Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Bundeskanzlei BK

Sektion Politische Rechte

Summary of the expert dialog

Redesign of Internet Voting Trials in Switzerland 2020

19th November 2020

Contents

Annex

- Summary of the replies to the questionnaire

1. Purpose

This document serves as a foundation for the task-force of the Confederation and the cantons to agree on recommendations for the future trials and later for implementing the recommendations. To that end, it summarizes the written discussions of the expert dialog that has been conducted in conjunction with the redesign of the internet voting trials in Switzerland.

2. Background

Under the Federal Council's mandate of 26 June 2019[1], the Federal Chancellery FCh in collaboration with various cantons conducted an expert dialog. The purpose of the dialogue was to provide the task-force of the Confederation and the cantons with a basis for drafting recommendations. Task-force members from the FCh and the cantons discussed various issues concerning internet voting with 23 experts[2] from the academic research community and the industrial sector, with the dialogue being focused on technical matters. Most of the experts were from a field of exact science, three of them had a background in social science.

To start, the FCh sent out a questionnaire[3] to the experts on 14 February 2020 containing around 60 questions. The FCh then conducted a moderated online discussion in writing using GitLab from 5 May to 17 July based on the responses to the questionnaire. The moderator provided the experts who participated with a summary showing the conclusions of the discussions on GitLab. The summaries of the discussions on GitLab and the summaries of the answers to the related questions of the questionnaire are contained in the present document. Find the summaries to all questions of the questionnaire in the document "Summary of the replies to the questionnaire" (annex 1).

3. Introductory Remarks

The document is organized thematically. Each of the sections 4 – 15 summarize the conclusions of one of the 12 discussion blocks.

Each section is divided into four sub-sections. The first sub-section gives a brief overview of how we proceeded to discuss the issues related to the discussion block. The second one summarizes responses to the related questions of the questionnaire (we received 15 written responses to the questionnaire by the experts or groups of experts, whereas not all responses contained answers to all questions). The third one reflects how the discussion blocks were introduced on the dialog platform. Finally, the questions and the summaries of the discussions are presented in the fourth sub-section. The summaries have been validated by the experts. Apart from corrected typos and unified spelling of terms, the introductions to the discussion-blocks and the summaries of the discussions are presented here without any changes.

To get a more complete picture, you might find additional information in the original questionnaire document as well as the document "Summary of the replies to the questionnaire", which was given to the experts as a means to prepare for the dialog on the platform.

Section 16 contains a recommendation of the experts that is not directly related to internet voting but to the use of technology affecting the conventional voting channels (in-person voting and voting by mail).

4. Block 1 - Effectiveness of Cryptography

4.1 Overview

Block one was discussed under the label "effectiveness of cryptography". It relates to questions 2.1.1 and 2.1.2 of the questionnaire. As there seemed to be consensus on most parts of these two questions, in block one we explicitly asked whether experts agree on the conclusions presented in the summary of

[1] www.bk.admin.ch > Political rights > E-Voting > Media releases

[2] www.bk.admin.ch > Political rights > E-Voting > Reports and studies

[3] www.bk.admin.ch > Political rights > E-Voting > Reports and studies

the questionnaire. We asked further questions in order to understand the answers in more detail and to get more specific guidance with regard to the redesign of the trials.

4.2 Summaries of the answers to the related questions of the questionnaire

4.2.1 **Question 2.1.1 of the questionnaire: "Crypto-Protocol Part 1"**

The effectiveness of the protocol depends on assumptions on the effectiveness of cryptographic building-blocks. Does it seem likely to you that building-blocks are flawed even if they comply with known standards?

Summary from 12 of 15 responses:

The answers do not contain any significant contradictions. Most experts would seem to relate to one of the two nuances: 10/12

- "Building-blocks are unlikely to be flawed, if they are used correctly"; or
- "Building-blocks are possibly flawed or likely to be flawed, given that they might not be used correctly."

Areas of concern related to wrong usage of building-blocks:

- Flawed implementation (this can involve errors in design-documents or code but also bad choices at instantiating from higher-level abstractions or when combining multiple building-blocks): 11/12
- Building-blocks particularly risk being flawed if they are not taken from widely accepted standards and if they are modified: 5/12
- Particularly zero-knowledge proofs (one expert also mentions mix-nets) for voting are complex and generally defined in research-papers and not taken up by widely accepted standards. Due to the generally high abstraction level, important considerations (e.g. setup, surrounding environment) are often not made explicit: 3/12
- Building blocks or their mode of operation might not address the real-world needs (trust assumptions, attacker capabilities) sufficiently unless carefully chosen: 4/12
- Regarding the real-world needs: Quantum computers or advances in cryptanalysis may at some point subvert the soundness of today's standard building blocks: 3/12

The following risk-limiting measures were proposed:

- Involve experts from cryptography at all relevant stages, particularly ensure vast scrutiny: 4/12
- Security proofs and their verification: 4/12
- Formal methods and automated proof-checking reduce the risk of a flawed proof and therefore of a flawed building-block: 3/12
- Closely relate choices to real-world needs, e.g. security proofs need to be checked against these needs: 3/12

How likely does it seem to you that such a flaw could be used for an undetected attack?

That depends on multiple factors. There were flaws observed in the code released in 2019 that would have allowed undetected attacks, given access to the relevant infrastructure: 2/12

4.2.2 **Question 2.1.2 of the questionnaire: "Crypto-Protocol Part 2"**

The effectiveness of the protocol also depends on whether it achieves the security goals in the defined trust model. Does it seem likely to you that the protocol and its security proof is flawed? How likely does it seem to you that such a flaw could be used for an undetected attack?

Summary from 10 of 15 responses:

The answers do not contain any significant contradictions. It can be taken from most answers that the complex and nonstandard nature of voting protocols underline or amplify the related conclusions from the question above (2.1.1 [here section 4.2.1]): 7/10

4.3 Introduction to Block 1 on the platform

Block 1 was presented as follows:

The first two questions of the questionnaire were about the effectiveness of cryptography (the specification of the full protocol as well as the underlying building-blocks). From the replies to the questionnaire, we conclude that the effectiveness of cryptography strongly hinges on its correct use.

Areas of concern related to wrong usage of cryptography:

1. Flawed implementation (this can involve errors in design-documents or code but also bad choices at instantiating from higher-level abstractions or when combining multiple building-blocks);
2. Building-blocks particularly risk being flawed if they are not taken from widely accepted standards and if they are modified;
3. Particularly zero-knowledge proofs and verifiable mix-nets for voting are complex and generally defined in research-papers and not taken up by widely accepted standards. Due to the generally high abstraction level, important considerations (e.g. setup, surrounding environment) are often not made explicit;
4. Building blocks or their mode of operation might not address the real-world needs (trust assumptions, attacker capabilities) sufficiently unless carefully chosen;
5. Regarding the real-world needs: Quantum computers or advances in cryptanalysis may at some point subvert the soundness of today's standard building blocks;

The following risk-limiting measures are effective:

6. Involve experts from cryptography at all relevant stages, particularly ensure vast scrutiny;
7. Security proofs and their verification;
8. Formal methods and automated proof-checking reduce the risk of a flawed proof and therefore of a flaw in the protocol;
9. Closely relate choices to real-world needs, e.g. security proofs need to be checked against these needs.

4.4 Questions and summaries of the discussions on the platform

4.4.1 Thesis validation

- *Do you agree with these conclusions? [The conclusions in section 4.3 of the present document]*
- *Is there anything important that would not fall under these points?*

The experts agree with the conclusions with some adjustments [underlined or crossed out below]:

The first two questions of the questionnaire were about the effectiveness of cryptography (the specification of the full protocol as well as the underlying building-blocks). From the replies to the questionnaire, we conclude that the effectiveness of cryptography strongly hinges on its correct use. Despite the challenges at getting it right, there is no way around cryptography when it comes to internet voting.

Areas of concern related to wrong usage of cryptography:

1. Flawed implementation (this can involve errors in design-documents or code but also bad choices at instantiating from higher-level abstractions or when combining multiple building-blocks);
2. Building-blocks particularly risk being flawed if they are not taken from widely accepted standards and if they are modified;
3. Particularly zero-knowledge proofs and verifiable mix-nets for voting are complex and generally defined in research-papers and not taken up by widely accepted standards. Due to the generally high abstraction level, important considerations (e.g. setup, surrounding environment) are often not made explicit;
4. Building blocks or their mode of operation might not address the real-world needs (trust assumptions, attacker capabilities) sufficiently unless carefully chosen;
5. Regarding the real-world needs: Quantum computers or advances in cryptanalysis may at some point subvert the soundness of today's standard building blocks;

The following risk-limiting measures are effective:

6. Involve experts from cryptography ~~at all relevant stages, particularly ensure vast scrutiny~~ at all stages that underlie the effectiveness of cryptography. Particularly ensure vast and continuous scrutiny. Furthermore, the latest advances in cryptanalysis must be taken into account;
7. Security proofs and their verification;
8. Formal methods and automated proof-checking reduce the risk of a flawed proof and therefore of a flaw in the protocol;
9. Closely relate choices to real-world needs, e.g. security proofs need to be checked against these needs.
10. Work toward rigorous standardization of building blocks of use both in internet voting systems and broader-interest communities (such as zero-knowledge proofs, verifiable mix-nets, and bulletin board protocols), and build on these standards once available.

4.4.2 **Number of Reviewers (related to thesis points (2), (3), (6), (7))**

If one independent cryptographer is mandated to verify the protocol specification and the proofs against the requirements, how likely is it that the protocol specification does not meet the requirements?

There are two things to be analyzed: (1) that the right statement is proven and (2) that the proof is correct. (1) is nontrivial both in the computational and the symbolic setting. (2) is nontrivial for the computational setting and trivial in the symbolic setting, at least when the proof is machine checked. Hence for both (1) and (2) in the computational setting, two cryptographers are better than one to avoid errors in the verification.

The fact, that (1) is nontrivial may be surprising, but it is a particular characteristic of cryptography.

Machine-checking proofs is helpful, but it comes with problems of its own and the tools existing today are not able to cover all areas of the cryptography used for an online voting system (Also see [section 4.4.5 of the present document]).

Assuming that the documentation is published, is there significant added value of appointing two independent cryptographers instead of one, and if so, should they work jointly or write separate reports?

Yes, there is significant value. They should work independently, but discussing the specifications or findings in a 2nd step or an iterative process would be beneficial. Two independent cryptographers is the minimum. Yet, additional reviewers would still be beneficial.

4.4.3 **Abstraction Level of Specification (related to thesis points (3), (6), (7))**

There is a protocol-specification at the top (the exposition which the security proofs relate to), the code at the bottom and specifications in between. Is it possible to express in words what maximum level of abstraction of the protocol specification seems admissible without introducing a significant risk that the lower level specification does not instantiate the protocol specification appropriately? (How specific should the security-proven protocol-specification be e.g. with regard to setup procedures, parameter generation?) Assume that the lower level specification is scrutinized by the same cryptographer as the protocol specification.

It is difficult to describe appropriate levels for different types of specifications.

The top specification of the protocol has to be a verifiable model that is abstract enough to allow reasoning about it.

The bottom specification has to be very detailed and include pseudo-code for every mathematical / cryptographic aspect in order to avoid any room for interpretation for the developer of the software as well as the people proving / checking it. Non-cryptographic aspects do not demand such a detailed specification.

There has to be a refinement relationship between the different levels of the specification down to the pseudo-code and eventually to the code level. This refinement relationship can only be established, when the step width is not too big. The ideal width is difficult to define and still a topic of research.

The choice of the implementation language plays an important role, since features like strong typing and flow control can be used in the low level specification. Language specific features and / or design patterns can make the review more difficult.

Setup procedures, parameter generation and everything that touches on the implementation of the cryptography has to be specified with no room for interpretation as well.

4.4.4 **Reviews of Operations (related to thesis points (1), (6))**

Is it relevant to appoint cryptographers at scrutinizing not just the specification and the code, but also the operational procedures, e.g. defined for printing, tallying and operating the control components and the verifier?

Yes, it is important to appoint cryptographers at scrutinizing the operational procedures wherever they are relevant to the cryptographic operations or the assumptions that the cryptographers make. It is also important that these expert have a deep understanding down to the physical layer.

4.4.5 **Formal Methods (related to thesis points (8))**

The VEleS requires a security proof based on symbolic methods. Would the formal methods and automated proof-checking you would think of meet that requirement? Would the requirement need to be more precise as to ensure that effective methods are implemented? What is the added value of the formal methods you might have thought of

- in terms of additionally reducing the risk that the protocol specification does not satisfy the requirements; and

- in terms of efforts / costs?

Checking proofs automatically brings very good assurance, as it is less prone to human errors. Symbolic proofs can and should be checked automatically, since this is well established. Computational proof used for online voting cannot yet be checked automatically with the software existing today. Therefore we still need manual checking.

Manual checking is also imperative since the two different kind of analyses - computational and symbolic - tend to find different attacks resulting from different flaws: The computational analysis is strong with respect to the cryptographic functionality, the automated symbolic analysis is more detailed with respect to the protocol's control flow and the possible ways that adversaries can interact with interleaved protocol runs.

Security proofs and checking the security proofs is an ongoing process. The proofs are established with regards to security definitions and models that are constantly being reviewed and updated themselves. As time passes, it is quite plausible a definition does not capture some important security aspect and that a different proof becomes necessary.

The efforts and costs for these proofs and the checks are very high, but there is no way around it.

4.4.6 **Breaking Cryptography (related to thesis points (4), (9))**

Assuming the real-world needs include protection from powerful organizations and state adversaries: how probable is it that such adversaries will rather find an attack based on the protocol specification (by finding a mistake or by breaking a cryptographic hardness-assumption) rather than the underlying specification, the code or the infrastructure? Assume reasonable key-lengths and no quantum computers. Relate your answer to the degree of scrutiny performed (e.g. number of reviewers). If you think it is anyway rather likely that attackers will find an attack based on the protocol specification, then why invest in cryptography at all?

Powerful organizations and state adversaries with sophisticated technical knowledge and resources have the realistic capability to find and secretly exploit the most subtle flaws in an electronic voting system successfully.

Well established and standardized cryptographic building blocks that have seen a lot of scrutiny are more likely to resist these attacks. It is therefore beneficial to use mature technologies, including cryptographic primitives that have been standardized, are widely deployed, and are supported by talent. Absent such maturity, it would be wise to support the creation of standardized cryptographic building blocks and well-maintained crypto libraries supporting internet voting.

Attackers are more likely to attack non-standardized aspects of the protocol, the implementation of the voting system or the infrastructure used to run the system. This is because these elements have seen less public scrutiny. The trust-assumptions of the system are also natural points for attacks, namely when they are too strong.

Another characteristic of the attackers in question here is their ability to keep weaknesses for themselves. It is therefore useful to incentify research that helps to reveal and fix weaknesses in the building blocks, but also the specification, the implementation and infrastructure used to run the system.

5. Block 2 - Diversity to support security and trust-building

5.1 Overview

The topic of block two was diversity to support security and trust-building: If one component is vulnerable, another component should compensate for the vulnerability. It relates to questions 2.1.4, 2.1.5 and 2.1.6 of the questionnaire.

On the discussion platform we tried to relate the subject more closely to the Swiss context (see introduction in section 5.3) in its current state and thereby to obtain more specific guidance with regard to the redesign of the trials.

5.2 Summaries of the answers to the related questions of the questionnaire

5.2.1 **Question 2.1.4 of the questionnaire: “Independence”**

The VEleS allows to assume that 1 out of 4 «control components» is trustworthy. The effectiveness of verifiability hinges on the degree to which these components are distinct. Using distinct components reduces the probability that all components share the same flaw that could be exploited to launch an unnoticed attack. Yet, the VEleS allows to use application-layer software from the same provider on each control-component. In practice, at the PIT 2019 the identical software from Scytl was run on all four *control components. How do you assess the added value and downsides of running software from different providers on the control components? Does the added security benefit offset the added complexity and the potential new attack vectors opened?*

Summary from 13 of 15 responses:

The answers show different opinions with regard to whether diversity justifies the added complexity:

It is important to run software from different providers in order to limit the impact of a flawed piece of software: 8/13

- It has also been mentioned that introducing independent software leads to scrutiny of the underlying system-specification, which increases the chances of detecting flaws: 2/8

There is another group that acknowledges the added value in running software from different providers, but gives priority to one properly implemented and reviewed software in order to avoid further complexity: 4/13

Using multiple systems multiply the attack-surface and should therefore be avoided: 1/13

5.2.2 **Question 2.1.5 of the questionnaire: “Independence”**

Similarly for «the auditors’ technical aid» that is used to verify the proofs underlying universal verifiability (sometimes called «verifier»): How do you assess the added value and downsides of running software on the auditors’ technical aid that was written by a different provider than the one of the voting system?

Summary from 13 of 15 responses:

Most experts refer to their replies to the question above or repeat their statements. The same arguments are basically valid here. One expert points out that the verification software is easier to write than the control components. Three experts come to a conclusion that differs from the one above. It is important to run software from different providers in order to limit the impact of a potentially flawed piece of software: 10/13

There is another group that acknowledges the added value in running software from different providers, but gives priority to one properly implemented and reviewed software in order to avoid further complexity: 3/13

Using multiple systems should be avoided: 0/13

5.2.3 **Question 2.1.6 of the questionnaire: “Independence”**

The VEleS requires operating systems and hardware to differ. As how relevant do you consider operating systems and hardware to differ? (In comparison to the application layer software?) Do you see any other machine components that could constitute a significant risk in case they do not differ across control-components / auditors’ technical aids? How do you assess independence created by separating duties at operating control components and auditors’ technical aids? How far could separation of duties go? What are the downsides?

Summary from 8 of 15 responses:

The answers show different opinions with regard to whether diversity justifies the added complexity related to using different operating system and hardware: 6/8

- It is important or advisable to use different operating systems and hardware in order to limit the impact of a flawed component, despite added complexity: 5/6
- There is added value in using different operating systems and hardware, but it can be difficult and does not need to be considered a priority: 1/6

The answers also show different opinions with regard to whether diversity justifies the added complexity related to using different components other than operating system and hardware: 4/8

- It is important or advisable to ensure diversification of components other than operating system and hardware in order to limit the impact of a flawed component: 3/4
- The possibilities of diversification of components other than operating system or hardware have their limits. It is conceivable to use shared components across the system to avoid the related difficulties: 1/4

Overall, the following components have been mentioned as possible subject to diversification:

- Compilers
- Java VM
- CPU
- GPU
- Standard libraries embedded in the operating systems
- Networking equipment
- System management controllers
- Peripherals, e.g. disks
- Not just different products, but also different vendors

Using different operating systems and hardware is less tricky than different application-layer software: 2/8

5.3 Introduction to Block 2 on the platform

Block 2 was presented as follows:

5.3.1 Introduction

Verifiability in Switzerland aims at detecting cases of large-scale fraud. Many computations cannot be verified by the voter but by the administrations or elected commissions. Their verification capabilities hinge on whether critical components they or their partners run are functioning correctly. Due to the trust model defined in Art. 5 in conjecture with chapters 4.1 and 4.3 of the annex, it must be possible to justifiably assume that at least one control component, at least one verifier and the printing office are functioning correctly. These critical components are introduced in the description below.

5.3.2 **Individual Verifiability**

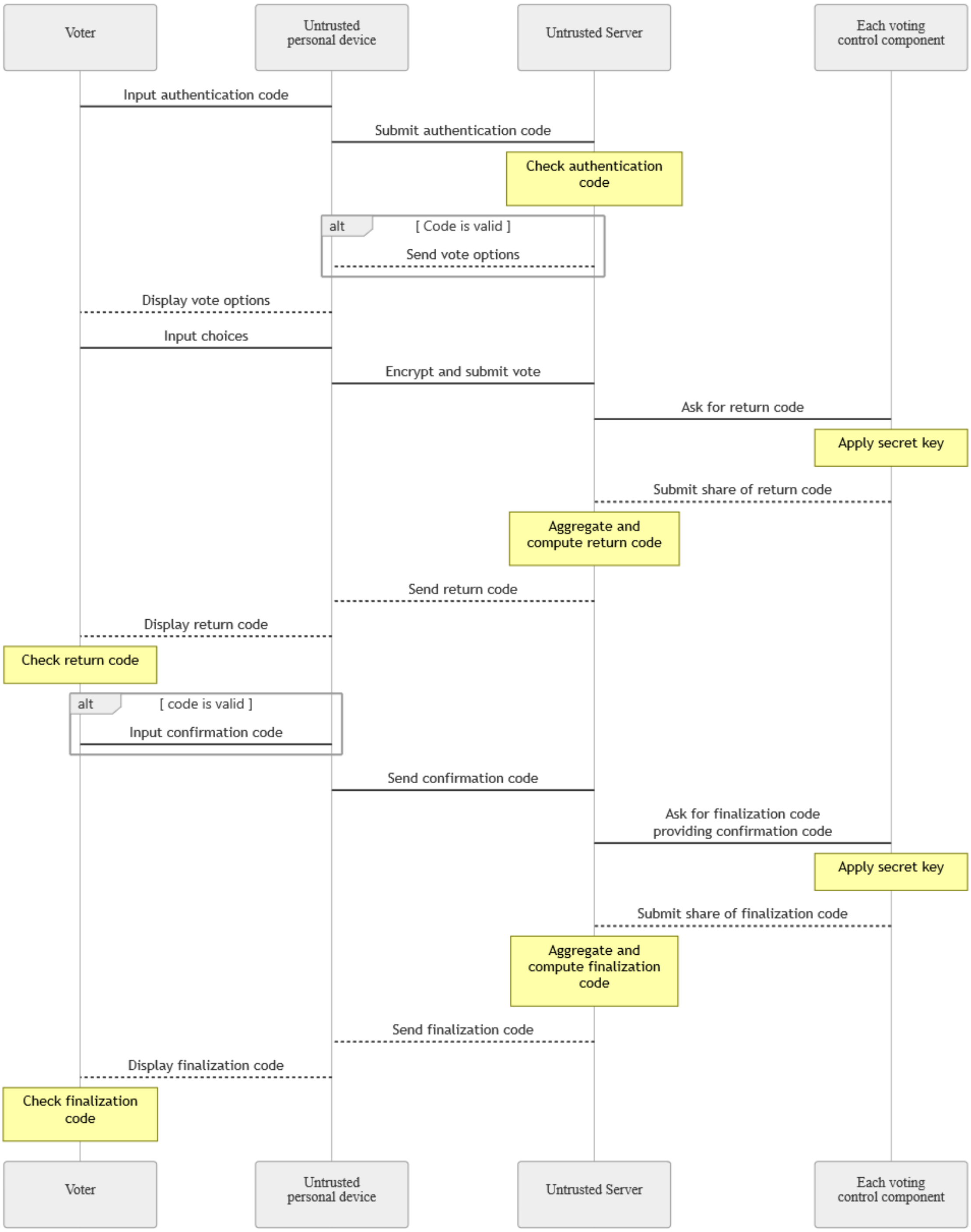
Voter
Untrusted personal device
Untrusted Server
Each voting control component
Input authentication code
Submit authentication code
Check authentication code
alt
[Code is valid]
Send vote options
Display vote options
Input choices
Encrypt and submit vote
Ask for return code
Apply secret key
Submit share of return code
Aggregate and compute return code
Send return code
Display return code
Check return code
alt
[code is valid]
Input confirmation code
Send confirmation code
Ask for finalization code providing confirmation code
Apply secret key
Submit share of finalization code
Aggregate and compute finalization code
Send finalization code
Display finalization code
Check finalization code
Voter
Untrusted personal device
Untrusted Server
Each voting control component

Figure 1

Figure 1 shows how “individual verifiability” is intended to work today. Any system compliant with the minimal requirements of the VEleS would work similarly. The voters open their voting material that allows them to vote at the polling station, by postal mail or through the internet. The voting material contains instructions for each voting channel. For internet voting, the voting card contains the codes that are needed to authenticate and to verify that votes have been cast-as-intended due to “individual verifiability” as defined in Art. 5 VEleS in conjecture with Art. 4 VEleS and Chapters 4.1, 4.2, 4.3 and 4.4 of the annex.

After entering their authentication code, the voters fill in their ballot by clicking on answers (yes/no/empty) in popular votes, or on lists and candidates in elections. After confirming, the vote is encrypted and sent to the voting server. Each of the four control components apply their distinct secret keys on the received data. They send back messages that allow the untrusted server to generate the return-codes related to the individual selections made by the voter. Due to the regulation, the content of the votes needs to remain secret, in particular the votes may not be decrypted. The return-codes are then displayed to the voters on their device. Voters are instructed to compare the displayed return-codes (distinct codes per answer and voter) with the ones received with the voting material and to report to the administration if the codes are not displayed correctly. The voters can then either change their mind and vote by postal mail or at the polling-station, or confirm the internet vote by entering their confirmation code. Similarly as above, the control components participate at generating the finalization code which is displayed to the voter for verification against the voting material. If the finalization code is not displayed correctly, voters can enter their confirmation code repeatedly until the finalization code is shown, or contact the administration.

Codes being displayed correctly holds the meaning that the vote will either be counted according the voter’s intension or, due to universal verifiability (introduced in section 4), the administration or an elected commission will detect fraud and start an investigation.

5.3.3 **Critical Components: Control-Components for individual verifiability**

One out of any of the four control components involved in generating the return-codes and the finalization code must function correctly, i.e. at least one needs to fulfill the relevant tasks as defined by the cryptographic protocol specification. No software errors or tampering may be suspected to cause a malfunction. If the assumption that at least one of the control components performs according to the protocol is not believed to hold, the return-codes do not have any meaning. The following list contains examples of the relevant functions underlying the effectiveness of individual verifiability:

- They must keep the keys they use to establish the return-codes secret. An attacker who knows these keys could manipulate votes and display to the voters the codes they expect to see.
- They may only apply their keys once per voter, otherwise attackers could learn all the codes, manipulate the votes and deceive the voters.
- They must make sure that the information received by the voters complies with the protocol. For example, if the return-codes are already pre-computed on the user platform, then the control components need to make sure that the pre-computation relates to the actual vote (this is done by checking a zero-knowledge proof delivered by the voting client). Otherwise, attackers could manipulate the vote but the precomputations would lead to the control components establishing the return-codes the voter expects to see.

5.3.4 **Universal verifiability**

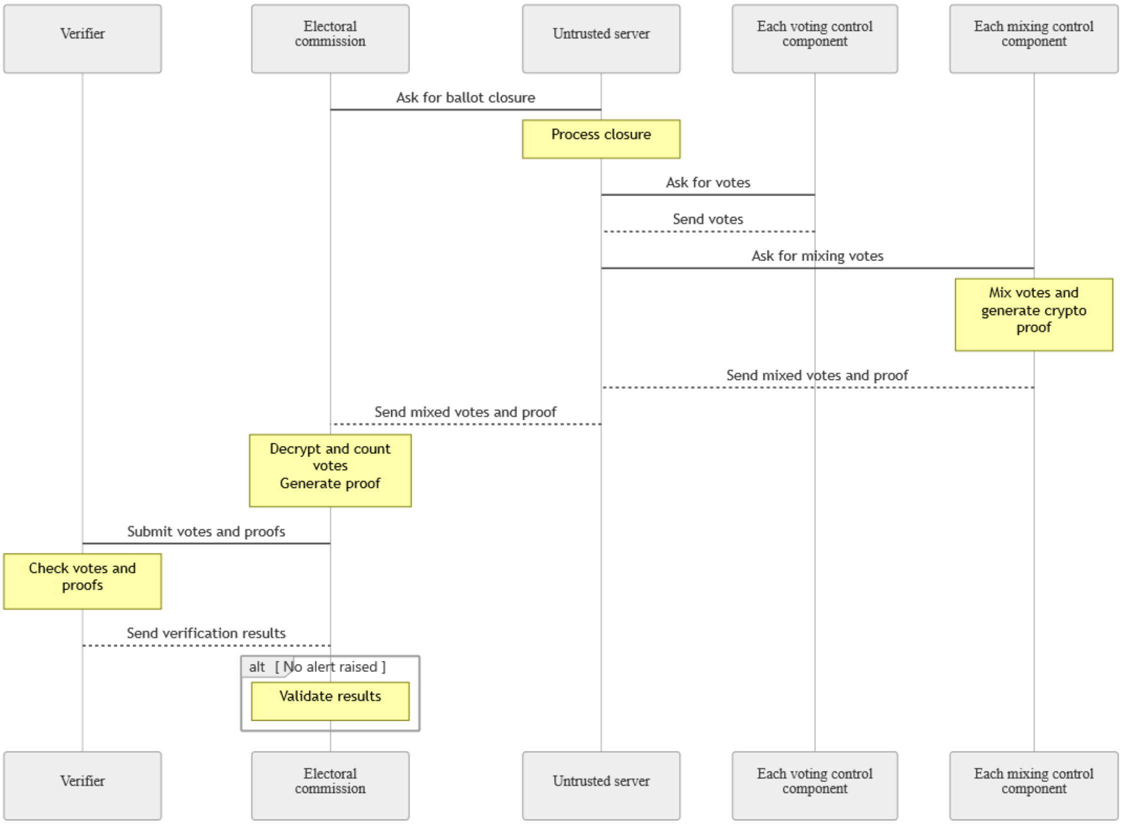
Verifier
Electoral commission
Untrusted server
Each voting control component
Each mixing control component
Ask for ballot closure
Process closure
Ask for votes
Send votes
Ask for mixing votes
Mix votes and generate crypto proof
Send mixed votes and proof
Send mixed votes and proof
Decrypt and count votes Generate proof
Submit votes and proofs
Check votes and proofs
Send verification results
alt [No alert raised]
Validate results
Verifier
Electoral commission
Untrusted server
Each voting control component
Each mixing control component

Figure 2

Figure 2 shows how “universal verifiability” is intended to work today (individual plus universal verifiability equals “complete verifiability”, as set out in the VEleS and the annex).

After the voting period is over, a set of at least four control- omponents sequentially mix all votes and change (re-randomize) their encryption. They provide a zero-knowledge proof that no votes have been changed despite changing their encryption. After mixing, the votes are decrypted and the correct decryption is proved using zero-knowledge proofs. The secrecy of the vote is granted, assuming that one of the control components will not reveal the permutation used at mixing.

Based on the data received from the control components active during the voting phase, the data received from the control components performing the mixing as well as the data received from the component used for the final decryption, the administration or an elected commission use the verifier to verify that all votes have been considered in the final tally correctly. To that end, the verifier checks the consistency of the lists of votes received from the control components (thus it verifies that the votes have been recorded-as-cast) and it verifies the proofs of correct mixing and decryption (thus it verifies that the votes have been tallied-as-recorded). Typically, the verifier is a device that would be run in the premises of a canton.

5.3.5 **Critical Components: Control Components for universal verifiability**

In order for the administration or an elected commission to detect manipulations, one out of any of the four control components must work correctly. The following list contains examples of functions or tasks the control components have to perform correctly, i.e. according to the cryptographic protocol:

- The control components that generate the return-codes must keep all votes or a digest until the end of the vote. If at least the correctly functioning control component keeps all votes, then the correctly functioning verifier will expose an inconsistency in the votes, in the case where other control components change or delete votes. However, if all control components delete the votes or change them in the same manner, this will not be detected by the verifier, i.e. there might be no detection based on well-analyzed cryptographic means.

- The control components that perform the mixing may not reveal the applied permutation. (This as well as the next point is to protect the secrecy of the vote, the verifier being able to detect manipulation does not hinge on these constraints.)
- Neither may they reveal the secret keys that they use for changing the encryption. (This is to protect the secrecy of the vote, the verifier being able to detect manipulation does not hinge on this.)

5.3.6 **Critical Components: Verifier for universal verifiability**

In order for the administration or an elected commission to detect manipulations, one verifier needs to work correctly. The following function has to be executed correctly, i.e. according to the cryptographic protocol:

- Verifiers need to raise an alarm if the lists of votes differ across the control components or if a proof of correct mixing or decryption does not hold.

5.3.7 **Critical Components: Printing Office for individual and universal verifiability**

The printing office is needed to produce the voting material, including the codes for authenticating and for verifying that their vote has been cast-as-intended ("individual verifiability"). The effectiveness of individual verifiability hinges on the return-codes, the confirmation code and the finalization code being secret and impossible to predict. In practice, the tasks of the Printing Office are divided. The generation of the codes and other parameters are performed in the cantonal premises. The physical printing is done by a printing company (see section 8.3).

The printing-office holds the codes in plain-text. At the latest when the codes are being printed, there can be no cryptographic means to protect these codes from being divulged. The codes have to be protected by organizational means.

The VEleS does not make a clear cut as to whether the printing office is allowed to generate parameters itself (including the codes) and whether it is sufficient to rely on organizational means to ensure the unpredictability of such parameters. In case the printing office is used to generate parameters, then the effectiveness of individual and possibly even universal verifiability additionally hinges on the quality of the printing office's random-number generator, its aggregating the inputs from the random-number generator correctly and printing the values accordingly.

In order for voters, the administration or an elected commission to detect manipulations the printing office needs to work correctly. The following list contains examples of functions or tasks the printing office has to perform correctly, i.e. according to the cryptographic protocol:

- Secret values, in particular the codes for verifiability and for authentication, must remain secret.
- Codes and if applicable other parameters must be generated correctly, i.e. according to the protocol specification.
- The generated values need to be further processed according to their generation, i.e. the printer must print these values correctly.

5.3.8 **Critical Components in practice**

The correct functioning of at least one control component and at least one verifier can be supported by:

- Scrutinizing the individual components
- Ensuring diversity among the components (different software and hardware)
- Increasing their number

- Keeping functionality simple and easy to analyze
- Appointing distinct groups of employees to operate each of the components
- Operating components in different premises

The correct functioning of the printing office can be supported by:

- Requiring the consent of at least 2 persons to obtain access that is further unnoticed
- Using offline machines when working with confidential information and destroying the data afterwards
- Scrutinizing the software and the processes

These points are addressed in the VEleS annex to a certain extent but not necessarily as a strict requirement.

In order to have something concrete to relate to, we summarize the features and modes of operation of the critical components as planned by the cantons when aiming for offering internet voting using system of Swiss Post. We point out that the features outlined here do not stand in conflict with the provisions of the VEleS and the annex in their current state.

5.3.8.1 Control components in practice

Control components as planned to be used during the voting phase:

- Implemented as physical servers (no virtualization)
- Segregated access rights
- Segregated networks
- Different operating systems and CPU
- Each to be operated in the premises of the provider
- Each to run the same application-level software from the voting provider
- Software to be scrutinized and published

Control components used for mixing the votes:

- The same as above holds for three of these control components
- The same holds for the fourth control component, except that it was meant to be operated offline in the premises of the canton

5.3.8.2 Verifier in practice

- One verifier was to be used
- Operated offline in the premises of the canton
- Due to the tight schedule, software from the same provider as the voting system was to be used. The software was to be scrutinized and published.

5.3.8.3 Printing office in practice

- The planned functionality attributed to the printing office can be distinguished as follows:
    - Parameter generation was to be done in the cantonal premises; software from the voting provider was to be used.
    - The actions of the actual printing office was reduced to decrypting the data received for printing, the printing itself and enveloping the voting cards. The cantons were to mandate their own printing service (the voting provider was not meant to print the voting cards).

- Some parameters underlying the effectiveness of verifiability were to be generated in a distributed way, but not all. For instance the return-codes as printed on the voting cards were to be selected based on random generator used by the component run in the cantonal premises.
- Parameter generation was not to be verified
- Parameter generation and printing was to be performed under the surveillance of at least two persons (four eyes principle)
- Private data was to be held on machines that were disconnected from any network (the computers and the printing machine were planned to be offline)

5.4 Questions and summaries of the discussions on the platform

5.4.1 Independent Software for Control Components used for return-code generation

In their answers to question 2.1.4 of the questionnaire, the invited experts acknowledge the added value in running software from independent providers. However opinions differ with regard to whether the risks emerging from the added complexity would overrule the added value with regard to verifiability. We would like to dig deeper in order to find out whether efforts should be put into introducing software from a different provider for at least one of the four control components used during the online phase (see section 3 and the first point in section 5).

Imagine Bob is an employee of a cantonal administration. It is his job to convince himself and others that internet voting is sufficiently secure. The control components used in the online phase are currently meant to run the same software. The software has been scrutinized to some degree and is currently being refactored. Auditing of the refactored software by independent engineers and cryptographers is planned and paid for, as well as publication of source code and documentation in order to allow for further scrutiny.

Bob really needs to get it right and Alice grants him additional funds. He thought about it and is now faced with a dilemma. One option would be to procure a new software for at least one of the control components and to have it scrutinized. Another option would be to invest in further examination or other risk-limiting measures of the existing software. He is not sure about the added complexity of the first idea. Although the software would be operated by the existing voting provider, still, the provider of the new software would constitute an additional actor and the new software would need to function well along with the existing one. On the other hand, diversity is a best practice in internet voting.

Would you advise Bob to procure the additional software or should he invest the money in other ways of limiting the risks related to the existing software?

Assume the underlying crypto-protocol is specified on a reasonable abstraction level and that a technical interface specification (data formats, timing constraints) is available.

Diversity in IT Security

It is a common understanding in the IT industry, that diversity leads to systems that are more resilient to attacks. The effect is very similar to a biological ecosystem, where monoculture is cheaper to maintain. But successful attacks in the form of fungus or any other form of pest is going to have a more devastating effect in a monoculture. It is true that investing more resources into the protection of the monoculture will have a positive effect. However, using the same amount of money for diversification, has a substantially better cost-benefit ratio when looking at the probability of a successful attack wiping out the whole ecosystem. And given the design of the internet voting system, an attacker has to wipe out all four control components and the verifier to manipulate a vote without being noticed. Therefore, the system is designed to get the biggest benefit possible out of diversity.

There has been a lot of research into this question for IT systems. The conclusion has been that diversification brings exponential cost-benefits while more hardening and more scrutiny will bring less and less security gain as investment increases. This follows a pattern of diminishing returns.

In 1984, Unix legend Ken Thompson received the Turing Award and held an acceptance speech that has proven to be a timeless classic. Published under the title of "Reflections on Trusting Trust", Thompson came to the conclusion, that "No amount of source-level verification or scrutiny will protect you from using untrusted code." Diversity is a relief for this problem, namely for Swiss internet voting. This is because it forces an attacker to exploit all four control components in parallel. This is (a) very expensive for the attacker and (b) could lead to the attack being detected, which is very risky for an attacker.

Hardening and Scrutiny Still Important

This does not mean, that hardening and scrutiny are futile. There still needs to be a high level of hardening and scrutiny, not the least because public trust will depend on these audits. Also formal verification methods bring a high level or reassurance to those parts of the system where they can be applied. This also contributes to the public trust. However, there comes a level of security, where diversification brings better benefits, namely in the long run, for unexpected attacks and for those parts of the system where formal verification cannot be applied.

Downsides of Diversity

Admittedly, diversity also comes with downsides. The costs of the independent implementation is the most obvious one. This includes the coding, the documentation, the integration, etc. At first sight it also includes a better specification, since independent developers will need to be able to write their code based on the specification alone. A single implementation can be developed based on a specification that has some ambiguity. But with a second, independent implementation, the specification is being validated by 3rd party developers, the ambiguities will surface and will have to be cleared out. This in turn leads to a better specification. So the additional costs that has to be put into the improvement of the specification (if any) is in fact an investment into the quality and the security of the system.

It would be useful to enforce conformance testing on the independent implementations. This would help in situations where shortcomings of the specifications were not noticed during the development.

Operation is also more expensive with multiple independent systems; namely when you also look at potential different behavior and how they ought to be reconciled as part of the operation. The latter however is but a small part of the operation, so we can still assume that the costs are more or less linear, while - as pointed out above - the benefits are exponential. So it is a worthwhile investment also in this regard.

There is also a bigger attack surface with multiple systems. But the more independent the control components are, the less can an attacker profit from one exploit when attacking the next one. The weaknesses are thus limited to individual control components and they no longer harm the security of the overall system. Of course, operation has to guarantee proper separation of the control components, so that one system cannot be used as a bridgehead to attack the next one.

Diversity as Core Requirement

All combined, the benefits of the diversity outweigh the costs by far.

So diversity is a core requirement and ideally every control component would be completely independent in every possible way. Several experts made it clear that they require a high degree of diversity to start to consider the system as trustworthy. Unfortunately, implementing four entirely separate control components is not feasible in real world. But there have to be rules to establish a minimum of diversity as a hard requirement.

Priorities

Even highly skilled attackers will not attack all levels of the system with equal ease. It seems likely that the application software of control components would be attacked first, since it is a custom implementation of software with limited public scrutiny. Afterwards, more widely used components, 3rd party components, would be attacked. The experts expect a powerful attacker to possess the means to exploit widely used software components (libraries, the Java virtual machine, operation systems and firmware, but also compiler etc.) and possibly also hardware components (CPU, mainboard, controllers etc.) However, exploits for widely used software and hardware components are very expensive to obtain and using them in an attack where they could be discovered makes it unattractive to use them. So, as a potential consequence, the OS and other 3rd party components would only be attacked if it brings an advantage over attacking all four control components together.

As a consequence, diversity for the control component application software must have the highest priority. This is also where the costs are the highest since it demands a separate implementation.

With the libraries, OS, firm- and hardware components, the situation is different, since these components do not have to be developed. Here the costs boil down to operation to a very wide extent. Therefore, the costs of the diversity are expected to be much lower in this area, especially for a system provider that has operation procedures in place for more than one operation system and for more than one type of server hardware already. Diversity is thus a cost-effective means of defense beyond the control component application software.

Components that Deserve Special Treatment

Two or three items are special and they deserve separate coverage: It should be required to use a different programming language and a different compiler for different control component implementations. This will bring different programming libraries and thus a welcome diversity. But there are shared libraries that are used by different programming languages. Their position is so dominant, they are very hard to replace with an independent implementation. E.g. OpenSSL, GMP etc. Special care has to be taken to guarantee diversity on this level.

Modern CPU level exploits like Spectre / Meltdown etc. can be avoided by working with dedicated hardware for the control components that are installed directly on the physical server. However, the hardware servers will still have an onboard management interface that serves as a backdoor into the system for setup and maintenance of the servers. Special care needs to be taken to close this access. But experience shows that you cannot be really sure you closed every backdoor in this regard. So beyond hardening, there also has to be diversity of the hardware, including CPU, mainboard, various controllers with their firmware etc. in order to mitigate the threat that comes from these management interfaces or supply chain attacks.

The Verifier

With all this being said, it goes without saying that diversity should also stretch to the verifier. There are four control components and there has to be a strong requirement to have multiple and diverse verification software to raise the cost of a successful attack.

In short

A high degree of diversity is a core requirement for a trustworthy internet voting system. The priority should be with the application software of the control components since an attacker is expected to attack this software first. But since all the 3rd party components (Shared libraries, OS, firmware, hardware, network devices etc.) are cheaper to diversify, diversification should extend to these parts of the system as well.

5.4.2 Independent Operating Systems for Control Components used for return-code generation

From the answers to question 2.1.6 in the questionnaire we take that experts tend to advise using different operating systems for the critical components. By relating to the above question, we would like to get a better understanding of the added value and the downsides.

Which puts the correct functioning of at least one control component to a greater risk: Using the same operating system or using the same application layer software on all four control components?

This discussion is captured in the summary of section 5.4.1.

5.4.3 Independent CPU for Control Components used for return-code generation

Same as question B [here in section 5.4.2] for different hardware.

Given physical servers for the critical components, which puts the correct functioning of at least one control component to a greater risk: Using the same operating system or using the same type of CPU in all four control components? Are other hardware components more critical than the CPU (find a list of proposed components in the summary to question 2.1.6 of the questionnaire [document ""Summary of the replies to the questionnaire""])?

This discussion is captured in the summary of section 5.4.1.

5.4.4 Control Components used for mixing and verifier

The previous questions are about the control components used for return-code generation. Here we would like to find out if your assessment is different when considering the other control components and the verifiers. (We will revisit the printing office in subsequent questions.)

Is your answer to questions A, B or C [here sections 5.4.1, 5.4.2 and 5.4.3] different when considering

-the control components used for mixing?

-the verifiers?

This discussion is captured in the summary of section 5.4.1.

6. Block 3 - Printing-Office (Diversity to support security and trust-building - Part 2)

6.1 Overview

The topic of block three was the printing office. It relates to question 2.1.3 of the questionnaire. The answers to that question reflect wide consensus.

Based on the conclusion that action should be taken, we asked the experts to relate their statements to two proposals for a stricter regulation in this domain (see section 6.3).

6.2 Summaries of the answers to the related questions of the questionnaire

6.2.1 Question 2.1.3 of the questionnaire: "Printing office"

For «individual verifiability» to be effective, the return-codes the voters receive by postal mail need to remain confidential. Since it is infeasible to protect the return-codes by cryptographic means from being divulged during and after printing, the trust-model allows the functionality of the printing office to be considered trustworthy. However, the VEleS is not clear about which functionality can formally be assigned to the printing office. Should the printing office only decrypt and print? Or could even some parameters be generated in the printing office? Formally assigning functionality to the printing office is intriguing, because that functionality could then formally be considered trustworthy and it would be obsolete to verify *the output using independent equipment. With the system of the Swiss Post, the generation of certain parameters has not been designed to be verifiable (formally, the generation of those parameters have been assigned to the printing office). How does this affect the effectiveness or the credibility of verifiability? Which enhancements would you propose?*

Summary from 12 of 15 responses:

The answers do not contain any significant contradictions.

Trusting the printing office at performing critical computations without verification is perceived as problematic, a majority seems to consider it highly problematic. Computations should be avoided or verified: 12/12

The confidentiality of parameters hinge on whether they are generated based on true randomness. Their appropriate generation in particular should not be subject to mere trust: 3/12

It is acknowledged that - in terms of the trust-model underlying the cryptographic protocol - the printer has to be trusted at least to the degree that printed values are not divulged. Therefore utmost care needs to be taken at defining the operations around printing: 7/12

To the same end, solutions could be explored that involve two independent printing services thus requiring only one to be trusted: 2/12

The trust-assumptions in the printing service are likely to impact the overall public trust in internet voting: 3/12

Due to its criticality, a governmental printing service might be reasonable: 2/12

One expert proposes to additionally discuss the trust-assumptions on the postal service with regard to the distribution of the voting material: 1/12

Also it has been proposed to investigate methods to offer verifiability that do not rely on printed return-codes: 1/12

6.3 Introduction to Block 3 on the platform

Block 3 was presented as follows:

6.3.1 Generation, verification and printing of parameters and codes

In their answers to question 2.1.3, mandated experts expressed that trusting the printing office at performing critical computations without verification is problematic. A majority seems to consider it highly problematic. Computations should be avoided or verified.

At some point, the task-force will recommend which measures to implement and by when. To that end, we make two propositions which we ask you to assess. The first one would take time for cantons and providers to implement. The second one is simpler but brings less value. It is meant as a possible mitigation to allow more time for the first proposal to be implemented.

6.3.2 Proposition A

1 - The notion of the printing service is replaced by “setup-components”, “setup-verifiers” and “printing components”.

2 - Functionality of the components

2.1 - Setup-components

- They generate the security-relevant parameters as defined by the cryptographic protocol, including the codes for individual verifiability.
- They use multiple random sources for generating random values. As an alternative for public parameters they could use a seed agreed upon by multiple persons.

2.2 - Setup verifiers

- They verify that the set-up components have generated the parameters correctly.

2.3 - Printing components (machines that receive the verified file containing the parameters to be printed, as well as the printing machine)

- They verify the origin of the verified file containing the parameters to be printed (signatures of setup-component and setup-verifier).
- They decrypt the file.
- If the file is not yet in a printable format, they create a printable file.
- They print the voting cards.

3 - Trust-assumptions (assuming one setup-component, one setup-verifier and one printing component)

3.1 - It is trusted that either the setup component generates the parameters correctly or the setup-verifier verifies the correct generation soundly.

3.2 - One out of four random sources is trusted to deliver values with sufficient entropy (control component or human chosen secret passphrase/dice roll).

3.3 - With regard to verifying secret parameters generated based on values established by persons: It may be assumed that persons will not be able to memorize very long values, e.g. the hash of the other persons' passphrase.

3.4 - The setup component or the setup-verifier is trusted not to leak secret data, unless by encoding data into their output.

3.5 - The printing component is trusted:

- To verify signatures soundly
- To print according to the verified printing file
- Not to leak secret data

3.6 - The channel between the setup-component and the setup-verifier is trusted. Channels leaving these components are untrusted.

4 - Operational requirements for "setup-components", "setup-verifiers" and "printing components"

4.1 - Before performing critical operations the following measures must be taken:

- Components have to be secured prior to operation
- Network connections must be removed physically
- The origin of the software to be installed has to be verified. The origin of the internet voting software has to be verified based on comparison with a signed published representation

4.2 - After performing critical operations the following measures must be taken:

- Components and any physical support holding critical data must be securely stored after use or the data securely erased
- Unless there is a significant reason, data is securely erased after use.

4.3 - Organizational measures need to enforce the consent and presence of a sufficient number of persons in the roles necessary as to ensure that the critical processes of the following kind are executed correctly:

- Operating with voting data
- Setting up the components
- Securing components or securely erasing data

4.4 - Data may only be transferred between the setup-component and the verifier by physical support which is then securely erased or the support securely stored.

4.5 - A number of voting cards must be compared with the verified file containing the parameters to be printed. This is to observe that the parameters are printed correctly.

6.3.3 Proposition B

1 - The notion of the printing service is replaced by “setup-components” and “printing components”.

2 - Functionality of the components

2.1 - Setup-components

- They generate the security-relevant parameters as defined by the cryptographic protocol, including the codes for individual verifiability.
- Unlike above, they can still generate secret values on their own. The generation of public values needs to be verifiable using a verifier as currently defined in the VEleS.

2.2 - Printing components (the machines that receive the file containing the parameters to be printed, as well as the printing machine)

- They verify the origin of the file containing the parameters to be printed (signatures of setup-component).
- They decrypt the file.
- If the file is not yet in a printable format, they create a printable file.
- They print the voting cards.

3 - Trust-assumptions (assuming one setup-component and one printing component):

3.1 - It is trusted that the setup component generates secret parameters correctly

3.2 - The setup component is trusted not to leak secret data.

3.3 - The printing component is trusted

- To verify signatures soundly
- To print according to the verified printing file
- Not to leak secret data

3.4 - Channels leaving the setup-component are untrusted.

4 - Operational requirements for “setup-components” and “printing component”:

4.1 - Before performing critical operations the following measures must be taken:

- Components have to be secured prior to operation
- Network connections must be removed physically
- The origin of the software to be installed has to be verified. The origin of the internet voting software has to be verified based on comparison with a signed published representation

4.2 - After performing critical operations the following measures must be taken:

- Components and any physical support holding critical data must be securely stored after use or the data securely erased
- Unless there is a significant reason, data is securely erased after use.

4.3 - Organizational measures need to enforce the consent and presence of a sufficient number of persons in the roles necessary as to ensure that the critical processes of the following kind are executed correctly.

- Operating with voting data
- Setting up the components

- Securing components or securely erasing data

4.4 - The functionality used in the software of the setup component for generating random values must enjoy special scrutiny.

4.5 - Measures must be put in place to ensure that sufficient entropy is used for generating random values.

4.6 - A number of voting cards must be compared with the verified file containing the parameters to be printed. This is to observe that the parameters are printed correctly.

6.4 Questions and summaries of the discussions on the platform

6.4.1 **Assessing Proposition A**

Assuming sufficient expertise at designing, implementing and scrutinizing a solution compliant with proposition A, how would you assess the added value? Which are important residual risks?

There is a strong sentiment among the experts, that the printing office is a severe security problem and it is very difficult to fix the situation as long as we rely on today's printing process.

Two propositions A and B were presented to the experts. Both propositions are very close to the existing setup, but they improve parameter generation and introduce some verification without adding too much overhead.

The two propositions were received with lukewarm approval. Nobody saw a fundamental flaw with them, but only a few experts confirmed that these propositions add some value. A possible conclusion is that an implementation makes sense if the propositions can really be found to raise security and if costs can be kept small.

Adrian Perrig proposed to print superfluous digits on the voting material (proposition C). The return-codes and the additional digits would form one long number. A second component would then print a mask with holes in it that corresponds with the voting material of the same voter. When putting the mask over the voting material, the relevant digits of the return-codes become readable. The superfluous digits are hidden. This proposition could make it harder for an attacker to steal voting material via the printing process. However, it makes the printing more difficult, since multiple elements have to be produced and matched for the same voter. The delivery of the mask would need to be discussed and there might be a usability problem with putting the mask over the voting material correctly.

Bryan Ford presented a proposal that involves an even more advanced multi step printing process (proposition D). The return-codes would be printed in multiple steps and glued onto the voting material in a way that makes sure only the voter will be able to read them. This proposition is much more complicated during the production, but it seems to reduce the usability problem with proposition C and also the transport is simpler than C.

All in all, the discussions around the printing office have not been conclusive during the dialog. Therefore, the Federal Chancellery and the Cantons are planning to commission a separate examination of the propositions A and B outside of the expert dialog in order to provide a near term perspective. In a second step, the propositions C and D will be examined as well. This discussion will also include representatives of printing offices in order to explore the mid- / long-term perspective.

6.4.2 **Assessing Proposition B**

Proposition B relies on one component generating values correctly. Do you see added value in adhering to the conditions in this proposition? Is there anything you would like to add?

This discussion is captured in the summary of section 6.4.1.

7. Block 4 – Public Bulletin Board

7.1 Overview

The topic of block four was public bulletin boards. It relates to question 2.1.7 of the questionnaire. There were differing opinions on this question in the answers to the questionnaire. Similar as with block 3, we asked the experts to relate their statements to two proposals. Unlike in block 3, the proposals this time outlined possible solutions rather than potential requirements for the regulation.

7.2 Summaries of the answers to the related questions of the questionnaire

7.2.1 Question 2.1.7 of the questionnaire: "Other Forms of Verifiability"

The trust model was defined under the assumption that the voters' computers should not be trusted with regard to manipulations of votes. At the same time, user-friendliness was a strong concern. This is why voting with return-codes was chosen. By assuming that voters have a device they trust, the remaining trust assumptions could be additionally relaxed: For instance voters could post and confirm their vote on a public bulletin board and also verify the proof allowing universal verifiability. Or they could send a signed digest of their vote to an electoral commission that checks that the encrypted vote has been recorded as cast, i.e. that the vote passed to the tally. How do you assess the added value and downsides of an additional verifiability service based on a device trusted by the voter, given that voters would need to transfer and evaluate cryptographic values using different equipment in order to benefit from individual and universal verifiability? Considering a solution where votes are posted to a public bulletin board, how do you assess long-term privacy issues?

Summary from 13 of 15 responses:

This is a valuable approach. Some experts of this group particularly see opportunities regarding public confidence and achieving more security in the long run: 6/13

- At the same time, long-term privacy risks need to be addressed: 3/6

No clear assessment. One expert raises concerns regarding the trustworthiness and user-friendliness of the voters' platform, the other one regarding long-term privacy: 2/13

Not advised, particularly in the absence of a dedicated voter device or if long-term privacy issues are not solved. One expert of this group questions the idea per se of putting the task of verification on the voter: 5/13

7.3 Introduction to Block 4 on the platform

Block 4 was presented as follows:

7.3.1 Introduction

Verifiability in Switzerland aims at detecting cases of large-scale fraud. Many computations cannot be verified by the voter but by the administrations or elected commissions. Their verification capabilities in return hinge on whether critical components they or their partners run are functioning correctly. Due to the trust-model defined in Art. 5 in conjunction with chapters 4.1 and 4.3 of the annex, it must be possible to justifiably assume that at least one control component, at least one verifier and the printing office are functioning correctly. Refer to the introduction in Block 2 [here section 5.3] for details.

Vast scrutiny in connection with the number and diversity of critical components as well as separation of duty at operations serve the detection of manipulated votes. Theoretically, there is no maximum to the extent to which these practices could be implemented. In reality, the possibilities will be limited.

In this section we discuss an additional means of verifiability based on two propositions. The goal is to allow the detection of manipulations even if full sets of critical components of the internet voting system are not functioning correctly. The propositions are based on publishing vote-related information on a public system-independent platform (public bulletin board; PB). Voters are given read-access to the platform so they can additionally perform verification independently of the return-codes.

We would like to get your assessment of the added value in terms of security and trust-building. Also we would like to discuss the caveats that have been brought forward in the answers to the questionnaire, i.e. with respect to the long-term secrecy of the vote as well as the user-friendliness of the verification steps performed by the voters. Indeed, it seems that expert-opinions currently differ when bringing the upsides and the downsides into relation (see summary of 2.1.7 in the questionnaire).

7.3.2 **Proposition A**

In a nutshell

In this proposition the voters can use a PB for individual and universal verifiability. With this alternative they do not have to trust the voting system. The trust-assumptions are shifted.

Recall from the introduction to Block 2 that voters are instructed to report to the administration cases where the return-codes are not displayed correctly. At that stage, voters can then still vote by postal mail or at the polling-station. If the codes are correct, they can confirm the internet vote by entering their confirmation code. Proposition A naturally integrates into this process: Additionally to verifying the codes, the voters can verify the correctness of their vote and the acceptance of the confirmation code on the public bulletin board (PB). Thereby voters verify that their vote has been cast-as-intended and recorded-as-cast, i.e. without relying on the printing office or the control components. After the voting phase, the results of mixing and decryption are also posted to the PB. Voters can verify that all votes (and in particular their vote) have been tallied-as-recorded. Voters can perform these verification steps an arbitrary number of times, possibly using different devices and different software. The parameters from the setup-procedure it takes to perform the verification as well as proofs of their correct generation are published on PB.

Functionality and trust-assumptions

The added possibility to detect systematic manipulations depends on whether a sufficient number of voters use at least one correctly functioning device at verifying the votes on the PB. The PB is designed to append signed messages received from the untrusted voting server and not to change or drop any content. The following trust-assumptions apply:

- The voting system (all control components, verifiers and printing-office components) is untrusted regarding verifiability.
- The PB is untrusted.
- A voter-device used to verify the votes is trusted
- There is a trusted broadcast-channel that can be used as a boot-strap pointer to the PB and to serve as a foundation for detecting and claiming malfunctions of the PB (e.g. unpersonalized paper voting-material, official gazettes of the administrations, newspapers, maybe someday a block-chain).

Description

The PB and the software for the voter-device are meant to be implemented and operated as easily as possible, e.g. much easier than a control component, and yet to bring added value with regard to detecting manipulations of votes.

1 - One or multiple server infrastructures are set up to constitute the PB. Each infrastructure returns either its full content, the content related to a voter (see 2c below) or the voting parameters at a given read-request along with a signed hash of the content. If the received contents differ between the infrastructures, the PB is considered not be functioning correctly and voters are instructed to inform the administration.

2 - The PB is initialized with

1. Encryption-scheme parameters, including the values that represent the voting options along with their text interpretation (e.g. question plus the answer or a candidate name).

2. Proofs of their correct generation.
3. Anonymous voter identifiers known to the individual voter (in current Swiss systems this could be a digest of the voting-card number which the voters enter to authenticate). Note that the detection of manipulated votes does not hinge on these identifiers or their pre-images being secret.
4. The PB's public signature key (a global one or one per infrastructure).

3 - A pointer to the PB (e.g. a URL) as well as a hash of the values on the PB are broadcasted through the trusted broadcast-channel. Based on this, the verification software is initialized and the public parameters are downloaded from the PB. (We assume for this example that the verification software runs on a mobile, the vote is cast on a laptop.)

4a - Before sending the vote to the server, the encryption of the vote and the random number used for encrypting the vote are displayed on the laptop, here we assume one QR-code, it could also be two QR-codes (one per value). The voters scan the QR-code with their mobile. Based on the random number and the parameters downloaded in step 3, the mobile displays the encrypted voting options as text. Voters can also use additional devices with different software to verify the encryption repeatedly. They cast the vote if they are satisfied with the verification result.

4b - Additional option: Prior to casting, voters have the option to create an arbitrary number of votes they do not intend to cast and verify the encryption of such a vote an arbitrary number of times. They can perform the verification with the same application on the same mobile or also with other devices possibly running different software. Once they are satisfied that the device they use for casting the vote encrypts votes correctly, they would create and cast the actual vote. It would be sufficient if the mobile only keeps the encryption of the vote, or in the case of two separate QR-codes, only to scan the one holding the encryption. The secret number would not need to be stored on the mobile. The contents of the QR-code might also be displayed as text for the voter to transmit to devices that cannot read QR-codes.

5 - Apart from generating and sending back the return-codes, the untrusted server signs the vote and posts it to the PB, assigned to the anonymous voter identifier. The PB accepts the entry if the signature is correct.

6 - Upon pressing a button, the application on the mobile phone directly downloads the contents from PB, signed by the PB. Based on the voting-card number (let's assume that in step 4 the voting card number was contained in the QR-code too), the mobile displays a text to confirm that the vote has reached the PB correctly.

7 - As in step 4, voters can repeat this step with other devices and/or other software. Once the voter is satisfied with the verification results, she enters her confirmation code.

8 - Apart from sending back the finalization code, the untrusted server signs a confirmation-message and posts it to PB, assigned to the anonymous identifier and the vote.

9 - Again upon pressing a button, the application on the mobile phone directly downloads the contents from PB and displays a text that the confirmation-message has been posted to the PB. This can again be verified multiple times using different equipment.

10 - After the voting period is over, the results from mixing, decrypting and possibly counting are posted to PB along with the proofs. The administration use the verifier to verify these proofs and additionally verify that the same votes are held by the control components. In case of inconsistencies they start an investigation. Voters could also download the full contents and verify that the votes have been tallied-as-recorded.

11 - Server infrastructures that constitute the PB that change or drop votes prior to tallying will be detected by voters that read the full PB using a correctly functioning device. This however hinges on the assumption that there is at least one correctly functioning server infrastructure. Voters can use the signed content obtained in the previous steps to prove that their vote should be counted.

12 - If we want to assume that none of the server infrastructures should be trusted, the individual voters would need to verify that their vote has not been dropped or changed prior to tallying. To that end, using the information obtained in steps 4, 6 and 9, voters could locate their vote in step 10 and make sure that

the encryption is correct. To address the case where all infrastructures show different content depending on who makes the request, the hash value of the full PB could be broadcasted through the trusted broadcast-channel. Again, voters can use the signed content obtained in the previous steps to prove that their vote should be counted.

7.3.3 **Proposition B**

In proposition A we assume that encrypted votes are posted to the PB. The encryptions will be decryptable in the far future. This could be addressed by posting perfectly hiding commitments to PB instead of the encryptions. This however would constitute a change to the existing Swiss systems which would require time. Further down we ask the question if it seems acceptable to yet post the encryptions, given anonymous identifiers. For the case where it might be concluded that this is not acceptable, we present proposition B as a possible intermediate mitigation.

In this proposition, the PB serves solely individual verifiability. However the public board cannot be used for decrypting the votes, i.e. not even in the long term.

To that end, not the votes are passed to the PB but a hash of the votes. The proofs of correct mixing and decryption are not posted to the PB. Thereby the benefit lies in the verification not relying on the control components or the printing office to function correctly. It does hinge on the verifier functioning correctly. This might seem acceptable, given that it is running offline and within an observed process which is typically performed in the cantons' premises.

7.4 Questions and summaries of the discussions on the platform

7.4.1 **Value of a Public Board**

Is a PB a valuable element of an internet voting system?

A Public Bulletin Board can be a valuable element of an internet voting system when several aspects receive the necessary attention during the design, implementation and introduction phases.

The list of these aspects consists chiefly of the following items:

- The Public Bulletin Board has to be designed carefully to fit into the overall design of the internet voting system.
- The trust assumptions should not be too weak and also not too strong. Using the appropriate threat models is an essential step.
- A vote-and-go capability would be helpful. Final agreement and monotonicity seem equally important.
- Any doubts with regards to the secrecy of the vote could have a very negative effect on the internet voting system as a whole.
- Dispute resolution is very important since an incorrect claim of voting fraud with reference to the Public Bulletin Board has to be refuted convincingly to experts and the general public alike.
- Communicating the role and the proper use of the Public Bulletin Board towards the voters is a very big undertaking on its own. This should involve ways to make the Public Bulletin Board and internet voting as a whole tangible for the voters so they can actually understand it.
- Extensive usability studies are essential for the success of the Public Bulletin Board.
- With regard to trials with Public Bulletin Boards, cantonal particularities should be taken into consideration.

Under the line, the Public Bulletin Board has the potential to improve security and trust. The security improvements strongly depend on how the board is implemented, used and the information displayed. Other than that, there are additional security risks that need to be addressed.

7.4.2 **Public Board as a means to gain trust**

Is a PB suitable to establish trust?

A Public Board can establish trust. If done right, the evidence and the transparency a Public Board provides will have a positive effect on trust of the voters into the system, namely in a long term perspective. A Public Board signals the authorities' commitment to transparency, which in turn has the potential to translate into increased public confidence into the electoral process.

It is very important to note, however, that a misdesigned, a poorly implemented, a badly documented or an otherwise not optimally supported Public Board will have a negative effect on the trust: Unless the Public Board and everything around it, is of a very high quality, it could very likely have a negative effect on trust. So the Public Board can only unleash its value if the simple metaphor is supported by good design, proper implementation and adequate operation that also convinces the experts.

It is likely that relatively few users would use the Public Board themselves, but instead rely on the checks executed by independent experts / political parties. But this is not necessarily a problem and one could think of establishing such a process officially, while still giving users access to the Public Board directly or via official and independent tools.

The Public Board is closely tied to the tools that make use of it, since very few voters will use it directly. The interface between the Public Board and these tools is therefore very important. If there is a failure with this interface, the Public Board will be broken from a user's perspective. That's why changes to the interface have to be done in a very careful way and the tools and their providers have to be supported adequately.

7.4.3 **Public Board and user-friendliness**

Does it seem likely that already or in the future a sufficient proportion of voters will use a correctly functioning device to perform the verification? How do you assess the potential user-friendliness?

It is very clear that user-friendliness is an important factor that decides on the degree of use by voters and also influences the proportion of voters that verify attentively enough to notice errors with the verification and also care to report them. However, requirements for user-friendliness in the context of internet voting must be specified before any thorough examination can be performed.

User interface studies are needed to assess various design and usage scenarios. It is also an area where testing of a design in a structured context like a citizen science project possibly in conjunction with internet voting trials could help studying the use of the public board and contribute back to its design.

Even if a mature and user-friendly design of the public board could be reached, there would still be the chance that an attacker could corrupt any particular voter's device(s) used for voting and / or verification. Likewise with social engineering attacks to steal secret information from the voter. It takes a sufficient proportion of voters that are non affected in order to detect large-scale fraud.

7.4.4 **Public Board communication**

How should this additional method for verifiability be communicated towards the voters and brought into relation with the current method, knowing that up to now voters were told that only the electoral board is able to decrypt the votes?

Effective communication with the voters is important. This is particularly true with regards to good practice when performing the verification (e.g. different device, separate network, bluetooth turned off) but also with regards to possible security risks that emerge from offering a new means of verifiability.

It would be useful if a scheme could be developed that does not require to display information (e.g. randomness) that is critical for the secrecy of the vote.

The difficulty to communicate the proper use of a public board effectively, makes it a hard to solve problem.

7.4.5 **Public Board and voting secrecy**

How likely does it seem that in the far future decrypted votes will be correlated with the voters' identities? Would we need to consider this a problem from societal perspective (as by now voting secrecy is guaranteed by Swiss law)?

Voting secrecy is a basic right, guaranteed by law. If a Public Board raises doubts about this guarantee, then this undermines not only trust in the Public Board but also in the internet voting channel and voting as a whole. The use of a Public Board raises some concerns regarding vote privacy. Depending on the cryptographic techniques used, advances in cryptanalysis, or due to bugs, the information on the board may reveal how a voter voted.

We can only speculate on the potential motivations for future attackers to attempt decryption or de-anonymization of past votes, and what harms such attacks could lead to, such as voter embarrassment or coercion. Nevertheless, it is important to understand the risks and trade-offs associated to using a Public Board, and different designs thereof.

Some Public Board designs publish encryption of votes. Here, secrecy of votes relies on the same set of assumptions which guarantee secret communication over the internet. Yet, these assumptions may be invalidated, and votes may be decrypted if quantum computers become a reality. If the board contains voter identifying information (e.g. required for auditing purposes) then the link between voters and their vote may be revealed. These designs may potentially be strengthened by using encryption schemes which are secure even against quantum computers -- such schemes are under development by the cryptographic community.

Other designs may hide the link between voters and their encrypted ballots so even if ballots get decrypted individual choices stay secret.

Finally, other designs aim to achieve "everlasting privacy". Instead of publishing encryptions of votes, such schemes publish only a so-called "perfectly hiding commitment", which registers the vote on the bulletin board, analogous to a hash of the encrypted vote, but which provably contains no information about the vote's content that could ever be decrypted or revealed no matter how powerful the attacker is. Of course, the overall guarantees for vote privacy still rely on the security of the rest of the building blocks.

7.4.6 **Public Board to determine voter turnout early**

According to both propositions, the full contents of the PB would be readable by the public at any given time. Thus, the preliminary turnout would be known before the voting period ends. This could be avoided by limiting access to the parameters and to the data related to the individual voters before the voting period ends. However, we wonder whether knowing the preliminary turnout should be considered an interesting feature rather than a problem. What do the social scientists think?

It is very important, that any form of Public Board does not leak preliminary results of a vote or election. Data that allows to deduce voter turnout is less of a concern, though. Also because this exists for voting by mail in some Swiss voting circles already.

7.4.7 **Public Board and support for voters**

Following the logic of querying a PB with independent software of own choice, would it be correct for the cantons or the provider not to offer support in case of problems during the verification process?

It is important to distinguish different support situations:

1. Support with using a verification software that is officially endorsed by the canton.
2. Support with using a 3rd party verification software.
3. Support for a voter when the software fails to verify the vote and a dispute arises.

(1) A canton needs to provide support for voters using officially endorsed software.

(2) In case of 3rd party software the canton should maintain a list and in case of problems the canton should be able to direct the voters to the provider supporting the software.

(3) If the role of the Public Board is to verify the correct operation of the vote by the system provider and the canton as well. Then the canton is not the best contact in this situation and there has to be an independent support provider. However it is difficult to design a solution that addresses this case adequately: If the bad reports are escalated before they are thoroughly confirmed, then the trust could be undermined without good cause. And if the support provider discourages the voter from pursuing a report, then a malfunction or fraud could go undetected. It is therefore important to design and implement proper dispute resolution for this and other situations where a dispute may arise.

7.4.8 Public Board vs. Independent Control Component (see Block 2) and adapted parameter generation (see Block 3)

Recalling Block 2 [here section 4], would you recommend Bob to introduce a PB or rather to procure new control component software or adapt the protocol to achieve parameter generation that is more trustworthy, assuming he could only pick one?

Under the line, the Public Bulletin Board has the potential to improve security and trust. The security improvements strongly depend on how the board is implemented, used and the information displayed. Other than that, there are additional security risks that need to be addressed.

Few experts were willing to address the hypothetical choice between more software diversity, more trustworthy parameter generation and the introduction of a Public Bulletin Board. One could conclude that it is a very hard choice or that it is a very hypothetical one.

Software or more generally implementation diversity brings a quantifiable benefit in security with linear costs. As discussed in 4A und 4B, a Public Bulletin Board can be a valuable element of an internet voting system which can establish trust.

8. Block 5 – Examinations Mandated by Government

8.1 Overview

The topic of block five was the mandated examinations of the internet voting system. It relates to questions 3.1 and 3.3 of the questionnaire (independent examinations) and also to question 2.1.8 (methods for implementing and deploying correct software). There was consensus on most parts of these questions in the questionnaire.

In block 5 we asked specific questions about the scope of the examinations and existing standards, because only few experts had given such answers. The third question in block 5 is about handling nonconformities. This subject was chosen because the experts had opposing opinions about it in the answers to the questionnaire.

8.2 Summaries of the answers to the related questions of the questionnaire

8.2.1 Question 2.1.8 of the questionnaire: “Correct implementation and protection from unauthorized access”

The software that was published in 2019 had security flaws. Which measures could be put in place in order to avoid these flaws (i.e. to make sure that the protocol is implemented according to its specification)? The VEleS did not set a strong focus on development and deployment procedures but rather on the final product. Do you know any standard that would likely lead to better procedures at development if required by the VEleS? Which measures could be put in place in order to ensure that the correct software is running (as examined and authorized based on the source-code), in particular on the trusted components? What could the role of independent experts from academia and industry be?

Summary from 14 of 15 responses:

The answers do not contain any significant contradictions. The following practices have been proposed:

- Security by design, particularly Microsoft SDL: 2/14
- OWASP Application Security Verification standard, FIPS, or Common Criteria: 1/14
- ISO 9000: 1/14
- Separation of duty: 6/14
- Best practices for deployment; e.g. verifying code using (public) hash / signature: 6/14
- Automated cross-checking and witness-cosigning techniques: 1/14
- Deterministic build practices as the Debian Linux distribution, use different compilers: 2/14
- Parallel implementations: 2/14
- Ensure transparency and oversight of the whole supply chain and the relevant development processes; the US trusted foundry program was mentioned: 1/14
- Involve highly skilled people from academia and industry at development: 3/14
- Open development: 3/14
- Security testing: 2/14
- Ensure quality of documentation and code; perfect alignment; reduce complexity; scrutiny by publishing: 2/14
- Supervision of developers, traceability of all actions in development and deployment based on unforgeable ledgers: 1/14
- Reduce dependencies on third-party libraries: 1/14
- Apply formal methods and verification (CC EAL 7 was mentioned): 4/14
- Measures to protect the host, in particular by using trusted execution environments to ensure the correct software is running: 3/14
- Supervision by governmental expert group: 2/14
- One expert points out the importance to consider that the abstract specification must reply to the real-world problem. If it fails to do so, all measures to ensure trustworthy implementation and deployment can be rendered obsolete: 1/14

8.2.2 Question 3.1 of the questionnaire

Which criteria should determine which persons or organizations are mandated with an examination? Please outline the scopes for examination you find important and relate your answer to these scopes. Given that internet voting is not standard technology, in which areas (e.g. software, operations/infrastructure, trusted components) does formal certification conducted by certification bodies seem reasonable?

Summary from 14 of 15 responses:

Criteria: The most cited criteria are expertise and experience. For scopes related to crypto the examiners should be academics. One expert notes that for standards based examinations they should be approved by a standardization body. Independence is also mentioned, as in independence from each other, from companies having a stake in internet voting and from the Cantons and the Confederation.

The following scopes are mentioned:

- ISO 27000
- Proofs of protocol/Crypto
- Model vs design
- Implementation vs design
- Crypto source code

- Software
- CC is not relevant
- limit to key components, do not overburden

Formal certification:

Most experts did not mention this topic in their answer. Those who did either agreed that formal certification was useful for the ISO 27000 scope or stated that they are not convinced by standardized certifications.

8.2.3 Question 3.3 of the questionnaire

Does the credibility of the outcome of an examination among the public (and experts from the public) hinge on which organization appoints the examination? Please relate your answer to individual areas of the scope (e.g. software, operations/infrastructure, trusted components).

Summary from 15 of 15 responses:

All experts who gave an answer noted that the credibility depends on who appoints the examination. They all agree that there should be different organizations for the different scopes and that they should be appointed by the government or an independent committee. The big four are explicitly excluded by one expert.

8.3 Introduction to Block 5 on the platform

Block 5 was presented as follows:

8.3.1 Introduction

The goal of the examinations is to verify that all the requirements set forth in the ordinance have been met. For example, this includes auditing the source code or reviewing voting procedures.

In this block we only discuss the examinations that are officially mandated. According to responses to question 3.3, the entity mandating the examinations should be the federal administration (often referred as the government in the responses) or an independent committee. Additionally there can be examinations run by the community, based on the publicly available information. These will be discussed in another block.

This block is based on responses to questions 3.1 and 3.3, regarding independent examinations. It also includes topics from question 2.1.8 regarding standards and methods for developing and deploying.

According to the responses, three topics seem interesting to discuss: the scope of the examination of the control components, the standards that could be used, and possible action if issues are discovered during an examination.

8.3.2 Breaking down VEleS Scope 4 (Software Examination)

The VEleS defines different scopes that must be thoroughly examined.

The current scopes defined by the VEleS (Art 7) and its annex are:

- Scope 1 - Examination of Cryptographic Protocol and proofs
- Scope 2 - Examination of functionality (software other than control components)
- Scope 3 - Examination of infrastructure
- Scope 4 - Examination of Control Components
- Scope 5 - Examination of protection against infiltration (intrusion test)

- Scope 6 - Examination of the print office (absence of leak during printing)

According to the answers in the questionnaire it seems that the existing examination of the software implementing the cryptographic protocol should be broken into two scopes. The main reasons are differences in the required skillsets, the methodology and the available standards.

One scope would be related to the correct implementation of the cryptographic protocol and the other related to generic software engineering. The software implementing the protocol is in Scope 4, which would now have two parts.

Scope 4.a Examination of correct implementation (alignment)

The goal of this examination is to verify the alignment of the different levels of abstraction. We consider the formal description of the protocol (used for the mathematical proofs), the specification of the system (used for implementing the system) and the resulting code itself

This scope would cover the following:

- Verify that the specification used to write the code correctly reflects the protocol defined in the formal description used to prove the security properties of the protocol. Analyze the impact of any elements that are added by the specification.
- Verify that the code correctly implements the specification. Analyze the impact of elements that are added by the code.
- Give an assessment of whether the code and the specification is written in a way that facilitates the examination of the correct implementation.

Qualifications for Scope 4.a

Experts in charge of this examination should be cryptographers, ideally the ones implied in the examination of the protocol. As discussed in the discussion of block 1, there would be the need for at least two experts.

Standards for Scope 4.a

No standards have been mentioned for this type of examination. Current research aims at generating the implementation automatically from the high level description, which would guarantee a complete alignment. It is not available yet.

Scope 4.b Examination of programming and deployment practices

This scope focuses on how the code is created and deployed.

- Examine the code and documentation that have been produced.
- Verify that the code and documentation is developed with a method that is demonstrably effective and auditable for developing secure code.
- Verify that the code is also deployed according to a demonstrably effective and auditable method for deployment of secure systems.

The first item is already in the VEleS. The other two are not explicitly part of VEleS. According to the responses to question 2.1.8, this would typically include examination of the build and deployment process, usage of different compilers, code signing or the analysis of dependencies.

Qualifications for Scope 4.b

Examiners for this scope should be experts in software development for secure systems.

Standards for Scope 4.b

In the VEleS the examination of the software (scope 2 and scope 4) is based on the Security Functional Requirements (SFR) of the Common Criteria (CC) Protection Profile (PP) for voting systems. Higher EAL levels are required for the control components as opposed to the untrusted server. While it refers to the requirements of the Common Criteria, this examination is not a formal Common Criteria certification. Other standards that were mentioned in the answers to 2.1.8 are the Microsoft Software Development Lifecycle, FIPS, OWASP Application Security Verification Standard and the Voluntary Voting System Guidelines (VVSG).

FIPS is already mentioned in 3.3.6 of the annex as one of the standards for secure algorithms and key lengths.

The answers to the questionnaire indicate that a formal certification according to a standard would not be interesting in the case of the examination of the software.

8.4 Questions and summaries of the discussions on the platform

8.4.1 Scopes for the examinations

Looking at the 6 scopes defined in the VEleS (listed in paragraph 2) is there a scope of examination that is missing?

Do you agree with the separation of scope 4 into two different parts, to add an emphasis on examining the correct implementation of the protocol by cryptographers and to extend the scope with programming and deployment practices?

The outlined scopes make sense. Yet applying this high level of scrutiny only to scope 4 and thus the Control Components is seen as an unnecessary prioritization. A successful attack on the other voting servers (scope 2 and 3) could severely undermine the confidence into the system regardless whether a control component detected a manipulation or not.

The examinations must also be conceived and performed from a holistic view. Performing the examinations only within the individual scopes without taking into account the full context could lead to important aspects being missed. A way to cover this would be a group of experts that is in charge of managing all the different transitions between the scopes.

Usability and performance have been mentioned as examples of external attributes of the software that should be taken up in the examination framework.

The decision on the assurance standard should not be determined by whether a component is declared trusted / untrusted in some model. As an example: Although untrusted servers that misbehave at authentication would be detected thanks to a trusted group of control components, failures in the authentication process are still bad and avoiding them deserves focus.

8.4.2 Standards for the examinations

Scope 2 (examination of functionality) and 4 (examination of control component) are based on elements of the Common Criteria internet voting Protection Profile and Assurance Levels 2 and 4 respectively.

Scope 3 (infrastructure) requires an ISO 27001 certification.

Scope 5 (penetration test) mentions OWASP.

Do you see aspects that could be borrowed from other standards and would be beneficial to the examinations?

Do you see a value in requiring a formal ISO 27001 certification as opposed to an examination based on that standard?

ISO 27002 seems to be broad enough to cover all the scopes of the examinations, which is desirable, but it comes with some limitations. Its risk based approach and its lack of detail in the domain of security and assurance requirements make it a poor choice for examination of the software.

The following frameworks/standards were mentioned as foundations for scope 4b: ISO 9000, Microsoft Secure development Lifecycle framework, ISO 27001 and ISO 27002 as well as Common Criteria.

Standards for security and privacy are under active development with new standards emerging regularly. It is important to keep a close eye on this area.

For scope 3 the NIST Cybersecurity Framework was mentioned.

There is a certain danger, that the disparity of control frameworks for the different scopes might cause a lack of consolidated view on the security of the system as a whole.

8.4.3 **Handling of non-conformity**

Imagine that non-conformities have been detected in one of the examinations and that an independent entity has run a risk analysis to identify the risk created by these non- conformities. In ISO 27001 certification, for example, it is often the case that a certification is given even if some less critical conformities have been detected. Would you agree that the system could still be used if the analysis shows that the risk is low enough? Consider that if zero risk is required, then any small issue could prevent the system from being used. Do you think that some scopes are more critical than others? If that is the case, do you see any scope for which no non-conformity should be tolerated? Here are some typical examples of issues that could be discovered:

-the code does not exactly implement the protocol, but there does not seem to be an impact.

-the development method does not conform in every aspect to an effective and auditable method, but the code is correct.

-the penetration test shows some missing best practices, but there is no exploitable vulnerability.

-the ISO audit shows a lack of documentation, which will only be fixed in 6 months.

-the examiners declare that they did not have enough time to complete the examination.

Based on the assessment of “risk” or “criticality”, it can be reasonable to accept non-conformities. However, the risk itself can be hard to assess. With regards to the risk, points of reference might be found in postal voting or in other systems that would likely allow a threat-agent to reach his goal to manipulate the outcome of a vote at a lower cost. The risks of not offering internet voting may also be taken into the equation, e.g. disenfranchisement of voters abroad or losing momentum and resources towards more secure solutions, thereby possibly also addressing the risks inherent to postal voting.

Situations where non-conformities become subject to acceptance should be avoided by allowing enough time for detecting / fixing them. Mechanisms should be in place to respond to non-conformities detected at the last-minute or during the election. Generally, the more the product (including the operating infrastructure, e.g. firewalls, PKI, …) is affected, rather than the processes around the product, the more important it is to fix a non-conformity. The following types of non-conformities have to be fixed:

If a protocol has a design error, this must be fixed as it would lead to an attack.

- If the protocol description (used for verification) and the design specification (used for implementation) are not conform then this must be fixed as otherwise the formal analysis says nothing about the actual system implemented.
- If something else has been implemented than described in the design specification, then this must clearly be fixed.
- If there are bugs on the implementation level, then they must be fixed too.

9. Block 6 – Development and Publication

9.1 Overview

The topic of block six was the development practice of the internet voting system and the publication of its specification, documentation and source code. It relates to questions 4.1, 4.2, 4.3, 4.4 and 4.11 of the questionnaire. In the answers to the questionnaire there was a consensus with a general feeling that the systems and their development should be very transparent. But the experts were not explicit on how open the development and the publication should be. In Block 6 the experts were thus asked about the scope of the publication and the license of the code or the development methodology.

9.2 Summaries of the answers to the related questions of the questionnaire

9.2.1 Question 4.1 of the questionnaire

- *How should the terms and conditions with regard to source code access be defined in order for them to enjoy credibility, i.e. to be accepted by the security community?*
- *Would incentives for participation at analyzing system documentation be reasonable?*
- *How could intellectual property concerns of the owner be addressed at the same time?*

Summary from 12 of 15 responses:

Several researchers talked about the Public Intrusion Test (PIT) instead of the source code publication (which ran as a separate project). Some of these responses can be read from a source code perspective as well. Some can't. They are listed and quoted under "No answer" and taken into consideration in question 4.7 (of the questionnaire) again.

Of those responses that really answered to the source code publication, a large group would like to see a transparent development process adopted. This would mean that all commits to the source code are visible; potentially in a slightly consolidated way. This does not necessarily mean that the result is software published under an open source license. Enforcing such a license is not very important for the experts. They just want to see the development of the code and they welcome a development model where the interested audience is welcome to submit criticism and proposals.

A second group of similar size does not touch on the development model, but they want to see a broad publication of the source code with as few restrictions as possible. Many experts touch on the question of intellectual property rights. However, for almost all of them the need for maximum transparency has more weight. Several experts estimate it very unlikely that a competitor would steal the code or they mention methods to make it even more unlikely.

One expert explains that specific parts of the source code might only be made available after registration and proof of qualification.

Some individual interesting ideas:

- Parts of the source code (e.g. verification, login, encryption) that are of public interest could be made available to everyone without registration to leave no doubt. If necessary, benefits can be linked to the registration (Q&A or similar).
- For critical parts (where property rights or similar apply) registration and proof of qualifications could be required.

9.2.2 Question 4.2 of the questionnaire

What should the scope / coverage of the published documentation be in order to achieve meaningful public scrutiny?

Summary from 15 of 15 responses:

There is a clear majority opting to publish as much as possible so experts in different areas find everything they need to assess the system. Documents mentioned are all specifications, anything that could help auditors and every document that would be given to a developer or maintainer of a system. One expert goes as far as to demand the publication of all mandated examination reports. It is unclear if the other experts did not think in this direction or if this is an individual position. Another one stated the same in response to 4.7 (of the questionnaire).

Minority positions include some who opt to make some documentation available on demand; another one who would like to delegate the definition of the scope to a new-formed body or committee and a last one who thinks the publication can be limited to the *internet* voting system and additional operational documentation can be withheld.

Some individual interesting ideas:

- As an exception to that, details about any peripheral security systems or additional measures deployed in production environments may however be excluded from this publication as they are not part of the internet voting system itself (which must be self-sufficient in terms of security) and are only used to increase security and apply “defense-in-depth” principle.
- However, complete traceability requires disclosure of almost all documentation, which is probably not desired by the operator. Here the way could be taken that qualified on demand receive appropriate documents.
- In depth documentation of the architecture and the components should be made available, on request, for any expert who wants to assess the trustworthiness of the system. Different levels of detailed documentation might be useful so that the expert is not lost in thousands of pages, but can request more in-depth documentation where needed.
- The new committee can develop such guidelines and a needs basis.

9.2.3 **Question 4.3 of the questionnaire**

When should the code and documentation be published considering the workflows (development, mandated examinations and authorization)?

Which indicators could be relevant?

Summary from 13 of 15 responses:

There are two strong groups present here. One group would like to see the code and the documentation very early and regularly updated in a continuous process. The second group would prefer thorough testing and possibly a certification before the publication.

The question about the indicator was not directly answered by anybody, however, indicators like "integration tests are complete" can be read as an answer to this question.

One expert sees a reputation damage when too many silly errors are published. Another one is afraid that a code review cannot be done on a moving target. This problem is not mentioned by other cryptographers and it may seem solvable if the review happens on tagged versions / release.

Some individual interesting ideas:

- In order for the publication to achieve the “transparency” objectives, the published code and documentation should correspond to the “production-ready” version.
- Before a new software release goes into production I suggest a PIT and other ways making the public aware of the new internet voting version.
- This depends in part on the complexity of the system. The more complex a system is, the more time it takes to check it.
- This is impossible too if the system is still undergoing change.
- Involve citizens somehow. Citizen Science namely for code verification.

- The new body can develop such guidelines and a needs basis.

9.2.4 **Question 4.4 of the questionnaire**

Is it appropriate for the preparation and publication of documents to go beyond the current requirements of the VEleS? (e.g. test data, instructions for simulated voting)

Summary from 11 of 15 responses:

There is unanimous consent that interested researchers need to be able to setup a complete election based on the published code, test data and accompanying documentation. One expert suggests to also provide preconfigured virtual machines so that researchers can save time.

9.2.5 **Question 4.11 of the questionnaire**

What additional transparency measures could promote security and / or trust? (Infrastructure inspections, publication of the minutes, for example, of an electoral commission, publication of the crypto proofs, etc.)

Summary from 12 of 15 responses:

The publication of detailed reports about infrastructure and other audit reports, possible at the moment of the vote / the publication of the results is seen as a measure that promotes trust. However, there are downsides to the publication as well. The common ground here is that the publication has to be of very high quality. Bad reports will undermine instead of supporting the trust. If published, the targets of the documents need to be clear. Detailed technical information should be addressed to the technically interested parties.

There is also a call for a public bulletin board among the responses of the experts. Other ideas include a strong formal function for existing roles like "Stimmenzähler" (vote counter) and "Wahlbeobachter" (election observer) which give authority to the results and brings witness co-signing with it.

One of the experts calls for an education initiative. He also thinks that the system should not be operated by a privately held company. Another expert does not really address this question, but he asks for strong infrastructure protection with IDS / WAF (he implies WAF, but does not mention it) as a deterrent.

Some individual interesting ideas:

- «... clear information campaigns should be carried in order to popularize and demystify the details about internet voting internals. Whilst the technical details may be inaccessible for a vast majority of the citizens, the general concepts should however be presented to and understood by all (not only to a specialized public).»
- «... the level of trust towards an internet voting solution may be increased if this solution is publicly owned and operated and not provided by a private company». «In theory, the more the better. In practice there are limits due to financial and timing constraints. It takes time and money to produce documents fit for publication, the public must be given time to look at them, there must be a response/remediation period, etc. Also, if published, the quality must be very high, or it will result in reputation damage and loss of trust. While this is ideally the case, in practice not all documents produced by companies and auditors are at this standard. Augmenting code with design documents and audit reports might be a reasonable compromise.»
- «Transparency measures do not necessarily increase trust, but everything that is not transparent is in a way suspicious. The question regarding the publication of documents such as the meeting minutes of an electoral commission is whether they are relevant for the public. If this is clearly not the case for a certain type of documents, we do not see any added value coming from releasing them to the public. Publishing a large amount of irrelevant documents could also be perceived negatively, for example as a sign of not having a clear view of the security-critical topics. The relevance question regarding a certain type of documents (or other transparency measures) can only be answered on a case-by-case basis after careful analysis.»

- «Transparency and voter inclusion in every stage. Note that the function of a "Stimmenzähler" and the possibility of a "Wahlbeobachter" is not to have the most secure version of a process, but to have the most democratic accountability of a process. The same principle should apply to electronic voting procedures.»
- «In my opinion, transparency always helps to strengthen trust in a system. As long as no rights or public interests are infringed and the law permits this. In this way, transparency was to be created where possible.»
- «Infrastructure and process inspections could of course help increase security and trust, as well as further public transparency provisions such as the use of trusted hardware attestation, mechanically-checkable formal verification of software, and witness cosigning or other online verification mechanisms as discussed above.»
- «I am not sure which measures would help specifically, but in general, the more transparency the better. Of course, more transparency implies a higher likelihood that problems are uncovered. The larger, harder question is how to deal with the unavoidable problems that will emerge in a way that increases trust.»
- «Publication of the content and results of system inspections by independent experts at election time would probably help as well.»
- «Based on past security systems, the presence some undisclosed verification / intrusion detection system has the potential to be a strong deterrent. Adversaries dislike uncertainties. If they know that some intrusion detection system is running, which may potentially catch an attack or discover the presence of a potential bias, they may shy away from attacking. Thus, my suggestion is to invest in a team that creates additional security measures to catch attackers.»
- «Everything should be made transparent. There cannot be trust if there is no transparency. Even meetings/minutes about the voting system should be made publically accessible. By hiding things, scoping trials, excluding experts, etc, an EMB risks to weaken public confidence in the electoral process.»
- «In my opinion the best thing are independent, trusted experts that have been involved in all the processes and thus can confirm that everything went well – or point out areas of improvement.»

9.3 Introduction to Block 6 on the platform

Block 6 was presented as follows:

The existing regulation mandates the publication of the source code after the 100% certification (VEleS article 7a, 7b). The idea is to gain transparency, a higher level of trust and also grow the number of people who are familiar with the system and online voting technology in general. It should also allow authorities to make early improvements in case errors are found. The publication of the source code of the Swiss Post / Scytl internet voting system combined with the public intrusion test led to global media attention. Several highly critical findings were identified in the source code.

The modalities of the publication of the source code in 2019 were criticized in the media and by politicians alike. Also the experts expressed strong criticism with the source code and its publication in the responses to the questionnaire: The source code and its documentation should to be more accessible, easier to read and easier to audit. The researchers should be able to compile the code, and they should be able to conduct a complete ballot on their own.

There is no agreement on the level and the extent of the proper transparency. Likewise, we don't know yet how to achieve continuing scrutiny.

In their answers to the questionnaire, some experts opt for the adoption of an open source license for all the code developed for the internet voting systems. A larger group of experts just wants to make sure everybody gets access to every part of the code and documentation needed to audit the system from various directions, possibly even on an on-demand base for some parts of the documentation.

The primary goal of the transparency measures is valuable scrutiny for the system. So far, the regulation only expects the publication of the source code of the finished system. For some experts this does not seem to be sufficient. They would also like to see the evolution of the system. There is thus a discussion if scrutiny would profit from regulating the development process in the direction of more openness and transparency. Furthermore, the rights and duties of the public with regard to the usage of the code (license choice) could have a potential to impact the obtained scrutiny as well.

During the years, the Geneva internet voting system published more and more of its source code with an open source license and adopted a more and more open development practice. However, there was very little contribution from 3rd party developers hitherto.

The Swiss Post / Scytl system has been developed in a closed mode so far and the publication led to numerous researchers examining the code after its publication. This may or may not have to do with the public intrusion test carried out in parallel (Also see block 7 for a discussion of the public intrusion test). However, an extensive publication of the source code doesn't guarantee that researchers and third party developers continually participate in the improvement of the code.

There is very little precedent with regulating publication of source code and system documentation in Switzerland. However, the new extension of the Epidemics Act contains the following passage on the Proximity-Tracing App: “The source code and technical specifications of all components of the PT-System are public. The machine-readable programs must have been demonstrably created from this source code.” (Epidemics Act, article 60, section 5e, change of June 19, 2020; inofficial translation of the official German text)

9.4 Questions and summaries of the discussions on the platform

9.4.1 Scope and quality of the publication

We assume the publication includes the source code of the voting systems, of the client code, server code, control components, and of the verifier.

VEleS 7a para. 3 excludes the following items from the obligation of publication: Source code of freely available and regularly updated systems that are in widespread use like OS, DBs, application servers etc. Is this reasonable?

Which parts of the documentation have to be published (please also think about specifications, namely the discussion 1C. Furthermore, would it be acceptable if some documents would only be made available on demand)?

If you think about setting up and conducting a complete ballot on your own: What changes if the test data and parameterization is published or not?

How big is the need to publish audit documents (including certification and penetration tests mandated by the system provider)?

What would be the essential reasons to ask for documentation and configuration of auxiliary systems and peripheral security measures (e.g. web application firewall configuration, processes to obtain administrative super-user access on production servers, system hardening information, patch levels, etc.)?

What would be your requirements if you had to decide if the publication of the source code, documentation, etc. was acceptable in terms of quality, readability and auditability?

Is a registration or an NDA acceptable to access the code and the documentation?

All developed source code and everything needed to understand, to install and to maintain the voting system has to be published. This is meant to allow researchers or auditors to run an election or vote in their own premises with exactly the same reproducible build of the software that is used by the official system.

Any part of the system that is not published will make it harder to find and expose vulnerabilities lurking there.

Being as transparent as possible will help to build confidence towards the voting system. Not necessarily because people will actually review the system themselves, but because they know that other people with the necessary know-how and interest have the possibility to do so.

There is a strong call for the publication of audit documents as well, yet it is not unanimous.

It is acceptable to use widely used standard components such as operating systems, databases, firewalls, etc. without publishing their source code when there is evidence, that these systems have seen a lot of scrutiny elsewhere and the voting system can profit from these efforts. Technical means to enforce the use of a specific, public and widely-used build of said software would be useful.

Publishing detailed patch levels of all sub-systems as well as configuration of peripheral elements of the voting systems such as firewalls would be welcome to several experts. But there are also experts who see this as an information leak that brings an advantage to the attackers, since they are no longer forced to run extensive probes to gather this information. Forcing them to probe can reveal useful information about the attackers in return, so this practice should not be seen as a mere “security by obscurity”.

Any form of NDA will massively reduce the number of qualified researchers looking at the code.

9.4.2 **Development mode as a means for transparency and public scrutiny**

Transparency and public scrutiny are seen as core requirements for an internet voting system and there is a desire to see transparency adopted as a general mindset.

What are the requirements to guarantee continuous interest of qualified security experts in the publicly available code and documentation? (please also see block 7 that touches on this topic)

Does the licensing scheme (closed source vs. open source) influence your answers? What would some kind of open source license change? Which aspects of the system (if any) would an open source license improve?

If we assume an advantage due to an open source license, which parts of the system profit the most from that advantage or where is it really essential to have such a license?

Which level of detail is necessary for researchers to see and understand the development of an internet voting solution? Is there a need to see every commit to the source code, the specifications and the documentation or are consolidated pull requests or tagged versions / minor releases sufficient?

For the experts, it is possible to publish source code with a proprietary license with access to the source code and to receive transparency and public scrutiny in return.

However, an open source system will lead to better results with the goals laid out in the VEleS article 7a, 7b such as transparency, scrutiny, public confidence into the voting system, attracting people and talent to become familiar with the system and voting technology in general. On top, it will also lead to an earlier discovery of bugs, which allows for early adjustments instead of costly repairs relatively late in the development cycle.

A proprietary license with access to the source code can help to reach these goals, but according to the experts, not to the same degree. An open source license is therefore very recommended for an internet voting system.

An open source license would also allow parts of the system to be reused in other open source software, thus making these elements standard components that profit from a wider scrutiny across several projects.

An open source license can also be used to create a community around a system. For one group of experts, this would be very valuable. For the other group of experts, the forming of such a community is not very likely and also not necessarily welcome. These experts are not convinced that a community is able to deliver the quality that is needed for an internet voting system.

9.4.3 **The point in time of the publication**

Assuming that neither public development nor an open source license is going to be enforced for the development of the online voting system, the control components and the verifier(s), what is the point in time when the code should be published?

After preliminary internal tests? Before certification or other forms of mandated examination are performed? After a first round of mandated examinations? After all the mandated inspections are over? Before production use?

With an open source approach, this question would be irrelevant.

If the code is developed in a closed source fashion, then it would be useful to perform a first round of mandated examinations before the publication. That way, less bugs will be discovered by the public and there is still enough time for independent researchers to review the code before it is being used actively.

9.4.4 **Assessing the level of public scrutiny and trust**

The degree of public scrutiny and the trust the society places into a system is hard to measure. Yet they are important pre-conditions for trustworthy internet voting.

How could public scrutiny and public trust be measured and what do you see as a minimum level?

There is no ready methodology to observe and measure the public scrutiny that an internet system has received.

However, there are many informal approaches that can help to shape a picture based on quantitative but also qualitative measurements. It would be useful to look into Science Technology Studies to develop a methodology that helps with an assessment of the situation.

10. Block 7 – Public Intrusion Test and Bug Bounty

10.1 Overview

The subject of block 7 was the bug bounty program. We received a lot of feedback on bug bounties or public intrusion tests in chapter four of the questionnaire. The topic of that chapter was transparency and building of tThe rust. The related questions are 4.1, 4.5, 4.6 and 4.7.

We wanted to have more specific feedback on the scope and of the governance of a bug bounty program and their relation to private penetration tests. We also wanted to know how the infrastructure of the provider could be tested, if the publication of the results could be delayed and what alternative ways there could be to motivate the public to participate at improving the systems.

10.2 Summaries of the answers to the related questions of the questionnaire

10.2.1 **Question 4.1 from the questionnaire**

Find the details in section 9.2.1.

10.2.2 **Question 4.5 from the questionnaire**

Under what conditions should public reactions be discussed?

1. *To whom should feedback be addressed? (System provider, administrations of the Confederation or the cantons, a new entity such as a scientific committee, common platform of multiple actors, etc.)*
2. *Which entities should be involved in the discussion?*

Summary from 15 of 15 responses:

There is a strong and diverse call for a committee involving various experts responding to the public reactions. Which organization should appoint the committee is not discussed in detail. The question, namely the first part of the question, was read differently by different experts. So the responses stretch from public reactions in the form of software weaknesses identified, or more general reactions by the public. Therefore the responses are also quite diverse and the committee is either very operational with discussing every individual bug report - or it is a more general advisory board that discusses reactions, assesses problems and addresses the wider public with documents.

Also, it is not entirely clear if said committee should give advice or have the power to stop an online voting system. What is clear is that all responses have to be transparent and made public; possibly after a certain delay. Minority opinions do not discuss a committee but rather emphasize the Federal Chancellery with regard to a technical lead position, they think reactions should generally be public by default or they read the question completely differently.

Some individual interesting ideas:

Feedback has to go to the maintainer of the system, the risk owners (election officials of the cantons), and as a matter of course, to the Confederation that authorizes the system.

A scientific committee is an interesting idea, but I would only give it advisory capacity. It is important not to dilute the political responsibility for allowing/denying a system to be in operation. Regular events can be held where questions from the public are discussed. Anyone can attend and participate in the discussion under certain rules.

A new unit (Public Private Partnership?) could be founded for the topic internet voting. This could also be used for other topics that affect all citizens (E-ID).

It is worth exploring how a citizen science initiative could be involved in the process.

In case only minor technical flaws are detected, these can be addressed in a revision of the software. If fundamental flaws are discovered, the project stakeholders may need to take strategic decisions.

The Federal Chancellery should then also be responsible for evaluating the comments, and take appropriate action, in the worst case, deauthorizing the use of a particular internet voting system. Most importantly, the Chancellery must be able to provide convincing answers to people crying wolf that the system doesn’t work. This means that the Chancellery should also be able to defend against unfounded accusations. Once the Chancellery has determined the right course of action, it will take next steps and involve the vendor, police, or academics.

For each major release I suggest holding a press conference and then a technical discussion with independent experts.

I feel that this is a question for Swiss administrators. My main suggestion is to think about the incentives of all people involved.

10.2.3 **Question 4.6 from the questionnaire**

Should the system providers publish existing / fixed security breaches? Through which channels? When?

Summary from 14 of 15 responses:

There is a very clear majority insisting on full transparency of every security-related fix. Most experts opt to use a standard public development platform such as github / gitlab and use that as a publication channel.

Most experts agree that security fixes are best published when the system has been patched (referring to an evolving industry best practice of a maximum delay of 3 months after the report). However several experts mention that this may be impractical in the internet voting context.

A few experts discuss the problems with security vulnerabilities reported during an ongoing vote. Another one, representing a minority position, opts to delegate this question to the committee that ought to be formed.

Individual interesting idea: Establishment of guidelines by the new body.

10.2.4 Question 4.7 from the questionnaire

Can security benefit from a PIT the way it was performed? Or a different form of bug bounty? Can public trust benefit from a PIT / bug bounty? Consider the scope restrictions of the PIT (e.g. social engineering, d-DOS, only external attacks) in the federal and cantonal requirements regarding public intrusion tests [5]. Should the restrictions be relaxed? Are there alternative, more appropriate ways to submit the untested elements to public scrutiny? What incentives should be provided to ensure participation?

Summary from 14 of 15 responses:

There is agreement that penetration tests and bug bounty programs complement each other. Mandated Pen-Test first, Bug bounty afterwards. There is a strong group of experts that see a lot of value in making it a permanent program, also because it would then be part of the normal security posture of the service and not a one-time happening.

A qualified majority of the experts think that restrictions should be dropped, maybe completely. A minority sees a bug bounty extending to DDos and social engineering as unpractical and resulting in harassment of employees of the system provider.

Alternative tests that also cover these areas (DDos and social engineering are mentioned by many experts) can be done in the form of mandated penetration tests with subsequent publication of the results (a credible Pen-Test report that covers DDoS in detail removes the pressure to include DDoS in the bug bounty program). A second option is to add smaller focused programs that include qualified bug bounty hunters (maybe some who have proven their quality in the main program) and invite them to conduct DDoS and social engineering tests in a controlled environment.

Two experts opt to give bounty hunters access to the backend systems as part of such a focused test. A hackathon has been mentioned as well.

One expert states that the bounties must be much larger. He also proposes an insurance model that could keep the budget relatively low but still guarantee large payouts (“Hydra framework”), thus setting the incentives in a way that all findings are reported.

Question 4.1 was also answered with the PIT in mind by many experts. One wants to put the Federal Chancellery in charge in the bug bounty, the other experts did not touch on this question. The other responses in 4.1 confirm the conclusions drawn in this section of 4.7.

10.3 Introduction to Block 7 on the platform

Block 7 was presented as follows:

10.3.1 Introduction

The main purpose of the bug bounty program is to motivate public scrutiny of the system. This goal is to obtain a system that is more secure, more transparent and more trusted. It also contributes to building a community of competent people who can scrutinize and evaluate the system.

The importance of scrutiny was already discussed in block 1 as means of making the crypto more difficult to break.

The Public Intrusion Test (PIT) of 2019 was a bounty program which put the spotlight on a specific scope for a limited time. It was not mandated by the VEleS and was run as a pilot project.

The source code was published to satisfy the requirements for transparency and scrutiny of the VEleS but was not in the scope of the 2019 PIT.

Mandated intrusion tests carried out by specialized companies are a complement. They are explicitly required by VEleS for Internet facing systems (scope 5, protection against infiltration) and implicitly for the back-end systems in scope 3 (ISO 27001 audit of infrastructure).

There are very different scopes that can be tested. One is made up of all standard infrastructure that supports the voting application: firewalls, web servers, databases, logging servers, DNS and so on. Another one is the tailor made software that implements the protocol: client side software in the browser, software on the control components, the voting server, the printing office, and others. Finally, documentation, physical security or social engineering can also be inspected or tested.

There are important differences between the scopes. For example, it only makes sense to test the supporting infrastructure that is actually provided by the voting service provider, as the security mainly depends on the configuration of the infrastructure. The security of the software or documentation can however be tested in virtual systems run by the testers themselves, as long as they run the same software with the same software configuration.

While it is possible to run a public intrusion test against the Internet facing infrastructure of the voting service provider it seems unrealistic to expect the provider to open up its backend infrastructure to the greater public for testing.

There are more scopes that cannot easily be tested by the public. Imagine the chaos if the public was invited to test denial of service attacks on the actual infrastructure of the service provider. The same holds for social engineering where real employees or voters would be attacked by the public. It also seems difficult to let the public attack the physical security of the system.

10.3.2 **Proposition of scopes and types of tests**

Based on the responses of the questionnaire and the above statements, it seems that the following would be a good way to organize tests:

- The time and scope limited PITs would become a continuous bug bounty program with a broader scope. Most tests could be run on local copies of the system. This implies that running a local copy of the system should be facilitated by the provider, e.g. by providing preconfigured containers or virtual machines. The testers could then for example also test the security of back-end elements like the control components. The internet facing infrastructure would be tested on a testing infrastructure set up in the same way as the productive infrastructure, although with lower requirements for availability. The specification of the system, the code or the documentation of processes would also be part of the scope of the bug bounty.
- The back-end infrastructure of the provider, denial of service attacks, social engineering or physical intrusion would not be in the scope of the public bug bounty program. They are part of the mandated private intrusion tests.

10.3.3 **Responsibilities**

In the responses of the questionnaire several experts saw the federal administration or an independent entity running the bug bounty program.

Bug bounties are typically run by the provider of the tested systems, for practical reasons.

This contradiction could be solved the following way:

- The federal administration defines the modalities of the bug bounty program (scope, terms and conditions, bounties, …) with the service provider and the Cantons.
- The provider operates the program. This includes assessing the reported bugs, paying the bounties and other operational tasks.
- The federal chancellery assesses the operation of the bug bounty program on a regular basis.

10.4 Questions and summaries of the discussions on the platform

The following are the questions that were asked in block 7 and the summaries of the dialogue that ensued, validated by the experts.

10.4.1 **Evolution of the PIT**

Do you agree that the 2019-PIT should evolve into an ongoing bug bounty program with a larger scope?

The scope would include everything that can be tested continuously by the public without creating inconveniences for the service provider (for example the back-end infrastructure). This would typically include Internet facing infrastructure reserved for testing, the software, the specification of the software and all security relevant documents (specification of the system, processes).

Do you see value in including a test infrastructure (reverse proxies, firewalls, web servers, …) set up identically to the production infrastructure into the scope of the bug bounty program?

Since the purpose of this test infrastructure is to test the front-end security, the back-end could not be identical to production. For example, it could not have the same level of redundancy. Would this reduce the value of the tests?

Should the value of the bounties be related to other bug bounty programs (up to tens or hundreds of thousands of dollars) or to the price paid for exploits on the public market (up to hundred thousands or millions of dollars)?

Do you agree that the bug bounty program should be operated by the service provider?

Do you agree that the federal administration should be in charge of defining the goals and modalities of the program and of assessing the program?

The publication of the code, mandated testing and a Bug Bounty Program are all seen as complementary. An ongoing Bug Bounty Program replacing the former Public Intrusion Test is seen as a welcome evolution.

The tests executed by the independent bounty hunters are useful to check the trust assumptions of the system and they support a defense in depth even in a world where everything is formally proven.

The target of the attack tests should not be the system used for the voting itself. Instead, the security researchers should have the options to attack a separate copy of the electronic voting system as well as a deployable version of the system that they download and run on their own premises.

The electronic voting system is divided in core elements and peripheral sub-systems. An attackable copy of the production system has to consist of the core elements and all the necessary peripheral sub-systems. It seems reasonable to concentrate on the voting here and skip auxiliary functionality (e.g. high availability). But there is also a danger in an attackable copy with a simpler setup, as this could lead to false claims of a successful attack against the bug bounty system. Such claims could be hard to refute.

There might also be reasons to pause the operation of this attackable copy during active voting / election periods. Alternatively, the support level of this target system could be reduced during these periods.

There are going to be limits as to which access is granted on the attackable system in order to test potential weaknesses against insider or social engineering attacks. Providing a ready to deploy version of the voting system can be a useful alternative that allows to test for these weaknesses. It has to be accompanied by a complete documentation of the used hard- and software as well as organisation units, their abilities, tasks, duties and standard operational procedures. This would allow a researcher to design and test an attack on his or her own premises while still shaping it according to the operation of the production voting system. Again, false claims of successful attacks against this local copy could be hard to refute.

The disclosure policy for the vulnerabilities should be covered by a policy that also encompasses other vulnerabilities of the system. This policy has to be defined by the Federal Administration.

There is no clear opinion whether the system provider should run the Bug Bounty Program itself or a third party partner; possibly commissioned by the Federal Administration. However, it is very important that the Federal Administration defines the goals and modalities and oversees the program.

The size of the bounties should be adaptive and grow with the maturity of the system.

10.4.2 **Private penetration tests and infrastructure**

Do you agree that independently of the bug bounty program (PIT), the Internet facing and internal infrastructure should be subjected to private penetration tests regularly and at each substantial modification? These tests would include controlled DDOS attempts.

Running Penetration Tests on a regular base is seen as standard industry practice that should be applied here as well. It adds a targeted method to the mix and complements a continuous Bug Bounty Program in order to get a more accurate picture of the security posture of a system.

If done right, penetration tests also allow to examine aspects of the system defenses that can hardly be tested in a bug bounty program (e.g. DDoS, social engineering, etc.).

All the reports as well as the responses of the system provider should be made public.

10.4.3 **Delaying the publication of the bugs**

Do you agree that the disclosure of a bug could be delayed by a limited time (e.g 90 days), while it is being assessed or while there is an ongoing vote that is vulnerable to the bug?

It is important to have a well defined and transparent disclosure process. The process and the individual decision about a publication should be based on industry best practices.

Yet, the high relevance of the electronic voting system and the clocked production use of the system make it a special case.

There are two conflicting desires that have to be balanced:

- Potential attackers should not get information about an active before it has been fixed.
- The public at large should get the information about a vulnerability so it can react accordingly.

Depending on the point in time, previous or scheduled use of the system (domestically or abroad), the severity of the vulnerability and the complexity of the remediation, the response should be different.

10.4.4 **Other ways of public participation**

Do you see other ways of motivating the general public to learn about and improve the security of the system?

Here are a few examples to start the discussion:

-Run a hackathon to develop voting clients that facilitate testing of the servers

-Run a hackathon to write the main parts of the crypto code of the control components in a different language

-Run a contest for improving the current solution (e.g. for a more secure printing office)

-Create a contest for social engineering plots and reward the best ones according to some defined criteria (e.g. most impact for least number of people fooled).

Specialized technical events like a hackathon can introduce more people to the electronic voting systems or raise the interest in the technical community.

For the public at large a different approach is needed. Finding a good mode or channel in this regard is very challenging.

Citizen science projects, or extended workshops involving a random representative selection of participants comprising a citizens' assembly, could help address wider audiences directly and/or through the media.

It would also be interesting to make the voting system tangible for ordinary people or to find roles to participate in the electronic voting process for them like the vote counters on a municipality level. This is meant to bring democratic accountability and ultimately contribute to the public confidence into the system.

11. Block 8 – Risk Management and Individual Risks

11.1 Overview

The topic of block 8 was risk management and individual risks. It relates to questions 2.2.1, 2.3.1, 2.3.2 and 2.3.3 as well as questions 6.1, 6.2, 6.5, 6.6, 6.7 and 6.8 of the questionnaire. Even though questions from chapter 2 of the questionnaire were not directly in the scope of the risk management described in chapter 6, some of them are related to new risks and mitigations that could be considered in future risk assessments. The mentioned questions from chapter 6 of the questionnaire (Risk management and action plan) were considered due to their lack of consensus calling for more discussion.

11.2 Summaries of the answers to the related questions of the questionnaire

11.2.1 Question 2.2.1 from the questionnaire

Are there any threats you feel are not covered by the basic threats in chapter 3.1 of the VEleS annex?

Summary from 13 of 15 responses:

The answers do not contain any significant contradictions.

- Threats to be considered
  - Accidents / human mistakes: 4/13
  - Social-engineering: 6/13
  - Planting trojan horse or backdoor: 1/13
  - Alleged attacks or malfunctions, framing attacks mentioned: 3/13
- Threat agents to be considered
  - State attackers: 3/13
- (Security) objectives to be considered
  - Trust in voting result: 1/13
  - User-friendliness: 2/13
- Weaknesses / assets to be considered
  - Zero-day vulnerabilities: 1/13
  - Printing office: 4/13
  - Postal service (for delivering voting material): 2/13
  - Ballot paper with regard to privacy (same as 3.1.4 but additionally applied to the objective “Protection of voting secrecy and non-disclosure of early provisional results”): 1/13
- Risks to be considered (defined by objectives, threats, threat agents, and weaknesses/assets)
  - Criminal organization or foreign adversary infiltrates the trusted printing office (physically, socially, or electronically) to exfiltrate the codes on mailed voter cards and compromise cast-as-intended security.
  - Criminal organization or foreign adversary infiltrates the trusted postal service (physically, socially, or electronically) to misdirect some percentage of ballots from selected neighborhoods to an alternate address where they are held or destroyed.
  - Criminal organization or foreign adversary offers a potentially-large number of voters money or cryptocurrency in exchange for installing malware or spyware on their devices, which verify that the voters cast votes the way the adversary prefers (largescale electronic vote-buying or coercion). An adversary could even carry out such an attack with almost complete anonymity with appropriate use of cryptocurrency and smart contract technologies.
  - Compromised web server or App store serves a compromised version of the voting Web app and/or native apps to users.

  - Compromised certificate authority, code-signing certificate, or developer signing keys used by an adversary to produce correctly-signed but compromised versions of the voting Web app and/or native app to distribute to users.
  - Network denial-of-service attacker prevents (targeted) users from casting votes electronically, forcing them to fall back to the mail or in-person process, in the expectation that many targeted users will give up and not vote at all due to the inconvenience. A network attacker who can disrupt the vote-casting process at the “critical moment” after the codes have been received but before the electronic vote has been confirmed would be particularly effective, since the voter cannot try again in this case. 2/13
- Further statements
  - Insider threats are by far the most underrated threat internet voting systems. They are covered in the list, however the related counter-measures deserve more insistence (separation of duties, development process).
  - Threats are difficult to identify generically, they must be identified on the grounds of a given system. Name threats based on basic components and data.
  - The list of threat-agents should not be limiting. All possible adversaries need to be considered (eavesdroppers, criminals, activists, insiders, parties, nation states, …)
  - Deserves more focus: Network-based attacks (preventing availability), PKI aspects, human aspects of printing and system administrators (especially also the formal verification of their processes), ensuring that voter has correct information or software obtained on-line (in case of OS vulnerability, TLS PKI vulnerability enabling MITM attack).

11.2.2 Question 2.3.1 from the questionnaire

Individual verifiability should allow voters to detect manipulations and, in case they have doubts, to choose another voting channel. Individual verifiability is also meant to allow the cantons to detect systematic fraud. This however hinges on the number of voters who check their return-codes and report observations that could imply a manipulation (i.e. a return-code not being displayed or being displayed incorrectly). Does it seem reasonable to believe that a sufficient number of voters will be capable of checking the codes according to instructions delivered in the voting material and to inform the administration in charge in case a code is not displayed or displayed incorrectly? What measures could be taken in order to maximize the number of voters who check their codes?

Summary from 15 of 15 responses:

- Yes, it seems reasonable to believe enough voters will follow the procedure. 6/15
- Some of these experts relate their conclusion to one or multiple of the following ideas or recommendations from the list further down:
  - P1: 2/6
  - P2: 2/6
  - P3: 3/6
  - P6: 1/6
  - P13: 1/6
- Pessimistic, but P3 could help: 1/15
- Some experts do not give an assessment. One expert is skeptical, whether putting this burden on the voters is reasonable at all: 8/15
  - P4: 2/8
  - P5: 3/8
  - P7-12: 1/8
- This list summarizes stated ideas or recommendations, independently of any assessment on their effect:
  - P1: The voting material has to be clear: 2/15

- P2: Online user experience needs to be appropriate, one mentions simplicity: 3/15
- P3: Ask the voters for an input that they can only give when checking the codes (the following examples were mentioned: The voting card could have a confirmation code per return-code, or repeat the code through a different channel, or display multiple codes to the voter and ask him to select the correct one), whereas two experts also acknowledge the extra burden on the voters: 4/15
- P4: Awareness campaigns: 2/15
- P5: Research / studies on the voters' behavior are necessary to answer this question: 4/15
- P6: Be aware that voters might check the first few codes on the top. Voters with wrong codes might believe they did something wrong and therefore be hesitant to expose themselves by calling the administration: 1/15
- P7: The notion of "sufficient needs" needs to relate to parameters that are likely to be unknown before the end of a vote: the margin of the actual outcome, the level of public distrust, issued complaints,..: 1/15
- P8: The procedure must in any case entail reporting wrong codes to the administration. Just instructing voters to vote by mail when seeing wrong codes beforehand is insufficient as to detect systematic fraud: 1/15
- P9: The rate of voters who check their codes is one thing. However, even a high rate brings no added value in case an attacker can predict who performs the check and who does not: 1/15
- P10: The entity who receives the reports of wrong codes needs to be trustworthy, i.e. should not risk being suspected of releasing falsified statistics on how many voters usually check/report and how many claims have been made: 1/15
- P11: Measures that would allow voters to seek assistance at verifying (i.e. while respecting the secrecy of the vote) could be explored, Helios, Scantegrity II or Prêt à Voter are references have such features: 1/15
- P12: Checking public boards might motivate more voters to verify, given that the experience would be more natural: 1/15
- P13: Switzerland has a substantial number of individuals who are concerned about internet voting fraud, and would thus perform active verification: 1/15

11.2.3 Question 2.3.2 from the questionnaire

The voters are also advised to check the TLS-Fingerprint in their browser. This aims at enabling them to detect connections to a wrong server. What measures could be taken in order to maximize the number of voters who check the fingerprint?

Summary from 14 of 15 responses:

- Advising voters to check the TLS-fingerprint is a good approach to prevent PKI-level attacks, given good instructions: 2/14
- Advising voters to check the TLS-fingerprint is unlikely to bring much benefit. Two experts recommend to advise voters to enter the URL correctly and check the pad-lock symbol. Another expert recommends using SCION to prevent corresponding attacks. One expert notes that the fingerprint or browsers might change on too short notice: 7/14
- A group of experts does not give an assessment. One expert notes that studies are needed in order to estimate the voters' behavior. Another one proposes to ask voters to identify the correct fingerprint between two similar ones. One expert states that all one can do is ask the voters to verify. Another one underlines the value of instructing voters to only use TLS connections (pad-lock symbol). The same expert advises certificate-pinning, HSTS as well as ensuring that all official websites that link to the system use TLS-connections. One expert is skeptical, whether putting this burden on the voters is reasonable at all. Finally, one expert proposes using external devices to check the domain: 5/14

11.2.4 **Question 2.3.3 from the questionnaire**

The voters must be given the possibility to verify that their client application is correct, i.e. that the correct encryption key is applied (the effectiveness of individual verifiability does not hinge on the correctness of the client application). This is to address the case where the client application is tampered with on the server side.

Which measures could meet this requirement in an effective way and what are the downsides? What measures could be taken in order to maximize the number of voters who check that they are running the correct client application?

Summary from 9 of 15 responses:

- Advising voters to check the correctness of the client application software is a good approach, given good instructions. One expert highlights that the solution could involve comparing fingerprints and checking signatures of the downloaded application against a reference. In this case, it would be important to make sure that a group is involved in signing, as to prevent malicious signatures: 2/9
- Advising voters to check the correctness of the client application is unlikely to bring much benefit. One expert states that the solution would be too difficult: 1/9
- A group of experts do not give an assessment. One expert states that a multitude of applications could be proposed, each one endorsed by different organizations. Another expert states that little can be done apart from installing browser extensions/plug-ins or external devices. Yet another one says voters should not be required to install any heavy weight applications. One expert notes that studies are needed in order to estimate the voters' behavior. Information campaigns and asking voters to verify is repeated from the statements to the previous questions. One expert is skeptical, whether putting this burden on the voters is reasonable at all. Finally, one expert states that trusted execution environments would be needed to obtain strong properties in this space (SGX or TrustZone on ARM devices): 6/9

11.2.5 **Question 6.1 from the questionnaire**

What should a continuous risk assessment process for internet voting consist of? How often should risk analyses be updated? Based on which input? Who should provide them? At which depth should risks be analyzed?

Summary from 11 of 15 responses:

- All experts agree on a continuous / regular (yearly or bi-yearly) assessment of the risks.
- A third think that the risk assessment has to be updated before every ballot as threats might change depending on the content of the ballot.
- A quarter think that it should be updated with (major) system change.
- Though there is no consensus about it, input for this assessment could come from:
  - Science
  - (Security) events
  - Federal entities
  - Public
  - Other countries
- A fifth recommend that a specific (scientific) body created to accompany internet voting activities takes the lead for the risk assessment.
- Some individual interesting ideas:
  - Publish the risks on a webpage to allow the public to review it and submit new ideas
  - Having an independent authority providing the risk analysis and then having the risk assessment done by the system provider and cantons based on it.

    - Consider conflict of interest when assigning responsibilities for doing the risk assessment

11.2.6 Question 6.2 from the questionnaire

What are the benefits and downsides of publishing the (dynamic) risk assessment?

Summary from 11 of 15 responses:

- Benefits:
    - 6/10 Trust-building
    - 3/10 Increased transparency
    - 3/10 Public involvement
    - 1/10 Allow wise decision (face the truth)
- Downsides
    - 6/10 Reveal weaknesses
    - 3/10 Brings controversy
- Half of the experts declare that they are in favor of publishing the risk assessment as long as it does not endanger the vote.
- Some individual interesting ideas:
    - The risk of revealing weaknesses is unfounded, as the adversary already knows about them.
    - Make a distinction between operational risk assessment and overall risk assessment of the system, which should be published.

11.2.7 Question 6.5 from the questionnaire

To what extent can risk analyses be better aligned with standard methodologies? Which one would you recommend?

Summary from 5 of 15 responses:

- Standard methodologies:
    - 2/5 ISO 27005
    - 2/5 NIST standards
    - 1/5 OCTAVE Allegro
    - 1/5 BSI Grundschutzhandbuch
    - 1/5 EU Risk assessment methodologies for Critical Infrastructure Protection
- Two experts mentioned that the benefit in using standards comes from having a structured and comprehensive way of analyzing assets, threats, vulnerabilities and risks.
- Some individual interesting ideas:
    - The quality of the risk analysis depends on the quality of those performing it
    - Systematic testing and review is necessary

11.2.8 Question 6.6 from the questionnaire

Who can / should support the public administrations of the Confederation or the cantons in its risk assessment (threat modelling, risks identification, etc.)? What could the role of science be?

Summary from 11 of 15 responses:

- A majority see a committee of experts as a good body to support the risk assessment process. However, the composition of such a body differs across experts:
  - Science representatives (research groups, teacher, Risk Center ETH)
  - Consultants (security, usability, software, statistics, economy, insurance)
  - Federal agencies
  - Election officials from cantons
- A third propose to assign different roles to the different kinds of experts. Scientist should take care of conceptual and technical aspects while consultants should take care of the concrete risk assessments, possibly with the help of federal agencies for threats and risks identification.
- Some individual interesting ideas:
  - The risk assessment process looks like a relevant subject for PhD theses, that could be run in collaboration with the Confederation
  - The problem does not lay in the identification or assessment of risks, but in the adequate definition of the scope, and the correct identification and effective implementation of adequate controls

11.2.9 Question 6.7 from the questionnaire

Some risks can only be assessed in knowledge of implementation details. At the same time, the public administration carries the risks that are in scope here. How should responsibilities be defined to ensure that the relevant aspects are dealt with effectively and credibly? Can the handling of certain issues be outsourced? To whom?

Summary from 6 of 15 responses:

Three solutions emerge here though no consensus exists:

- The Confederation, in collaboration with the different stakeholders, issues a core security concept that inventories threats, proposes mitigation controls in a comprehensive manner and defines the responsibilities for implementation.
- As this task needs sovereign capabilities, the cantons are responsible for it. They can however get help from outside if they lack the competencies. (status quo)
- The committee of experts is in charge

11.2.10 Question 6.8 from the questionnaire

Would it be meaningful to have a risk analysis from the canton focusing on the canton's processes and infrastructure and a separate analysis from the system provider focusing on its processes and infrastructure? How to ensure comprehensiveness and consistency in this set up?

Summary from 8 of 15 responses:

- Half of the experts says that the splitting could be done but in a different way. The idea would be to have a group (Confederation, cantons, providers, external experts) defining a neutral risk concept. The practical risk assessment, by the canton and/or operator, would then focus on residual risks specific to their organization and the implementation of the controls and its effectiveness.
- A quarter of the experts mention that splitting the risk analysis could pose problems because, in general, security does not compose well. The risks of two systems taken separately do not cover the interactions between the systems. Some kind of additional analysis would be needed to identify new risks that come from these interactions.

- Some individual interesting ideas:
    - Only one risk analysis by the cantons
    - Risks for each canton should be considered in one forum

Questions not covered in block 8

The following questions have not explicitly been further elaborated within the current block. The may however have been indirectly discussed in other blocks.

- Question 2.3.4 on Quantum computers
- Question 2.3.5 on voters' platform hardening
- Question 2.3.6 on vote buying and coercion
- Question 6.3 on supply chain risks
- Question 6.4 on criteria for prioritizing action plan
- Question 6.9 on Octave Allegro proof of concept

11.2.11 **Question 2.3.4 from the questionnaire**

How, if at all, do you think the developments in the area of quantum-computing should be addressed? Who / which organizations will be able to use these computers if anyone ever? When do you expect quantum computing to become relevant? Assume that encryption and soundness of proofs must hold for months (the latter with regard to premature results being known), the secrecy of the vote (which person voted how) should at least last for decades. Regarding the secrecy of the vote you may assume that no personal voter data (i.e. names) is transmitted through the internet.

Summary from 8 of 15 responses:

- Most experts do not see a threat to integrity in the coming years. However, privacy deserves attention, given that data could be collected today and evaluated later, as four experts highlight. One expert notes that one must always assume that the encrypted votes might leak. Two experts highlight that breaking voter privacy would require to establish a link to the voter. This could be managed by withholding the personal data of the voters, that however would have to be done in an effective way: 6/8
- It is unclear whether quantum computers will exist in the near future or if they already exist. Therefore, it is not possible to determine when a post-quantum cryptographic redesign is necessary: 1/8
- Cryptographic schemes that offer resilience against quantum computers are being researched. These developments should be taken into consideration for the future. Three experts imply either that they are not sufficiently matured yet or complicated to implement. Three experts seem to recommend to first get it right with conventional cryptography: 6/8

11.2.12 **Question 2.3.5 from the questionnaire**

The voters' platforms hold the votes in plaintext. In some cases in the past, voters were instructed to use a platform they trust to cast their vote. To what degree can voters influence their level of protection from malware, e.g. by following the guidelines from MELANI?

Summary from 13 of 15 responses:

- These measures make sense. However, almost all experts point out that they do not eliminate the risks related to corrupted voter platforms. Particularly, resourceful attackers must be considered to be capable of circumventing these measures. Two experts explicitly doubt that many will follow the guidelines. One expert sees general public information campaigns on cybersecurity more promising than election-related instructions. One expert highlights that trusted execution environments such as SGX would offer security even in the presence of malware. Another expert highlights that code-voting (each choice on the ballot is expressed by entering a distinct code) would remedy the problem, at the cost of usability: 12/13

- One expert did not assess the MELANI-guidelines, but brought to attention that rates of malware infection can be high: 1/13

11.2.13 Question 2.3.6 from the questionnaire

*Despite postal voting being used by more than 90% of the voters, vote-buying and coercion are not considered to be a concern in Switzerland. Do you think internet voting would in practice be likely to increase the number of successful attempts of vote-buying or coercion?***Summary from 15 of 15 responses:**

- The risks are not or not much higher with internet voting. One expert clearly highlights the scalability of vote-buying in the absence of a resilient voting protocol. However, he locates the parameters as to whether vote-buying happens rather in societal than technical aspects: 7/15
- The risk is higher with internet voting, mainly due to anonymity and increased scalable technical feasibility: 8/15

11.2.14 Question 6.3 from the questionnaire

How could supply chain risks be properly handled by the cantons towards the system provider and by the system provider toward its contractors?

Summary from 6 of 15 responses:

Half of the experts think it is a tricky problem that the cantons might not be able to handle.

Most of the proposed solutions fall under ISO 27002 controls (contractual clauses, penalties, criteria for choosing a provider, careful planning and verification of the deliveries).

Some individual interesting ideas:

- Cantons build and operate their own system
- Modularize system with different providers
- Redundant infrastructure through national datacenters
- "chip-x-ray" system developed by PSI for checking hardware

11.2.15 Question 6.4 from the questionnaire

Which criteria for prioritizing action plan measures are relevant and in what order (including significance, urgency, feasibility, electorate impacted)?

Summary from 7 of 15 responses:

- Criticality 3/7
- Urgency 3/7
- Feasibility 3/7
- Effort (Cost) / Benefit (Risk reduction, effect on trust) 2/7
- Impact 2/7
- Some individual interesting ideas:
  - Prioritization is useless, each and every action has to be implemented
  - One mentioned minimization of negative impact on public trust as objective, as another one mentioned maximization of public trust

11.2.16 **Question 6.9 from the questionnaire**

Attached to this questionnaire is a proof of concept (as an illustration) of a risk analysis based on the Octave Allegro methodology. Would this methodology be appropriate to handle the risks from the cantons' / system provider's point of view? Do you see any weakness / strong points in this methodology?

Summary from 5 of 15 responses:

- A bit more than a half are in favor of the methodology, a bit less than a half are against it.
- Strengths:
  - Recognized as a straightforward, pragmatic and popular methodology for IT
  - Risk management
  - Clear and objective
  - Lightweight and easy to use
  - Treats technology as well as people
- Weaknesses:
  - Absence of links between threats and controls
  - Too lightweight
  - Will likely miss subtleties arising from the particular adversary model and problems in the actual implementation or configuration
  - Scope rather different from that of an internet voting platform as it is designed to assess information security risks within a large organization
- Some individual interesting ideas:
  - Develop an individual internet voting risk assessment tool
  - Use a catalogue of generic threats mapped with 27002 controls
  - Methodology is not critical in the process of securing a system

11.3 Introduction to Block 8 on the platform

Block 8 was presented as follows:

Risks management is a central element for the security of the internet voting system and its processes. As per art. 3 VEleS, by the means of a risk assessment, the canton must document in detailed and understandable terms that any security risks are within adequate limits. To do so, they are to address the security objectives described and by no mean minimising risks must be dependent on keeping security-relevant information on the system and its operation secret.

11.3.1 **Organization**

The necessary knowledge of the system, conflicts of interest and comprehensive and effective risk coverage are the issues at the centre of the organisation of risk analysis in the complex context of internet voting in Switzerland. Consider the following actors:

Actor	Role	Reputation risks	Financial risks	Technical risks	Legal risks
Confederation	Legal requirements Authorisation	Yes	No	No	Yes
Cantons	Implementer	Yes	Yes	Yes	Yes
System provider	Cantons' contractor	Yes	Yes	Yes	Yes

As each of these actors has a different perspective on risks, each of them has to manage the risks it owns. Cantons and system provider also share most of the critical information assets but have separate infrastructures, thus weaknesses and strengths.

In order to solve this puzzle, several solutions emerged from the answers to the questionnaire, none with a majority:

1. The Confederation, in collaboration with the different stakeholders (which may also include representatives of the science, security experts, a committee of experts etc.), issues a core security concept that inventories threats, proposes mitigation controls in a comprehensive manner and defines the responsibilities for implementation.
2. As this task needs sovereign capabilities, the cantons are responsible for it. They can however get help from outside if they lack the competencies. (status quo)
3. A committee of experts fully independent from the confederation and cantons is created and in charge of doing the risk analysis.

It is quite difficult to imagine a committee of experts that is fully independent from the confederation and cantons. The question of who will define its mandate and effectively mandate it arises, who if not the Confederation or the Cantons. The second solution could be perceived as the same as the first one. However, they differ in the fact that, in the second one, the cantons are responsible for the whole risk management but in the first one, the different stakeholders agree upon responsibilities. In the current context, the first solution appears to be a better match.

11.3.2 Concept

A possible implementation of the solution 1 would define the responsibilities as follows:

- Confederation
  - Establishing a core security concept in collaboration with the different stakeholders
  - Establishing its own risk analysis and action plan
  - Reviewing submitted risk analyses and action plans
  - Monitoring its risks
- Cantons
  - Establishing their own risk analysis and action plan
  - Reviewing the system provider's risk analysis and action plan
  - Monitoring their own risks
- System provider
  - Establishing its own risk analysis and action plan
  - Monitoring its own risks

The process of risks management could be schematised according to the two following figures:

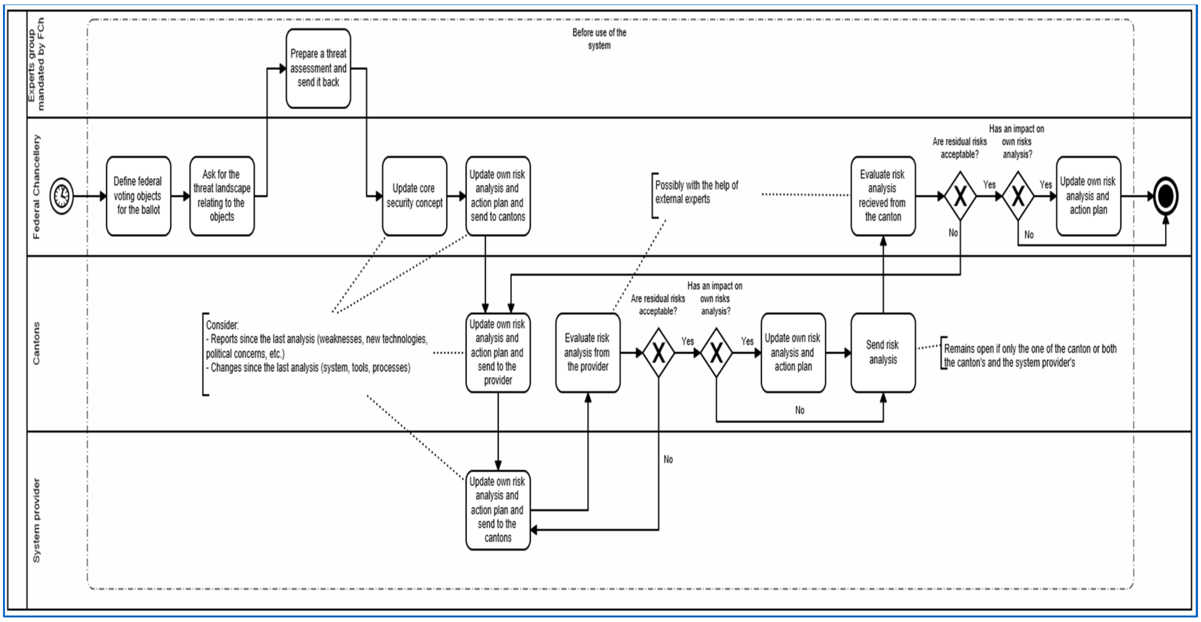
Experts group mandated by FCh
Federal Chancellery
Cantons
System provider
Before use of the system
Prepare a threat assessment and send it back
Define federal voting objects for the ballot
Ask for the threat landscape relating to the objects
Update core security concept
Update own risk analysis and action plan and send to cantons
Possibly with the help of external experts
Evaluate risk analysis recieved from the canton
Are residual risks acceptable?
Has an impact on own risks analysis?
Yes
Yes
No
No
Update own risk analysis and action plan
Consider:
- Reports since the last analysis (weaknesses, new technologies, political concerns, etc.)
- Changes since the last analysis (system, tools, processes)
Update own risk analysis and action plan and send to the provider
Evaluate risk analysis from the provider
Are residual risks acceptable?
Has an impact on own risks analysis?
Yes
Yes
Update own risk analysis and action plan
Send risk analysis
No
Remains open if only the one of the canton or both the canton's and the system provider's
No
Update own risk analysis and action plan and send to the cantons

Figure 3 - *Before using the system (before voting period)*

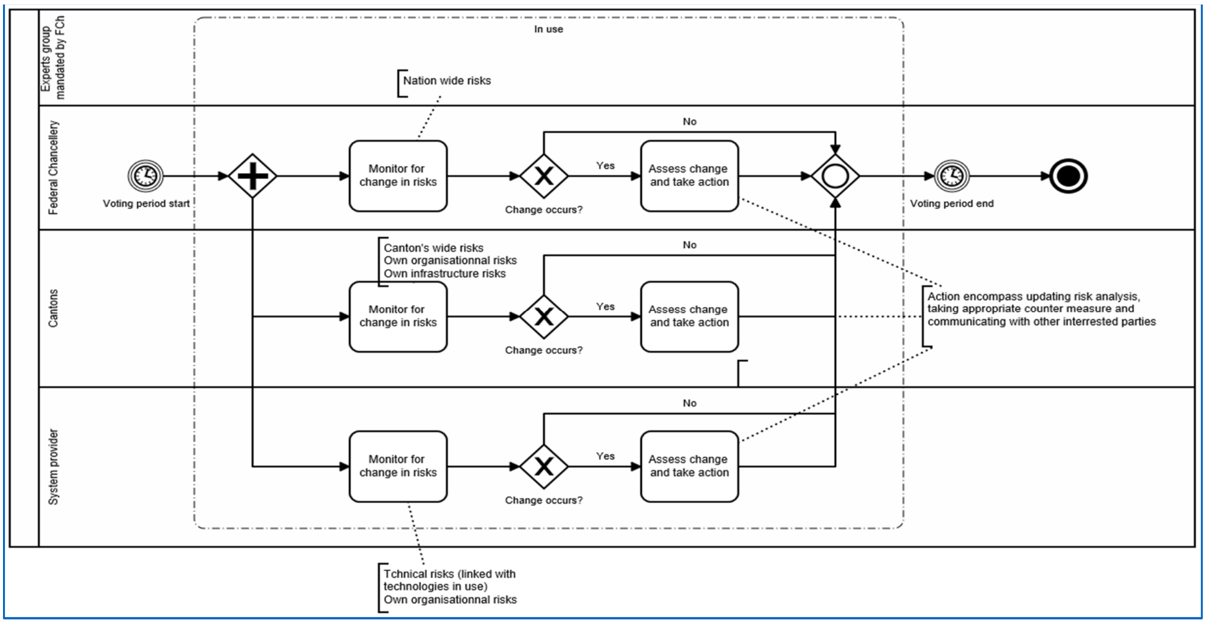
Experts group mandated by FCh
Federal Chancellery
Cantons
System provider
In use
Nation wide risks
Voting period start
Monitor for change in risks
Change occurs?
Yes
No
Assess change and take action
Voting period end
Canton's wide risks
Own organisationnal risks
Own infrastructure risks
Monitor for change in risks
Change occurs?
Yes
No
Assess change and take action
Action encompass updating risk analysis, taking appropriate counter measure and communicating with other interrested parties
Monitor for change in risks
Change occurs?
Yes
No
Assess change and take action
Tchnical risks (linked with technologies in use)
Own organisationnal risks

Figure 4 - *While using the system (voting period)*

11.3.3 **Update**

A regular update of the risk assessments is a requirement in the information security standards. However, recommended frequency of those updates looks like “regularly” or “whenever necessary”. The question here is what does “whenever necessary” mean in the context of internet voting in Switzerland. While some would define this point in time as whenever the system changes or whenever a new threat arises, the most relevant time in the context of internet voting in Switzerland is before each ballot as the object of the vote has an impact on the threat landscape. This review will only affect a part of the risk assessment, thus taking less time than doing a full review. However, enough time is to be planned for this activity, as some actions might have to be taken to mitigate updated risks. Voting usually takes place four times a year (every 3 months), this frequency can also be understood as regularly. As only a specific part of the risk assessment impacted by the ballot itself and events that have happened since the last ballot will be reviewed each time, it might also be appropriate to review the whole risk assessment once a year as per ISO 27001 prescription.

11.3.4 **Methodology**

The most important part in risk analysis is a comprehensive and systematic approach for analysing assets, threats, vulnerabilities and risks. As long as a methodology offers ways to achieve this approach, it can be used to assess an internet voting system's risks.

Methodologies meeting these requirements could be:

- ISO 27005
- NIST standards
- OCTAVE Allegro
- BSI Grundschutzhandbuch

11.3.5 **Publication**

The publication of a risk assessment comes with benefits:

- Trust-building
- Increased transparency
- Public involvement
- Allow wise decision (face the truth)

But also downsides:

- Revealing of weaknesses
- Controversy potential

The publication of the risk analysis requires resources, first of all for the formatting of it and the writing of the information allowing its better understanding. Secondly, to ensure that the questions and remarks that a publication generates are answered.

11.3.6 **In practice**

The VEleS already mentions a list of threats that are expected to be considered in the risk analysis. However, according to the answers we received, some threats seem to be underrated:

- Social-engineering
- Accidents / human mistakes
- Alleged attacks or malfunctions, framing attacks
- Planting trojan horse or backdoor

The state attacker threat agent is also mentioned to be missing.

Moreover, a particular attention should be paid to:

- Printing office
- Postal service (for delivering voting material)
- Zero-day vulnerabilities

In the answers given in the questionnaire, some specific risks have been suggested to be considered:

- Criminal organisation or foreign adversary infiltrates the trusted printing office (physically, socially, or electronically) to exfiltrate the codes on mailed voter cards and compromise cast-as-intended security.

- Criminal organisation or foreign adversary infiltrates the trusted postal service (physically, socially, or electronically) to misdirect some percentage of ballots from selected neighbourhoods to an alternate address where they are held or destroyed.
- Criminal organisation or foreign adversary offers a potentially large number of voters money or cryptocurrency in exchange for installing malware or spyware on their devices, which verify that the voters cast votes the way the adversary prefers (largescale electronic vote buying or coercion). An adversary could even carry out such an attack with almost complete anonymity with appropriate use of cryptocurrency and smart contract technologies.
- Compromised web server or App store serves a compromised version of the voting Web app and/or native apps to users.
- Compromised certificate authority, code-signing certificate, or developer signing keys used by an adversary to produce correctly signed but compromised versions of the voting Web app and/or native app to distribute to users.

  Network denial-of-service attacker prevents (targeted) users from casting votes electronically; forcing them to fall back to the mail or in-person process, in the expectation that many targeted users will give up and not vote at all due to the inconvenience.

11.4 Questions and summaries of the discussions on the platform

11.4.1 Organisation - One organisation, three alternatives

How do you assess the assumption that the solution 1 (the confederation prepares a core security concept then all actors prepare their risk assessment under consideration of it) is a better match considering the context? If it is not an adequate one, which one would be a better choice and why?

How do you assess the concept of the solution 1 proposed in the organisation chapter?

Who should be responsible for assessing (possibly with the help of external experts) the coverage of the risk analyses from the cantons and system provider? Do you see a risk that there would be gaps between the coverage of the analyses?

The solution 1 is preferred by the experts.

The leading role of the Confederation in this setup allows for a homogeneous, consolidated picture that helps to compare the situation in different cantons. There is less risk of gaps in this approach than with the other proposed solutions.

11.4.2 Update - Frequency and timing

How do you assess the statement made in the update chapter (targeted update every 3 months + full update annually)? Would an additional review of the risks be useful?

The experts welcome a full annual update of the risk assessment and also a renewed targeted risk assessment every three months. However, they also agree that three months is somewhat steep and it might make sense to scale back to a less aggressive rhythm after a couple of years.

Major new versions of the internet voting system should also bring a full update of the risk assessment regardless of where they fall into the update cycle.

11.4.3 Methodology – Standardisation

Is there a benefit in standardising the approach for all stakeholders or, on the contrary, would it be better to let each stakeholder chose the methodology he thinks is the most appropriate independently, provided that it matches the criteria mentioned in chapter 3?

The experts call for common security concepts that will result in a baseline risk analysis and homogeneity, that will allow to compare Cantons.

However, it would also be useful to leave room for Cantons to innovate and move beyond this baseline.

11.4.4 Publication - What, with which benefit and how

The publication of the core security concept would allow for public feedback on this topic. Is there a real risk that the publication could benefit a potential attacker, as he probably already knows about the weaknesses? Does it outweigh the benefit of the feedback? Would it be wise to publish further risk assessments (FCh's, cantons' or system providers')? On demand or systematically?

According to the experts, the core security concept should be published in order to obtain valuable feedback.

The detailed risk assessment should not be published immediately for several reasons. However, it may be useful to publish consolidated figures or statistical information as well as a description of the governing body, the rules and formats for the publication of the detailed risk assessment.

A more detailed assessment could then be published after a reasonable delay, or when a new finding has been fixed, based on these rules.

11.4.5 In practice - Particular risks

Do you agree that the proposed risks are relevant in the context of internet voting in Switzerland?

All the proposed risks are relevant and represent a good start, but the list is far from complete.

11.4.6 In practice - Mitigations depending on voters' support

Part of the risk mitigation measures relies on the voter (individual verifiability, checking a certificate fingerprint, checking the hash of JavaScript files) and more of this kind of measures might come. Given the high frequency of votes, that proper training material is provided to the voter along with the voting material and that an easily accessible channel exists to report problems (i.e. Report button on the voting website), what would still be needed to ensure effectiveness of such measures?

Risk mitigation measures relying on voters seem to be effective.

Measures to establish and improve confidence into the voting system (namely block 7) are likely to bring a positive effect here as well. And usability tests will also contribute to this goal.

Education campaigns, namely for students and teenagers before they reach voting eligibility is being named as another useful initiative.

12. Block 9 – Risk Limiting Audits and Plausibility Checks

12.1 Overview

Block nine covered two topics that are losely related: Risk Limiting Audits and Plausibility Checks on election results. The block relates to question 4.12 of the questionnaire.

Risk Limiting Audits are meant to support elections with a paper trail. As several experts mentioned them in the questionnaire we put them up for discussion. The idea of plausibility checks was not covered fully in the responses of the questionnaire, so it was put up for discussion again.

12.2 Summaries of the answers to the related questions of the questionnaire

12.2.1 Question 4.12 of the questionnaire

Which statistical plausibility checks method could be meaningful (e.g. comparison with other voting channels)? Is the publication of the results and method of any benefit? Do you see downsides?

Summary from 14 of 15 responses:

Many experts see the plausibility checks as a technique that is very close to the risk limiting audits that are known to detect election fraud. However, one expert has ruled them out in the specific Swiss case because there is no paper trail to base them upon.

Other experts see risk limiting audits as an option.

No expert really named a plausibility check method like the question asked. The statistical plausibility checks are given less value because different voting shares are expected on the electronic channel.

Most experts think that such a difference between the channels destroys the option to detect fraud that way. However, no expert explained how an attacker would be able to model the fraud without knowing the results of said voting districts in advance. Also there is no mentioning of experience with the results and the role of politologists interpreting the voting shares and making sense of the plausibility checks and thus establishing a definition which divergence is acceptable / explainable and where fraud would start.

There is a minority position who thinks that these plausibility checks should only be for internal use, since there is a big risk to get the fine line between false positive and false negative wrong.

One expert makes it very clear that plausibility checks can never be used as evidence, only as marker where to look.

Some individual interesting ideas:

«The problem with statistical plausibility checks based on election results only is the definition of useful threshold values, which determine the borderline for triggering an alarm. The selection of these values is a trade-off between generating false positives (values selected too loosely) and false negatives (values selected too strictly). Both cases are highly problematical, because manipulations are either not detected when they exist or suspected when they don't exist. This can create very confusing situations, for example by triggering an alarm even if universal verification has been successful. To avoid such situations, we recommend conducting statistical plausibility test at most as an additional measure in the internal monitoring of the system.»

«Statistical plausibility checks based on election results should not be confounded with another type of post-election audit called risk limiting audit. This technique helps to increase the credibility of the election result by manually checking statistical samples of paper ballots. As such, they can only be applied to an electronic voting system with a paper trail and are therefore not relevant for the Swiss case.»

«... the statistical methods can at best suggest where to look for election fraud and electoral problems, but it can never be used as evidence to identify election fraud and electoral problems. The same finding holds here as well. It is the evidence generated during an election that needs to be inspected. With the right auditing framework this will be automatically done, rendering statistical plausibility checks unnecessary.»

12.3 Introduction to Block 9 on the platform

Block 9 was presented as follows:

12.3.1 Risk Limiting Audits

Risk Limiting Audits are a method to check the results of elections and votes. They provide statistical indications that the outcome of elections conducted with voter-verified and machine-scanned paper ballots are correct. See https://risklimitingaudits.org for a good introduction into the step by step process.

While designed for physical voting, the concept can also be used for electronic voting systems, where a paper trail exists (See Stark/Teague 2014[4]). However, there is no paper trail with the online voting systems used in Switzerland. Yet several experts mentioned Risk Limiting Audits or the idea of a derived and adopted method in the questionnaire. It is therefore an idea worth exploring.

12.3.2 **Plausibility Checks**

Plausibility Checks are meant to detect any intentional or unintentional error with the entire voting process. For voting fraud, this is based on the idea that it is inherently difficult for an attacker to predict the result and tailor the fraud to blend in with the rest of the results.

Switzerland executes at least four national votes per year and depending on the residence of a voter also several votes on the canton and municipality level. During the years a very big pool of historical voting data has been accumulated. This could be a collection of information to draw upon when working with plausibility checks.

In Spring 2020, one of the largest voting frauds on record in Switzerland was discovered in the city of Frauenfeld in the canton of Thurgau. 100 paper ballots were misattributed to a different party. A simple plausibility check brought the con to light after the betrayed party formally complained. The ballots in question were quickly identified during the subsequent investigation.

Some sort of plausibility checks are established in most Swiss cantons, but they vary in the methods used depending on the administration.

In the responses to the questionnaire, several experts expressed doubts about the plausibility checks and especially cross-channel plausibility checks for various reasons.

According to those responses,

- there is a danger that low thresholds might lead to false positives that would undermine the trust in the whole system.
- statistical checks are not necessary, since they can not be used as evidence to identify fraud or problems anyways.
- individual and universal verifiability bring hard evidence and in the light of this evidence, the plausibility checks are not necessary.
- there is going to be a socio-cultural difference between voters using paper based and internet based voting, which will lead to different results on the different channels.

The idea of different outcomes across the different channels sounds convincing. Yet previous research shows that this is not necessarily the case (see Vassil 2016[5] for some insights based on Estonian elections). Maybe the socio-cultural differences and the differences between municipalities can be integrated into the statistical model: It is not readily understandable how gaps visible between different counting circles (there are more than 2000 municipalities in Switzerland) on paper would disappear completely when looking at the votes submitted via the internet.

Cross-channel plausibility checks (including comparison between and within municipalities) make it harder for an attacker to predict the results and to perform a fraud in a successful way. If the results have to pass the plausibility checks, then this forces the attacker to commit the fraud across different channels. This is more difficult and more expensive than the manipulation of a single voting channel. The introduction of voting channel diversity would therefore contribute to the attack resilience of the entire voting system.

[4] https://www.usenix.org/system/files/jets/issues/0301/overview/jets_0301-stark.pdf

[5] https://authors.library.caltech.edu/71713/1/1-s2.0-S0740624X1630096X-main.pdf

12.4 Questions and summaries of the discussions on the platform

12.4.1 Risk Limiting Audits and other forms of audits based on sampling

Do you see a way to adopt the paper-trail based Risk Limiting Audits or a similar method to double-check the results of internet voting in Switzerland?

If yes, please elaborate.

It seems unlikely that Risk Limiting Audits designed for paper ballots can be carried over directly to the current generation of internet voting systems.

12.4.2 Plausibility Checks

Do you think political scientists and statisticians can develop a statistical model that will detect hard to explain differences between multiple municipalities and between the electronic and the conventional voting channels? Can you briefly sketch such a model?

Would such a statistical method and its use contribute to the trust into internet voting?

Despite a large amount of historical voting data, there is relatively little data available for Swiss internet voting, namely for domestic voters. A derived statistical model might therefore be weak in the beginning. But do you think a growing share of internet voters and more data accumulated throughout the years would lead to improved models and solve the false positive problem in the long run?

In other words: Are plausibility checks something that could get better and better the more we use them?

Yes, it is possible to develop such a statistical model. The quality of the plausibility checks depends very much on the model and its inputs, which is expected to identify large scale tampering of ballots with a higher likelihood than surreptitious tampering on a smaller scale.

But it is unlikely that a careful and surreptitious manipulation on a smaller scale would be detected.

It is likely that said statistical model would improve over time.

Analyses based on statistical models cannot identify electoral fraud themselves. They can only point to electoral irregularities and trigger an investigation to find hard evidence.

13. Block 10 – Forensic Readiness

13.1 Overview

The topic of this block was forensic readiness. It relates to questions 7.1, 7.6 and 7.7 of the questionnaire. The content of this block is based on one hand on the answers to the questionnaire in the domain of crisis management and incident response as well as interviews conducted with digital investigation professionals. We focused on the incident response and forensic readiness because of the experts and expertise involved in the dialog.

13.2 Summaries of the answers to the related questions of the questionnaire

13.2.1 Question 7.1 from the questionnaire (Key elements in crisis management)

What are the key elements in crisis management when it comes to internet voting? How should the fact that internet voting involves multiple actors be taken into consideration?

Summary from 9 of 15 responses:

Mentioned key elements are:

- 6/9 Defined processes (encompassing detection of attacks, termination of attacks, assessment of the damages and impact on results, recovery, communication and prevention)
- 4/9 Action and communication plans must be ready beforehand

- 3/9 Dedicated team
- 2/9 Direct links with internet voting actors
- 2/9 Links with feds (GovCERT, MELANI, ChF, NDB) possibly also private companies

A third of the experts highlights the key role of detection of an attack in crisis management and one of them proposes to set up a monitoring that includes what is happening within the network and servers, feedback from the voters (e.g., lack of access or individual verifiability checks failed) and feedback from election auditors.

Some individual interesting ideas:

- Get advice from others (e.g. BAG, FCh's e Strategic Management Support Section)
- Do not forget the political aspect of the crisis
- Inform voters ongoing instead of after the vote
- Review the plans for every election
- There may be a serious incident without anything obviously having gone wrong, which is precisely what verification is meant to prevent

13.2.2 Question 7.6 from the questionnaire

*What would the process of investigating an incident (potential manipulation of votes, intrusion in the voting system, distribution of malware on voters' platform, etc.) look like?***Summary from 6 of 15 responses:**

Investigation has not been understood in the same way by all experts. A third of them recommend leaving it to the police. However, apparently, for the police to be involved, one would have to fill in a complaint, which cannot be done for each incident. The other two thirds have understood investigation of incident in a broader way, which is what was meant in the question.

3/6 experts propose the following structure for investigating an incident:

- Identification / Detection
- Containment / Eradication / Termination
- Recovery / Impact assessment
- Lessons learned

3/6 experts recommend involving external experts specialized in the area.

Some individual interesting ideas:

- In order to efficiently investigate the internet voting system in case of an incident, forensic readiness makes a significant difference. Forensic readiness means that the whole system has been conceived and developed keeping in mind that problems will occur and that investigating tools must be built on top of the system itself.
  Forensic readiness requires trustworthy traceability (e.g. ledger-based traceability using blockchain technology) and detailed logs in order to investigate unusual or suspected events.
- If the audit fails, then, incident investigation will become an important part of the election
- Possible incidents as well as response playbooks and investigative measures should be – as much as possible – defined beforehand and be ready before the incident happens.
- This needs to be outsourced to an external company that specializes in this topic. This should not be left to individual cantons or to the FCh.

13.2.3 **Question 7.7 from the questionnaire**

What are the requirements and stakeholders for digital forensics and incident response?

Summary from 4 of 15 responses:

Half of the experts mention the need to have a system designed for digital forensics, notably tools and immutable logs. The logs could be secured by a blockchain technology or other crypto methods like Merkle trees. The tools should be available to efficiently identify, authenticate, classify, analyze, integrate, interpret and evaluate digital traces.

Some individual interesting ideas:

- Technical capabilities for digital forensics and incident response should be provided by the vendor/operator of the platform (as well as the hosting provider if different). However, it is important that an independent third party is involved in order to prevent any conflict of interest.
- It may be required to establish partnerships with private companies that would be capable of assisting in the investigations. They are to be trained on the solution.
- A system that is designed for digital forensics might present a tradeoff with respect to its privacy guarantees.
- It is important that technical means are in place to assure that recounts can be done and that the system has enough redundancy to cross check the results in a manner understandable to the general public.
- If the servers are out of the jurisdictions of the Swiss police it may be very difficult to collect the information

Leftover questions

The crisis management part of the chapter 7 will be developed outside of the dialog. The questions 7.2 to 7.5 as well as some auxiliary feedbacks along the dialog will be considered in this work. Answers to question 7.8 have been considered in the current block, however the question in itself and especially the prosecution part have not really been answered and need competencies that were not available in the dialog. Question 7.9 is more of a political question that need to be discussed at a different level.

13.2.4 **Question 7.2 from the questionnaire**

What are the right events and thresholds for an activation?

Summary from 7 of 15 responses:

There is no consensus on this question. The only idea that is shared among experts is that one has to consider events that could have an impact on the results.

The proposed events can be part of 3 categories:

- Evidence of errors
- Evidence of malfeasance
- Evidence of significant public concerns

And the sources for those events could be but not limited to:

- The internet voting hotline
- Secret service or cybersecurity firm
- Monitoring at the canton/system provider (system log files, physical access log file, behavior of employees, server integrity checks, etc.)

Some individual interesting ideas:

- All irregularities should be communicated
- There needs to be a continual situation awareness cell that monitors the election environment. The cell should regularly brief the chief election officers (cantons/federal level), who should be able to trigger crisis management (political act).
- The definition prepared beforehand includes a reflection on which incidents are considered as “benign” and can be ignored or handled automatically (e.g. banning an IP address which has triggered automatic attack detection mechanisms) and those that must – in any case – be discussed and handled by the crisis management committee.

13.2.5 **Question 7.3 from the questionnaire**

Who should be involved in crisis management, with which role?

Summary from 6 of 15 responses:

Here again, no consensus except that there should be a defined group with broad competencies (e.g. technique, communication, voters’ behavior expertise, official representation).

The group could consist of:

- 2/6 Federal chancellery
- 2/6 Cantons
- 2/6 System provider / operator
- 2/6 Communication expert (could come from cantons and/or FCh)
- 2/6 Independent technical expert

2/6 experts propose to have the Federal chancellery leading this group and the system provider/operator for monitoring their systems and people and investigating.

Some individual interesting ideas:

- The independent technical actor is responsible for validating or challenging the results of this investigation (and thus prevent any conflict of interests).
- There should be liaisons people, for example to the Swiss telecommunication service, GovCert, and the cybersecurity office of the government/military, etc.
- There should be a national voting expert, who can frame and explain a voter’s reactions.

13.2.6 **Question 7.4 from the questionnaire**

How should the communication be organised (internally and externally)?

Summary from 4 of 15 responses:

3/5 experts plead for a quick and transparent communication to the public in case of an incident.

2/5 experts think that the external communication should occur through the cantonal authorities.

Regarding communication inside of the crisis team, one recommends that they are collocated and another one that clear communication channels are established with frequent update on the situation. Both are pursuing a common understanding and vision of the situation.

Some individual interesting ideas:

- Technical communication must be distinguished from public communication

For external communication, it is recommended to employ a communication expert, who can explain the problems that occurred in a greater political context

13.2.7 **Question 7.5 from the questionnaire**

Are there already structures that should be involved in crisis management (e.g. GovCERT)?

All experts that have answered this question are clearly or somewhat in favor of the involvement of such entities. GovCERT, MELANI could be involved.

Some individual interesting ideas:

- A few identified private companies capable of providing expertise and/or manpower could be on the stakeholders' list. For this last category, it could be interesting to assess the possibility of relying on existing organisms like Swiss Cyber Experts (SCE).
- Governments often have intrusion detection systems integrated into the national network. The US, for example, uses a Suricata based Alert sensor system.

13.2.8 Question 7.8 from the questionnaire

In practice, is it possible to investigate and prosecute a case in an effective and efficient way? If not, what measures could be taken?

Summary from 5 of 15 responses:

2/5 experts write that while investigation is something achievable, prosecution is quite complex and depends on external factors (e.g. international cooperation and laws).

Some individual interesting ideas:

- Disrupting the attack or making it much more difficult to achieve in the future perturbs the attacker. Perturbing the attacker without being able to prosecute him, understanding his/her motivations, discovering the target of his/her attack can also be considered as successful results of the digital investigation of an internet voting system.
- Experts in crisis management must be involved and available at any time, particularly during a voting period.

Require that each file generated, ballot boxes, log files etc., are digitally signed by a person responsible for it (this requires that Switzerland has access to a national ID infrastructure).

13.2.9 Question 7.9 from the questionnaire

How should the validity of election or voting results be handled in case of an incident? Which investigation outcomes or further circumstances could allow or would need to prohibit declaring a voting result as valid?

Summary from 8 of 15 responses:

5/8 experts mention that the validity of the results depends on the impact the attack had on them. The interpretation of results here is not clear. It could be the exact count of votes for an option or another or the outcome of the vote (acceptance or refusal of a proposition or the elected person). If one choses the first interpretation, the result is only valid if the investigation can prove that the count has not been altered. With the second interpretation, the result is only valid if the investigation (which is to be done at least if the share of internet voting votes is big enough to change the outcome) can prove that the outcome would not have changed even if votes have been altered. The latter interpretation seems to be the most widely shared.

2/8 experts write that this is a political question that needs deliberation and acceptance by all stakeholders.

Some individual interesting ideas:

- The trustworthiness of the system can be measured by how trusted/accepted its remediation processes are. If they are politically accepted, the system is trustworthy enough for the democratic electorate, if not, it should not be used, as it cannot settle political conflict.
- In case of a security breach, an attack or some malfunctioning, it is important to precisely assess the potential impact in a worst case scenario.
- It would be good to integrate the robustness to failure and recovery into the voting system.
- If auditing the evidence succeeds, then the result is valid.

- Ideally, the VEleS should state exactly what has to happen in each case of incident/failure.

I would recommend to create a catalogue of possible auditing failures, and possible incident reports, and then determine for each the appropriate course of action. One therefore circumvent a paralyzed government that relies on the courts to determine the election result in the case of an incident.

13.3 Introduction to Block 10 on the platform

Block 10 was presented as follows:

While the VEleS and its annex contain requirements for the security of the system, they do not contain particular requirements regarding digital forensic and incident response further than maintaining logs.

Individual and universal verifiability are at the core of the system to detect manipulations but an efficient way to react on what they might uncover is now to be defined.

Forensic readiness is defined as the ability of an organisation to maximise its potential to use digital evidence whilst minimising the costs of an investigation (See Rowlingson 2004). In the specific case of internet voting, forensic readiness helps in detecting and investigating suspicions of manipulation thus increasing confidence in the outcome of the ballot.

13.3.1 Forensic and data collection

A timely detection is crucial, thus an efficient monitoring is to be put in place. A monitoring could include what is happening within the network and servers (technical monitoring), feedback from the voters (e.g., lack of access or individual verifiability checks failed) and feedback from election auditors (perhaps based on the output of the "verifier" introduced in discussion block 2). However, when it comes to investigating a case, the most critical part is data. The data you have collected before using the system as a reference of a clean system, the data you collect while using the system to be able to trace the actions back and the data you collect after the use of the system as a state of play.

13.3.2 Data collection Swiss army knife

As mentioned above, data is the cornerstone of investigation. You have to gather a lot of data even though part of it might not be useful or even looked at in the end. It is difficult to predict which data is to be collected and which not. It highly depends on the system and on the techniques of the attacker. A way to check if you have collected the right data and the right amount of data is to simulate the investigation of an attack. Note that collecting everything possible can also affect the performance of the system, the storage you need and the number of people you need to analyse the collection. Thus, a balance is to be found. We have gathered here a collection of tools that could be useful in this context.

The monitoring of the voting system and its platform should at least encompass centralized and secured logs as per required in the VEleS' annex. However, logs are as good as what they contain. As the set of information that are to be logged highly depends on the system in use and its architecture, it is not possible to define it in a generic way and a specific analysis should be conducted for each system. Decisions need to be taken based on the risk analysis and reflected in it. It should be possible to track an event through the logs, thus they have to be consistent with each other (e.g. time stamp, event id, etc.).

Aside from logs, an integrity check should run regularly on all files of the system that are deemed critical, including but not limited to registry files and operating system files. However, these checks suppose that you keep a reference base and maintain it along with all updates you make on the system, which can be resource consuming.

A third tool in this Swiss army knife would be a forensic agent running on the servers. This agent would be able, for example, to take "images" of some parts of the server like its memory, its disks and its running processes. Some further tools, like malware research and detection tools, could help analysing those data and identify suspicious patterns in memory. The forensic agents are powerful but they open a new area for threats as they have high privileges on the computer they run on and even if they act in a read only mode, one could exploit a vulnerability in the agent to do whatever he wants on the server.

The use of out-of-band management equipment could also be leveraged in monitoring the system. As a security measure, it allows to cut the management of the elements of the system from the common network. You can also use it to get the configuration files and other data from the elements of the system that are linked to it without having those data going through the common network. As for the forensic agent, this kind of equipment is double-edged and can be abused if not properly secured.

Finally, network traffic capturing can also help investigating a case. In this case, the frontend and the backend have to be considered. However, having an all-time capturing might generate a huge amount of data, not to mention that someone should also be able to analyse those data. A compromise would be to have the whole system ready for network traffic capturing but only activate it when a suspicion of abuse has been detected by other monitoring systems.

All these tools also raise some concerns on one hand regarding the way they transfer their results to a centralized monitoring system and on the other hand regarding personal data protection. As for the first area of concern, networks and firewall have to be carefully set up and the “out of band” management has to be properly secured in order for an attacker not to be able to use the same channels to compromise the system. The second area of concern is a bit trickier and depending on the data that are collected, a data protection officer should be involved to ensure a proper handling of data that could be sensitive like IP addresses. The data on which vote secrecy depends should be handled with a “secret” classification level, other data could be considered as “confidential”.

13.4 Questions and summaries of the discussions on the platform

13.4.1 Forensic readiness and confidence

Do you think investigations based on a forensic ready system is likely to provide convincing explanations in the case where anomalies are detected or suspected (e.g. inconsistent control components)?

Forensic readiness anticipates that problems might or will occur. It adds forensic consistency to a system. Broadly speaking, forensic readiness is achieved via the careful integration of tamper-resistant log files and by the use of forensic tools that help to observe and analyse the operation of a system or its components.

Carefully planning the content of the log files and securing their creation, transfer and storage is a must. This applies to peripheral components that are not trusted in the system architecture as well as the core components of the system that are trust-critical.

It is important to note that no secret may be logged and that advances in cryptography could allow to decrypt encrypted information in the future.

The situation with the forensic tools and the out of band management is different: Their installation and use is welcome on the peripheral components. But it is critically important that none of the tools allowing to expose secrets are installed or running on the trust-critical components of the voting system. There always has to be a part of a voting system whose operations is unobservable so that vote secrecy and verifiability can be maintained.

It may be beneficial to limit the components that are allowed to run on a component via a hardware-attestation mechanism. This could be used as a proof that none of the tools that are dangerous to the security of the system are running on the individual components.

It may be worthwhile to decommission and snapshot a system that is no longer in operation or has strong suspicions of compromise in order to do a forensic analysis.

13.4.2 Criteria for forensic readiness

Do you know of a system that is fully forensic ready? What are the criteria for such a system? How should it be (publicly) scrutinised to ensure confidence in the results?

We did not get any answers to this question. An overall summary for this block is being provided in the discussion of Forensic readiness and confidence.

13.4.3 **Minimal Swiss army knife**

Which minimal subset of the above-mentioned tools would be necessary to properly investigate incidents? Do you see any critical tool missing?

We did not get a lot of answers to this question. An overall summary for this block is being provided in the discussion of Forensic readiness and confidence.

13.4.4 **Effective monitoring and vote secrecy**

How could we set up the monitoring so that vote secrecy is still acceptably protected? Consider especially the data that are collected and the time during which they could be retained. Are there data that are useful for digital forensic and that would not hinder vote secrecy in the sense of how one voted (e.g. IP address)?

We did not get a lot of answers to this question. An overall summary for this block is being provided in the discussion of Forensic readiness and confidence.

13.4.5 **Capabilities and limits of forensic tools**

The forensic tools can have many capabilities (e.g. taking memory snapshots, monitoring running processes, looking into system files, imaging disks, etc.). Do you think that some of those capabilities should be limited to a subset on certain parts of the system (e.g. control-components) and on which parts? On the other hand, do you see any capabilities that would be particularly useful on some parts of the system?

An overall summary for this block is provided in the discussion of forensic readiness and confidence.

14. Block 11 – Big Picture

14.1 Overview

The topic of block 11 was the concluding Big Picture at the end of the dialog. The experts were invited to share their outlook on the future development. It was loosely based on the opening question 1.1 of the questionnaire but with the whole dialog in mind.

14.2 Summaries of the answers to the related questions of the questionnaire

14.2.1 **Question 1.1 of the questionnaire**

You visit an imaginary country where internet voting is widely used. Given your background, you want to assess to which degree internet voting in that country may be considered «trustworthy» with respect to past and future votes. (Assume the possibilities that technology currently offers, i.e. no futuristic devices. Assume that coercion / vote buying are not a problem.)

Which are the most important questions you would need answers to? (Think for example of roles and responsibilities, operations, system, scrutiny)

Which are the most important answers you need in order to conclude that internet voting is trustworthy?

How does the origin of the answers influence your conclusion? Which piece of information would need to originate from which source in order for you to consider it relevant for your assessment (e.g. information on system features, third party examinations or voting procedures)?

Then relate your statements to the Swiss case. Which key-elements from the answers above do you observe in Switzerland? Which existing key-elements could be improved? What is missing? Can you identify any low hanging fruits? Where can you spot alternative approaches you find interesting?

We would like to understand the reasoning behind your views in detail.

Please give extensive explanations. If this requires writing extra pages, please do so.

The answers to this open question were not summarized due to their strong diversity.

14.3 Introduction to Block 11 on the platform

Block 11 was presented as follows:

The first internet voting trials were conducted over 15 years ago. The low-scale trial phase has allowed to learn and adapt as slowly more and more cantons joined in. Once the trials are resumed only a small number of cantons are likely to offer internet voting and that with only limited fraction of the electorate participating.

The expert dialog gives us an important foundation at redefining the trial phase commissioned by the Federal Council. Now that the discussions on the platform are almost over, we would like to ask you to relate your personal conclusions and issues you find important to the Swiss trials.

All experts are asked to share their thoughts.

14.4 Questions and summaries of the discussions on the platform

14.4.1 The Big Picture

How do you assess the current situation in Switzerland? Please distinguish between the level of requirements in the VEleS (technical, scrutiny, transparency, ...) and the fulfillment of these requirements.

What would be the most important next steps and when should they be implemented?

Where would we ideally stand after the next 15 years?

All experts are asked to share their thoughts.

For the experts, this dialog marks a milestone. The government authorities have invited scientists to discuss several pending questions around internet voting for the first time in an official process. This dialog has been very fruitful and should serve as a start of an ongoing exchange.

The experts identified several areas where the currently available internet voting systems are lacking or where the current Swiss regulation is prone to be updated in terms of security, privacy, verifiability, transparency and scrutiny.

The experts have addressed many problems that should be tackled immediately. Bigger technical challenges can only be implemented in a mid- and long-term perspective. They include the creation of formally verified voting components, more diversity and redundancy of the trust-critical system components and to use simplicity as a basic design principle.

Another challenge in the mid- and long-term perspective is a Public Bulletin Board. Whether and how a public board should be introduced requires further analysis.

Yet the dialog has also shown that a lot has been achieved during the past 15 years of internet voting trials in Switzerland. There is no need to abandon all future plans with internet voting. Security is a continuous improvement process and this dialog is a single step on this path.

For the experts, the dialog focused on technology to a very wide extent. The question of trust and how a society could start to trust a new technology remains to be discussed in the future. Raising doubts about the trustworthiness of an internet voting system or the results of a vote is one of the biggest residual risks with internet voting: One politician who claims that a vote has been manipulated can do as much damage as an actual attack.

It is very important that the design, the development and operation of internet voting does become more transparent. Yet it also has to become more inclusive at the same time. It has to include political stakeholders as well as broader groups of the voting population to find more acceptance.

According to the experts' big picture, the dialog focused on internet voting and left the existing physical voting channels aside. However, the existing voting channels are also prone to attacks; not the least via the auxiliary electronic system components. A holistic view on the possible attack vectors is thus necessary. This should lead to a holistic effort to improve voting security with a risk-based approach.

After all, there is no perfectly secure voting system and the voters should have a transparent view on the known weak spots with the voting process. You can not rule out fraud completely nor can you rule out technical errors. But you can make it very hard and expensive to cheat. And you can prepare yourself to be able to detect an attack as soon as possible. By establishing a strong track record, you can build trust.

Some experts expressed the necessity for strong dispute-resolution procedure that require the Internet Voting system to produce broadly accepted and immutable evidence that can be independently checked to settle claims of malfunction and cyber attacks.

If the experts look 15 years ahead, then internet voting is established and the population has confidence that the voting system produces the correct results. For some of the experts, said process will take a bit longer. For other experts, it is much less clear that Internet voting will earn the trust necessary for being established.

The next 15 years are therefore key since many problems are still unresolved. But this dialog has shown viable or at least potential solutions to many if not most of them. Continuing the dialog with subject matter experts as well as representatives of the population is thus a useful and promising method to move forward.

15. Block 12 – Future Dialog

15.1 Overview

The topic of block 11 was the future dialog. It is based on question 5.1 of the questionnaire and explores idea of a continuing expert dialog about internet voting. The block took up experiences with the two phases of the written dialog at the base of this report and puts it in perspective.

15.2 Summaries of the answers to the related questions of the questionnaire

15.2.1 Question 5.1 of the questionnaire

Which are the conditions to be met in order to ensure that independent experts (particularly experts from science) participate? Which measures must / could be taken to meet or promote these conditions and thereby participation?

1. Participation in «public scrutiny»

2. Participation in examinations mandated by the public administrations of the Confederation or the cantons or the system providers

3. Supporting the public administration in the further course of the trial phase, e.g, at implementing the measures currently being defined in the course of the redesign

Summary from 14 of 15 responses:

1. For participation in «public scrutiny» there is an agreement that access to documentation and code as well as the possibility to publish the results are important. Some additional ideas are to organize workshops, found research or give bug bounties.
2. Examinations are clearly seen as a job that must be paid for. Some experts suggests creating a committee or testing institute to manage examinations.
3. This work is also seen as having to be paid for. One proposal is to involve a random group of voters.

Creating a European network of experts was also suggested in general.

15.3 Introduction to Block 12 on the platform

Block 12 was presented as follows:

We aim at continuing and structuring the collaboration with independent experts in various domains, e.g. at conceiving, implementing and scrutinizing the systems, risk-, incident- and crisis-management as well as communication. The foundation of a successful collaboration of administrations, providers and independent experts depends on a common understanding achieved by regular exchange.

We believe that the discussions on the platform have contributed to mutual understanding and there would be much more to talk about. The final day of the dialog is approaching. Of course, many of the participants will stay in touch after the dialog and discuss internet voting in other settings and that is a good thing. Nevertheless, we would like to find a reasonable setting that promotes exchange between administration and experts in the future.

15.4 Questions and summaries of the discussions on the platform

15.4.1 The Future Dialog

Would it make sense to run a dialog platform continuously?

Do you see another way of having a continuous dialog?

Or should dialog be rather event driven?

What kind of events could give a good frame to promote exchange on Swiss internet voting?

The dialog via the questionnaires and here on the platform is seen as well-organized and effective.

The experts find it very beneficial to continue the conversation in a scientific advisory committee.

The extent of the authority of the committee is not entirely clear: It could have a function with overseeing elections and votes, but it should rather not have regulating power.

Identifying the relevant topics, the preparation of the questions and the moderation of the dialog are seen as key success factors.

Several experts felt uneasy and pressured with the asynchronous written dialog. For other experts it felt beneficial to be able to time the contributions oneself and to think about arguments before posting a response. A mix of written dialog and on-site or remote meetings is likely a balanced compromise.

The use of the gitlab platform for the written dialog was adequate. Yet a lighter approach via a mailing list could also be an option. Gitlab as well as a mailing list come with the problem how to organise the current state of the knowledge and the consensus in a dynamic overview. A wiki might be a beneficial replacement or complementary component in this regard.

Next to a scientific committee, there could also be a non-expert citizen's advisory board that supports various tasks like public trust-building, usability testing and refinement as well as education methods and materials.

16. Beyond Internet Voting

Several experts objected to the internet voting focus of the questionnaire and online dialog. The following statement approved by the experts is meant to introduce a broader perspective:

The expert dialogue focused on internet voting. Yet online voting is not the only use of electronic systems in the voting process, nor are the electronic systems the only part of the entire voting process that could be attacked successfully. A paper from Christian Killer and Burkhard Stiller[6] from 2019 examined the physical voting process and identified several potential weaknesses. We encourage the Federal and cantonal authorities to look into these problems in order to remedy the identified vulnerabilities.

[6] https://files.ifi.uzh.ch/CSG/staff/killer/extern/publications/EVOTEID19-Killer-Stiller.pdf

However, we also recognize that this paper is only part of the story. For example, Killer & Stiller did not analyze how vulnerabilities in the existing postal voting system could become a threat to election security once internet voting is introduced, and vice versa. These "cross-channel" interaction risks should be examined carefully. Killer & Stiller also did not examine important threat areas that can affect postal and internet voting channels alike, such as voter coercion or vote-buying, a risk that appears most scalable and attractive to foreign adversaries when applied to internet voting (see "On-Chain Vote Buying and the Rise of Dark DAOs", 2018[7]), but has also proven a realistic threat to postal voting even in mature democracies (see "Election Fraud in North Carolina Leads to New Charges for Republican Operative", 2019[8]).

Finally, the inherent risks of deploying internet voting must be carefully balanced against the bigger-picture risks of *not* having it, such as the potential disenfranchisement of citizens abroad whose postal services are slow, unreliable, or untrustworthy, or the systemic risk of lock-in to a postal voting technology monoculture that may present few clear paths towards greater long-term security, transparency, or convenience.

[7] https://hackingdistributed.com/2018/07/02/on-chain-vote-buying/

[8] https://www.nytimes.com/2019/07/30/us/mccrae-dowless-indictment.html