E2E verifiable elections can be characterized by an *ideal functionality*. This ideal functionality, called $\mathcal{F}_{\textsf{e2e}}$ and presented in Figure 6.1, recognizes and interacts with the election authority EA, the set of eligible voters $V_1, \ldots, V_n$ and the auditor AU. These are "ideal-world" entities; a "real-world" implementation of $\mathcal{F}_{\textsf{e2e}}$ may involve more parties that will enable the implementation to realize the ideal functionality.

13: Dan parked this here, because it doesn't belong in the "requirements" chapter, but it may not really belong here either. Still, it or some variant of it probably does. Do with it what you will...

$\mathcal{F}_{\textsf{e2e}}$ accepts a number of commands from the election authority EA, the voters and the auditors. At the same time it informs the (ideal world) adversary of certain actions that take place and is influenced by the adversary to perform certain actions. The ideal functionality keeps track of which parties are corrupted and may act according to their corruption status.

$\mathcal{F}_{\textsf{e2e}}$ has two parameters:

1. A function $f : (X \cup \{\perp\})^n \to E$ that defines the election function, where X defines the set of all possible ways for an individual voter to vote and E is the set of all possible election results. The notation X^n denotes all possible strings of length n over the alphabet X. The symbol $\perp$ stands for "undefined." The election function f is invariant with respect to $\perp$, i.e., $f(\perp, x) = f(x)$ for all x.
2. A relation Q that defines the level of sensitivity to manipulation permitted by $\mathcal{F}_{\textsf{e2e}}$. In particular, for two possible election results T and T' we say that $Q(T, T')$ holds if and only if T' is sufficiently close to T. For the most strict version of $\mathcal{F}_{\textsf{e2e}}$ we define Q to be the equality relation over E (that is, $Q(T, T')$ holds if and only if $T = T'$).

The ideal functionality $\mathcal{F}_{\textsf{e2e}}^{f,Q}$ captures the following set of security characteristics:

- Provided the EA is not corrupted, the adversary is incapable of extracting the voters' selections.
- Provided the EA is not corrupted, all votes are recorded and tallied according to the election function $f(\cdot)$.
- Even if the EA is corrupted, a set of well defined votes are assigned to the voters of the election (however, such votes may deviate from the original voters' intent). The votes cannot be manipulated when the EA is honest.
- Even if the EA is corrupted, the functionality consistently returns the same tally result to all parties that request it. Furthermore, the functionality always tests the reported tally according to the predicate Q and reports the outcome, hence any substantial (according to Q) deviation from the recorded tally will be detectable by all honest parties.
- The functionality preserves voter intent and, in case of vote manipulation, the voter or an auditor can use the unique receipts provided in the completion of ballot-casting to test whether voter intent was manipulated by a corrupt EA. Any party may use those receipts, hence verification is "delegatable."

Functionality $\mathcal{F}_{\mathsf{e2e}}^{f,Q}$

The functionality recognizes and interacts with the following parties: (i) the election authority EA; (ii) the eligible voters $\mathcal{V} = \{V_1, \ldots, V_n\}$; (iii) the auditor AU; and (iv) the adversary $\mathcal{A}$. It is parameterized by the relation Q over E and the election function $f : (X \cup \{\perp\})^n \to E$.

- Upon receiving an input $(\texttt{Create}, sid, B)$ from the EA, record the tuple (sid, B) such that sid is the election identifier and B is a string defining the ballot of the election. Send $(\texttt{Create}, sid, B)$ to the adversary $\mathcal{A}$.
- Upon receiving an input $(\texttt{Deliver}, sid)$ from EA, deliver (B, s_i) to each voter V_i, where s_i is some voter-specific information that is provided by the adversary $\mathcal{A}$.[a]
- Upon receiving an input $(\texttt{Vote}, sid, a)$ from V_i, select a unique identifier vid and record the tuple (vid, a) provided $a \in X$. If EA is honest, send $(\texttt{Vote}, sid, vid)$ to the adversary $\mathcal{A}$; if EA is corrupted, send $(\texttt{Vote}, sid, vid, V_i, a)$ to $\mathcal{A}$.[b]
- Upon receiving $(\texttt{RecordVote}, sid, vid, b)$ from $\mathcal{A}$, verify that a tuple (vid, a) has been previously recorded and then record the tuple (V_i, a, b) provided that (i) V_i is a voter that has not previously been assigned a vote,[c] (ii) the value a is a valid choice consistent with the ballot description B, and (iii) the value b is unique amongst received $\texttt{RecordVote}$ messages. Finally, return $(\texttt{Receipt}, b)$ to V_i.
- Upon receiving an input $(\texttt{Tally}, sid)$ from the EA, collect all recorded inputs $\{(V_j, a_j, b_j)\}_{j \in \tilde{\mathcal{V}}}$, where $\tilde{\mathcal{V}}$ is the set of voters that voted successfully, and set $a_j = \perp$ for all $j \notin \tilde{\mathcal{V}}$. Compute $T = f(\langle a_1, \ldots, a_n \rangle)$ and return $(\texttt{Tally}, T)$ to $\mathcal{A}$.
- Upon receiving $(\texttt{RecordTally}, sid, \mathcal{M}, \hat{T})$ from $\mathcal{A}$, where $\mathcal{M}$ can be parsed as a polynomial-size circuit, set $\langle a'_1, \ldots, a'_n \rangle = \mathcal{M}(a_1, \ldots, a_n)$ and if $\exists j : (a'_j \neq a_j)$ and EA is honest then ignore the message. In any other case, record $(\texttt{Result}, T', \hat{T})$ and $\langle a'_1, \ldots, a'_n \rangle$, where T' is the election result calculated as $T' = f(\langle a'_1, \ldots, a'_n \rangle)$.
- Upon receiving $(\texttt{ReadTally}, sid)$ from any party, return $(\texttt{Result}, \hat{T}, Q(\hat{T}, T'))$.
- Upon receiving $(\texttt{Audit}, b)$ from from any party, recover the triple (V_j, a_j, b_j) such that $b_j = b$ and return 1 if and only if $(a'_j = a_j)$ and 0 otherwise.

[a] In some systems, voters may request this information actively; hence, $\mathcal{F}_{\mathsf{e2e}}$ will be passive and will not deliver the ballots. In such cases the adversary will adaptively provide the s_i values.

[b] In some systems the voter identity V_i can be leaked to the adversary during ballot casting.

[c] In some systems the voter is allowed to change his/her mind and hence vote multiple times.

Figure 6.1: The ideal functionality $\mathcal{F}_{\mathsf{e2e}}^{f,Q}$.

Consider now a protocol π that implements syntactically the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$ (i.e., has the same I/O characteristics as $\mathcal{F}_{\mathsf{e2e}}$). Following standard notation and terminology we have the following:

Definition 6.1.1 *Let f be an election function and Q a predicate over the election results. The protocol π implements $\mathcal{F}_{\mathsf{e2e}}^{f,Q}$ provided that, for all adversaries $\mathcal{A}$, there is a simulator $\mathcal{S}$ so that for all environments $\mathcal{Z}$ it holds that*

$$\mathsf{Exec}_{\pi,\mathcal{A},\mathcal{Z}} \approx \mathsf{Exec}_{\mathcal{S},\mathcal{Z}}^{\mathcal{F}_{\mathsf{e2e}}^{f,Q}} .$$

Note that the protocols that will be considered in practice may utilize simpler ideal functionalities. In such cases, the protocol π implements $\mathcal{F}_{\mathsf{e2e}}$ conditional on the existence and availability of these other functionalities. Such functionalities include “authenticated channels”, a “write-only bulletin board”, etc.

Party corruption. As stated, the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$ enables the adversary $\mathcal{A}$ to corrupt parties by issuing special $(\texttt{Corrupt}, P)$ messages. Given such a message, the ideal functionality $\mathcal{F}_{\mathsf{e2e}}$ will divulge to the adversary the complete I/O transcript from the interface between $\mathcal{F}_{\mathsf{e2e}}$ and P. We distinguish between static and adaptive corruptions. In the case of static corruptions all messages $(\texttt{Corrupt}, P)$ are delivered at the onset of the execution, while for adaptive corruptions they can be delivered at any time. For brevity we do not explicitly include the actions taken for `Corrupt` messages in the description of the functionality.

In the real world, the corruption of an entity expresses an action taken by the adversary that results in the complete control of the entity's computing environment. A corrupted voter, specifically, loses privacy completely and the adversary may take any action on her behalf. For example, if corruption of a voter happens prior to ballot casting the adversary may vote on her behalf, while if corruption of a voter happens after ballot casting the adversary will learn her choice. If the adversary corrupts the EA, it may try to manipulate some voters' ballots however to the extent permitted by $\mathcal{F}_{\mathsf{e2e}}$ (no matter how many parties are corrupted, $\mathcal{F}_{\mathsf{e2e}}$ always has the "upper hand").

6.1.1 Claims Regarding $\mathcal{F}_{\mathsf{e2e}}$

Claim 1 *Assuming the* EA *is not corrupted, the ideal functionality* $\mathcal{F}_{\mathsf{e2e}}^{f;Q}$ *leaks no information about how honest voters vote, except for information that is revealed from the partial tally of the votes of the honest voters (according to* f*).*

Claim 2 *The adversary may delay the recording of an honest voter's ballot, however when it is recorded the voter obtains a receipt that enables her to verify that her vote has been properly recorded and tallied.*

Claim 3 *The receipt each voter obtains after her vote is recorded is unique; assuming the* EA *is honest, each receipt is independent of the voter's choice and hence can safely be passed to, e.g., a third party auditor* AU.

Claim 4 *When the* EA *is corrupted it is possible for the adversary to manipulate all the votes (via computational manipulation* $\mathcal{M}(a_1, \ldots, a_n) = (a'_1, \ldots, a'_n)$*) and even provide an incorrect tally* $\hat{T}$*; nevertheless, the ideal functionality ensures that honest parties are notified about whether the reported tally* $\hat{T}$ *and the recorded tally* T' *satisfy the relation* Q*, i.e., it returns* $Q(T', \hat{T})$*.*

6.1.2 Security Properties Not Captured by $\mathcal{F}_{\mathsf{e2e}}$

We have intentionally omitted a number of security aspects from this specification of the ideal end-to-end functionality $\mathcal{F}_{\mathsf{e2e}}$:

- **Denial of service attacks.** The ideal functionality as written enables the adversary to prevent voters from completing the ballot-casting protocol and prevent the tally from becoming available. From a definitional point of view, expressing the mitigation of such attacks is feasible by assuming certain qualities of the underlying communication and message passing mechanisms employed in the implementation. One way to extend the functionality to capture such mitigation is to oblige the adversary to deliver the (`RecordVote`) and (`RecordTally`) messages by certain deadlines. In order to do this formally, a notion of time would have to be introduced in the model, for example by introducing a global clock functionality.

- **Coercion via corrupting voters.** Even though $\mathcal{F}_{\mathsf{e2e}}$ does not permit coercion via the receipts it provides, the adversary may still achieve coercion by corrupting a voter (e.g., hacking into the voter's computer). If this happens after the ballot-casting protocol, $\mathcal{F}_{\mathsf{e2e}}$ reveals the choice of the voter and hence the voter may be vulnerable to coercion. Addressing this in the model is feasible by further restricting the information that is divulged when voter corruption takes place. Various intermediate levels of corruption may be considered, e.g., is the voter capable of erasing some information? rewriting some information? etc.

- **Sybil attacks.** The set of voters $V_1, \ldots, V_n$ is predetermined and integrated into the functionality $\mathcal{F}_{\mathsf{e2e}}$. Hence, the adversary cannot manipulate the list of voters. It follows that $\mathcal{F}_{\mathsf{e2e}}$ is applicable to the setting where the list of voters is predetermined, assumed to be public, and immune to tampering by the adversary.

Chapter 7

Architecture (Joe K./Dan) (15%)

Chapter 8

System Specification (Joe K./Dan) (15%)

Chapter 9

Verification and Validation (Joe K./Dan/Adam) (20%)

9.1 Requirements and Scenarios

9.2 Methodology

9.3 Technologies

9.4 Interpreting Results

Chapter 10

Feasibility (*Unassigned*) (25%)

10.1 Threats and Security Risks

10.2 Availability

14: Should this be “accessibility”?

10.3 Usability

10.4 Legal Frameworks and Politics

10.5 LEO Considerations

10.6 Cost

10.6.1 Design and Development

10.6.2 Operational

10.6.3 Integration with Local Election Systems and Processes

Chapter 11

Conclusion (Joe K./Susan) (0%)

11.1 Results

11.2 Recommendation (YES or NO???)

11.3 Next Steps

11.3.1 Political/Legal Challenges

11.3.2 Research Challenges

11.3.3 Engineering Challenges

11.3.4 Business Opportunities

Appendix A

BON Representation of E2E VIV Requirements (Dan/Joe K.) (50%)

BON (from external files) will appear here. Currently it is just dumped in a somewhat reasonable order, but it will be cleaned up and brought up to date.

```
scenario_chart E2EVIV_REQUIREMENTS
indexing
  title: "Requirements for End-to-end Verifiable Internet Voting Systems.";
  editor: "Joe Kiniry <kiniry@galois.com>", "Daniel M. Zimmerman <dmz@galois.com>";
  created: "16 July 2014";
  revised: "April 2015"
explanation
  "Functional and non-functional requirements for end-to-end \
 \ verifiable internet voting systems.  Requirements consisting of two \
 \ or more sentences are in fact stipulating multiple, related \
 \ requiremenets in a since scenario.  We index requirements from one, \
 \ thus SYSTEM_AND_DATA_ACCESS_CONTROL requirement 1 is 'Only persons \
 \ appointed by the electoral authority shall have access to the \
 \ central infrastructure, the servers and the election data.'."
end

scenario_chart TECHNICAL_REQUIREMENTS
indexing
  partof: "E2EVIV_REQUIREMENTS";
explanation
  "General technical requirements for digital elections systems."
end

scenario_chart NON_FUNCTIONAL_REQUIREMENTS
indexing
  partof: "E2EVIV_REQUIREMENTS"
explanation
  "General non-functional requirements of digital voting systems."
end

scenario_chart ACCESSIBILITY_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "General requirements for accessibility of digital election systems."
```

```
scenario
  "MANDATORY_ACCESSIBILITY_TESTING" -- @ref Kiniry/Zimmerman
description
  "Accessibility testing for disabled and abled voters shall be performed, \
 \ and the reports of the testing made public. The system must achieve \
 \ satisfactory accessibility testing results before being used in a \
 \ binding election."

-- @ref Rec(2004)11 Accessibility
scenario "UNIVERSAL_ACCESSIBILITY" -- @ref Rec(2004)11 Appendix III, A. 61.
description
  "Measures shall be taken to ensure that the relevant software and \
 \ services can be used by all voters and, if necessary, provide access \
 \ to alternative ways of voting."

scenario "ACCESSIBILITY_STAKEHOLDERS" -- @ref Rec(2004)11 Appendix III, A. 62.
description
  "Users shall be involved in the design of e-voting systems, \
 \ particularly to identify constraints and test ease of use at each \
 \ main stage of the development process."

scenario "USER_FACILITIES_FOR_ACCESSIBILITY" -- @ref Rec(2004)11 Appendix III, A. 63.
description
  "Users shall be supplied, whenever required and possible, with \
 \ additional facilities, such as special interfaces or other \
 \ equivalent resources, such as personal assistance."

scenario "COMPLEMENT_ACCESSIBILITY_TECHNOLOGIES" -- @ref Rec(2004)11 Appendix III, A.
    64.
description
  "Consideration shall be given, when developing new products, to \
 \ their compatibility with existing ones, including those using \
 \ technologies designed to help people with disabilities."

scenario "ACCESSIBLE_VOTING_OPTIONS" -- @ref Rec(2004)11 Appendix III, A. 65.
description
  "The presentation of the voting options shall be optimised for the \
 \ voter."
end

scenario_chart ASSURANCE_REQUIREMENTS
indexing
  partof: "NON_FUNCTIONAL_REQUIREMENTS"
explanation
  "General non-functional assurance requirements which increase system \
 \ and election assurance."

scenario
  "CLIENT_ENVIRONMENTS" -- @ref David Jefferson
description
  "Client side software (applications, apps, scripts, etc.) should be \
 \ free of known bugs on a wide range of platform and software stack \
 \ combinations intended to be usable as voting terminals."

scenario
  "AUTHENTICATION_RESILIENCE" -- @ref David Jefferson
description
  "There must be no way to automate forging or invalidation of \
 \ voter authentications without compromising the cryptographic \
 \ protocols or secrets used in the system."

scenario
  "OPEN_DOCUMENTATION" -- @ref David Jefferson
description
  "All aspects of the design, architecture, algorithms and \
 \ documentation for the entire Internet voting system (not just the \
 \ E2EV core) should be published and available for free download by \
 \ anyone."

scenario
  "DOCUMENTATION_CONSISTENCY" -- @ref David Jefferson
description
  "As the system changes, all documentation must be kept up to \
 \ date.  No new version of an E2EV Internet voting system may be \
 \ certified until all documentation is up to date."

scenario
  "OPEN_SOURCE" -- @ref David Jefferson
description
  "The source code, build scripts, issue tracking system, security \
 \ features, and related development information for the entire \
 \ Internet voting system (all versions for all platforms) shall be \
 \ made publicly available for free download and inspection by \
 \ anyone."

scenario
  "SOURCE_LICENSE" -- @ref David Jefferson
description
  "The source code for all parts of the E2EV Internet voting system \
 \ shall be made publicly available under a license that permits \
 \ anyone to download the code and build, instrument, and test it."

end

scenario_chart AUDITING_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "General requirements pertaining to auditing systems and digital \
 \ election systems."

-- @ref Rec(2004)11 Audit, I. General
scenario "AUDIT_SYSTEMS" -- @ref Rec(2004)11 Appendix III, E. I. 100.
description
  "The audit system shall be designed and implemented as part of the \
 \ e-voting system.  Audit facilities shall be present on different \
 \ levels of the system: logical, technical and application."

scenario "AUDITING_COMPLETENESS" -- @ref Rec(2004)11 Appendix III, E. I. 101.
description
  "End-to-end auditing of an e-voting system shall include recording, \
 \ providing monitoring facilities and providing verification \
 \ facilities."

-- @ref Rec(2004)11 Audit, II. Recording
scenario "AUDIT_SYSTEM_BASELINE" -- @ref Rec(2004)11 Appendix III, E. II. 102.
description
  "The audit system shall be open and comprehensive, and actively \
 \ report on potential issues and threats."

scenario "AUDIT_SYSTEM_DATA" -- @ref Rec(2004)11 Appendix III, E. II. 103.
description
  "The audit system shall record times, events and actions, including: \
 \ a. all voting-related information, including the number of eligible \
 \ voters, the number of votes cast, the number of invalid votes, the \
 \ counts and recounts, etc.; b. any attacks on the operation of the \
 \ e-voting system and its communications infrastructure; c. system \
 \ failures, malfunctions and other threats to the system."

-- @ref Rec(2004)11 Audit, III. Monitoring
scenario "AUDIT_SYSTEM_EVIDENCE" -- @ref Rec(2004)11 Appendix III, E. III. 104.
description
  "The audit system shall provide the ability to oversee the election \
 \ or referendum and to verify that the results and procedures are in \
 \ accordance with the applicable legal provisions."

scenario "AUDIT_DATA_SECURITY" -- @ref Rec(2004)11 Appendix III, E. IIi. 105.
description
  "Disclosure of the audit information to unauthorized persons shall \
 \ be prevented."

scenario "AUDIT_DATA_SECRECY" -- @ref Rec(2004)11 Appendix III, E. III. 106.
description
  "The audit system shall maintain voter anonymity at all times."

-- @ref Rec(2004)11 Audit, II. Verifiability
scenario "AUDIT_SYSTEM_CAPABILITY" -- @ref Rec(2004)11 Appendix III, E. IV. 107.
description
  "The audit system shall provide the ability to cross-check and \
 \ verify the correct operation of the e-voting system and the accuracy \
 \ of the result, to detect voter fraud, and to prove that all counted \
 \ votes are authentic and that all votes have been counted."

scenario "AUDIT_SYSTEM_FOR_LEGAL_COMPLIANCE" -- @ref Rec(2004)11 Appendix III, E. IV.
    108.
description
  "The audit system shall provide the ability to verify that an \
 \ e-election or e-referendum has complied with the applicable legal \
 \ provisions."

-- @ref Rec(2004)11 Audit, II. Other
scenario "AUDIT_DATA_VALIDITY" -- @ref Rec(2004)11 Appendix III, E. V. 109.
description
  "The audit system shall be protected against attacks that may \
 \ corrupt, alter or lose records in the audit system."

scenario "AUDIT_DATA_CONFIDENTIALITY" -- @ref Rec(2004)11 Appendix III, E. V. 110.
description
  "The electoral authority shall take adequate steps to ensure that the \
 \ confidentiality of any information obtained by any person while \
 \ carrying out auditing functions is guaranteed."

scenario
  "LOG_BASICS" -- @ref David Jefferson
description
  "The Internet voting system should keep detailed logs of all \
 \ relevant activity."
```

```
scenario
  "LOG_IMMUTABILITY" -- @ref David Jefferson
description
  "Log entries must be unmodifiable once written."

scenario
  "LOG_COMMITMENT" -- @ref Ron Rivest
description
  "Log entries must accurately reflect the commitment character \
 \ of elections and the relationships among election events \
 \ (e.g., ballot, vote, voter, and election state transitions)."

scenario
  "LOG_DATA_COMPLETENESS" -- @ref David Jefferson
description
  "The log data should be as complete as possible, consistent with \
 \ maximum possible vote privacy."

scenario
  "PRIVACY_VS_FRAUD_TRADEOFF" -- @ref David Jefferson
description
  "If there is a tradeoff between vote privacy and the identification \
 \ of the perpetrators of fraud, the decision should be made in favor \
 \ of vote privacy."

scenario
  "VOTER_LIST" -- @ref David Jefferson
description
  "The list of voters who voted online should be published."
end

scenario_chart AUDITING_REQUIREMENTS_VERIFICATION
indexing
  partof: "AUDITING_REQUIREMENTS"
explanation
  "Requirements specific to auditing verifiable elections."

scenario
  "VERIFICATION_PARTIAL_FAILURE" -- @ref David Jefferson
description
  "The system, in the event that it does not verify the online \
 \ votes cast, must be capable of giving an upper bound on the \
 \ number of ballots that may have been affected."

scenario
  "VERIFICATION_SOURCE" -- @ref David Jefferson
description
  "Official verification applications, like the voting software itself, \
 \ must be published in source form along with documentation, build \
 \ directions, and a standard cryptographic hash of the source code."
end

scenario_chart AUTHENTICATION_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "General requirements relating to the authentication of principles \
 \ (both computers and humans) involved in any digital election \
 \ system."

scenario
  "VOTER_AUTHENTICATION" -- @ref David Jefferson
description
  "The voting service must by itself securely authenticate the voter \
 \ (verify identify the voter and verify his/her registration and/or \
 \ eligibility according to law to vote in the election) before \
 \ allowing him/her to cast a ballot (or modify or replace a \
 \ previously cast ballot)."

scenario
  "NO_THIRD_PARTY_AUTHENTICATION" -- @ref David Jefferson
description
  "Authentication must not be done through third party intermediaries \
 \ such as Facebook, iCloud, Google, Yahoo, Amazon, etc. that offer \
 \ authentication services."

scenario
  "SECRET_AUTHENTICATION_SHARED_SECRETS" -- @ref David Jefferson
description
  "Authentication for remote voting systems must not use personal \
 \ information, government or commercial account identifiers, etc."

scenario
  "AUTHENTICATION_DATA_UPDATES" -- @ref David Jefferson
description
  "Authentication secrets must be changeable or revokable at \
 \ any time at the behest of either the voter or election \
 \ officials."

scenario
  "AUTHENTICATION_DATA_REFRESH_PERIODICITY" -- @ref David Jefferson
description
  "All voter authentication secrets must be changed at least once in \
 \ every election cycle."

end

scenario_chart CERTIFICATION_FUNCTIONAL_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "Requirements relating to the functional certification of digital \
 \ election systems and elections."

scenario "AUTOMATED_TESTING" -- @ref Kiniry/Zimmerman
description
  "Each functional requirement must have an associated set of automated \
 \ tests that provide evidence that the requirement is fulfilled."

scenario "ELECTION_PROTOCOL_PROOFS" -- @ref Kiniry/Zimmerman
description
  "The election protocol shall have associated formal proofs of correctness \
 \ and security."
end

scenario_chart CERTIFICATION_NON_FUNCTIONAL_REQUIREMENTS
indexing
```

```
  partof: "NON_FUNCTIONAL_REQUIREMENTS"
explanation
  "Requirements relating to the non-functional certification of \
 \ election systems and elections."

-- @ref Rec(2004)11 Certification
scenario "CERTIFICATION_PROCESSES" -- @ref Rec(2004)11 Appendix III, F. 111.
description
  "The electoral authority shall introduce certification processes that allow \
 \ for any ICT (Information and Communication Technology) component to \
 \ be tested and certified as being in conformity with technical \
 \ requirements."

scenario
  "CERTIFICATION_PARTIES_COMPETENCE" -- @ref David Jefferson
description
  "Any E2EV Internet voting system should be certified by competent \
 \ professionals."

scenario
  "CERTIFICATION_REPORT_TRANSPARENCY" -- @ref David Jefferson
description
  "Any and all certification reports issued by certification \
 \ professionals must be public, whether they recommend \
 \ certification or not."

scenario
  "RECERTIFICATION_CONDITIONS" -- @ref David Jefferson
description
  "Any time there is a change in the voting system client or server \
 \ side or the E2EV system, all of the requirements must \
 \ be re-established and recertified.  Changes that mandate \
 \ re-certification include, but are not limited to: new supported \
 \ hardware platforms, OS’s, browsers, etc.; bug fixes and security \
 \ patches to voting client and/or server; changes or upgrades to \
 \ voting client or server in response to detected bugs or security \
 \ vulnerabilities, changes in law, or changes in threat environment."

scenario
  "RECERTIFICATION_PERIODICITY" -- @ref David Jefferson
description
  "The requirements must be re-established and recertified every \
 \ election cycle even if there are no changes."

scenario
  "VALIDATION_PLATFORM_COVERAGE" -- @ref David Jefferson
description
  "The system must be extensively tested on a wide range of platform \
 \ and software combinations."

scenario
  "PUBLIC_VALIDATION_PLATFORM_COVERAGE_RESULTS" -- @ref David Jefferson
description
  "All test procedures and results for platform coverage must be public."

end

scenario_chart EVOLVABILITY_REQUIREMENTS
indexing
```

```
  partof: "NON_FUNCTIONAL_REQUIREMENTS"
explanation
  "General requirements on the evolvability of digital election \
 \ systems."

scenario
  "ELECTORAL_AUTHORITY_UPDATE"
description
  "The electoral authority has the right and ability to update \
 \ election systems to conform to changes in applicable law, \
 \ available techology, or the system threat model."

end

scenario_chart FUNCTIONAL_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "General functional requirements for digital election systems."

scenario
  "CASTING_ATOMIC" -- @ref David Jefferson
description
  "Ballot casting shall be atomic with respect to server failures."

scenario
  "DETERMINISTIC_VOTING_PROCESS" -- @ref David Jefferson
description
  "If a server side failure occurs, no voter's balloting can be \
 \ left in an unknown state."

scenario
  "BALLOT_FINAL_STATES" -- @ref David Jefferson
description
  "Either a ballot is securely and completely cast and the \
 \ voter is marked as having voted, or no ballot is recorded and the \
 \ voter is not marked as having voted."

scenario
  "VOTE_RECORD_MONOTONICITY" -- @ref David Jefferson
description
  "If the system and the law allows a voter to cast multiple votes \
 \ with only the last one counting, or to cast a partial ballot with \
 \ the option of modifying it later, then each voting session must be \
 \ atomic with respect to server failures. If a failure occurs during the \
 \ voter's last session, then the votes cast as of his or her previous \
 \ session will count."

scenario
  "RECEIPT_FREEDOM" -- @ref David Jefferson
description
  "There must be no way for voters to prove to another party any \
 \ information regarding how they voted in any race (beyond what is \
 \ mathematically deducible from the final distribution of votes)."

scenario
  "VALID_BALLOT_PROVENANCE" -- @ref David Jefferson
description
  "Once it is determined that a ballot will be counted, the ballot \
 \ shall be irrevocably separated from the identification of the \
 \ voter who cast it."

scenario
  "MULTI_BALLOT_RECORD" -- @ref David Jefferson
description
  "If the voting system permits voters to modify or replace their \
 \ previously cast ballots, only the latest vote by each voter in \
 \ each race shall be counted in the final tally."

scenario
  "NO_DOUBLE_VOTE" -- @ref David Jefferson
description
  "But for systems supporting MULTI_BALLOT_RECORD, the voting system \
 \ shall not record more than one vote for any voter in any race."

scenario
  "NO_ADVERTISING" -- @ref David Jefferson
description
  "The voting system client must not display or permit the display of \
 \ any advertising or commercial logos in the window that contains the \
 \ voting session, other than those of the election jurisdiction \
 \ itself."

scenario
  "NO_EXTERNAL_LINKS" -- @ref David Jefferson
description
  "The voting system client must not display any links to other sites \
 \ except for help in the mechanics of voting."

end

scenario_chart INTEROPERABILTY_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "General requirements on the interoperability of digital election \
 \ systems."

-- @ref Rec(2004)11 Interoperabilty
scenario "OPEN_STANDARDS" -- @ref Rec(2004)11 Appendix III, B. 66.
description
  "Open standards shall be used to ensure that the various technical \
 \ components or services of an e-voting system, possibly derived \
 \ from a variety of sources, interoperate."

scenario "EML" -- @ref Rec(2004)11 Appendix III, B. 67.
description
  "The Election Markup Language (EML) shall be used whenever possible \
 \ for e-election and e-referendum applications."

scenario "DATA_LOCALIZATION" -- @ref Rec(2004)11 Appendix III, B. 68.
description
  "In cases that imply specific election or referendum data \
 \ requirements, a localization procedure shall be used to accommodate \
 \ these needs."

scenario
  "OPEN_LOG_FORMATS" -- @ref David Jefferson
```

```
description
  "The log data and documentation of its meaning and format shall be \
 \ available for public download so that anyone can download, inspect, \
 \ and publish concerns based on the logs."
end

scenario_chart LEGAL_REQUIREMENTS
indexing
  partof: "NON_FUNCTIONAL_REQUIREMENTS"
explanation
  "General legal requirements relating to legal matters and digital \
 \ election systems."

-- @ref Rec(2004)11 Universal Suffrage
scenario -- @ref Rec(2004)11 Appendix I, A. I. 1.
  "USABLE_UI"
description
  "The voter interface of an e-voting system shall be understandable and \
 \ easily usable."

scenario -- @ref Rec(2004)11 Appendix I, A. I. 2.
  "UNIMPEDED_REGISTRATION"
description
  "Possible registration requirements for e-voting shall not pose \
 \ an impediment to the voter participating in e-voting."

scenario -- @ref Rec(2004)11 Appendix I, A. I. 3.
  "MAXIMIZE_DISABLED_ACCESSIBILITY"
description
  "E-voting systems shall be designed, as far as it is practicable, to \
 \ maximize the opportunities that such systems can provide for persons \
 \ with disabilities."

scenario -- @ref Rec(2004)11 Appendix I, A. I. 4.
  "REMOTE_ONLY_SUPPLEMENTARY"
description
  "Unless channels of remote e-voting are universally accessible, they \
 \ shall be only an additional and optional means of voting."

-- @ref Rec(2004)11 Equal suffrage
scenario -- @ref Rec(2004)11 Appendix I, A. II. 5--6.
  "AT_MOST_ONE_VOTE_PER_VOTER"
description
  "The e-voting system shall ensure that at most one electronic vote from \
 \ each voter is included in the final tally."

scenario -- @ref Rec(2004)11 Appendix I, A. II. 7.
  "VALID_TALLY"
description
  "Every vote deposited in an electronic ballot box shall be counted, and \
 \ each vote cast in the election or referendum shall be counted only once."

scenario -- @ref Rec(2004)11 Appendix I, A. II. 8.
  "VOTE_AGGREGATION"
description
  "Where electronic and non-electronic voting channels are used in the same \
 \ election or referendum, there shall be a secure and reliable method to \
 \ aggregate all votes and to calculate the correct result."
```

```
-- @ref Rec(2004)11 Free suffrage
scenario -- @ref Rec(2004)11 Appendix I, A. III. 9.
  "FREE_SUFFRAGE"
description
  "The organization of e-voting shall secure the free formation and \
 \ expression of the voter’s opinion and, where required, the \
 \ personal exercise of the right to vote."

scenario -- @ref Rec(2004)11 Appendix I, A. III. 10.
  "REFLECTIVE_VOTING_PROCESS"
description
  "The way in which voters are guided through the e-voting process \
 \ shall be such as to prevent their voting precipitately or without \
 \ reflection."

scenario -- @ref Rec(2004)11 Appendix I, A. III. 11.
  "FLEXIBLE_VOTING_PROCESS"
description
  "Voters shall be able to alter their choice at any point in the \
 \ e-voting process before casting their vote, or to break off the \
 \ procedure, without their previous choices being recorded or made \
 \ available to any other person."

scenario -- @ref Rec(2004)11 Appendix I, A. III. 12.
  "NO_VOTER_MANIPULATION"
description
  "The e-voting system shall not permit any manipulative influence to \
 \ be exercised over the voter during the voting."

scenario -- @ref Rec(2004)11 Appendix I, A. III. 13.
  "BLANK_VOTE"
description
  "The e-voting system shall provide the voter with a means of \
 \ participating in an election or referendum without the voter \
 \ exercising a preference for any of the voting options, for example, \
 \ by casting a blank vote."

scenario -- @ref Rec(2004)11 Appendix I, A. III. 14.
  "CONCLUSION_OF_VOTING_PROCESS"
description
  "The e-voting system shall indicate clearly to the voter when the \
 \ vote has been cast successfully and when the whole voting procedure \
 \ has been completed."

scenario -- @ref Rec(2004)11 Appendix I, A. III. 15.
  "IMMUTABLE_VOTES"
description
  "Except in systems supporting MULTI_BALLOT_RECORD, the e-voting system \
 \ shall prevent the changing of a vote once that vote has been cast."

-- @ref Rec(2004)11 Secret suffrage
scenario -- @ref Rec(2004)11 Appendix I, A. IV. 16.
  "SECRET_SUFFRAGE"
description
  "E-voting shall be organized in such a way as to exclude at any \
 \ stage of the voting procedure and, in particular, at voter \
 \ authentication, anything that would endanger the secrecy of the \
 \ vote."
```

```
scenario -- @ref Rec(2004)11 Appendix I, A. IV. 17.
  "ANONYMOUS_VOTES"
description
  "The e-voting system shall guarantee that votes in the electronic \
 \ ballot box and votes being counted are, and will remain, anonymous, \
 \ and that it is not possible to reconstruct a link between the vote \
 \ and the voter."

scenario -- @ref Rec(2004)11 Appendix I, A. IV. 18.
  "NO_INDIRECT_SECRECY_VIOLATION"
description
  "The e-voting system shall be so designed that the expected number \
 \ of votes in any electronic ballot box will not allow the result to \
 \ be linked to individual voters."

scenario -- @ref Rec(2004)11 Appendix I, A. IV. 19.
  "NO_SECRET_SUFFRAGE_SIDE_CHANNEL"
description
  "Measures shall be taken to ensure that the information needed \
 \ during electronic processing cannot be used to breach the secrecy of \
 \ the vote."

scenario
  "NO_NDAS_FOR_STUDY" -- @ref David Jefferson 22-6-2014
description
  "No nondisclosure agreement or any other contract shall be required \
 \ to download and study the Internet voting system."

scenario
  "NO_NDAS_FOR_AUDIT" -- @ref David Jefferson 22-6-2014
description
  "No nondisclosure agreement or any other contract shall be required \
 \ to download, instrument, build, test, and publish test results for \
 \ an E2EV Internet voting system."

end

scenario_chart MAINTENANCE_REQUIREMENTS
indexing
  partof: "NON_FUNCTIONAL_REQUIREMENTS"
explanation
  "General requirements relating to the maintainence of digital election \
 \ systems."

scenario
  "ELECTORAL_AUTHORITY_PATCH"
description
  "The electoral authority has the right and ability to patch \
 \ election systems to correct flaws discovered in the algorithms, \
 \ implementation, or deployment."

end

scenario_chart OPERATIONAL_REQUIREMENTS
indexing
  partof: "NON_FUNCTIONAL_REQUIREMENTS"
explanation
  "General operational requirements for digital election systems."
```

```
-- @ref Rec(2004)11 Notification
scenario "ELECTION_TIMETABLES" -- @ref Rec(2004)11 Appendix II, I. 36.
description
  "Domestic legal provisions governing an e-election or e-referendum \
 \ shall provide for clear timetables concerning all stages of the \
 \ election or referendum, both before and after the election or \
 \ referendum."

scenario "ELECTION_PERIOD" -- @ref Rec(2004)11 Appendix II, I. 37.
description
  "The period in which an electronic vote can be cast shall not begin \
 \ before the notification of an election or a referendum.  Particularly \
 \ with regard to remote e-voting, the period shall be defined and made \
 \ known to the public well in advance of the start of voting."

scenario "EVOTING_OUTREACH" -- @ref Rec(2004)11 Appendix II, I. 38.
description
  "The voters shall be informed, well in advance of the start of \
 \ voting, in clear and simple language, of the way in which the \
 \ e-voting will be organised, and any steps a voter may have to take \
 \ in order to participate and vote."

-- @ref Rec(2004)11 Voters
scenario "VOTER_VERIFIABLE_VOTER_REGISTER" -- @ref Rec(2004)11 Appendix II, II. 39.
description
  "There shall be a voters’ register that is regularly updated.  The \
 \ voter shall be able to check, as a minimum, the information that is \
 \ held about him/her on the register, and request corrections."

scenario "ONLINE_VOTER_REGISTER" -- @ref Rec(2004)11 Appendix II, II. 40.
description
  "The possibility of creating an electronic register and introducing \
 \ a mechanism allowing online application for voter registration \
 \ and, if applicable, for application to use e-voting, shall be \
 \ considered.  If participation in e-voting requires a separate \
 \ application by the voter and/or additional steps, an electronic, \
 \ and, where possible, interactive procedure shall be considered."

scenario "VOTER_REGISTRATION_ELECTION_OVERLAP" -- @ref Rec(2004)11 Appendix II, II.
    41.
description
  "In cases where there is an overlap between the period for voter \
 \ registration and the voting period, provision for appropriate \
 \ voter authentication shall be made."

-- @ref Rec(2004)11 Candidates
scenario "ONLINE_CANDIDATE_NOMINATION" -- @ref Rec(2004)11 Appendix II, III. 42.
description
  "The possibility of introducing online candidate nomination may be \
 \ considered."

scenario "PUBLIC_CANDIDATE_LIST" -- @ref Rec(2004)11 Appendix II, III. 43.
description
  "A list of candidates that is generated and made available \
 \ electronically shall also be publicly available by other means."

-- @ref Rec(2004)11 Voting
scenario "MULTIPLE_CHANNELS_ONE_VOTE" -- @ref Rec(2004)11 Appendix II, IV. 44.
description
```

```
  "Where remote e-voting takes place while polling stations are open, \
 \ the system shall be so designed that it prevents any voter from \
 \ voting more than once."

scenario "VOTING_PERIOD_INVARIANT" -- @ref Rec(2004)11 Appendix II, IV. 45.
description
  "Remote e-voting may start and/or end at an earlier time than the \
 \ opening of any polling station.  Remote e-voting shall not continue \
 \ after the end of the voting period at polling stations."

scenario "UNIVERSAL_VOTER_HELP" -- @ref Rec(2004)11 Appendix II, IV. 46.
description
  "For every e-voting channel, support and guidance arrangements on \
 \ voting procedures shall be set up for, and be available to, the \
 \ voter.  In the case of remote e-voting, such arrangements shall also \
 \ be available through a different, widely-available communication \
 \ channel."

scenario "FAIR_VOTING_OPTIONS" -- @ref Rec(2004)11 Appendix II, IV. 47.
description
  "There shall be equality in the manner of presentation of all voting \
 \ options on the device used for casting an electronic vote."

scenario "VOTING_OPTIONS_ONLY" -- @ref Rec(2004)11 Appendix II, IV. 48.
description
  "The electronic ballot by which an electronic vote is cast shall be \
 \ free from any information about voting options, other than that \
 \ strictly required for casting the vote.  The e-voting system shall \
 \ avoid the display of other messages that may influence the voters' \
 \ choice."

scenario "FAIR_VOTING_OPTION_INFORMATION" -- @ref Rec(2004)11 Appendix II, IV. 49.
description
  "If it is decided that information about voting options will be \
 \ accessible from the e-voting site, this information shall be \
 \ presented with equality."

scenario "BINDING_ELECTION_CLARITY" -- @ref Rec(2004)11 Appendix II, IV. 50.
description
  "Before casting a vote using a remote e-voting system, voters' \
 \ attention shall be explicitly drawn to the fact that the e-election \
 \ or e-referendum in which they are submitting their decision by \
 \ electronic means is a real election or referendum.  In case of \
 \ tests, participants shall have their attention drawn explicitly to \
 \ the fact that they are not participating in a real election or \
 \ referendum and shall, when tests are continued at election times, \
 \ at the same time be invited to cast their ballot by the voting \
 \ channel(s) available for that purpose."

scenario "REMOTE_RECEIPT_FREEDOM" -- @ref Rec(2004)11 Appendix II, IV. 51.
description
  "A remote e-voting system shall not enable the voter to be in \
 \ possession of a proof of the content of the vote cast."

scenario "SUPERVISED_VOTE_RECEIPT_FREEDOM" -- @ref Rec(2004)11 Appendix II, IV. 52.
description
  "In a supervised environment, the information on the vote shall \
 \ disappear from the visual, audio or tactile display used by the \
 \ voter to cast the vote as soon as it has been cast.  Where a paper \
 \ proof of the electronic vote is provided to the voter at a polling \
 \ station, the voter shall not be able to show it to any other per- \
 \ son, or take this proof outside of the polling station."

-- @ref Rec(2004)11 Results
scenario "SECRET_INTERMEDIATE_TALLY" -- @ref Rec(2004)11 Appendix II, V. 53.
description
  "The e-voting system shall not allow the disclosure of the number of \
 \ votes cast for any voting option until after the closure of the \
 \ electronic ballot box.  This information shall not be disclosed to \
 \ the public until after the end of the voting period."

scenario "NO_ITALIAN_ATTACK" -- @ref Rec(2004)11 Appendix II, V. 54.
description
  "The e-voting system shall prevent processing information on votes \
 \ cast within deliberately chosen sub-units that could reveal \
 \ individual voters’ choices."

scenario "DECODING_LATENCY" -- @ref Rec(2004)11 Appendix II, V. 55.
description
  "Any decoding required for the counting of the votes shall be \
 \ carried out as soon as practicable after the closure of the voting \
 \ period."

scenario "TALLY_OBSERVATION" -- @ref Rec(2004)11 Appendix II, V. 56.
description
  "When counting the votes, representatives of the competent electoral \
 \ authority shall be able to participate in, and any observers able to \
 \ observe, the count."

scenario "TALLY_RECORD" -- @ref Rec(2004)11 Appendix II, V. 57.
description
  "A record of the counting process of the electronic votes shall be \
 \ kept, including information about the start and end of, and the \
 \ persons involved in, the count."

scenario "INTEGRITY_VIOLATION_RECORD" -- @ref Rec(2004)11 Appendix II, V. 58.
description
  "In the event of any irregularity affecting the integrity of votes, \
 \ the affected votes shall be recorded as having their integrity violated."

-- @ref Rec(2004)11 Audit
scenario "SYSTEM_AUDITABILITY" -- @ref Rec(2004)11 Appendix II, VI. 59.
description
  "The e-voting system shall be auditable."

scenario "SYSTEM_AUDITS_IMPACT" -- @ref Rec(2004)11 Appendix II, VI. 60.
description
  "The conclusions drawn from the audit process shall be applied in \
 \ future elections and referenda."

scenario
  "OPEN_SYSTEM" -- @ref David Jefferson
description
  "The e-voting system must function correctly as an open system, \
 \ where large parts (the mix of client hardware and software in \
 \ fact) are unknown, unsecured, uncertified, and completely out \
 \ of control of election officials."
```

```
scenario
  "SUPPORTED_CLIENTS" -- @ref David Jefferson
description
  "Operators of voting systems must document exactly what client \
 \ configurations are required or supported, including: \
 \  - versions of hardware platforms (PCs, mobile devices, etc.) \
 \  - versions of specific operating systems for those platforms \
 \  - versions of specific browsers, plugins, protocols, or \
 \    other software applications, apps, components, and plugins."

scenario
  "CLIENT_INTERFERENCE" -- @ref David Jefferson
description
  "Operators of voting systems must document exactly which common \
 \ components, plugins, or other software interfere with voting (e.g., \
 \ flash blockers, popup blockers, script blockers, etc.)."

scenario
  "MANDATORY_CLIENT_TECHNOLOGY" -- @ref David Jefferson
description
  "Operators of voting systems must document exactly what configuration \
 \ choices the voter must make to successfully vote (e.g., mandate \
 \ Javascript)."

scenario
  "PRIVACY_ENHANCING_VOTER_OPTIONS" -- @ref David Jefferson
description
  "Operators of voting systems must document exactly what configuration \
 \ choices the voter might wish to make to more strongly protect \
 \ his/her vote privacy; e.g., disable cookies, run privacy-protecting \
 \ browser plugins, vote from virtual machine that is later destroyed, \
 \ log out of social networks, disable remote control and remote \
 \ administration tools, disable incoming connections, etc."

scenario
  "BREADCRUMBS_USER_ADVICE" -- @ref David Jefferson
description
  "Users may be advised to turn off browser history data, cookies, \
 \ logging data, and other tools that might retain a record of the \
 \ vote transaction whether the vote data itself or metadata."

end

scenario_chart PROCEDURAL_REQUIREMENTS
indexing
  partof: "NON_FUNCTIONAL_REQUIREMENTS"
explanation
  "General procedural requirements for digital electoin systems."

-- @ref Rec(2004)11 Transparency
scenario "VOTER_COMPREHENSION_AND_CONFIDENCE" -- @ref Rec(2004)11 Appendix I, B. I.
    20.
description
  "The electoral authority shall take steps to ensure that voters understand and \
 \ have confidence in the e-voting system in use."

scenario "PUBLIC_SYSTEM_FUNCTION" -- @ref Rec(2004)11 Appendix I, B. I. 21.
description
  "Information on the functioning of an e-voting system shall be made \
 \ publicly available."

scenario "VOTER_PRACTICE" -- @ref Rec(2004)11 Appendix I, B. I. 22.
description
  "Voters shall be provided with an opportunity to practice any new \
 \ method of e-voting before, and separately from, the moment of \
 \ casting an electronic vote."

scenario "OBSERVER_INVARIANTS" -- @ref Rec(2004)11 Appendix I, B. I. 23.
description
  "Any observers, to the extent permitted by law, shall be able to be \
 \ present to observe and comment on the e-elections, including the \
 \ establishing of the results."

-- @ref Rec(2004)11 Verifiability and accountability
scenario "DISCLOSURE_OBLIGATIONS" -- @ref Rec(2004)11 Appendix I, B. II. 24.
description
  "The components of the e-voting system shall be disclosed, at least \
 \ to the competent electoral authorities, as required for verification \
 \ and certification purposes."

scenario "CERTIFICATION_OBLIGATIONS" -- @ref Rec(2004)11 Appendix I, B. II. 25.
description
  "Before any e-voting system is introduced, and at appropriate \
 \ intervals thereafter, and in particular after any changes are made \
 \ to the system, an independent body, appointed by the electoral \
 \ authorities, shall verify that the e-voting system is working \
 \ correctly and that all the necessary security measures have been \
 \ taken."

scenario "RECOUNT_SUPPORTED" -- @ref Rec(2004)11 Appendix I, B. II. 26.
description
  "There shall be the possibility for a recount.  Other features of the \
 \ e-voting system that may influence the correctness of the results \
 \ shall be verifiable."

scenario "RERUN_SUPPORTED" -- @ref Rec(2004)11 Appendix I, B. II. 27.
description
  "The e-voting system shall not prevent the partial or complete \
 \ re-run of an election or a referendum."

-- @ref Rec(2004)11 Reliability and security
scenario "RELIABILITY_AND_SECURITY" -- @ref Rec(2004)11 Appendix I, B. III. 28.
description
  "The electoral authority shall ensure the reliability and \
 \ security of the e-voting system."

scenario "NO_FRAUD_OR_INTERVENTION" -- @ref Rec(2004)11 Appendix I, B. III. 29.
description
  "All possible steps shall be taken to avoid the possibility of fraud \
 \ or unauthorized intervention affecting the system during the whole \
 \ voting process."

scenario "SYSTEM_AVAILABILITY" -- @ref Rec(2004)11 Appendix I, B. III. 30.
description
  "The e-voting system shall contain measures to preserve the \
 \ availability of its services during the e-voting process.  It shall \
 \ resist, in particular, malfunction, breakdowns or denial of service \
 \ attacks."
```

```
scenario "SYSTEM_GENUINE_AND_CORRECT" -- @ref Rec(2004)11 Appendix I, B. III. 31.
description
  "Before any e-election or e-referendum takes place, the competent \
 \ electoral authority shall satisfy itself that the e-voting system \
 \ is genuine and operates correctly."

scenario "SYSTEM_AND_DATA_ACCESS_CONTROL" -- @ref Rec(2004)11 Appendix I, B. III. 32.
description
  "Only persons appointed by the electoral authority shall have access \
 \ to the central infrastructure, the servers and the election \
 \ data.  There shall be clear rules established for such \
 \ appointments.  Critical technical activities shall be carried out by \
 \ teams of at least two people.  The composition of the teams shall be \
 \ regularly changed.  As far as possible, such activities shall be \
 \ carried out outside election periods."

scenario "OPEN_BALLOT_BOX_INVARIANTS" -- @ref Rec(2004)11 Appendix I, B. III. 33.
description
  "While an electronic ballot box is open, any authorised intervention \
 \ affecting the system shall be carried out by teams of at least two \
 \ people, be the subject of a report, and be monitored by \
 \ representatives of the competent electoral authority and any \
 \ election observers."

scenario "VOTES_INVARIANTS" -- @ref Rec(2004)11 Appendix I, B. III. 34.
description
  "The e-voting system shall maintain the availability and integrity \
 \ of the votes.  It shall also maintain the confidentiality of the \
 \ votes and keep them sealed until the counting process.  If stored or \
 \ communicated outside controlled environments, the votes shall be \
 \ encrypted."

scenario "SEALED_VOTES_VOTER_RELATION" -- @ref Rec(2004)11 Appendix I, B. III. 35.
description
  "Votes and voter information shall remain sealed as long as the data \
 \ is held in a manner where they can be associated.  Authentication \
 \ information shall be separated from the voter’s decision at a \
 \ pre-defined stage in the e-election or e-referendum."

scenario
  "VERIFICATION_FAILURE_PROCEDURES" -- @ref David Jefferson
description
  "There must be clear technical and legal procedures for how to \
 \ proceed in the event that voters can prove that their votes were \
 \ not received accurately or counted, or if the official election \
 \ verification application does not verify that the Internet part of \
 \ the election was correct."

end

scenario_chart SYSTEM_OPERATIONAL_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "General system operational requirements for digital election \
 \ systems."

-- @ref Rec(2004)11 Systems Operation
```

```
scenario "PUBLIC_SYSTEM_MANIFEST" -- @ref derived from Rec(2004)11 Appendix III, C.
    69.
description
  "The electoral authority shall publish an official manifest of the \
 \ software used in an e-election or e-referendum.  It may exclude \
 \ from the public manifest data protection software for security \
 \ reasons. At the very least the manifest shall indicate the software \
 \ used, the versions, its date of installation and a brief description. \
 \ A procedure shall be established for updating the manifest to reflect \
 \ changes to the installed software."

scenario "PRIVATE_SYSTEM_MANIFEST" -- @ref derived from Rec(2004)11 Appendix III, C.
    69.
description
  "The electoral authority shall maintain a manifest of all software, \
 \ including data protection software, used in the system. This manifest \
 \ shall contain at least the same information as the public manifest. \
 \ A procedure shall be established for updating the manifest to reflect \
 \ changes to the installed software."

scenario "MANIFEST_ACCURACY" -- @ref derived from Rec(2004)11 Appendix III, C. 69.
description
  "It shall be possible for the electoral authority to check the installed \
 \ software against the system manifests at any time."

scenario "SYSTEM_FAILOVER_INVARIANTS" -- @ref Rec(2004)11 Appendix III, C. 70.
description
  "Those responsible for operating the equipment shall draw up a \
 \ contingency procedure for system failures.  Any backup system shall \
 \ conform to the same standards and requirements as the original system."

scenario "DATA_BACKUP_INVARIANTS" -- @ref Rec(2004)11 Appendix III, C. 71.
description
  "Sufficient backup arrangements shall be in place and be permanently \
 \ available to ensure that voting proceeds smoothly.  The staff \
 \ concerned shall be ready to intervene rapidly according to a \
 \ procedure drawn up by the electoral authority."

scenario "SYSTEM_INVARIANTS_DURING_ELECTION" -- @ref Rec(2004)11 Appendix III, C. 72.
description
  "Those responsible for the equipment shall use special procedures to \
 \ ensure that during the polling period the voting equipment and its \
 \ use satisfy requirements.  The backup services shall be regularly \
 \ monitored."

scenario "PRE_ELECTION_CERTIFICATION_INVARIANTS" -- @ref Rec(2004)11 Appendix III, C.
    73.
description
  "Before each election or referendum, the equipment shall be checked \
 \ and approved in accordance with a protocol drawn up by the \
 \ electoral authority.  The equipment shall be checked to ensure that \
 \ it complies with technical specifications.  The findings shall be \
 \ submitted to the electoral authority."

scenario "FORMAL_CONTROL_PROCEDURE" -- @ref Rec(2004)11 Appendix III, C. 74.
description
  "All technical operations shall be subject to a formal control \
 \ procedure.  Any substantial changes to key equipment shall be \
 \ performed with advance notice."
```

```
scenario "PHYSICAL_SECURITY_OF_SYSTEMS_INVARIANTS" -- @ref Rec(2004)11 Appendix III, C
    . 75.
description
  "Key e-election or e-referendum equipment shall be located in a \
 \ secure area and that area shall, throughout the election or \
 \ referendum period, be guarded against interference of any sort and \
 \ from any person.  During the election or referendum period a \
 \ physical disaster recovery plan shall be in place.  Furthermore, any \
 \ data retained after the election or referendum period shall be \
 \ stored securely."

scenario "INCIDENT_RESPONSE_INVARIANTS" -- @ref Rec(2004)11 Appendix III, C. 76.
description
  "Where incidents that could threaten the integrity of the system \
 \ occur, those responsible for operating the equipment shall \
 \ immediately inform the electoral authority, which will \
 \ take the necessary steps to mitigate the effects of the \
 \ incident.  The level of incident that shall be reported shall be \
 \ specified in advance by the electoral authority."

scenario "OPERATIONAL_TRANSPARENCY" -- @ref Kiniry/Zimmerman
description
  "A report containing every manifest change, every data or system \
 \ invariant violation, every control procedure violation, and every \
 \ physical security violation shall be prepared and made public by \
 \ the electoral authority after every election."

end

scenario_chart RELIABILITY_REQUIREMENTS
indexing
  partof: "SYSTEM_OPERATIONAL_REQUIREMENTS"
explanation
  "General reliability requirements for any internet election system."

scenario
  "GENERAL_MTBF" -- @ref David Jefferson
description
  "The entire voting service (server side) must have a proven MTBF of \
 \ >168 hours (1 week) under peak expected voting loads the entire \
 \ time."

scenario
  "LIVE_ELECTION_MTBF" -- @ref David Jefferson
description
  "MTBF validation must be demonstrated in multiple tests of \
 \ actual mock elections."

scenario
  "MTBF_CONTRA_DDOS" -- @ref David Jefferson
description
  "MTBF requirements apply only during normal peak operation, not \
 \ during attacks (e.g., DDoS)."

scenario
  "SYSTEM_RECOVERY_TIME" -- @ref David Jefferson
description
  "If service goes down for any reason other than regional natural \
 \ disaster or malicious attack, service must be restored in no more \
 \ than 10 minutes."

scenario
  "UPTIME" -- @ref David Jefferson
description
  "The system must have three nines (99.9%) uptime."

scenario
  "FAILURE_VALIDATION" -- @ref David Jefferson
description
  "Uptime must be demonstrated by failures in actual mock election \
 \ situations, e.g. tested by sudden loss of power to any server."

scenario
  "MIRRORED_FAILOVER_SERVICE" -- @ref David Jefferson
description
  "The system must have a warm spare in a second data center that can take \
 \ over in case of major failure."

scenario
  "FAILOVER_STAFFING" -- @ref David Jefferson
description
  "The system must be staffed at all times to guarantee the 10 minute \
 \ recovery time."

scenario
  "OPERATION_UNDER_DDOS" -- @ref David Jefferson
description
  "In a federal election the voting system must remain available even \
 \ during a large distributed denial of service attack.  It must be \
 \ able to continue correct operation during a sustained DDoS attack \
 \ on any combination of server side IP addresses (whether at the \
 \ primary server data center or its ISP) at a total level of 100 Gb/s \
 \ with no more than 15s degradation of response time to voters during \
 \ the attack."

scenario
  "DDOS_REFRESH_PERIODICITY" -- @ref David Jefferson
description
  "The DDoS threshold (initially 100 Gb/s) should be evaluated every \
 \ election cycle to see if it has to be raised due to newer \
 \ DDoS attack technologies."

scenario
  "DDOS_ATTACK_VALIDATION" -- @ref David Jefferson
description
  "The ability to survive a DDoS attack must be actually demonstrated \
 \ in the actual network configuration to be used prior to each \
 \ federal election."

scenario
  "DDOS_LOCAL_ELECTION" -- @ref David Jefferson
description
  "Reduced DDoS defense requirements might be acceptable for \
 \ non-federal elections."

end
```

```
scenario_chart SECURITY_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS";
explanation
  "General security requirements for digital elections systems."

-- @ref Rec(2004)11 Security, I. General requirements
scenario "NO_DATA_LOSS" -- @ref Rec(2004)11 Appendix III, D. I. 77.
description
  "Technical and organizational measures shall be taken to ensure that \
 \ no data will be permanently lost in the event of a breakdown or a \
 \ fault affecting the e-voting system."

scenario "VOTER_PRIVACY_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. I. 78.
description
  "The e-voting system shall maintain the privacy of \
 \ individuals.  Confidentiality of voters' registers stored in or \
 \ communicated by the e-voting system shall be maintained."

scenario "SYSTEM_SELF_CHECKS" -- @ref Rec(2004)11 Appendix III, D. I. 79.
description
  "The e-voting system shall perform regular checks to ensure that its \
 \ components operate in accordance with its technical specifications \
 \ and that its services are available."

scenario "SYSTEM_ACCESS_CONTROL" -- @ref Rec(2004)11 Appendix III, D. I. 80.
description
  "The e-voting system shall restrict access to its services, \
 \ depending on the user identity or the user role, to those services \
 \ explicitly assigned to this user or role.  User authentication shall \
 \ be effective before any action can be carried out."

scenario "DATA_PROTECTION" -- @ref Rec(2004)11 Appendix III, D. I. 81.
description
  "The e-voting system shall protect authentication data so that \
 \ unauthorized entities cannot misuse, intercept, modify, or otherwise \
 \ gain knowledge of any of this data.  In uncontrolled \
 \ environments, authentication based on cryptographic mechanisms is \
 \ advisable."

scenario "UNIQUE_IDENTIFICATION" -- @ref Rec(2004)11 Appendix III, D. I. 82.
description
  "Identification of voters and candidates in a way that they can \
 \ unmistakably be distinguished from other persons (unique \
 \ identification) shall be ensured."

scenario "OBSERVATION_DATA" -- @ref Rec(2004)11 Appendix III, D. I. 83.
description
  "E-voting systems shall generate reliable and sufficiently detailed \
 \ observation data so that election observation can be carried \
 \ out.  The time at which an event generated observation data shall be \
 \ reliably determinable.  The authenticity, availability and \
 \ integrity of the data shall be maintained."

scenario "TIME_SYNCHRONIZATION" -- @ref Rec(2004)11 Appendix III, D. I. 84.
description
  "The e-voting system shall maintain reliable synchronized time \
 \ sources.  The accuracy of the time sources shall be sufficient to \
 \ maintain time marks for audit trails and observations data, as well \
 \ as for maintaining the time limits for registration, nomination, \
 \ voting, or counting."

scenario "SECURITY_COMPLIANCE_RESPONSIBILITY" -- @ref Rec(2004)11 Appendix III, D. I.
    85.
description
  "The electoral authority has overall responsibility for compliance \
 \ with these security requirements, and such compliance shall be assessed by \
 \ independent bodies."

-- @ref Rec(2004)11 Security, II. Requirements in pre-voting stages
scenario "LISTS_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. II. 86.
description
  "The authenticity, availability and integrity of the voters' \
 \ registers and lists of candidates shall be maintained.  The source of \
 \ the data shall be authenticated.  Provisions on data protection shall \
 \ be respected."

scenario "CANDIDATE_PROCESS_TIME_PROVENANCE" -- @ref Rec(2004)11 Appendix III, D. II.
    87.
description
  "The fact that candidate nomination and, if required, the decision \
 \ of the candidate and/or the electoral authority to accept a \
 \ nomination has happened within the prescribed time limits shall be \
 \ ascertainable."

scenario "VOTER_PROCESS_TIME_PROVENANCE" -- @ref Rec(2004)11 Appendix III, D. II. 88.
description
  "The fact that voter registration has happened within the prescribed \
 \ time limits shall be ascertainable."

-- @ref Rec(2004)11 Security, III. Requirements in the voting stage
scenario "ELECTION_DATA_INTEGRITY_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. III
    . 89.
description
  "The integrity of data communicated from the pre-voting stage \
 \ (e.g., voters' registers and lists of candidates) shall be \
 \ maintained.  Data-origin authentication shall be carried out."

scenario "BALLOT_AUTHENTICITY_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. III.
    90.
description
  "It shall be ensured that the e-voting system presents an authentic \
 \ ballot to the voter.  In the case of remote e-voting, the voter shall \
 \ be informed about the means to verify that a connection to the \
 \ official server has been established and that the authentic ballot \
 \ has been presented."

scenario "CAST_VOTE_TIME_PROVENANCE" -- @ref Rec(2004)11 Appendix III, D. III. 91.
description
  "The fact that a vote has been cast within the prescribed time \
 \ limits shall be ascertainable."

scenario "CONTROLLED_SYSTEMS_AND_VOTE_INTEGRITY" -- @design derived from Rec(2004)11
    Appendix III, D. III. 92.
description
  "Election equipment under the control of the electoral authority \
 \ shall be protected against influence that could modify the vote."
```

```
scenario "UNCONTROLLED_SYSTEMS_AND_VOTE_INTEGRITY" -- @ref Kiniry/Zimmerman
description
  "The integrity of the vote must not depend on the security of election \
 \ equipment not under the control of the electoral authority."

scenario "NO_BREADCRUMBS" -- @ref Rec(2004)11 Appendix III, D. III. 93.
description
  "Residual information holding the voter’s decision or the display of \
 \ the voter’s choice shall be destroyed after the vote has been \
 \ cast.  In the case of remote e-voting, the voter shall be provided \
 \ with information on how to delete, where that is possible, traces \
 \ of the vote from the device used to cast the vote."

scenario "ELIGIBILITY_IMPLIES_VOTE_VOTER_INVARIANTS" -- @ref Rec(2004)11 Appendix III,
     D. III. 94.
description
  "The e-voting system shall at first ensure that a user who tries to \
 \ vote is eligible to vote.  The e-voting system shall authenticate \
 \ the voter and shall ensure that only the appropriate number of votes \
 \ per voter is cast and stored in the electronic ballot box."

scenario "VOTE_CHOICE_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. III. 95.
description
  "The e-voting system shall ensure that the voter’s choice is \
 \ accurately represented in the vote and that the sealed vote enters \
 \ the electronic ballot box."

scenario "END_OF_VOTE_PERIOD_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. III. 96.
description
  "After the end of the e-voting period, no voter shall be allowed to \
 \ gain access to the e-voting system.  However, the acceptance of \
 \ electronic votes into the electronic ballot box shall remain open \
 \ for a sufficient period of time to allow for any delays in the \
 \ passing of messages over the e-voting channel."

-- @ref Rec(2004)11 Security, IV. Requirements in post-voting stages
scenario "DATA_COMMUNICATION_INTEGRITY_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D
    . IV. 97.
description
  "The integrity of data communicated during the voting stage \
 \ (e.g. votes, voters’ registers, lists of candidates) shall be \
 \ maintained.  Data-origin authentication shall be carried out."

scenario "TALLY_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. IV. 98.
description
  "The counting process shall accurately count the votes.  The counting \
 \ of votes shall be reproducible."

scenario "BALLOT_BOX_AND_TALLY_INVARIANTS" -- @ref Rec(2004)11 Appendix III, D. IV.
    99.
description
  "The e-voting system shall maintain the availability and integrity \
 \ of the electronic ballot box and the output of the counting process \
 \ as long as required."

scenario "ADVERSARY_RESOURCES" -- @ref Kiniry/Zimmerman
description
  "The e-voting system shall be designed and tested with the assumption \
 \ that an adversary has a budget of $10 per voter per election, which they \
 \ can apply toward any critical subset of votes/voters of their choosing."
end

scenario_chart E2EVIV_SECURITY_REQUIREMENTS
indexing
  partof: "SECURITY_REQUIREMENTS";
  author: "David Jefferson <d_jefferson@yahoo.com>";
  created: "22 June 2014";
  reviewer: "Joe Kiniry <kiniry@galois.com>";
  reviewed: "16 July 2014"
explanation
  "General security requirements for end-to-end verifiable internet \
 \ election systems."

 -- These are requirements for embedding an E2EV system in an Internet
 -- voting environment.  They are over and above the requirements for
 -- the core E2EV itself.  We do not consider usability or accessibility
 -- requirements here.  Some of these requirements will make
 -- accessibility and usability more difficult to achieve.  Still, these
 -- are requirements, and if they cannot be met, or cannot be met
 -- simultaneously with usability and accessibility requirement, then we
 -- have to recommend not implementing an E2EV Internet voting system.

scenario
  "NATIONAL_SECURITY" -- @ref David Jefferson
description
  "If used in federal elections, an Internet voting system is also a \
 \ national security system, and thus must be subject to the highest \
 \ security requirements."

scenario
  "FEDERAL_REQUIREMENTS" -- @ref David Jefferson
description
  "Any Internet voting system used in a public primary or general \
 \ election in the U.S. for federal or state legislative, executive, \
 \ or judicial office, or recall election, or statewide initiative or \
 \ referendum, must meet all of the requirements in this document."

scenario
  "LOCAL_REQUIREMENTS" -- @ref David Jefferson
description
  "Reduced security requirements might be appropriate for county, \
 \ municipal, or other kinds of elections"

scenario
  "AUTOMATED_REGISTRATION_FRAUD" -- @ref David Jefferson
description
  "Automated registration fraud must not be possible."
 -- Eligibility & Registration (online registration, automated
 -- registration fraud, and change of credentials): DJ doesn’t know yet
 -- what to write here regarding requirements.  But obviously any
 -- automated registration fraud can be used to affect the outcome of
 -- elections.

scenario
  "CLIENT_SIDE_AUTHENTICITY" -- @ref David Jefferson
description
  "There must be a means by which any third party can determine if the \
 \ client-side software is genuine."
 -- Authentication of service: Not sure what requirement should be
 -- here.  The intent is to somehow ascertain that the E2EV software
 -- on the client-side is genuine.  Presumably that E2EV software will
 -- authenticate the remote server.

scenario
  "AUTHENTICATION_INDEPENDENCE" -- @ref David Jefferson
description
  "The security of authentication must not be affected by \
 \ any potential breach of any public or commercial databases."

scenario
  "ZERO_KNOWLEDGE_AUTHENTICATION" -- @ref David Jefferson
description
  "It should not be possible for an attacker to impersonate voters \
 \ even if the entire server database used for authentication is \
 \ compromised."

scenario
  "AUTHENTICATION_CREDENTIAL_REESTABLISHMENT" -- @ref David Jefferson
description
  "In some cases of security breach it must be possible to require all \
 \ voters in a jurisdiction to re-establish credentials."
end

scenario_chart PRIVACY_REQUIREMENTS
indexing
  partof: "SECURITY_REQUIREMENTS"
explanation
  "General privacy requirements for end-to-end verifiable internet \
 \ election systems."
-- violations of vote privacy are not generally detectable
-- violations of vote privacy are irreversible
-- violations of vote privacy enable vote coercion and vote selling
-- vote privacy cannot be verified by testing; it can only be ascertained by expert
    analysis of architecture and code

scenario
  "E2E_VOTE_PRIVACY" -- @ref David Jefferson
description
  "Vote privacy must be preserved end-to-end insofar as mathematically \
 \ possible."

scenario
  "VOTE_PRIVACY_INVIOLATE" -- @ref David Jefferson
description
  "Vote privacy cannot be waived by voters."

scenario
  "MALWARE_PRESENCE" -- @ref David Jefferson
description
  "Vote privacy must not be violated even in the presence of arbitrary \
 \ malicious code on the client platform, including phony client \
 \ software, malicious client wrappers, MITM code between the user and \
 \ the E2EV interface, malicious browser plugins or scripts, \
 \ keyloggers, etc."
 -- This requirement will seriously complicate the user interface an
 -- usability of the system, but is absolutely essential.
```

```
scenario
  "REMOTE_MONITORING" -- @ref David Jefferson
description
  "Voting should not be permitted from client platforms known to have \
 \ remote monitoring software installed that could be used to monitor \
 \ or log voting activity and that cannot be turned off by the voter. \
 \ (All mobile platforms had, and probably still do have, such remote \
 \ monitoring software.)"

scenario
  "CLIENT_SIDE_CHANNELS" -- @ref David Jefferson
description
  "The client software of the voting system must not send data to any \
 \ IP address except those associated with the vote server and the \
 \ basic infrastructure servers of the Internet."

scenario
  "SOCIAL_MEDIA_SIDE_CHANNELS" -- @ref David Jefferson
description
  "The client should not provide any information to third parties, \
 \ e.g., Facebook, Twitter, etc. regarding the act of voting."

scenario
  "NO_TRACKING" -- @ref David Jefferson
description
  "There must be no tracking devices or tracking logic in the vote \
 \ client."

scenario
  "NO_BREADCRUMBS_DETAILS" -- @ref David Jefferson
description
  "The client software must leave no files or other persistent data on \
 \ the platform regarding the vote transaction but for an optional \
 \ file containing information needed for subsequent verification that \
 \ the voter’s ballot is included in the election canvass: no cookies \
 \ or other session files, no temporary files."

scenario
  "TRANSIENT_DATA_CLEANUP" -- @ref David Jefferson
description
  "The client software should explicitly erase (i.e., overwrite) all \
 \ transient copies of vote-transaction data, e.g. data in registers, \
 \ caches, RAM, and virtual memory."

scenario
  "FORENSICALLY_SECURE" -- @ref David Jefferson
description
  "It should not be possible even for client-side forensic tools to \
 \ retrieve any information regarding the voting transaction after the \
 \ voting session is ended."

scenario
  "REMOTE_ADMINISTRATION_FORBIDDEN" -- @ref David Jefferson
description
  "The voting system should not support platforms that have remote \
 \ administration or remote control tools installed that cannot be \
 \ turned off by the voter."

scenario
```

```
  "INVULNERABLE_TO_ELECTION_MALWARE" -- @ref David Jefferson
description
  "The voting system must not be vulnerable to malware designed to \
 \ modify votes before they are input to the E2EV system."
 -- This will seriously complicate the human interface and usability
 -- of the voting system, but is absolutely essential.  Malware can be
 -- in many forms: completely phony or "alternative" client app,
 -- client wrapper, client-side MITM, browser plugin, client APT, etc.

scenario
  "CLIENT_SYSTEM_AUTHENTICATION" -- @ref David Jefferson
description
  "The voting system server must authenticate that it is communicating \
 \ with a genuine vote client during a voting session."
 -- This will complicate, but not eliminate, the possibility of
 -- client-side malware.  @see CLIENT_SIDE_AUTHENTICITY.

scenario
  "PENETRATION_ATTACKS" -- @ref David Jefferson
description
  "Deny penetration attacks. (DJ doesn’t know what to write about \
 \ this.)"

scenario
  "APT_ATTACKS" -- @ref David Jefferson
description
  "Deny advanced persistent threat attacks.  (DJ doesn’t know what to \
 \ write about this.)"

scenario
  "INSIDER_ATTACKS" -- @ref David Jefferson
description
  "Something about insider attacks being impossible.  (DJ doesn’t know \
 \ what to write about this.)"

scenario
  "COERCION_PREVENTION" -- @ref David Jefferson
description
  "There must be no way for voters to prove to another party any \
 \ information regarding how they voted in any race beyond what is \
 \ mathematically deducible from the final distribution of votes."
  -- @see RECEIPT_FREEDOM

scenario
  "SOFTWARE_INDEPENDENCE" -- @ref Ron Rivest
description
  "The system must witness software independence: the tabulation \
 \ record must not rely solely on software."

scenario
  "DIGITAL_EVIDENCE_NOT_A_RECEIPT"
description
  "Digital evidence (e.g., photographing a ballot or video recording \
 \ the casting process) of the voting process must not violate receipt \
 \ freedom."
end

scenario_chart CERTIFICATION_AND_RECERTIFICATION_REQUIREMENTS
indexing
```

```
  partof: "SECURITY_REQUIREMENTS"
explanation
  "General security requirements relating to certification of digital \
 \ elections systems."
end

scenario_chart USABILITY_REQUIREMENTS
indexing
  partof: "TECHNICAL_REQUIREMENTS"
explanation
  "General usability requirements of digital elections systems."

scenario
  "MANDATORY_USABILITY_TESTING" -- @ref Kiniry/Zimmerman
description
  "Usability testing for disabled and abled voters shall be performed, \
 \ and the reports of the testing made public. The system must achieve \
 \ satisfactory usability testing results before being used in a \
 \ binding election."

scenario
  "VOTE_CONFIRMATION" -- @ref David Jefferson
description
  "If a voter receives the final ‘Thank you for voting’ confirmation, \
 \ then she/he can be certain the ballot was recorded."

scenario
  "UNCERTAIN_VOTER_REVOTE" -- @ref David Jefferson
description
  "If the voter is uncertain about the state of their ballot, he/she \
 \ is free to attempt to vote again."

end
```

Bibliography

[1] Ben Adida. Helios: Web-based Open-Audit Voting. In *USENIX Security*, 2008.

[2] Ben Adida, Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a University President using Open-Audit Voting: Analysis of real-world use of Helios. In *USENIX EVT/WOTE*, 2009.

[3] David Bismark, James Heather, Roger M. A. Peel, Steve Schneider, Zhe Xia, and Peter Y. A. Ryan. Experiences Gained from the first Prêt à Voter Implementation. In *First International Workshop on Requirements Engineering for e-Voting Systems (RE-VOTE)*, pages 19–28. IEEE, Aug 2009.

[4] Philippe Bulens, Damien Giry, and Olivier Pereira. Running mixnet-based elections with Helios. In *USENIX EVT/WOTE*, 2011.

[5] Craig Burton, Chris Culnane, James Heather, Thea Peacock, Peter Y. A. Ryan, Steve Schneider, Sriramkrishnan Srinivasan, Vanessa Teague, Roland Wen, and Zhe Xia. Using Prêt à Voter in Victorian State elections. In *USENIX EVT/WOTE*, 2012.

[6] Richard Carback, David Chaum, Jeremy Clark, John Conway, Aleksander Essex, Paul S. Herrnson, Travis Mayberry, Stefan Popoveniuc, Ronald L. Rivest, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. Scantegrity II Municipal Election at Takoma Park: The First E2E Binding Governmental Election with Ballot Privacy. In *USENIX Security*, 2010.

[7] Carter Center. Internet Voting Pilot: Norway's 2013 Parliamentary Elections. http://www.cartercenter.org/resources/pdfs/peace/democracy/Carter-Center-Norway-2013-study-mission-report2.pdf, Mar 2014.

[8] David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, and Alan T. Sherman. Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes. In *USENIX EVT*, 2008.

[9] David Chaum, Richard T. Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. Scantegrity II: End-to-End Verifiability by Voters of Optical Scan Elections Through Confirmation Codes. *IEEE Transactions on Information Forensics and Security*, 4(4):611–627, Dec 2009.

[10] David Chaum, Ben Hosp, Stefan Popoveniuc, and Poorvi L Vora. Accessible voter-verifiability. *Cryptologia*, 33(3):283–291, 2009.

[11] David Chaum, Peter Y. A. Ryan, and Steve Schneider. A Practical Voter-Verifiable Election Scheme. In Sabrinade Capitani di Vimercati, Paul Syverson, and Dieter Gollmann, editors, *Computer Security – ESORICS 2005*, volume 3679 of *Lecture Notes in Computer Science*, pages 118–139. Springer Berlin Heidelberg, 2005.

[12] Alex Delis, Konstadina Gavatha, Aggelos Kiayias, Charalampos Koutalakis, Georgios Sotirellis, Elias Nikolakopoulos, Lampros Paschos, Mema Roussopoulou, Pavlos Vasilopoulos, Thomas Zacharias, and Bingsheng Zhang. Pressing the Button for European Elections 2014: Public attitudes towards Verifiable E-Voting In Greece. https://drive.google.com/file/d/0B-mtbRwyPn_SdnpMRzBKcEZWUm8/view?usp=sharing, June 2014.

[13] Aleks Essex, Jeremy Clark, Richard Carback, and Stefan Popoveniuc. Punchscan in practice: an E2E election case study. In *Proceedings of Workshop on Trustworthy Elections*, 2007.

[14] Kristian Gjøsteen. The Norwegian Internet Voting Protocol. In Aggelos Kiayias and Helger Lipmaa, editors, *E-Voting and Identity*, volume 7187 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2012.

[15] Rop Gonggrijp, Willem-Jan Hengeveld, Eelco Hotting, Sebastian Schmidt, and Frederik Weidemann. RIES - Rijnland Internet Election System: A Cursory Study of Published Source Code. In PeterY.A. Ryan and Berry Schoenmakers, editors, *E-Voting and Identity*, volume 5767 of *Lecture Notes in Computer Science*, pages 157–171. Springer Berlin Heidelberg, 2009.

[16] How to Vote: Wombat Voting System. http://www.wombat-voting.com/how-to-vote.

[17] Engelbert Hubbers, Bart Jacobs, and Wolter Pieters. RIES - Internet Voting in Action. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International*, volume 1, pages 417–424. IEEE, 2005.

[18] Engelbert Hubbers, Bart Jacobs, Berry Schoenmakers, Henk van Tilborg, and Benne de Weger. Description and analysis of the RIES internet voting system. In *Report of the Eidhoven Institute for the Protection of Systems and Information*. Faculty of Mathematics and Computer Science Eindhoven University of Technology, June 2008.

[19] OSCE/ODIHR Election Assessment Mission Report. http://www.osce.org/odihr/elections/netherlands/24322?download=true, Nov 2006.

[20] Stefan Popoveniuc and Ben Hosp. An introduction to Punchscan. In *IAVoSS Workshop On Trustworthy Elections (WOTE 2006)*, pages 28–30. Robinson College United Kingdom, 2006.

[21] Stefan Popoveniuc and Ben Hosp. An Introduction to PunchScan. In David Chaum, Markus Jakobsson, RonaldL. Rivest, PeterY.A. Ryan, Josh Benaloh, Miroslaw Kutylowski, and Ben Adida, editors, *Towards Trustworthy Elections*, volume 6000 of *Lecture Notes in Computer Science*, pages 242–259. Springer Berlin Heidelberg, 2010.

[22] Noel Runyan. Improving access to voting: A report on the technology for accessible voting systems. *Retrieved October*, 1:2008, 2007.

[23] P.Y.A. Ryan, D. Bismark, J. Heather, S. Schneider, and Zhe Xia. Prêt à voter: a voter-verifiable voting system. *Information Forensics and Security, IEEE Transactions on*, 4(4):662–673, Dec 2009.

[24] Security Review: Helios Online Voting. https://cubist.cs.washington.edu/Security/2009/03/13/security-review-helios-online-voting/, Mar 2009.

[25] Georgios Tsoukalas, Kostas Papadimitriou, Panos Louridas, and Panayiotis Tsanakas. From Helios to Zeus. In *USENIX EVT/WOTE*, 2013.

[26] Filip Zagórski, Richard T. Carback, David Chaum, Jeremy Clark, Aleksander Essex, and Poorvi L. Vora. Remotegrity: Design and Use of an End-to-End Verifiable Remote Voting System. In Michael Jacobson, Michael Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *Applied Cryptography and Network Security*, volume 7954 of *Lecture Notes in Computer Science*, pages 441–457. Springer Berlin Heidelberg, 2013.