IEEE International Conference on Blockchain and Cryptocurrency
Toronto, Canada, Virtual Conference, May 3, 2020

Practical Introduction to Blockchain-based Remote Electronic Voting

Christian Killer, Bruno Rodrigues, Eder Scheid, Muriel Franco, Burkhard Stiller
Communication Systems Group CSG
Department of Informatics IfI
University of Zürich UZH
[killer,rodrigues,scheid,franco,stiller]@ifi.uzh.ch

Universität Zürich[UZH]

CSG
Communication Systems Group

ifi

ToC

- Electronic Voting (EV)
  - Definitions and Notions with a Global Overview on EV
  - History of Switzerland and REV
  - Focus on Remote Postal Voting (RPV) in Switzerland
- Remote Electronic Voting (REV)
  - REV Requirements and Properties
  - Building Blocks
    - Homomorphic Encryption, ZKP, PBB
- BC-based REV
  - Provotum Voting Protocol and Evaluation
- Summary

11:30

12:30

Break

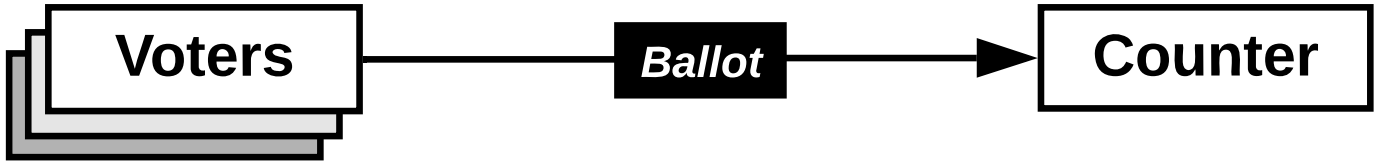
12:30

13:30

Electronic Voting (EV)

- **Definitions and Notions with a Global Overview on EV**
- History of Switzerland and REV
- Focus on Remote Postal Voting (RPV) in Switzerland

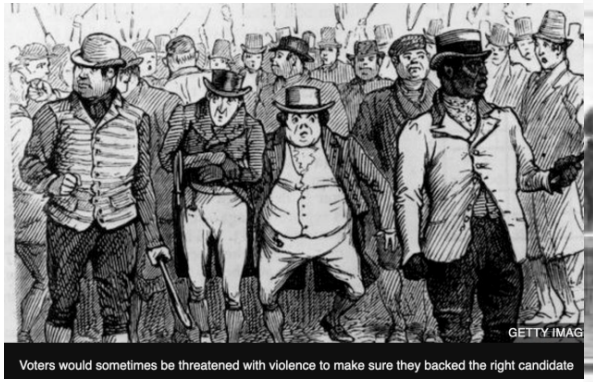
Are elections just counting?



- Main difficulties in practice
  - No one is trusted
    - Anyone is a potential adversary
  - The need for evidence
    - Any observer should be able to verify the election data
  - The secret ballot
    - Voters should not be able to prove how they voted to anyone, even if they wish to do so

The Secret Ballot

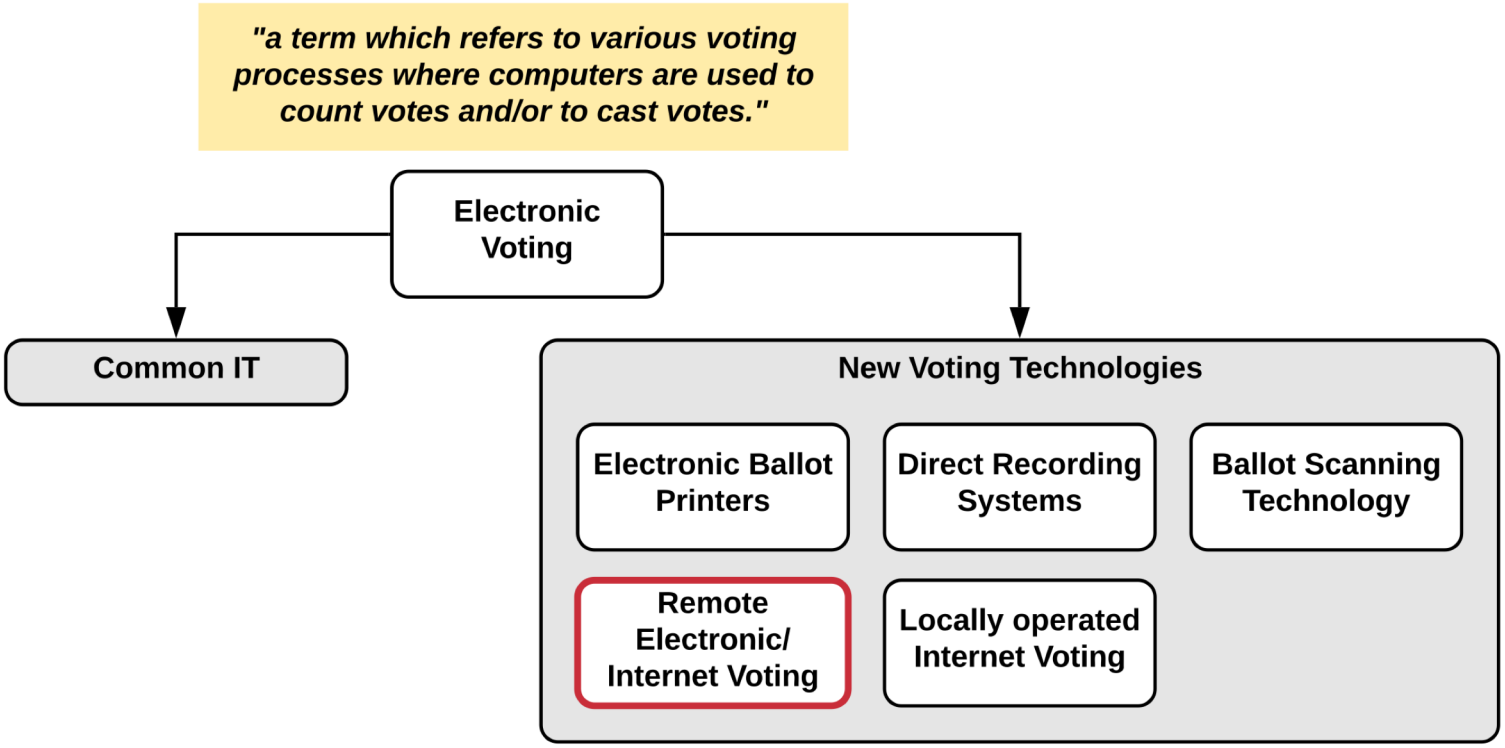
- Mid 19th century, the “Australian ballot” was introduced as a means of deterring coercion
- Today, society takes for granted this process of voting privately within a public environment, where observers can enforce privacy, but great innovation in its time
- In the early 1980s, cryptographers began to propose new election designs with new properties and integrity guarantees



Voters would sometimes be threatened with violence to make sure they backed the right candidate

Terminology

- **Remote Electronic Voting** (REV) or Internet Voting
  - Vote cast in a **remote, uncontrolled** environment and ***not*** in a voting booth.



Source: https://www.bbc.com/news/uk-england-leeds-31630588

E-Voting World Map

Brazilian government reiterates e-voting security

The electronic system has been in place for over two decades "without a single proven fraud occurrence," said minister.

BBC Home News Sport Reel More

NEWS

Technology

E-voting experiments end in Norway amid security fears

No plans

Discussion with concrete plans for Internet voting

Internet voting (legally binding) with other voting technologies

Stopped use of voting technologies

Discussion and/or voting technology pilots

Ballot scanners and/or Electronic Voting Machines (legally binding)

Estonia

i-Voting saves over

11,000

working days per election

44%

of Estonians use i-Voting

844

years of working time saved

From

2005

Estonia uses i-Voting

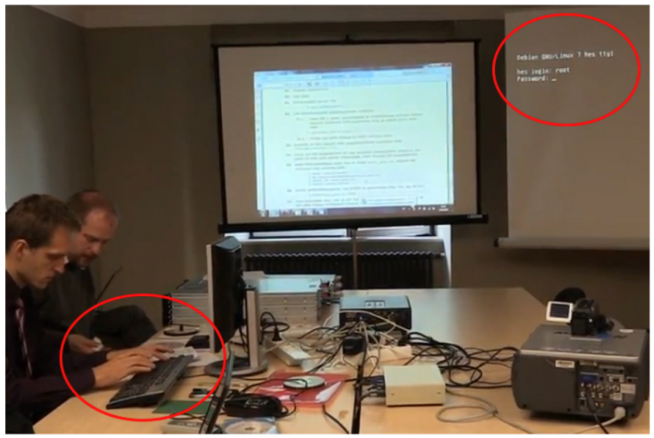


Figure 7: **Keystrokes reveal root passwords** — Videos posted by officials during the election show operators typing, inadvertently revealing root passwords for election servers.

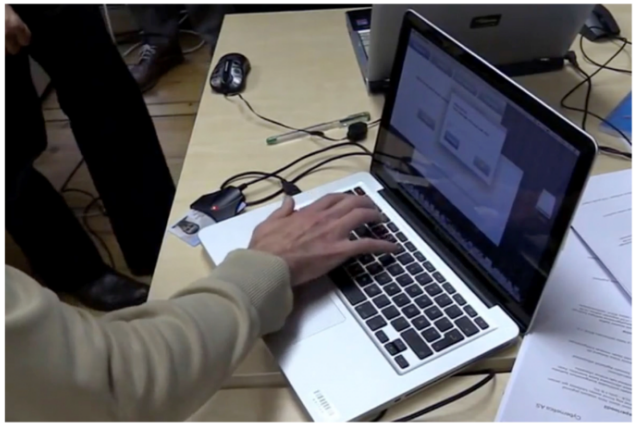


Figure 8: **Video shows national ID PINs** — During pre-election setup, someone types the secret PINs for their national ID card in full view of the official video camera.

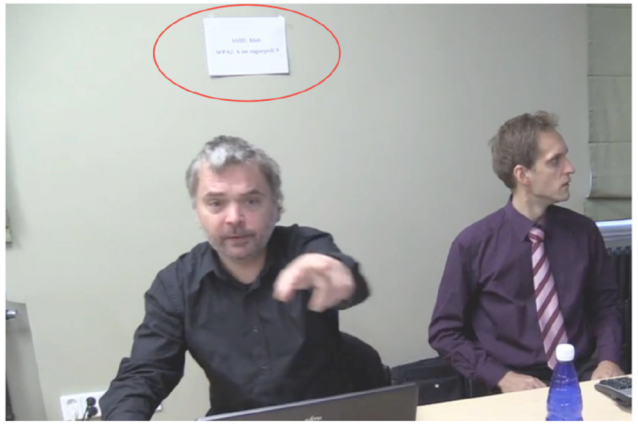


Figure 9: **Posted Wi-Fi credentials** — The official video of the pre-election process reveals credentials for the election officials' Wi-Fi network, which are posted on the wall.

Source: Security Analysis of the Estonian Internet Voting System, Drew Springall et al., November 2014

Electronic Voting (EV)

- Definitions and Notions with a Global Overview on EV
- **History of Switzerland and REV**
- Focus on Remote Postal Voting (RPV) in Switzerland

Brief History of Swiss Voting Channels

- A tradition of direct democracy (Landsgemeinde)
  - Introduction of “Remote Postal Voting” in the 1970’s which was generalized in the 1990’s
  - 1998: the Federal Council proposes to study Internet Voting (I-Voting) and by the year 2000, official start of the Vote “électronique” project with three pilot cantons (GE, NE, ZH)

Image: *Samuel Trümpy Photography, https://www.gl.ch/landsgemeinde.html/216*

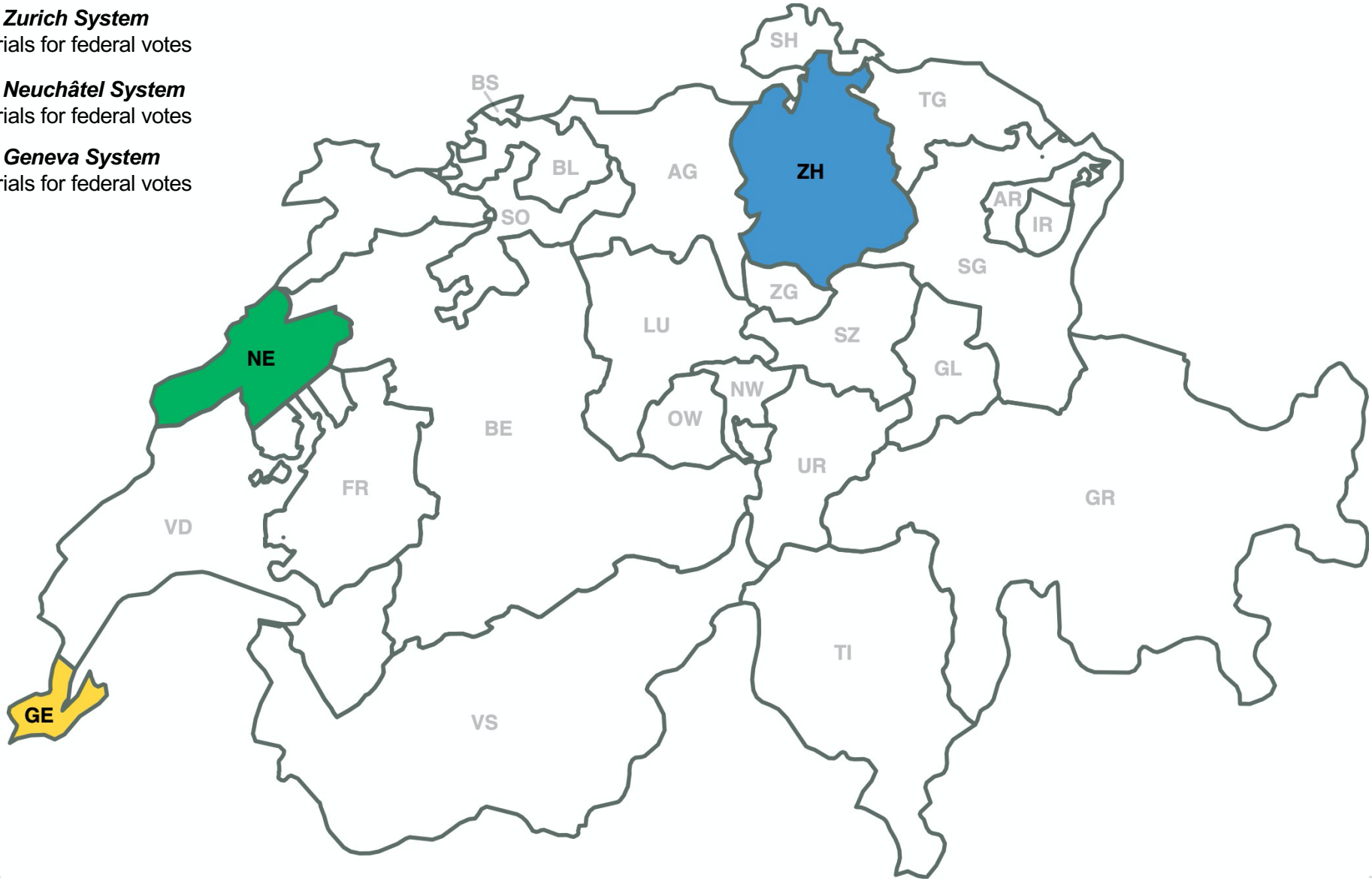
 Source: https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/chronik.html

First Trials, Initial Pilot Phase

2004 ***Zurich System***
first trials for federal votes

2005 ***Neuchâtel System***
first trials for federal votes

2004 ***Geneva System***
first trials for federal votes



ifi

Extension of Trials

- In 2006, the Federal Government opened the door for all cantons to use *limited* internet voting:
  - 20% of the cantonal electorate (later extended to 30%);
  - 10% of the federal electorate

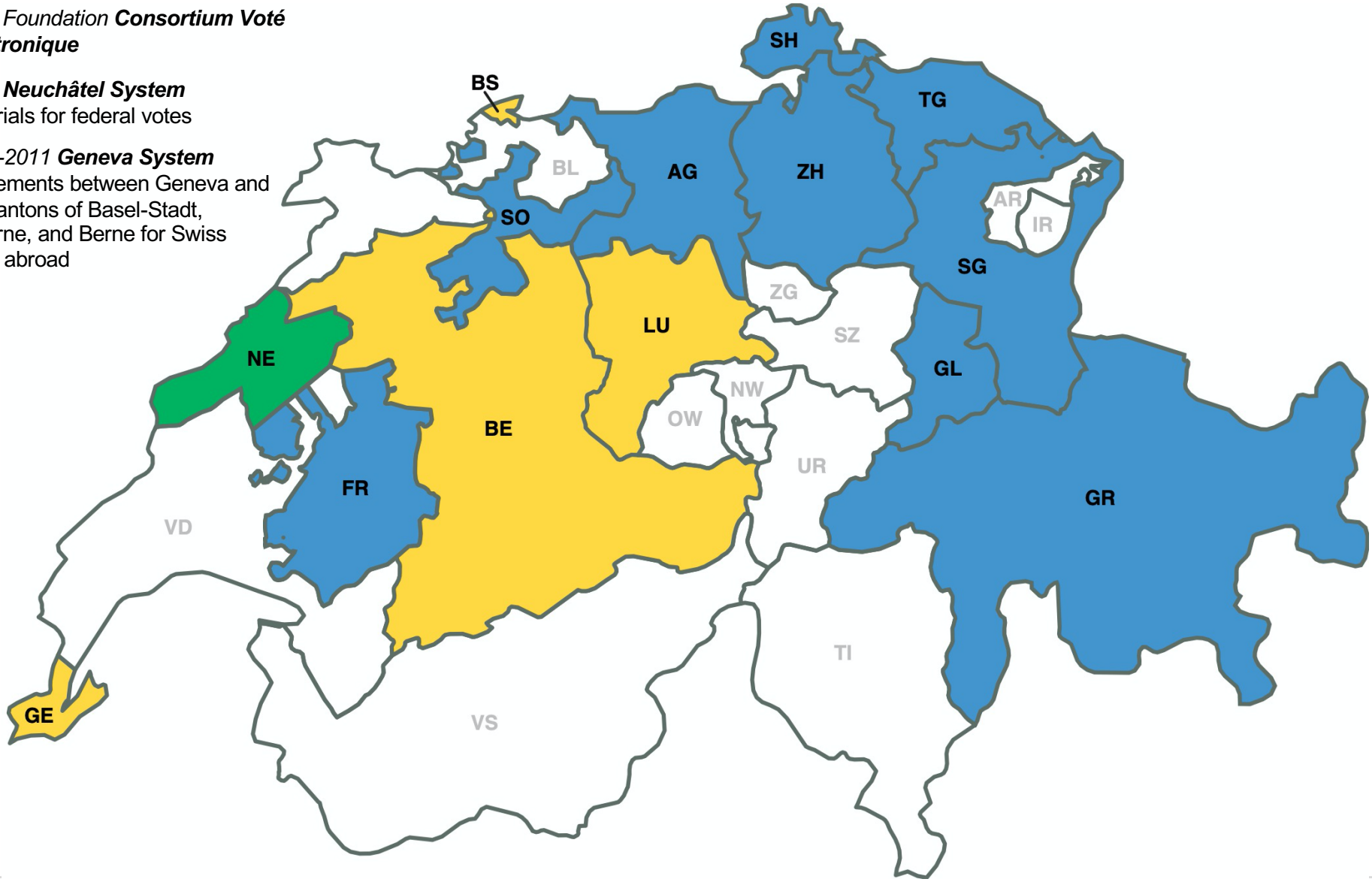
- Geneva and Zurich (Vote Consortium) started offering their system to other cantons
  - Some drawbacks were also experienced:
    - Regulatory issues in Geneva (2005 – 2007)
    - Technical issues in Zurich (2011 – now)

Extension of Swiss-wide Pilots

2009 Foundation ***Consortium Voté Electronique***

2005 ***Neuchâtel System*** first trials for federal votes

2009-2011 ***Geneva System*** Agreements between Geneva and the cantons of Basel-Stadt, Lucerne, and Berne for Swiss living abroad



ifi

Situation as of 2011

- Geneva System
  - Canton owns the SW and operates the system
  - Developed by Geneva IT Department
  - Voting Protocol by Bern University of Applied Sciences
- Zürich / Vote Électronique Consortium System
  - Canton owns the SW, external company (Unisys) operates it
  - Developed by Unisys
- Neuchâtel System
  - Canton owns the SW and operates the system
  - Developed by Scytl
  - Embedded in “Guichet unique“, as one of many offerings

New Swiss Regulations in 2014 (VEleS)

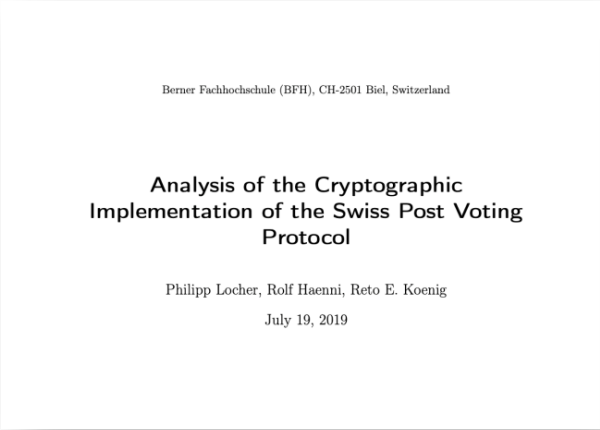
- In 2013 new regulations regarding REV in CH
  - Federal Chancellery Ordinance on Electronic Voting
    «161.116 - Verordnung der BK über die elektronische Stimmabgabe vom 13. Dezember 2013 (VEleS)»
- Framework for authorization of voting systems with three different levels, linked to the amount of allowed electorate

Level	Limit(1)	Requirements	Certification
1	30% cantonal 10% Swiss	Functional and security • BSI Common Criteria PP for Internet Voting Products • Council of Europe standards	• None, only ChF experts revisions of documentation (risk assessment, security architecture…) and security tests of system (functional and infrastucture)
2	50% cantonal 30% Swiss	Individual verifiability • Cast-as-intended verifiability • Recorded-as-cast verifiability	• Security: Common Criteria EAL 2 • Cryptography: Security and formal proofs (printing, server and tally are trusted) • Infrastructure and printing offices
3	No limit	Complete verifiability • Individual verifiability • Counted-as-recorded verifiability • Control components (2 or 4)	• Security: Common Criteria EAL 2 + EAL 4 (control components) • Cryptography: Security and formal proofs (printing and one control component are trusted) • Infrastructure and printing offices

Source: Defining a national framework for online voting and meeting its requirements: the Swiss experience, Jordi Puiggalí, Adrià Rodríguez Pérez, E-VOTE ID 2018

Publication of Source Code in 2019

- Publication of the source code in February 2019, researchers have uncovered significant security flaws in the new, fully verifiable, system provided by Swiss Post (sVote, developed by Scytl)
  - One of the security issues relates to individual verifiability and thus to the system already used. As a result, the system was not available for the vote on 19 May 2019.


Berner Fachhochschule (BFH), CH-2501 Biel, Switzerland

Analysis of the Cryptographic Implementation of the Swiss Post Voting Protocol

Philipp Locher, Rolf Haenni, Reto E. Koenig

July 19, 2019


Auditbericht

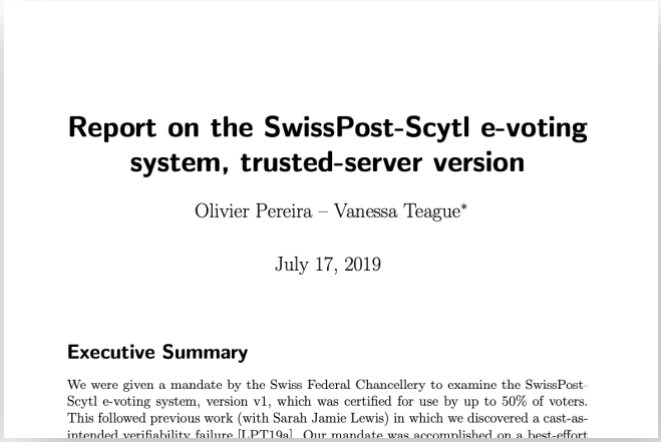
Sicherheitsüberprüfung betriebliche Aspekte E-Voting-System der Post

Kunde: Bundeskanzlei

Datum: 28.06.2019

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

oneconsult


Report on the SwissPost-Scytl e-voting system, trusted-server version

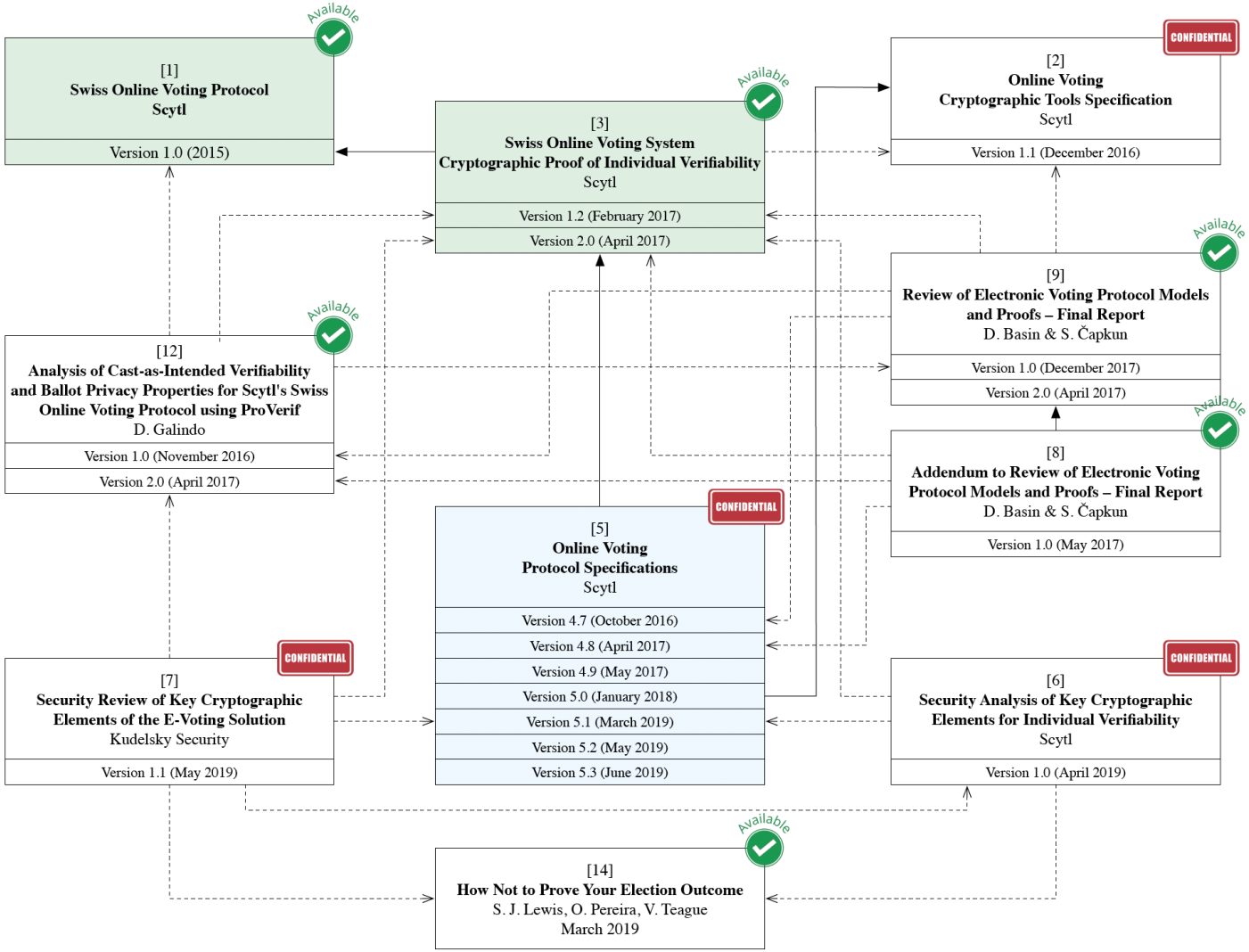
Olivier Pereira – Vanessa Teague*

July 17, 2019

Executive Summary

We were given a mandate by the Swiss Federal Chancellery to examine the SwissPost-Scytl e-voting system, version v1, which was certified for use by up to 50% of voters. This followed previous work (with Sarah Jamie Lewis) in which we discovered a cast-as-intended verifiability failure [LPT19a]. Our mandate was accomplished on a best-effort

sVote Dependencies



Source: Analysis of the Cryptographic Implementation of the Swiss Post Voting Protocol, Philipp Locher, Rolf Haenni, Reto E. Koenig, https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/berichte-und-studien.html

Current Situation in Switzerland

COUNCIL OF THE SWISS ABROAD

Expat Swiss push for e-voting despite reluctance at home

The expat Swiss assembly has given the green light for an online petition calling for the introduction of e-voting by 2021.

By Urs Geiser in Visp

UNIVERSAL VERIFIABILITY

Hackers uncover ‘significant’ flaw in Swiss Post e-voting

Hackers reported a major bug in the new Swiss Post’s e-voting system as part of a public intrusion test. Swiss Post has resolved the issue.

Mar 12, 2019 - 14:03

ONLINE DEMOCRACY

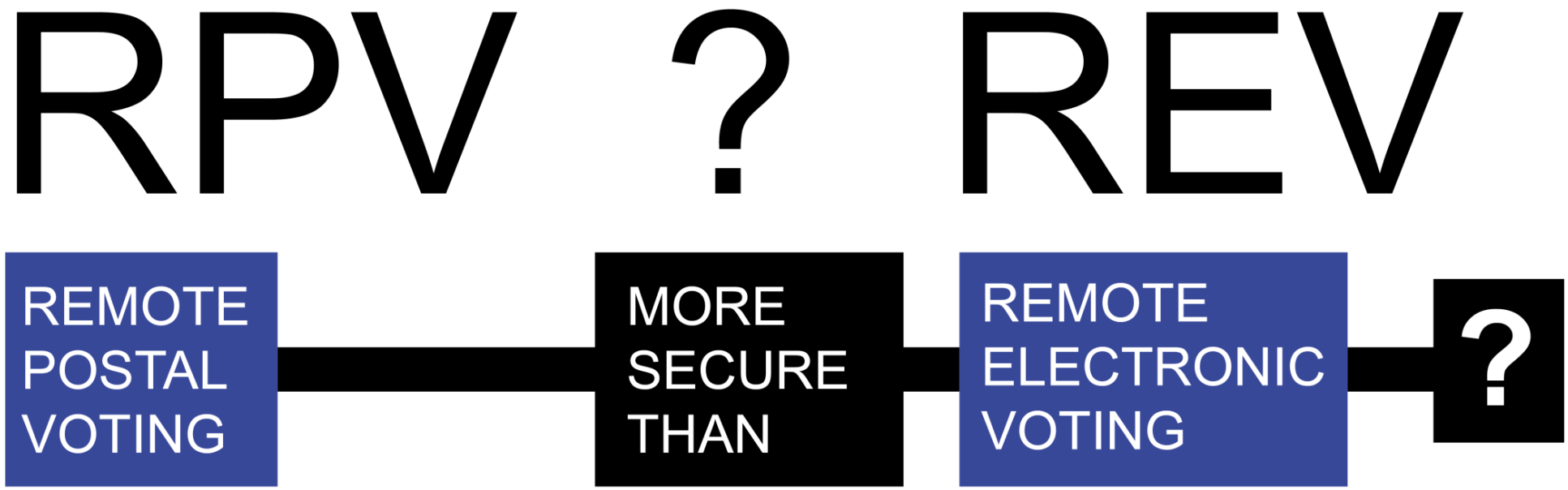
Opposition against e-voting project gathers pace

A committee of politicians and IT experts launches an initiative aimed at banning online voting for at least five years in Switzerland.

Electronic Voting (EV)

- Definitions and Notions with a Global Overview on EV
- History of Switzerland and REV
- **Focus on Remote Postal Voting (RPV) in Switzerland**

Comparing “Systems“



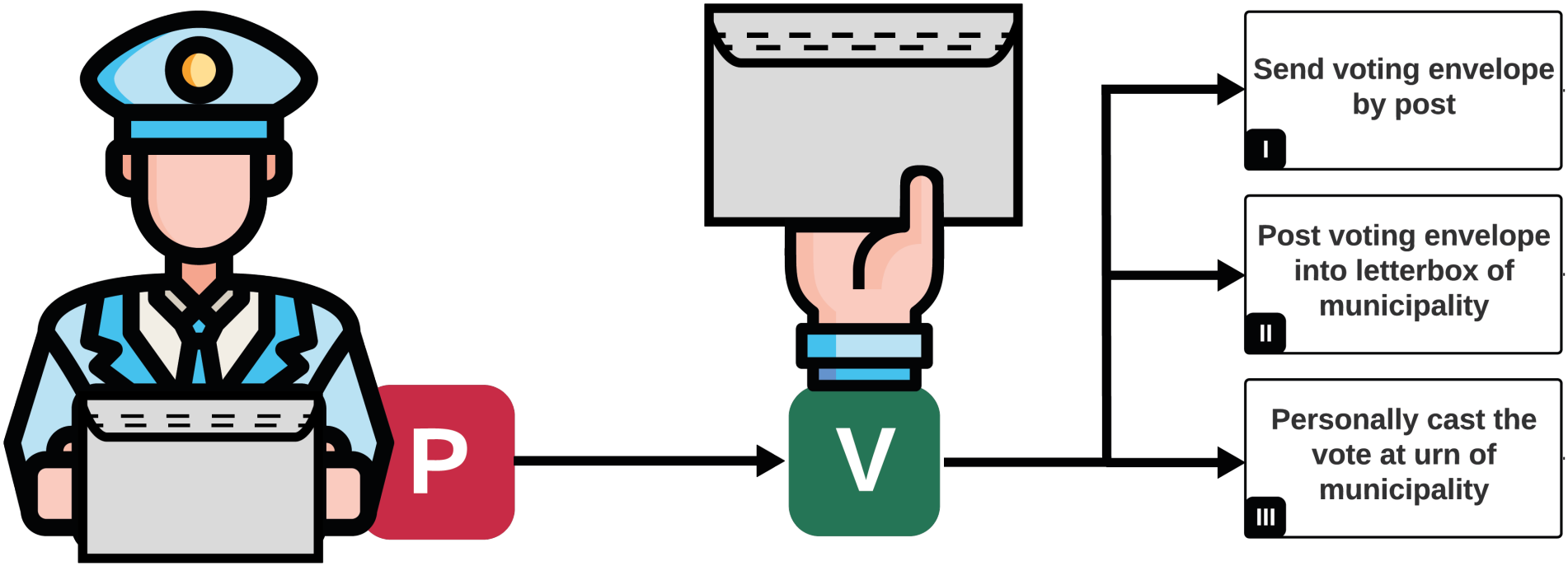
The Swiss RPV Case

- The Swiss RPV is **fragmented and difficult to generalize**, due to federalism in Switzerland, autonomy, and involvement of many external suppliers

- *Goal:*
  To identify weaknesses of RPV to allow for "hardening" of the RPV through security and risk assessment.
  - **Disclaimer:** Focus on generalization, may not cover all cantons and processes exactly, leaves room for exceptions.
  - Many exchanges with Swiss authorities and external suppliers

[KS19] C. Killer, B. Stiller: The Swiss Postal Voting Process and Its System and Security Analysis, E-VOTE-ID 2019

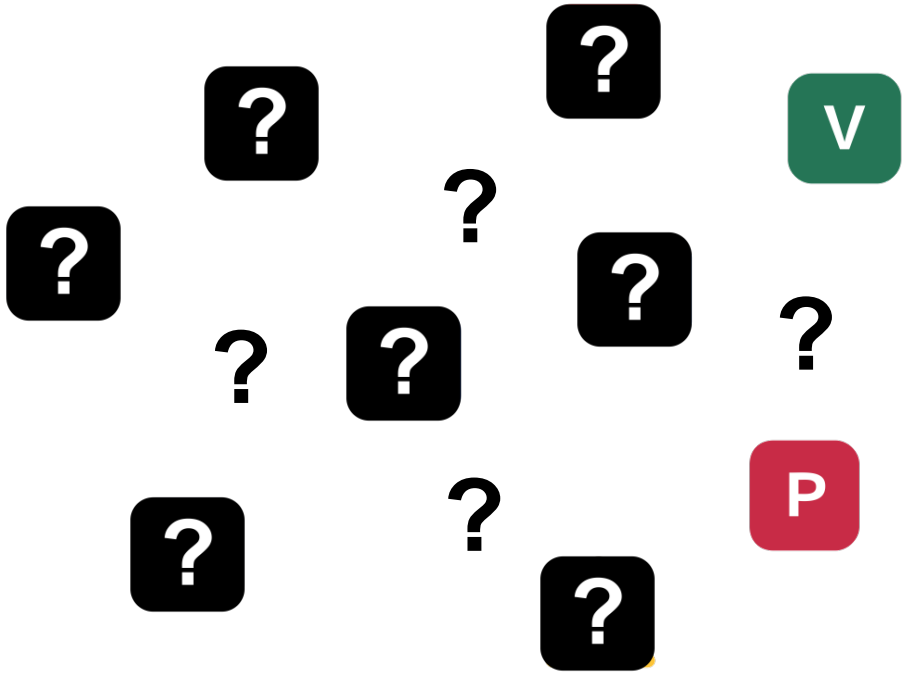
RPV From a Voter's Perspective



RPV: Remote Postal Voting

PVPF: Postal Voting Process Flow

Identification of Stakeholders

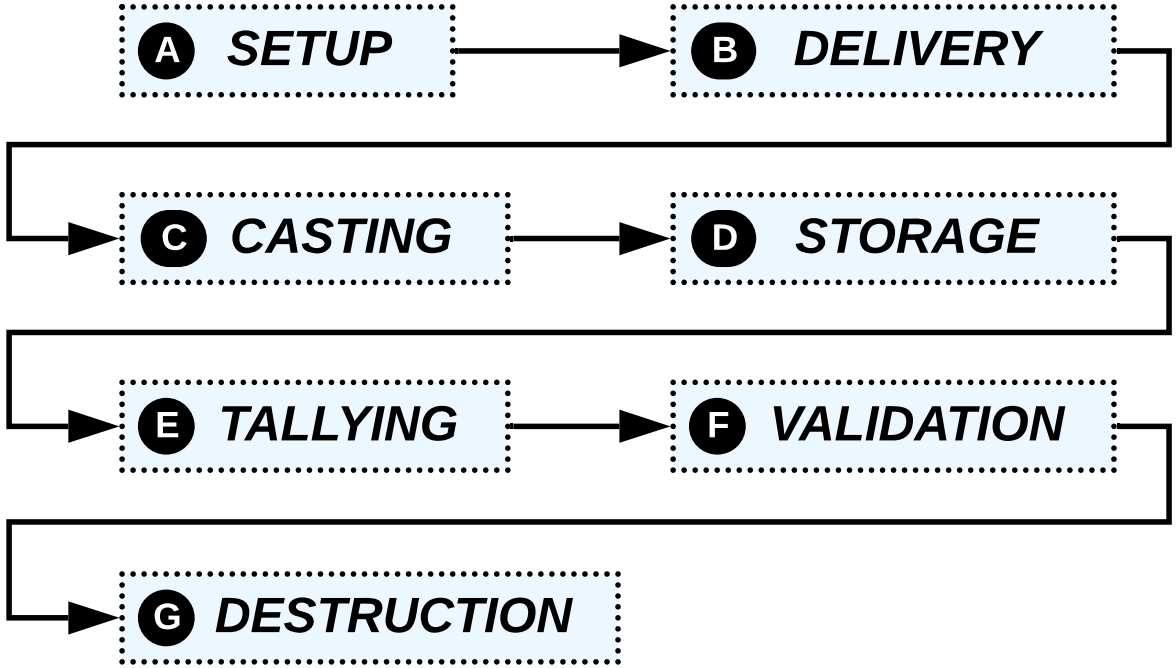


- Federal Government
- Federal Chancellery
- Cantonal Government
- Municipality
- Municipal Election Office
- Eligible Voter
- The Swiss Post
- External Supplier
- Security Threat

PVPF Phases

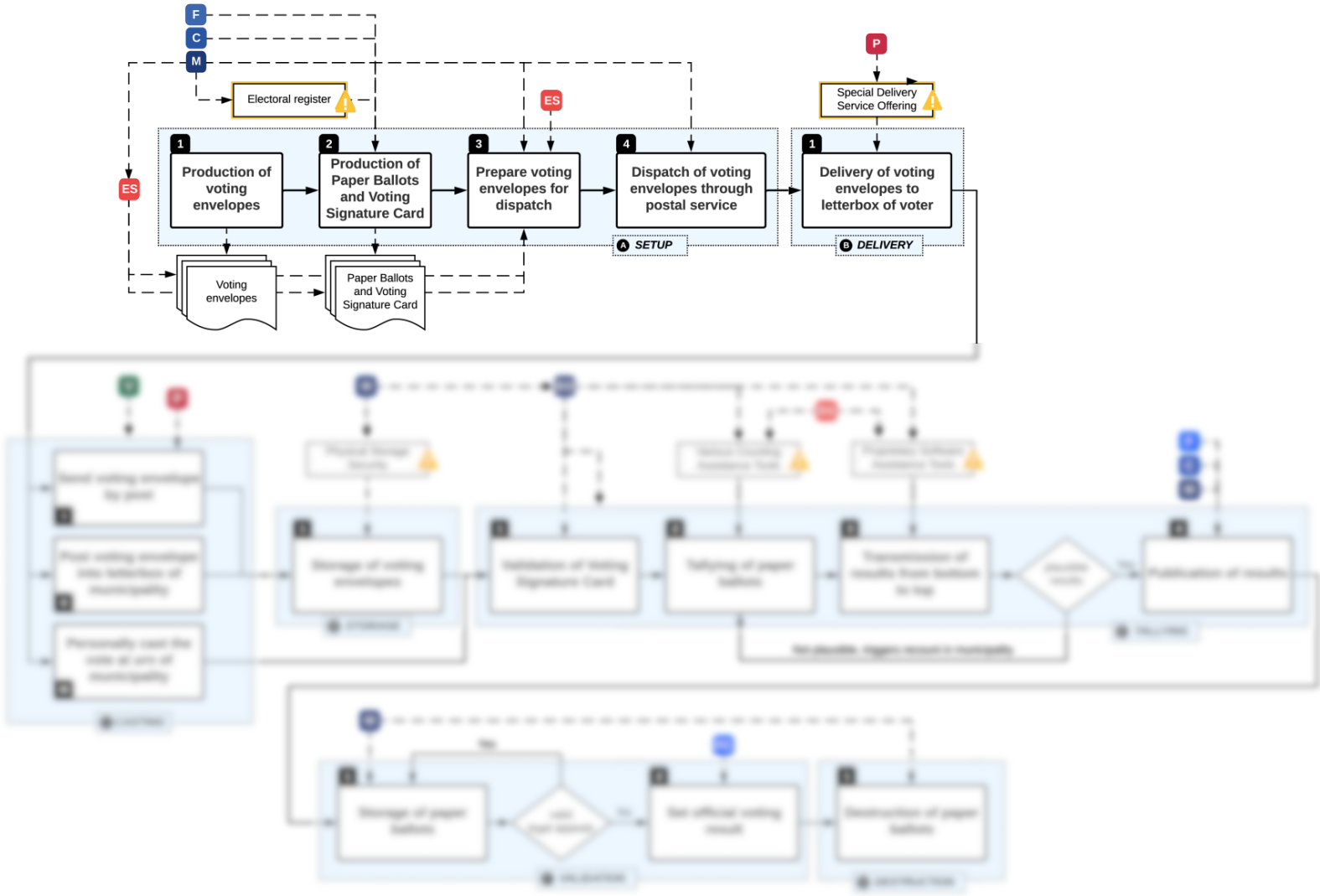
- Divided into phases A to G with various stakeholders

F Federal Government
FC Federal Chancellery
C Cantonal Government
M Municipality
EO Municipal Election Office
V Eligible Voter
P The Swiss Post
ES External Supplier
Security Threat



PVPF: Postal Voting Process Flow

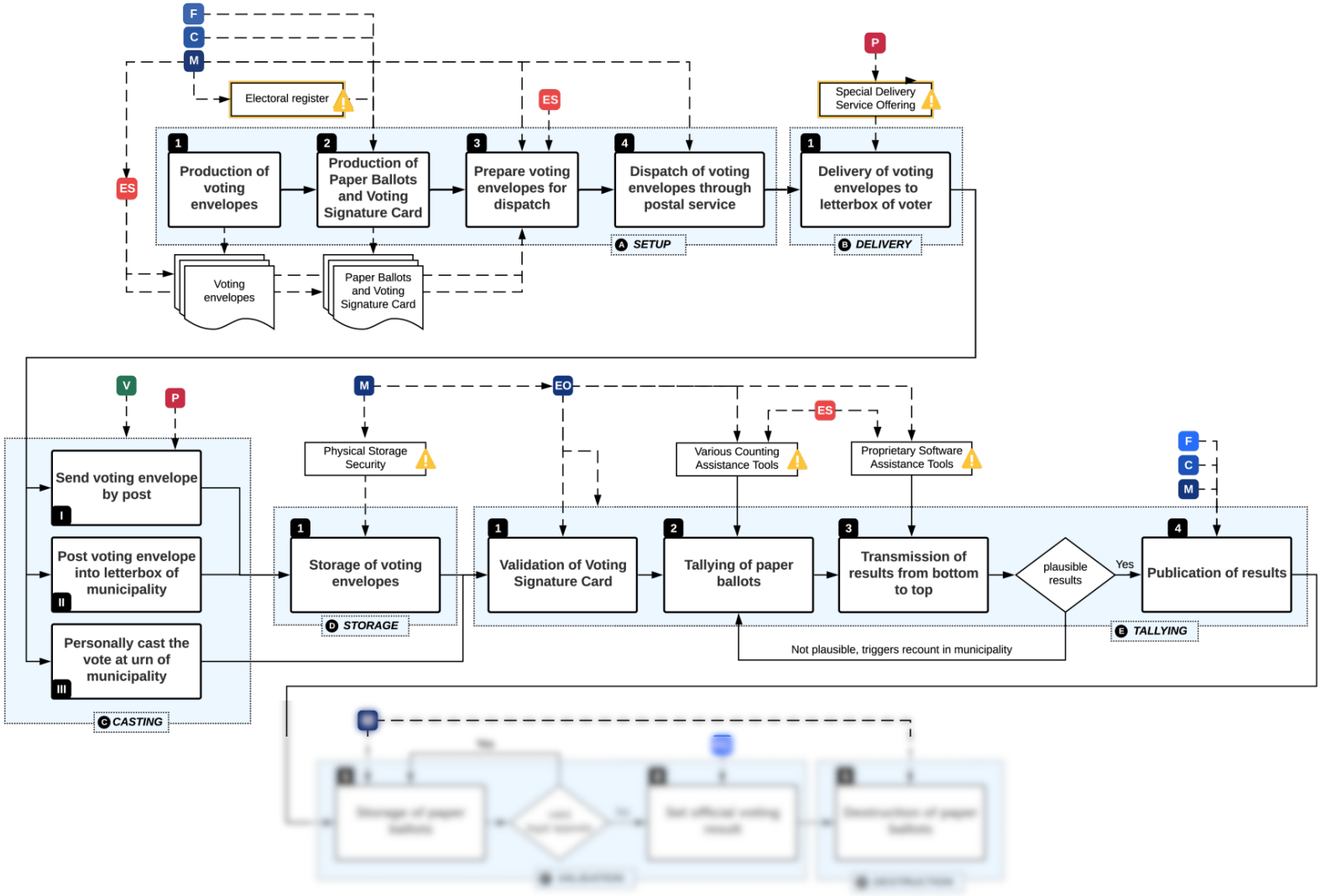
PVPF in Detail (1)



[KS19]

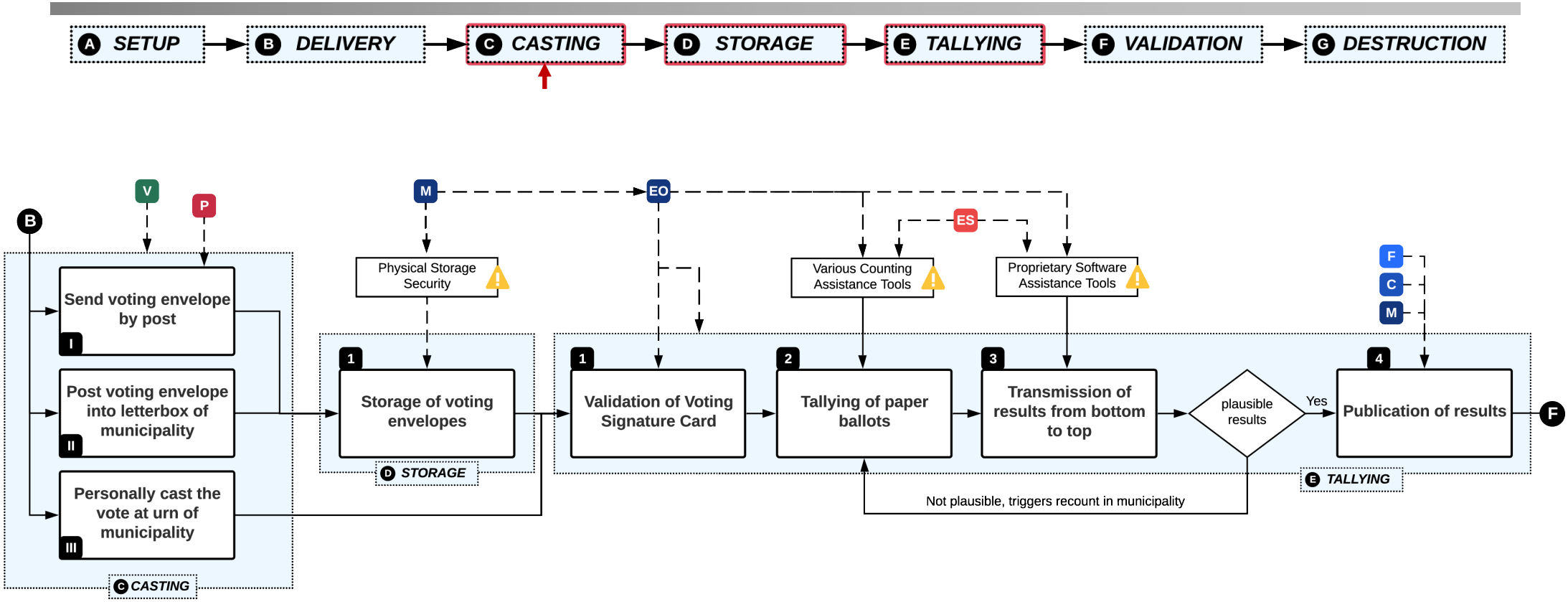
PVPF: Postal Voting Process Flow

PVPF in Detail (2)



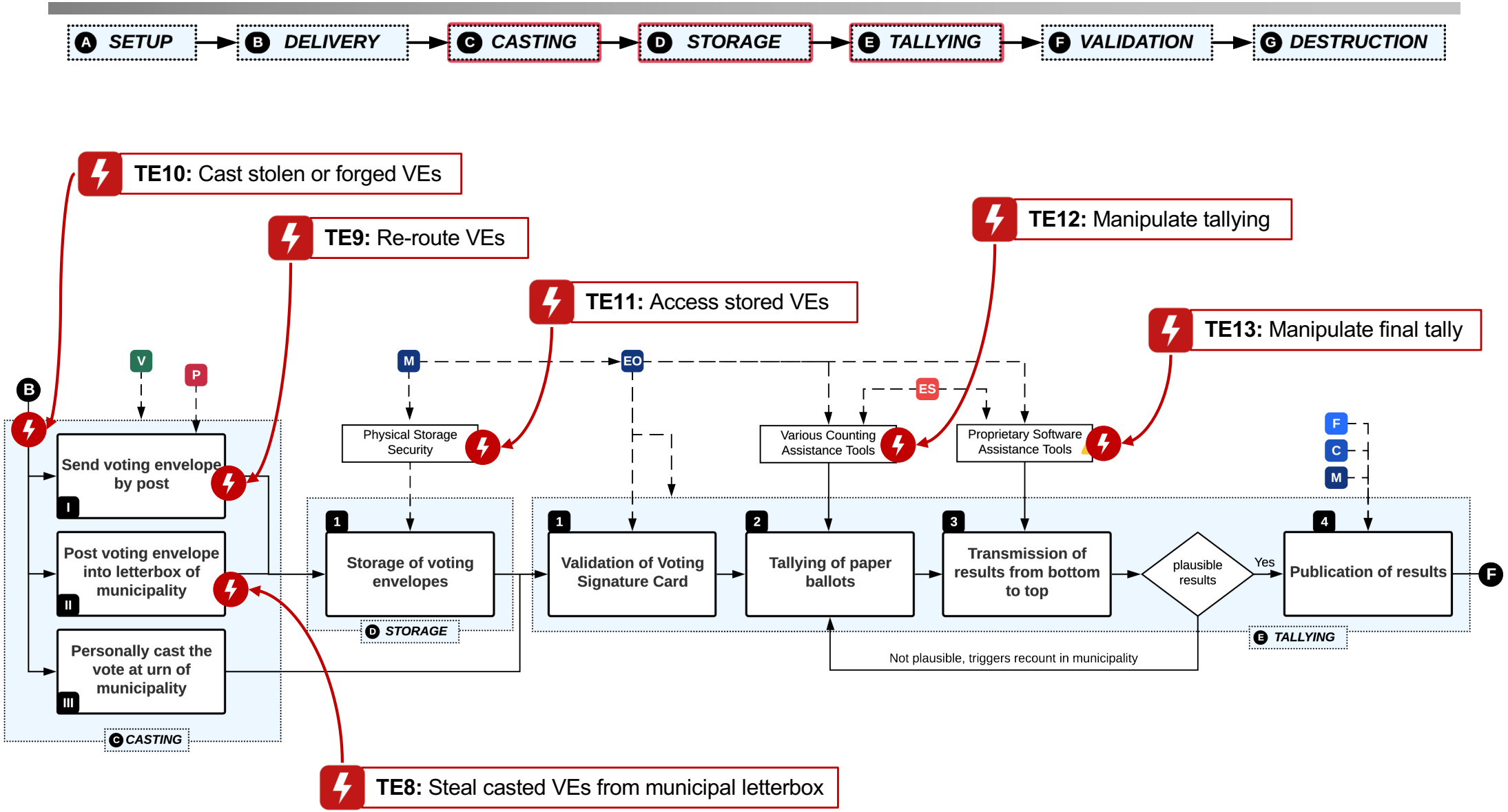
PVPF: Postal Voting Process Flow

C: Casting, D: Storage, E: Tallying

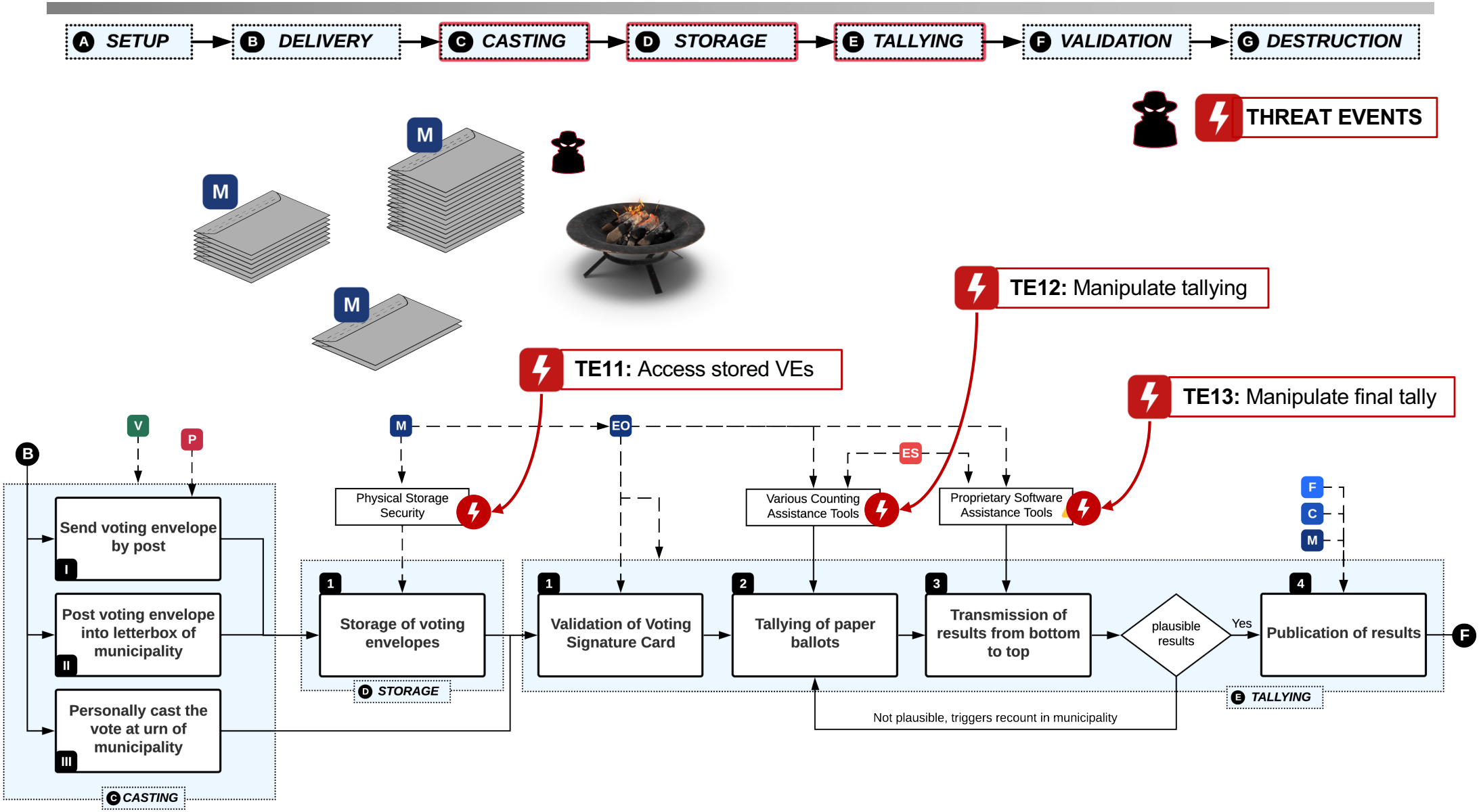


C: Casting, D: Storage, E: Tallying

THREAT EVENTS

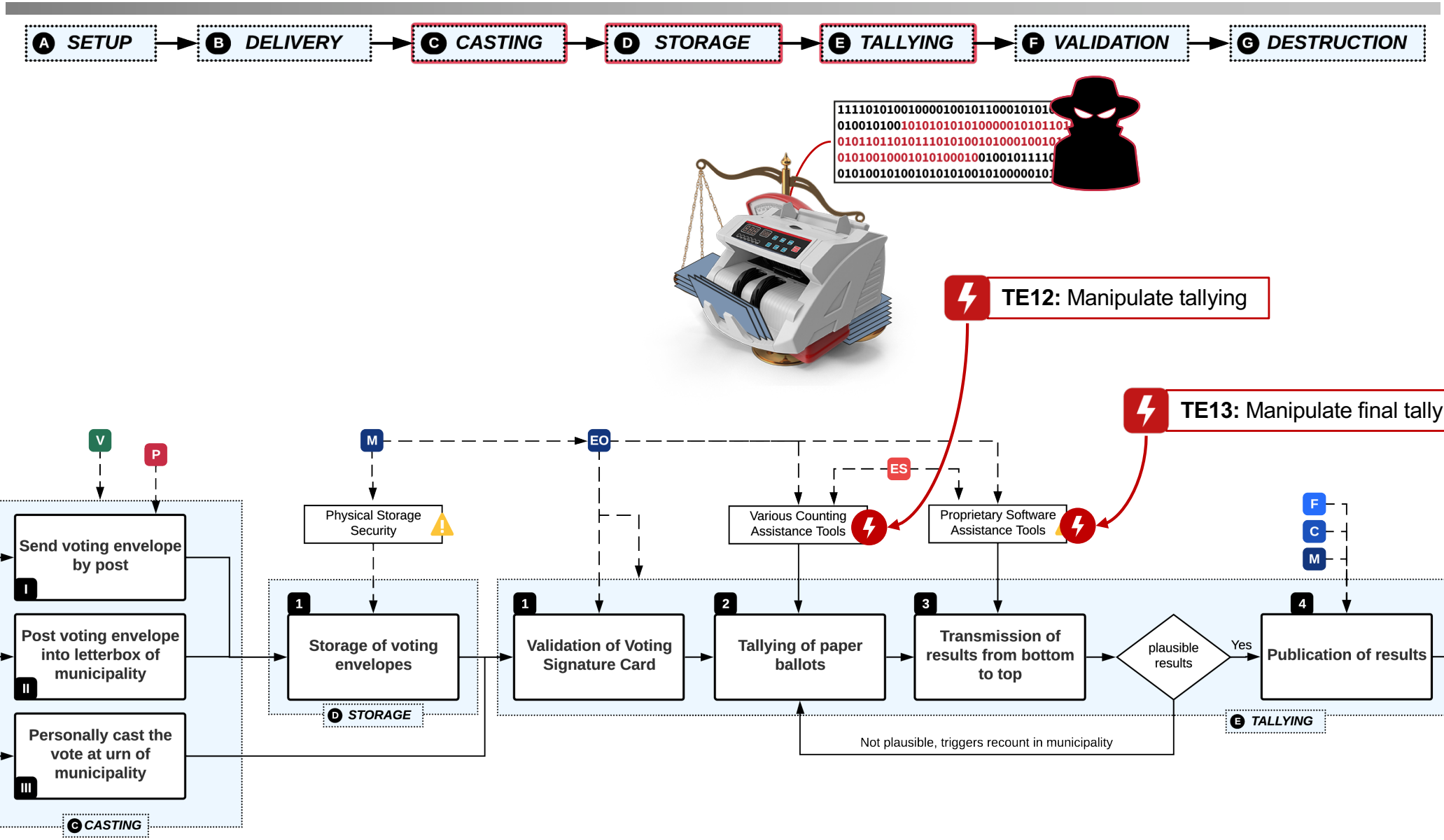


C: Casting, D: Storage, E: Tallying



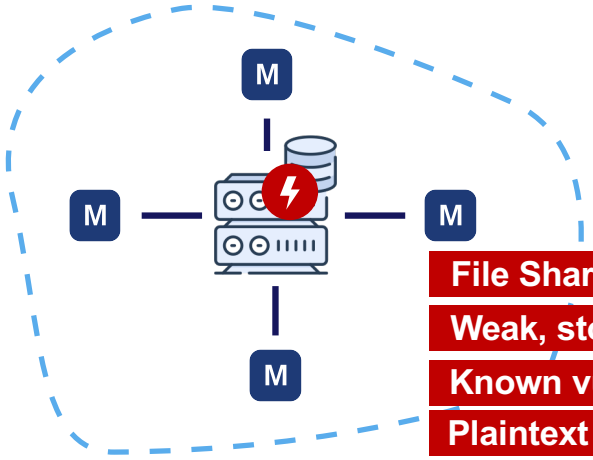
C: Casting, D: Storage, E: Tallying

THREAT EVENTS



C: Casting, D: Storage, E: Tallying

THREAT EVENTS



Phishing

Physical Access

File Shares

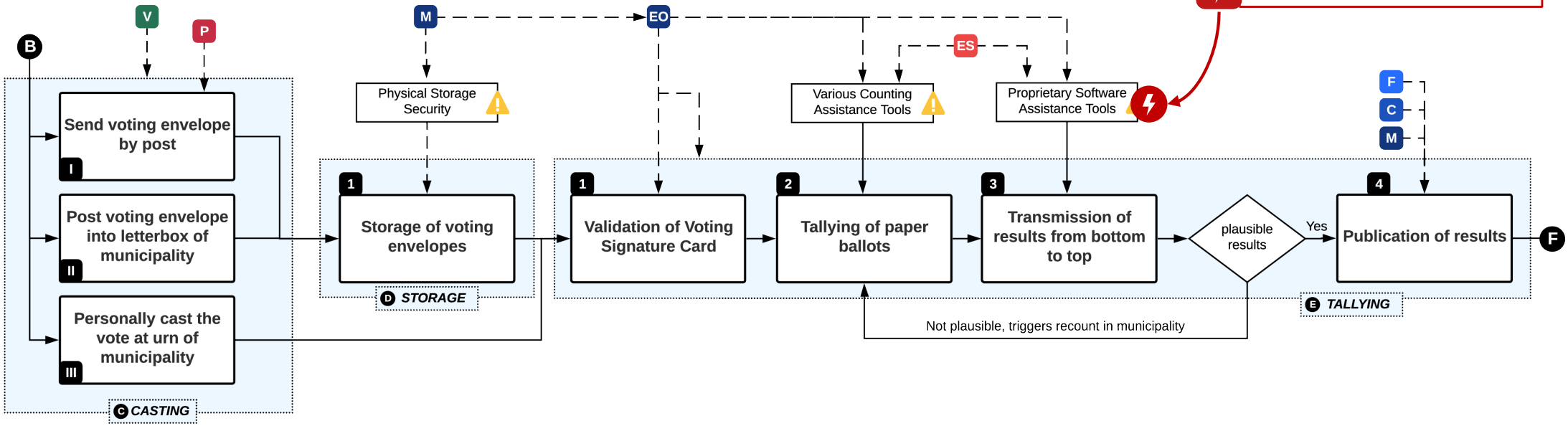
Weak, stored or default credentials

Known vulns

Authentication Protocol Abuse

Plaintext connections

TE13: Manipulate final tally



ifi

Conclusions on the Swiss RPV

- The Swiss RPV system
  - Heterogeneous process with cantonal variations 😐
  - Substantial trust in third parties ☹
  - Distribution of trust and physical decentralization ☺

- *How can a new design of an electronic voting system distribute trust?*

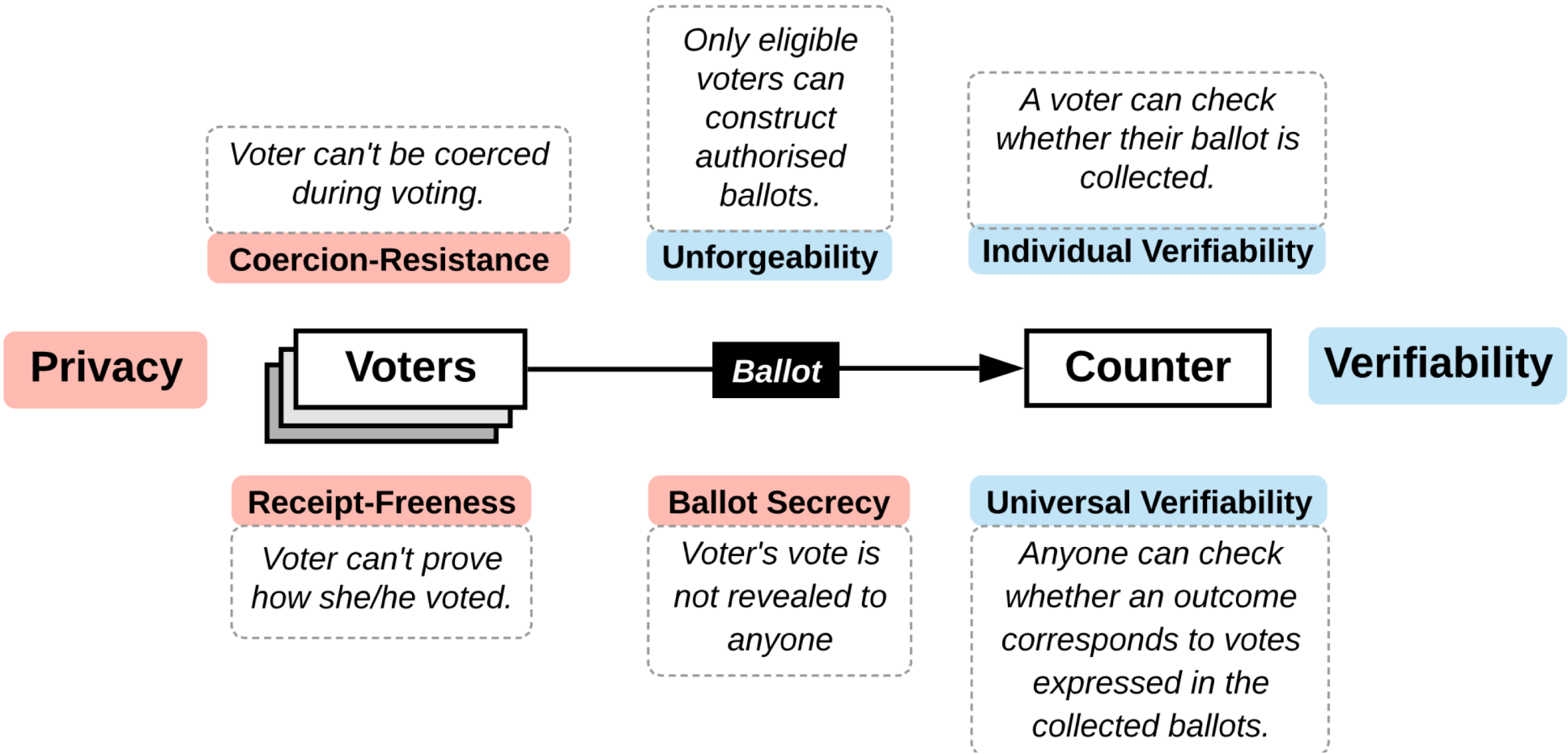
Remote Electronic Voting (REV)

- **REV Requirements and Properties**
- Security Building Blocks
  - Homomorphic Encryption, ZKP, PBB

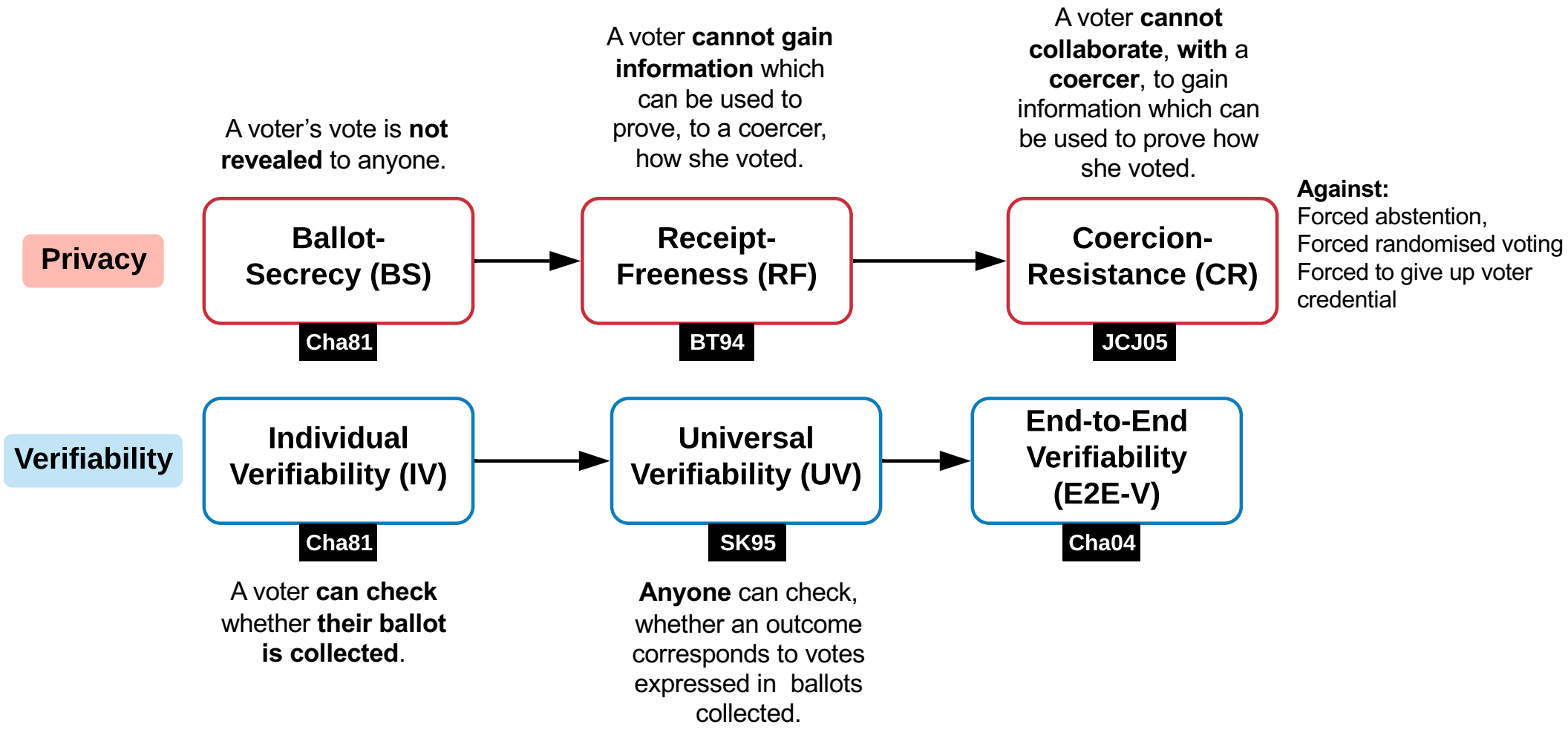
REV Requirements

- Privacy
  - A vote must not identify a voter and any traceability between the voter and its vote must be removed
- Verifiability
  - A voter should be able to verify if its vote was correctly recorded and accounted for in the final vote tally
- Absolute privacy (no information leaks) cannot coexist concurrently with verifiability (sufficient information leaked to verify the result)
  - Goal of voting system design is to find a way to enable as much privacy and verifiability simultaneously as possible

Voting and Electoral Processes

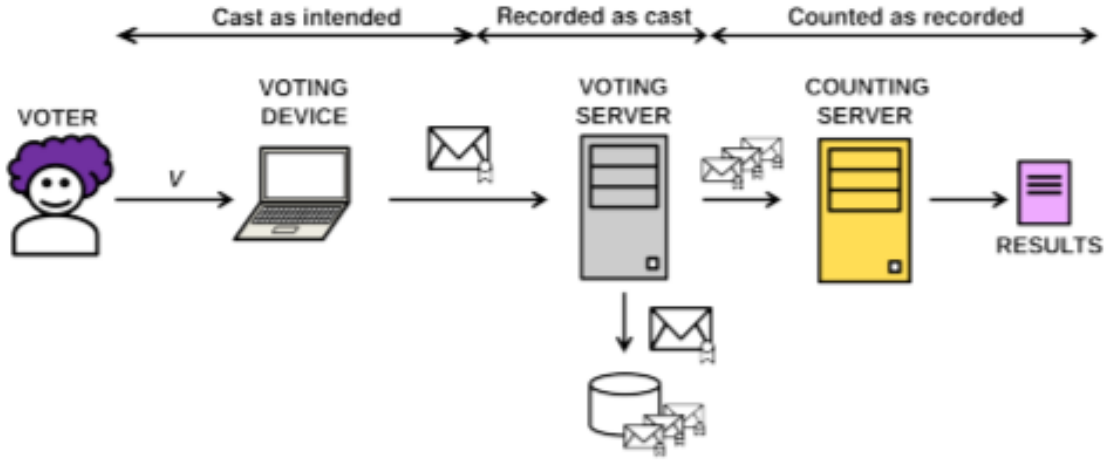


Privacy and Verifiability Notions



End-to-end Verifiability

- End-to-end Verifiability “consist” of:
  - Cast-as-Intended
    - The voter’s choice was correctly denoted on the ballot by the system
  - Recorded-As-Cast
    - The voter’s ballot was received the way she cast it
  - Count As Recorded
    - The Voter’s ballot counts as received



Fairness and Unforgeability

- Fairness
  - concerns the impossibility to deduct a partial tally, *before the end of the vote casting period*.
  - REV schemes should not allow for intermediate results, which potentially influence voters that have not voted yet.
- Eligibility Verifiability
  - anyone can check that each vote in the election outcome was cast by a registered voter and there is at most one vote per voter
- Unforgeability
  - Only eligible voters can construct authorized ballots.

Software Independence

- Software Independence
  - A voting system is software-independent if an undetected change or error in its software cannot cause an undetectable change or error in an election outcome

- Addresses the difficulty of verifying
  - Complex software in a completely electronic voting systems

- Problem: providing assurance that the voting software is in fact correct
  - Verify the election results, not the voting system

Refinements of Software Independence

- Weak Software Independence
  - A voting system that is ***weakly*** software-independent conforms to the basic definition of software-independence, that is, ***there is no recovery mechanism***

- Strong Software Independence
  - A voting system is ***strongly*** software-independent if an undetected change or error in its software cannot cause an undetectable change or error in an election outcome, and moreover, a detected change or error in an election outcome (due to change or error in the software) can be corrected without re-running the election

REV Building Blocks

- Secret-Ballot Voting Schemes, *i.e., voting protocols describing the process*
- Cryptographic Primitives, *i.e., procedures used in the protocols to assure privacy and verifiability*
- Public Bulletin Board (PBB), *i.e., ballot box saving all results and logs of the procedures, making the process verifiable*

Secret Ballot Voting Schemes

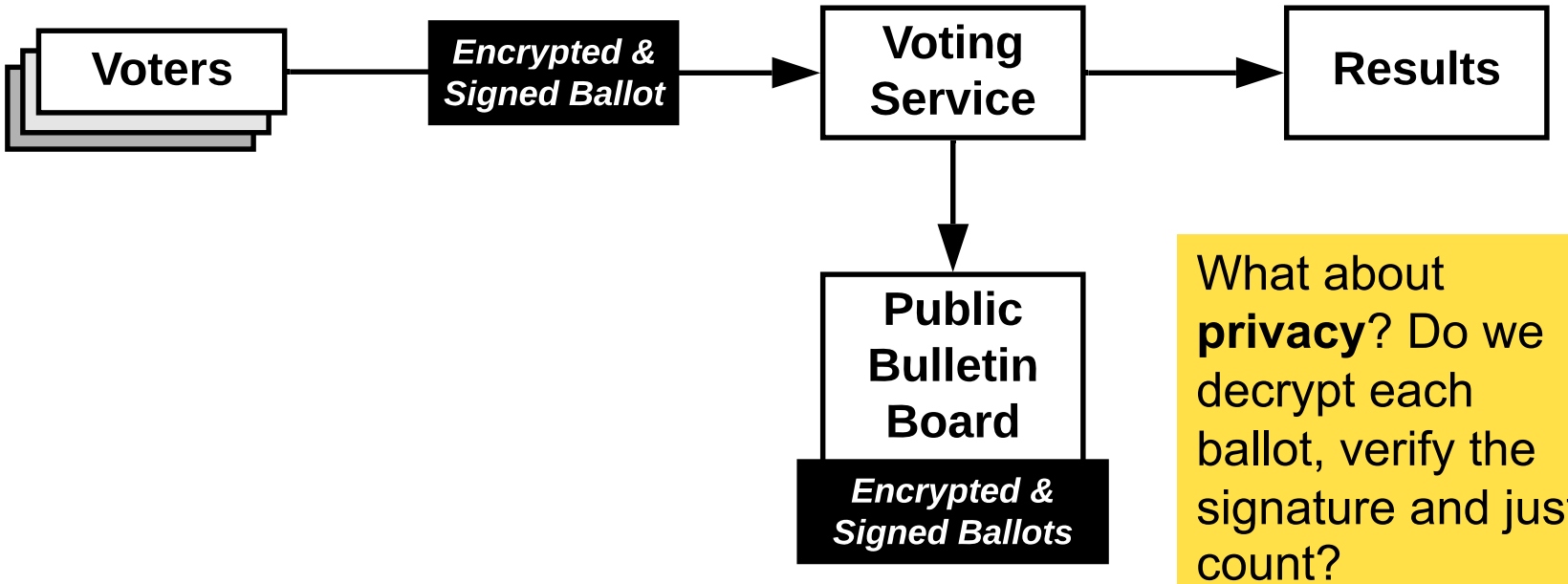
scheme	privacy	verifiability	attacks	based on
FOO92 [32]	BP	UV	KR05 [44]	BS
CGS97 [25]	BP	UV	–	HE
BT94 [15]	RF	–	HS00 [37]	HE
NR94 [58]	RF	–	–	MC
SK95 [68]	RF	UV	MH96 [50]	MN
Oka96 [60]	RF	–	Oka97 [61]	BS
Oka97 [61]	RF	IV, UV	–	BS
HS00 [37]	RF	IV,UV	–	MN
LK00 [46]	RF	UV	LBD04 [45]	HE
BFP01 [8]	RF	UV	Hir01 [36],JCJ05 [42]	HE
MBC01 [48]	RF	–	LK02 [47], JCJ05 [42]	HE
LK02 [47]	RF	UV	–	HE
LBD03 [45]	RF	UV	–	MN
ALBD04 [4]	RF	IV, UV	–	MN
JCJ05 [42]	RF, CR	UV	–	MN
MN06 [52]	EBP*, RF	UV	–	CS
MN07 [54]	EBP*, RF	UV	–	CS, HE

BP: Ballot Privacy
RF: Receipt-Freeness
CR: Coercion-Resistance
EBP: Everlasting Ballot-Privacy
IV: Individual Verifiability
UV: Universal Verifiability

BS: Blind Signatures
HE: Homomorphic Encryption
MC: Multiparty Computation
MN: Mixnet
CS: Commitment Scheme

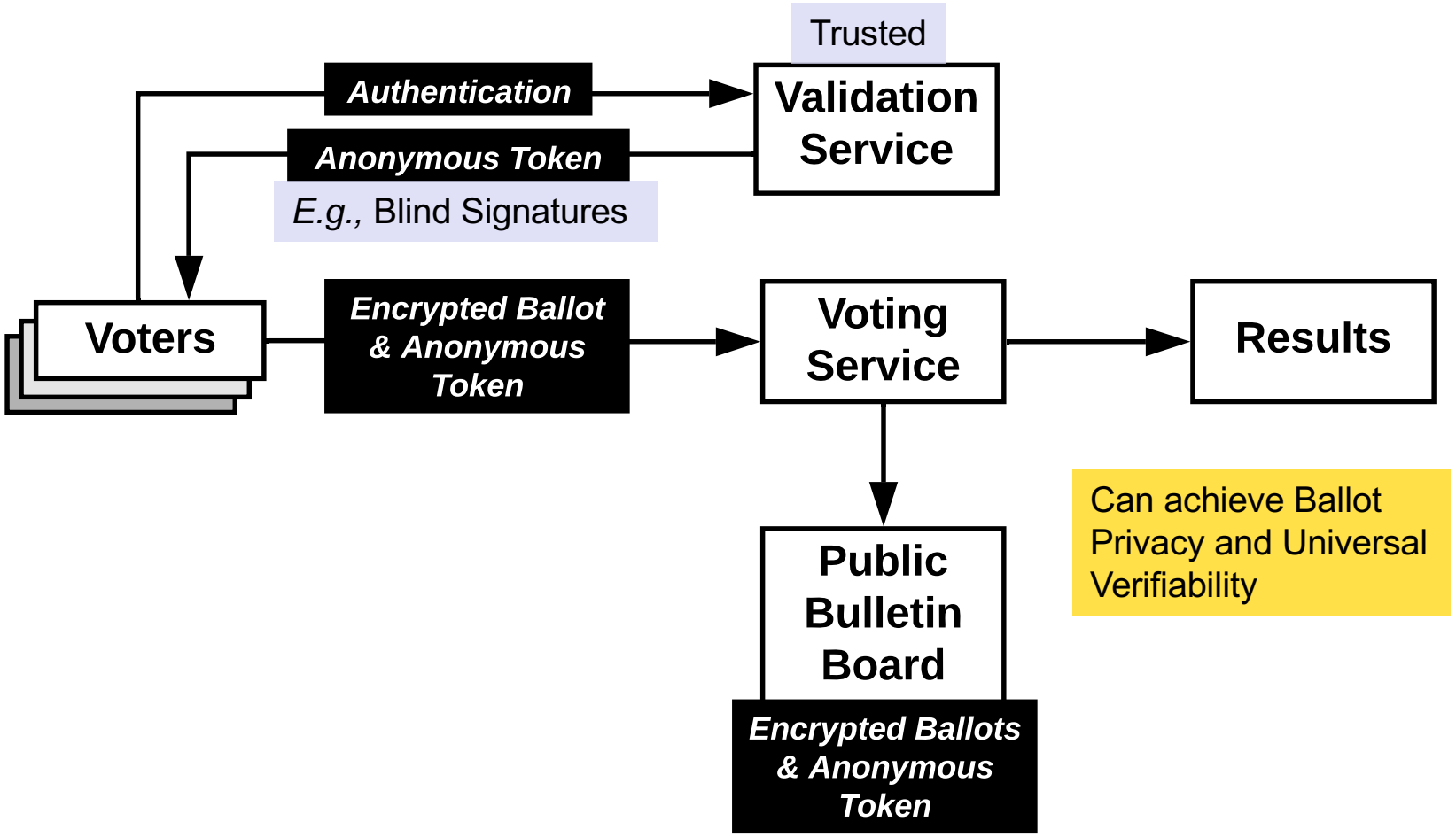
Based on H. Jonker : "Privacy and Verifiability in Voting Systems" 2013

Encrypt and Sign

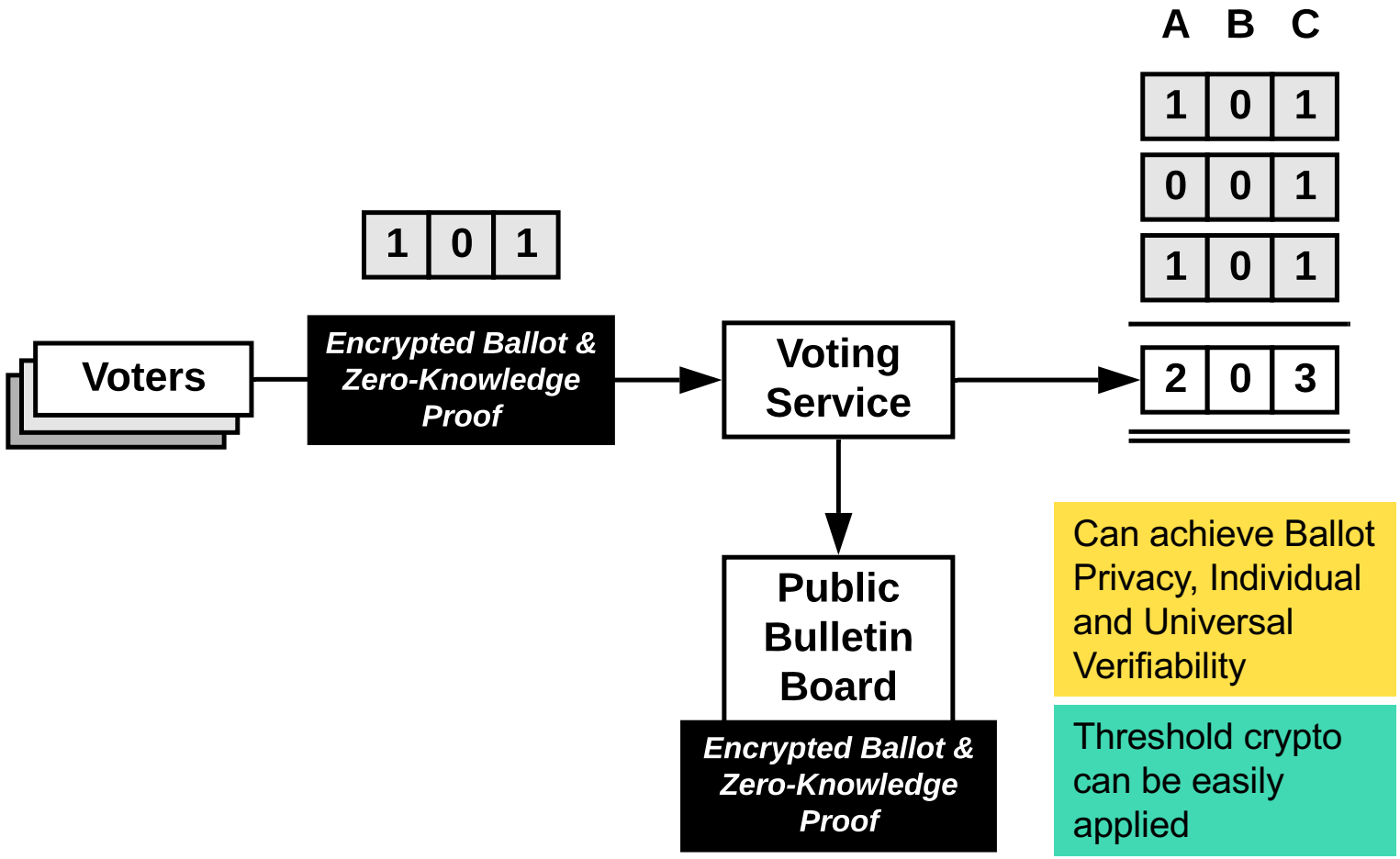


What about **privacy**? Do we decrypt each ballot, verify the signature and just count?

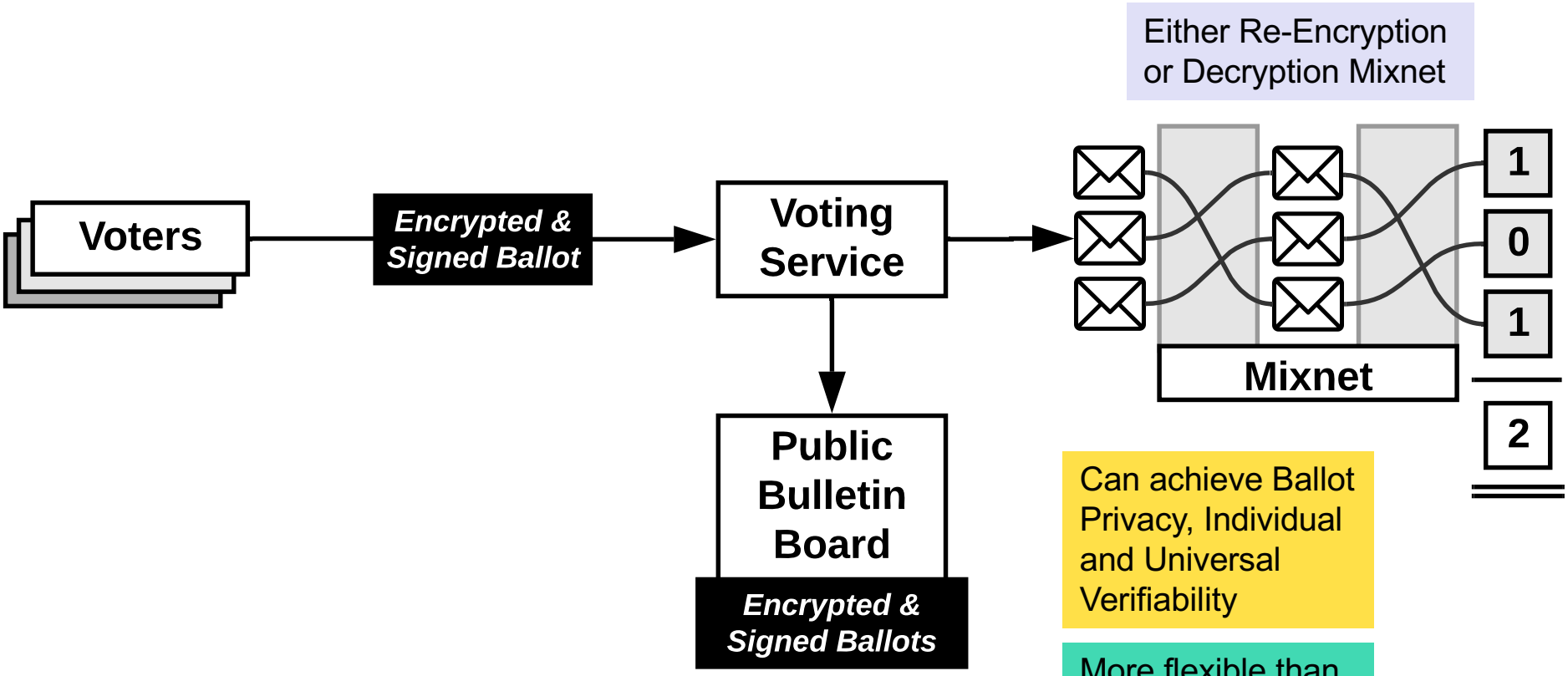
Two Agencies Model



Homomorphic Tallying



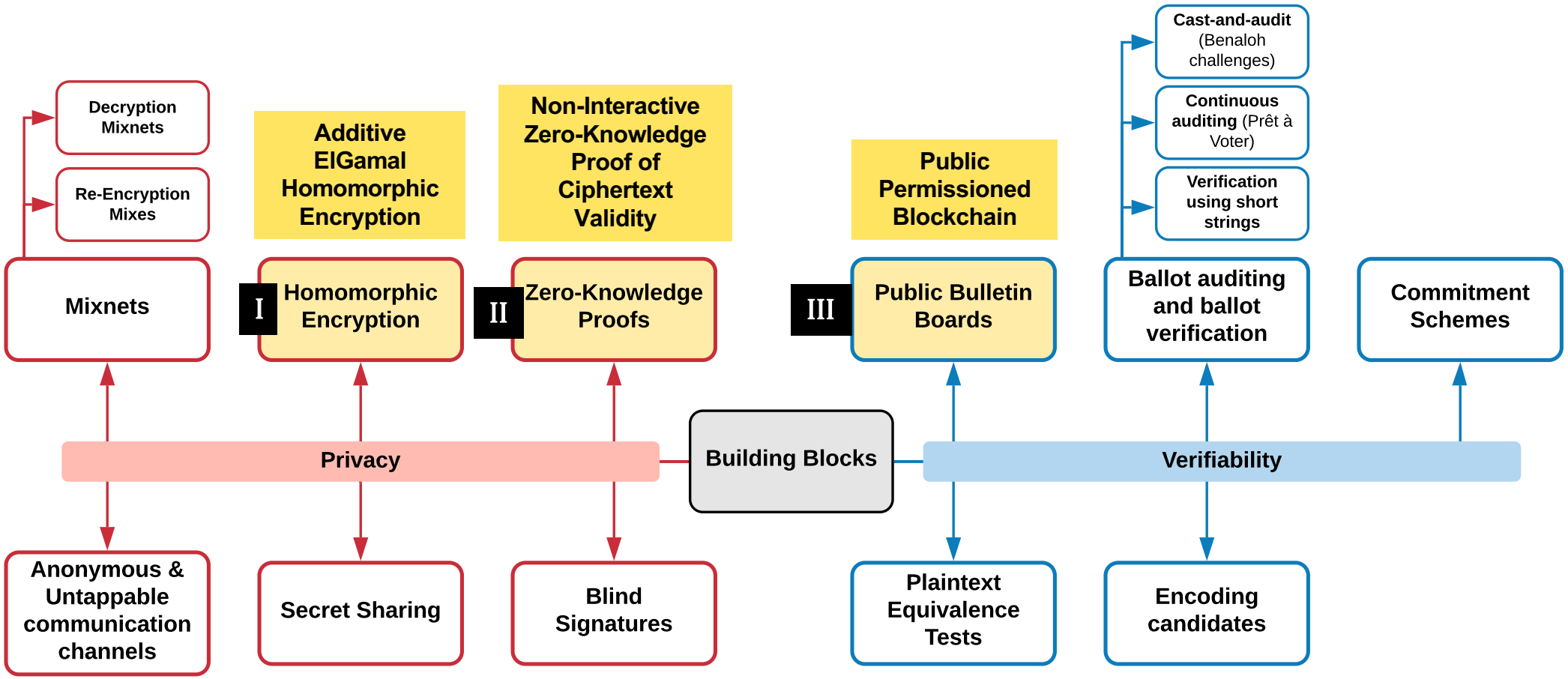
Mixnet


Either Re-Encryption or Decryption Mixnet
Voters
Encrypted & Signed Ballot
Voting Service
Public Bulletin Board
Encrypted & Signed Ballots
Mixnet
1
0
1
2
Can achieve Ballot Privacy, Individual and Universal Verifiability
More flexible than HE, supports write-ins

Remote Electronic Voting (REV)

- REV Requirements and Properties
- **Building Blocks**
  - Homomorphic Encryption, ZKP, PBB

Building Blocks for REV Schemes



Homomorphic Encryption (1)

- A public key cryptosystem

$$E = (KeyGen, Encrypt, Decrypt)$$

  - Is homomorphic if there are these additional operations:
    - An operation $+$ on the message space
    - An algorithm Add that takes a public key pk and two ciphertexts $c1, c2$ and outputs another ciphertext s
    - The correctness condition is that for any message $m1, m2$ the following returns $d = m1 + m2$:

$$(pk, sk) \leftarrow KeyGen()$$
$$c_1 \leftarrow Encrypt(pk, m_1)$$
$$c_2 \leftarrow Encrypt(pk, m_2)$$
$$c = Add(pk, c_1, c_2);$$
$$d = Decrypt(sk, c);$$

RED: **Secrets**, should never be divulged to anyone.
BLUE: **public** information, are known to everyone

Homomorphic Encryption (2)

- **Homomorphic Encryption (HE)**

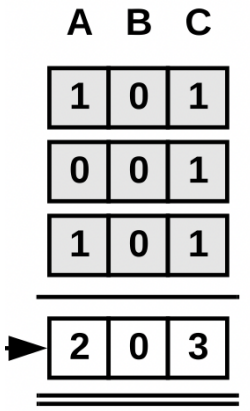
$$enc_k(m_1) \otimes enc_k(m_2) = enc_k(m_1 \oplus m_2)$$

  - $\otimes$ *appropriate operation on ciphertext messages*
  - $\oplus$ *appropriate operation on plaintext messages*
- Allows two encrypted messages to be combined meaningfully (by using $\oplus$)
- The actual operations represented by $\otimes$ and $\oplus$ depend on the used cryptographic system.
  - A variant of ElGamal, called exponential ElGamal, provides **additive** homomorhic properties.

Homomorphic Encryption (3)

- **In REV, HE used to tally votes**
- Every eligible voter V can encrypt his or her $vote$ using the election public key pk, resulting in the encrypted vote: $enc_{pk}(vote_V)$.
- No individual votes need to be decrypted, because the votes can be added together homomorphically.
- Thus, the final, encrypted tally is

$$enc_{pk}(vote_1) \otimes \ldots \otimes enc_{pk}(vote_n) = enc_{pk}(vote_1 \oplus \ldots \oplus vote_n)$$



Proving Ciphertext Validity II

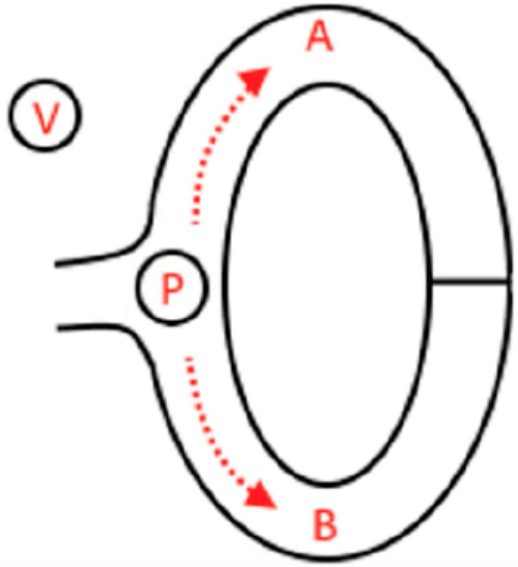
- But how encryption malleability tackled?
  - *I.e.* What happens if 100 instead of 1 is encrypted?
  - In search of a solution to assure the validity of the ciphertext, while assuring Ballot Privacy at the same time
- ***Zero Knowledge Proofs!***
  - Each voter proves that she cast a valid vote, without revealing her/his vote
  - Authorities prove that they know their secret keys (that match the election public keys), without revealing their secret keys
  - At the end of the election, authorities prove that they have tallied correctly (decrypted the results correctly), again without revealing their secret keys

Zero Knowledge Proofs II

- **Zero Knowledge Proof** (ZKP) is a proof of a statement s which reveals nothing about statement s, except that it is true
  - With the use of a ZKP system, a prover P can convince a verifier V via an interaction that some statement s is true.
- **Non-interactive ZKP** (NIZKP)
  - This relies on the prover generating a (to him) unpredictable challenge (which cannot be tweaked) based on his commitment and public knowledge, such as the hash of the commitment.
  - *In REV: NIZKPs enable a voter to verify that her vote is valid, being able to prove to an outsider what she voted.*

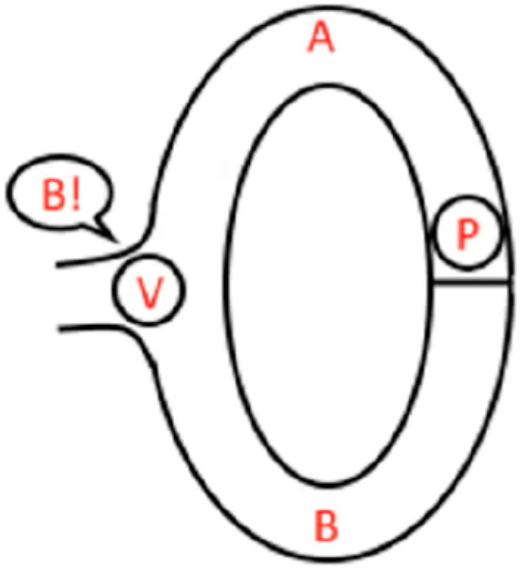
Ali Baba‘s Cave (1) II

- The cave has a single entry and is shaped like a ring with two paths (A and B) and contains a hidden door separating the two paths
- Peggy (P – “prover”) knows the password to open the hidden door but does not want to reveal it
- To prove to Victor (V – “verifier”) that Peggy knows the password, Peggy enters the cave and chooses a path



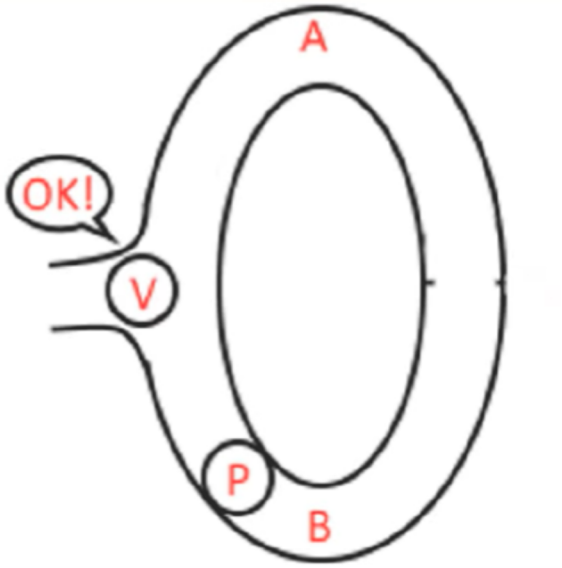
Ali Baba‘s Cave (2) II

- Next, Victor enters the cave and demands that Peggy exits the cave using path A or B, chosen at random
  - On one hand, since P knows the password, it is trivial for her to pass the test
  - On the other hand, would P had not known the password, she would have had a 50% chance only of returning from the path demanded

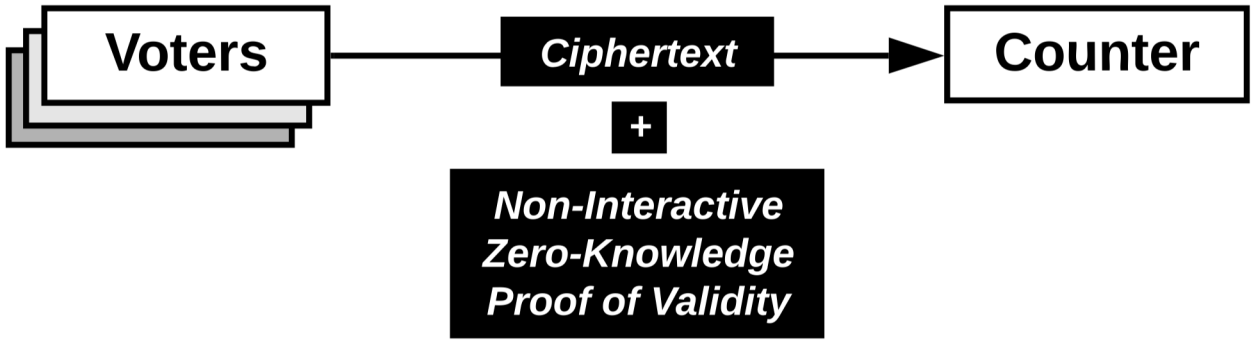


Ali Baba‘s Cave (3) II

- Victor can challenge Peggy multiple times, thereby, decreasing the probability of Peggy being a fraudulent “prover”
  - For each challenge the probability decreases by a factor of two
  - Therefore, Victor can repeat this process as many times as deemed necessary, until the desired level of confidence is obtained



Achieving Privacy with HE and ZKPs II



- Ballot Privacy achieved by encrypting with the public election key, and proving validity with NIZKPs
  - Thus, no individual ballot needs to be decrypted and authorities keep their secret election keys private, while homomorphically tallying the final result

- *How can voters verify election data?*

Public Bulletin Board (PBB) III

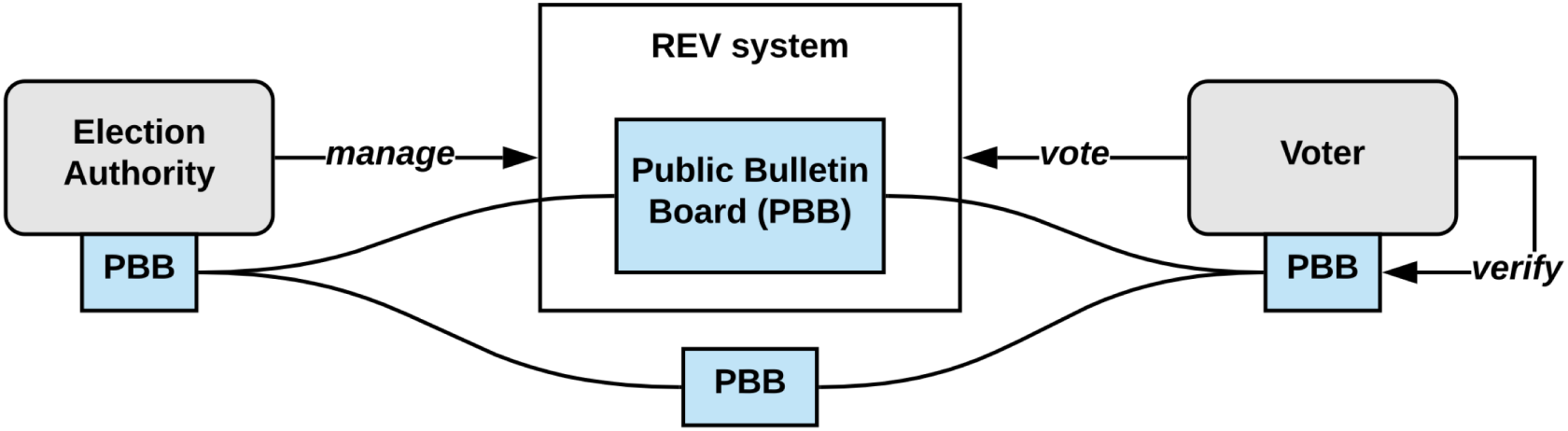
- PBB serving as a
  - **(1) shared**, **append-only** data structure
  - **(2) public** and searchable by anyone
  - **(3) consistent** view for anyone accessing its information

- In practice, a **public permissioned Blockchain (BC)** fulfills those theoretical requirements

[KRM20]

Public Permissioned Blockchain (BC) III

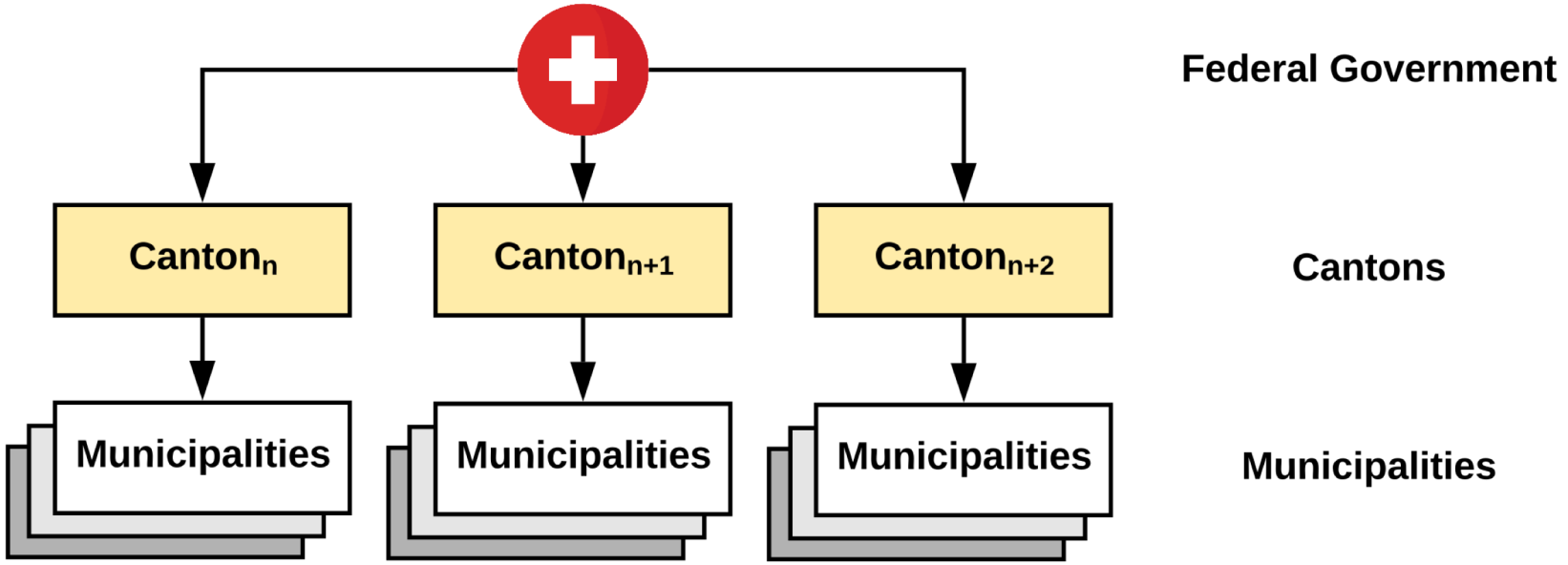
- **Deployment** of a **Public Permissioned BC**
  - Read operations: Public
  - Write operations: Permissioned



[KRM20]

The Swiss Scenario

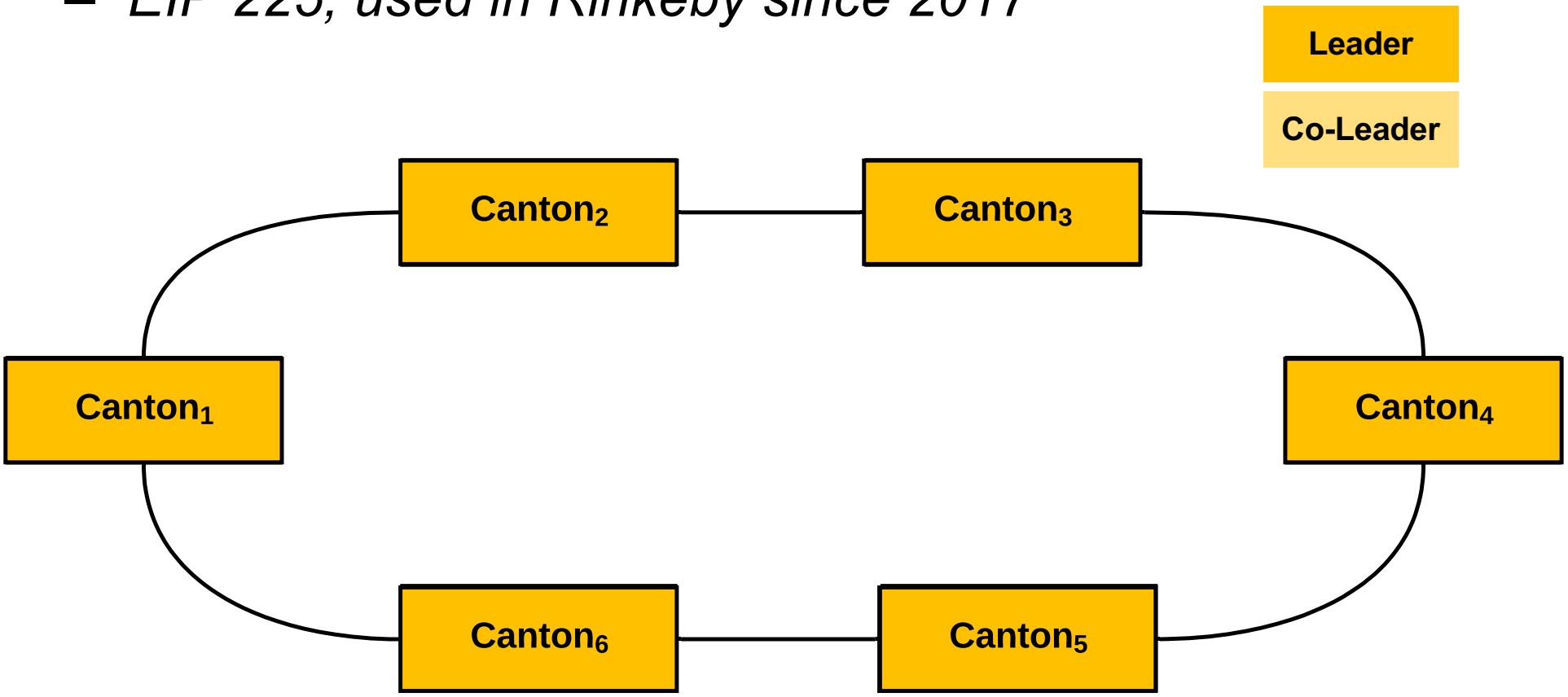
- The Federalism of Switzerland serves as a suitable framework for a **decentralized** BC-based REV system



[KRM20]

Proof-of-Authority (PoA) Consensus

- Consensus Algorithm: Clique Proof-of-Authority (PoA)
  - *EIP 225, used in Rinkeby since 2017*



Blockchain-based REV

- Instructions for local deployment (Practical Part)
- **Provotum Stakeholders, Voting Protocol and Evaluations**

Practical Part

- To follow along the next steps on your own system, please refer to the guide provided at
  - http://bcbev.ch/icbc20
  - Project implemented in Typescript and Solidity, organized as Monorepo, dockerized environment

- Prebuilt docker application available, by executing

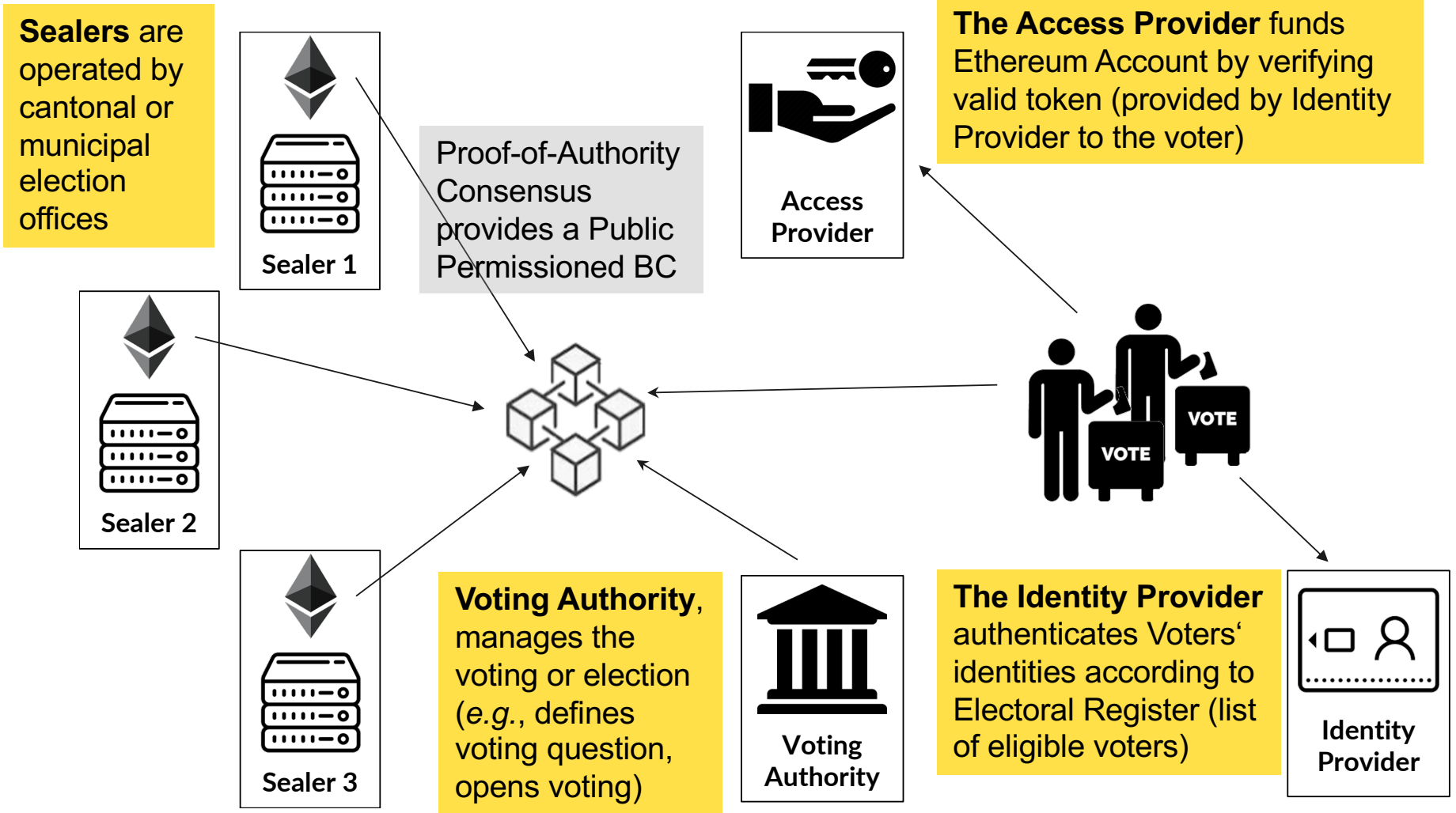
```
./docker-prebuilt-up.sh
```

  - Works on Windows or Linux, not on macOS (docker network)
  - ➢ Make sure to set up your github.json with an access token, otherwise you cannot pull from GitHub Packages

Blockchain-based REV

- Instructions for local deployment (Practical Part)
- **Provotum Stakeholders, Voting Protocol and Evaluations**

BC-based REV System: *Provotum*



Provotum 2.0 Identity Provisioning

REGISTRATION

PAIRING

KEY GENERATION

VOTING

TALLYING

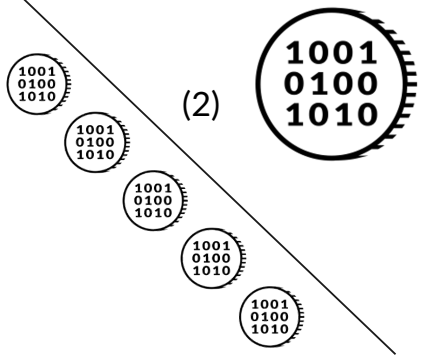
RESULT

Identity Provisioning not part of the official voting steps and

performed

off-chain

A shuffled list of tokens is provided to the Access Provider



Tokens will be used as a means to assure eligibility

Electoral Register (list of eligible voters) is provided to the Identity Provider, who generates a token for every voter

(1)

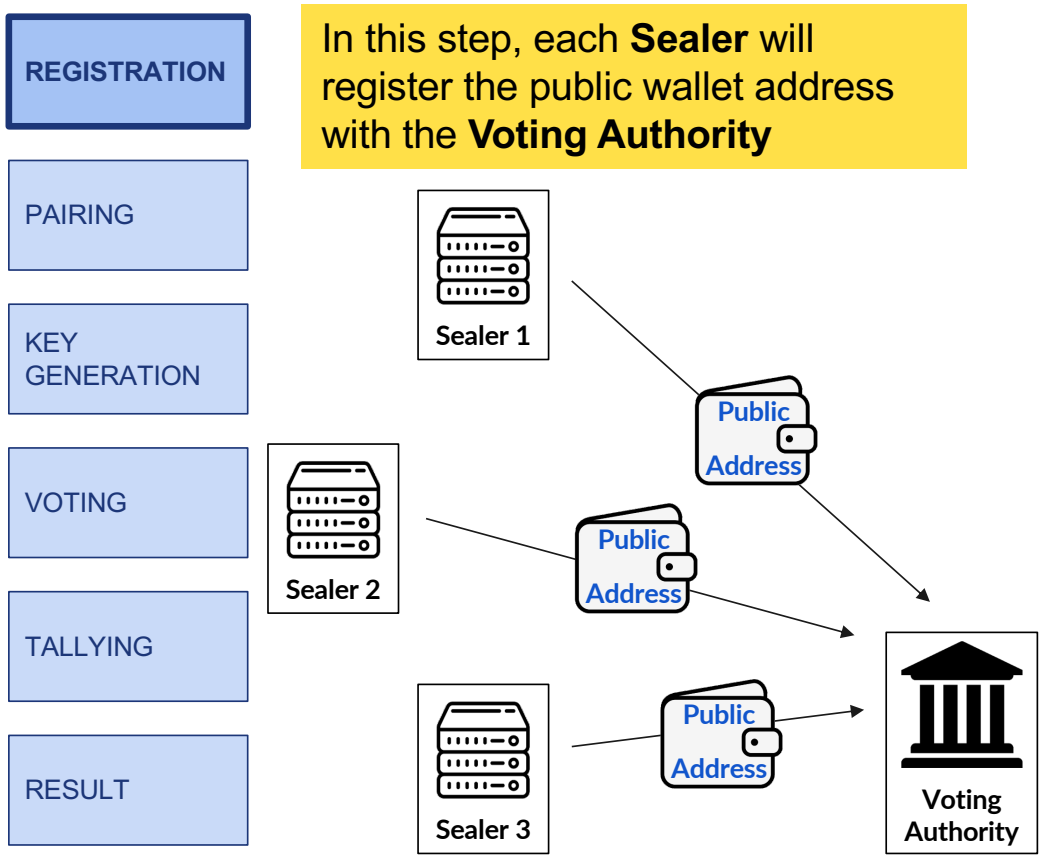
Disclaimer: Out of scope of the voting protocol, experimental process only to make the system run!

Voting Protocol: Registration (1)

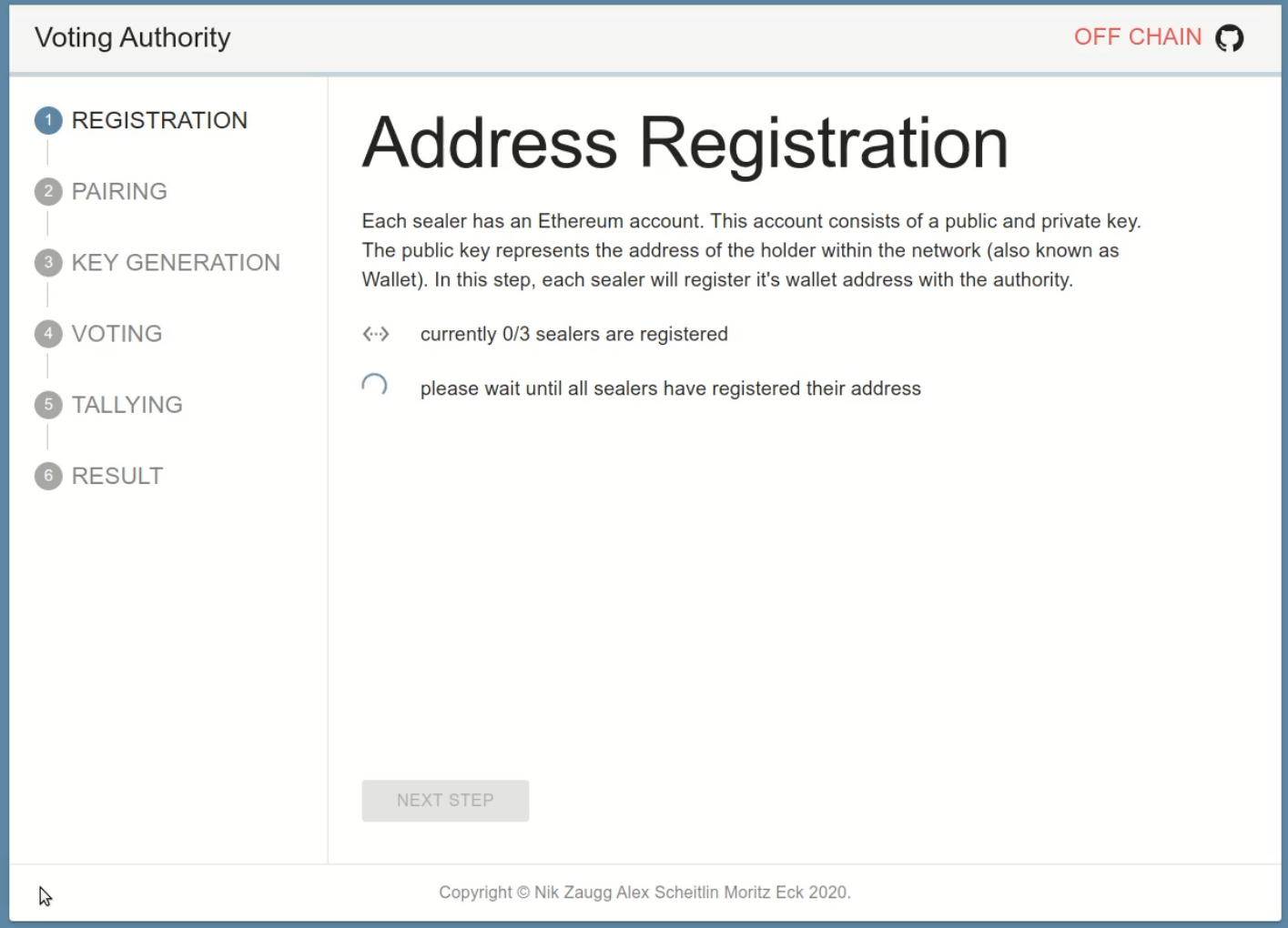
Sealer Sign-Up and Automated PoA Bootstrapping

performed

off-chain

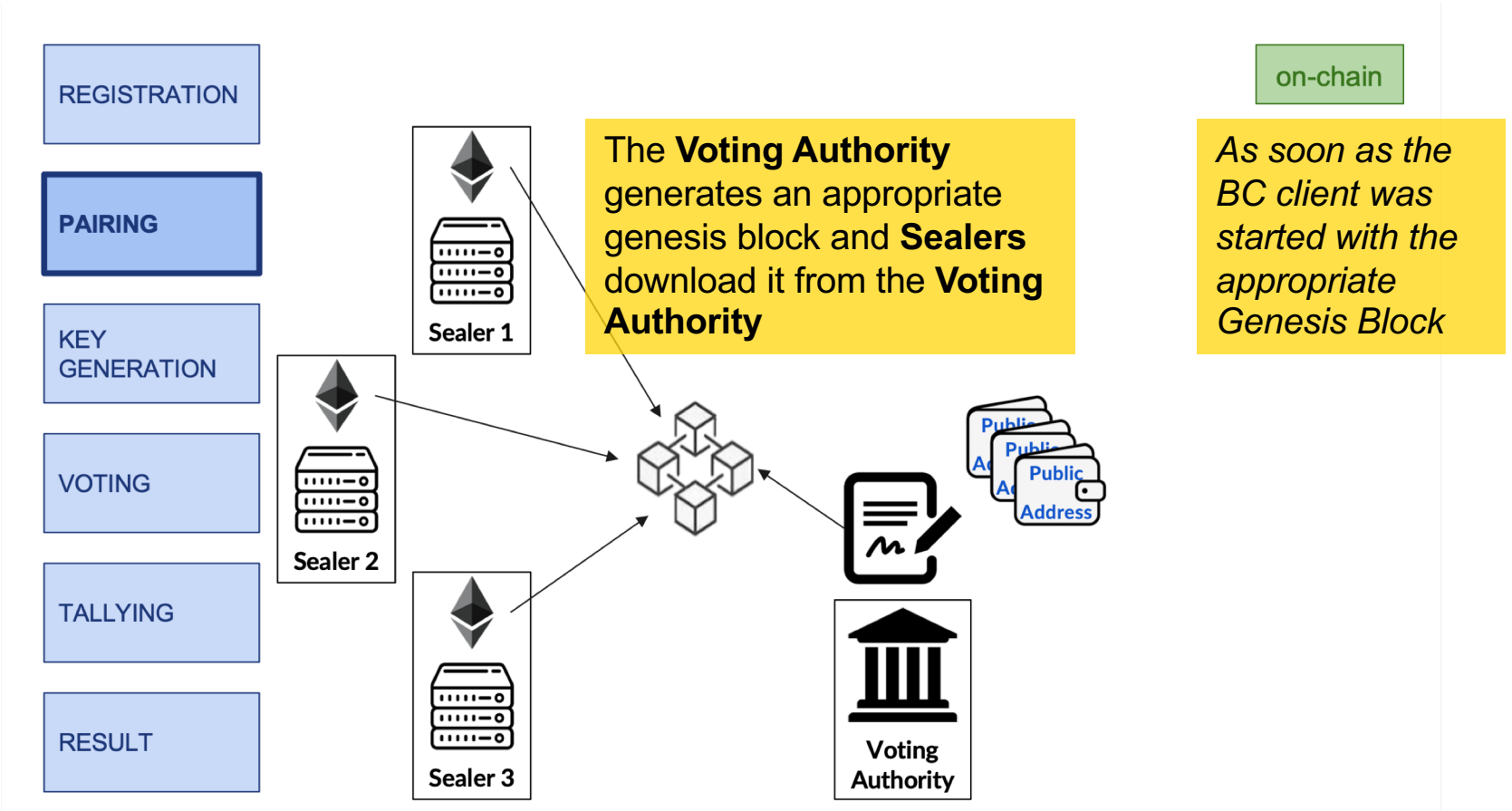


Voting Protocol: Registration (1)

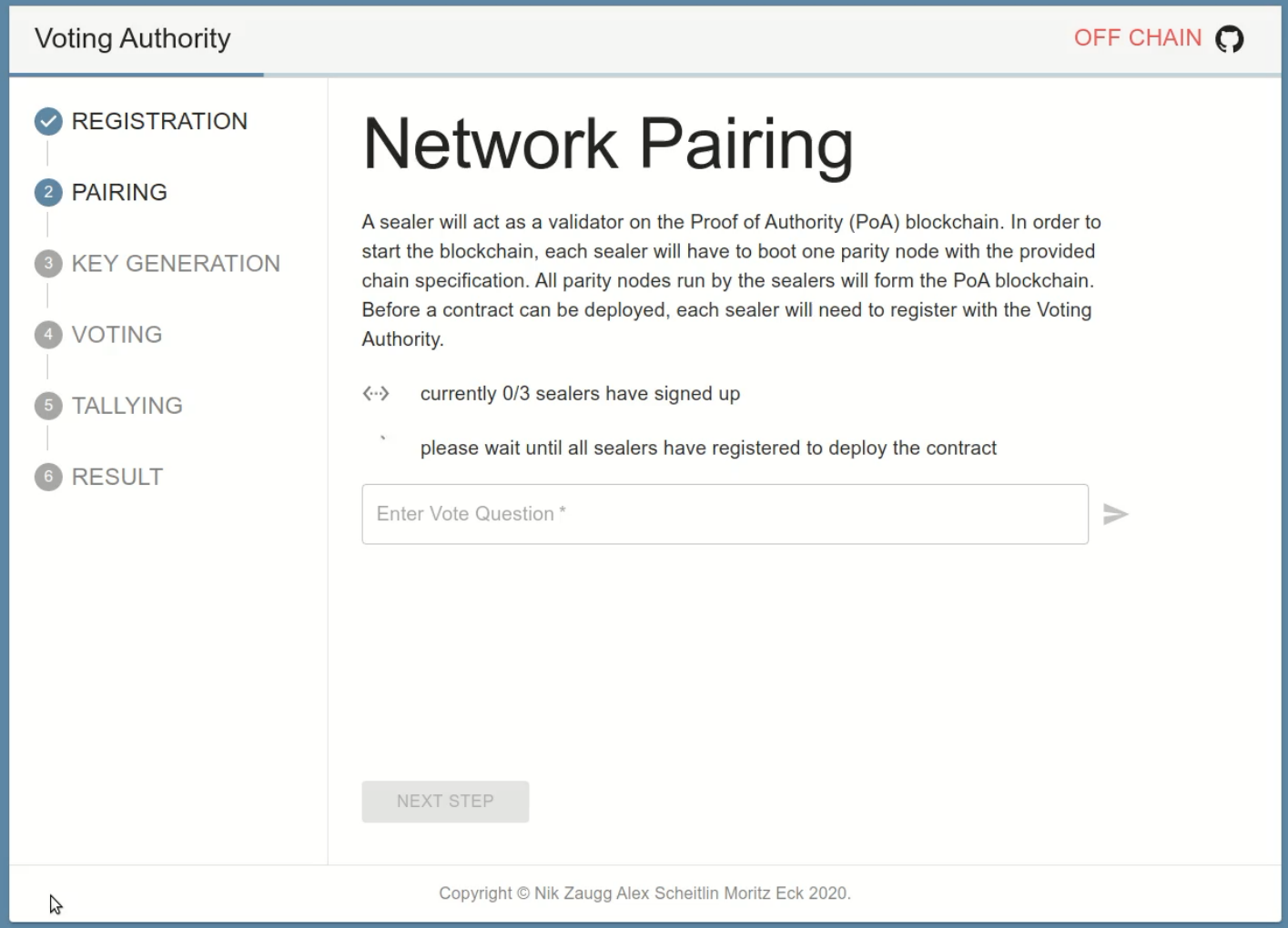


Voting Protocol: Pairing (2)

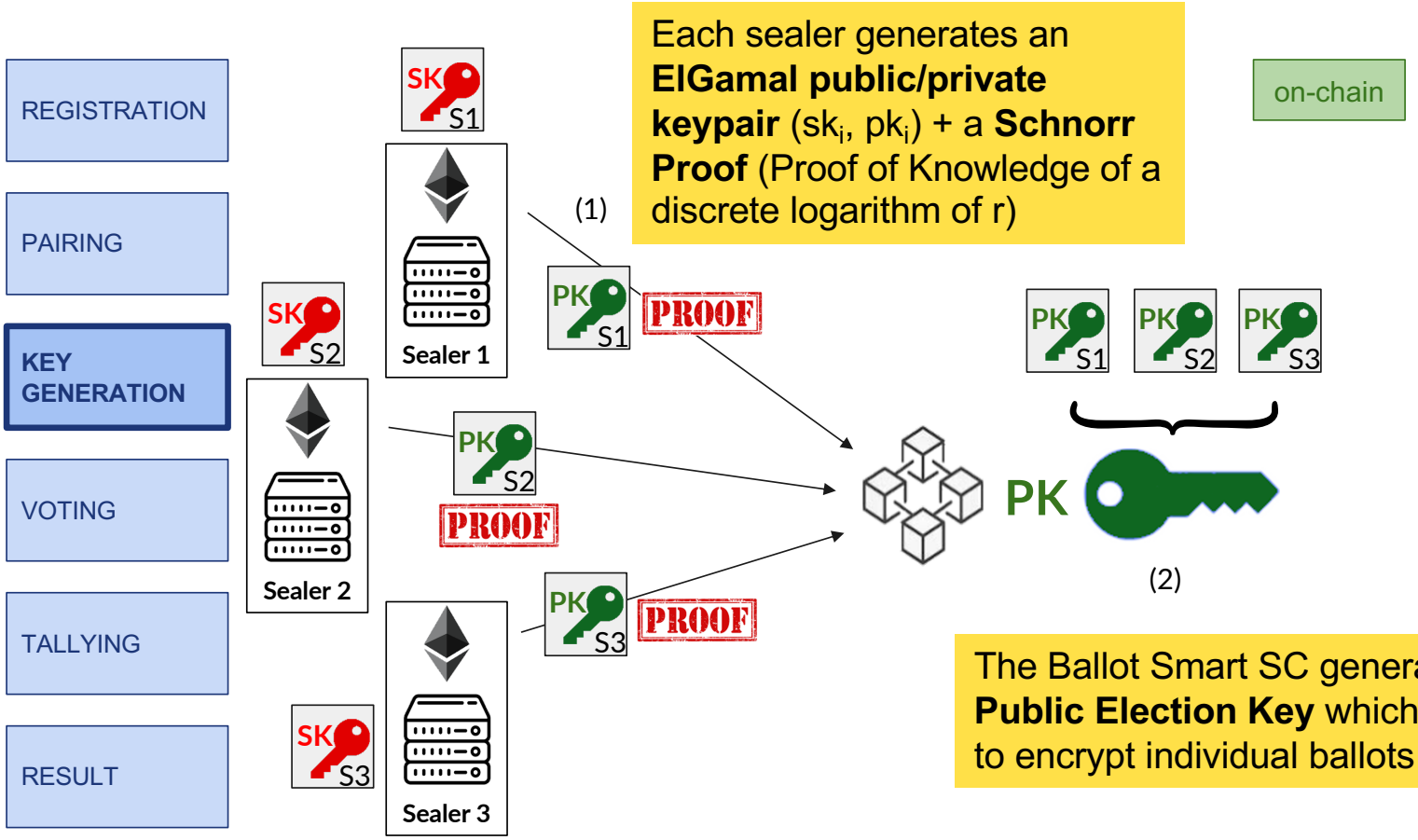
PoA Bootstrapping, where the appropriate Genesis Block is retrieved from the Voting Authority



Voting Protocol: Pairing (2)



Voting Protocol: Key Generation (3)



Voting Protocol: Key Generation (3)

Ballot.sol

```
// submit the public key share from the distributed key generation
function submitPublicKeyShare(uint256 key, uint256 proof_c, uint256 proof_d)
    external
    returns (bool, string memory)
{
    // accept key shares only if state is KEY_GENERATION
    // and the public key does not yet exist
    require(votingState == VotingState.KEY_GENERATION, 'Need state
    KEY_GENERATION.');
    require(!IS_PUBKEY_SET, 'Public key shares can only be submitted if the
    public key is not yet set.');
    require(
        keyGenProofVerifier.verifyProof(proof_c, proof_d, key, msg.sender),
        'Key Generation Proof is not correct.'
    );

    // check if this address has already submitted a share
    bool sealerAlreadySubmitted = false;

    for (uint256 i; i < election.publicKeyShareWallet.length; i++) {
        if (election.publicKeyShareWallet[i] == msg.sender) {
            sealerAlreadySubmitted = true;
        }
    }

    PublicKeyShare memory publicKeyShare = PublicKeyShare(key, KeyShareProof
    (proof_c, proof_d));

    if (!sealerAlreadySubmitted) {
        // add sealer address to array
        election.publicKeyShareWallet.push(msg.sender);
    }

    // add or replace the share
    election.pubKeyShareMapping[msg.sender] = publicKeyShare;

    return (true, 'Key Generation Proof is valid.');
}
```

- Each sealer submits his/her public key share
  - The Schnorr Proof about the knowledge of the secret key is verified
- If the proof is valid and the sealer has not submitted yet, the public key share is stored

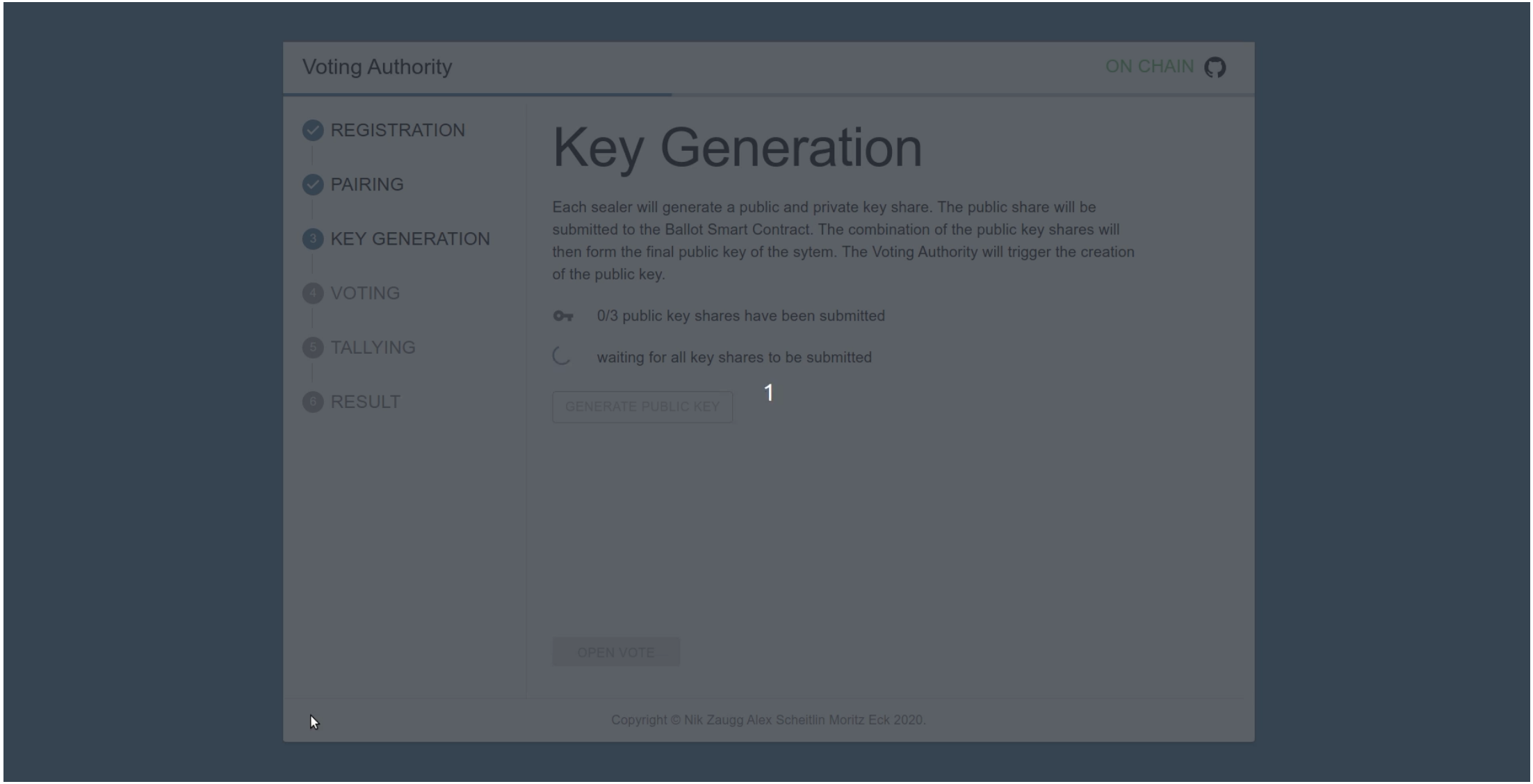
Voting Protocol: Key Generation (3)

Ballot.sol

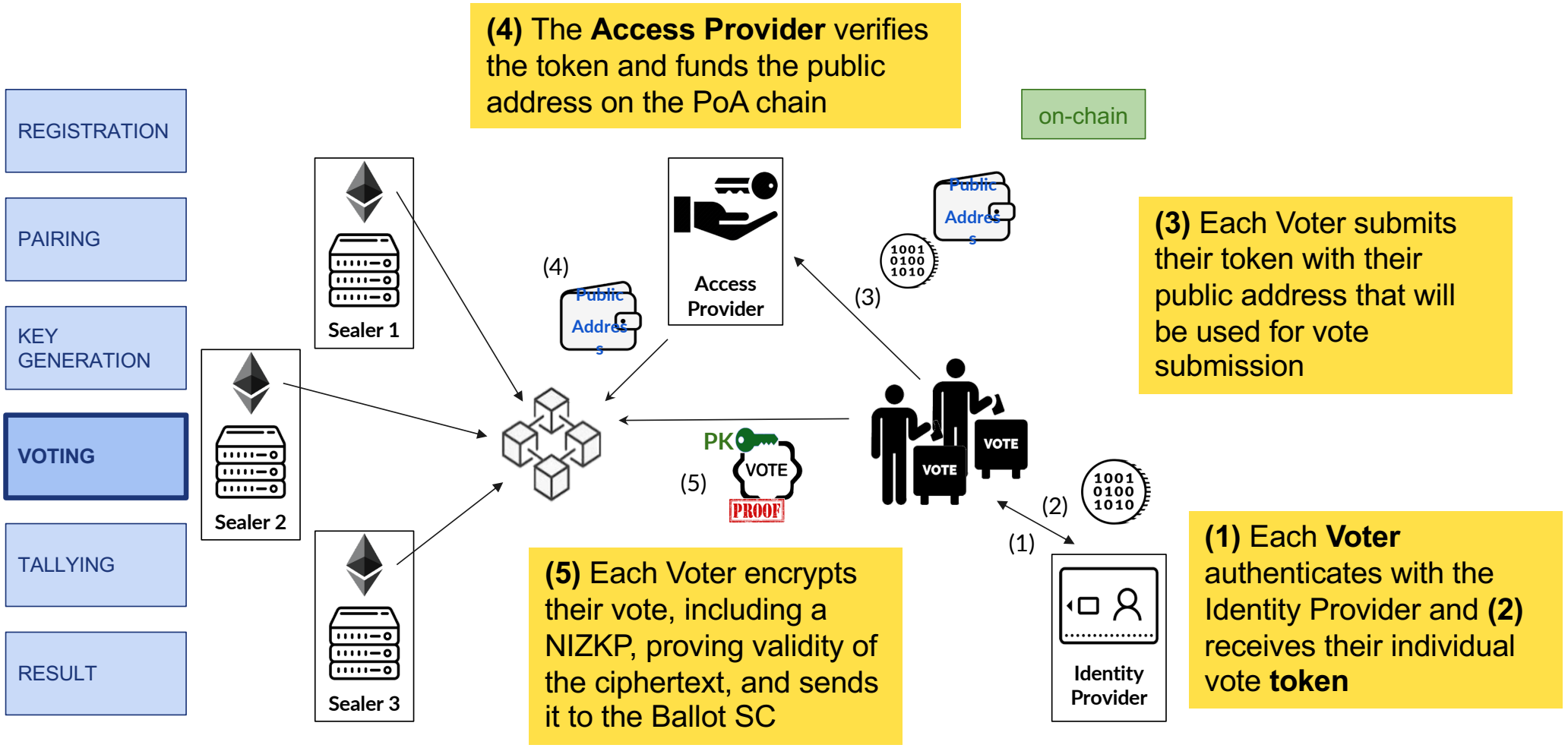
```
// generates the public key of the elgamal crypto system
// - the public key is generated from all submitted public key shares by
the sealer nodes
// - their product forms the public key
function generatePublicKey() external onlyOwner {
    // public key can only be generated once
    require(!IS_PUBKEY_SET, 'The public key is already set.');

    // every sealer needs to have published it's public key share
    require(
        election.publicKeyShareWallet.length == NR_OF_AUTHORITY_NODES,
        'Public key shares !== number of authorities.'
    );

    // set an initial key (here, we take the first)
    address firstSealerAddress = election.publicKeyShareWallet[0];
    uint256 key = election.pubKeyShareMapping[firstSealerAddress].share;

    // form the product of all public key shares
    for (uint256 i = 1; i < election.publicKeyShareWallet.length; i++) {
        address addr = election.publicKeyShareWallet[i];
        key = key.modMul(election.pubKeyShareMapping[addr].share,
        systemParameters.p);
    }

    // set the public key
    publicKey = key;

    IS_PUBKEY_SET = true;

    // trigger the creation of the verifiers (for proof verification)
    // they depend on the system parameters and public key, that's why
    // they are created only once the public key is set
    createVerifiers();
}
```

- As soon as all public keys pk_i had been submitted by the Sealers
- The public election key is calculated by forming a product of all public keyshares.

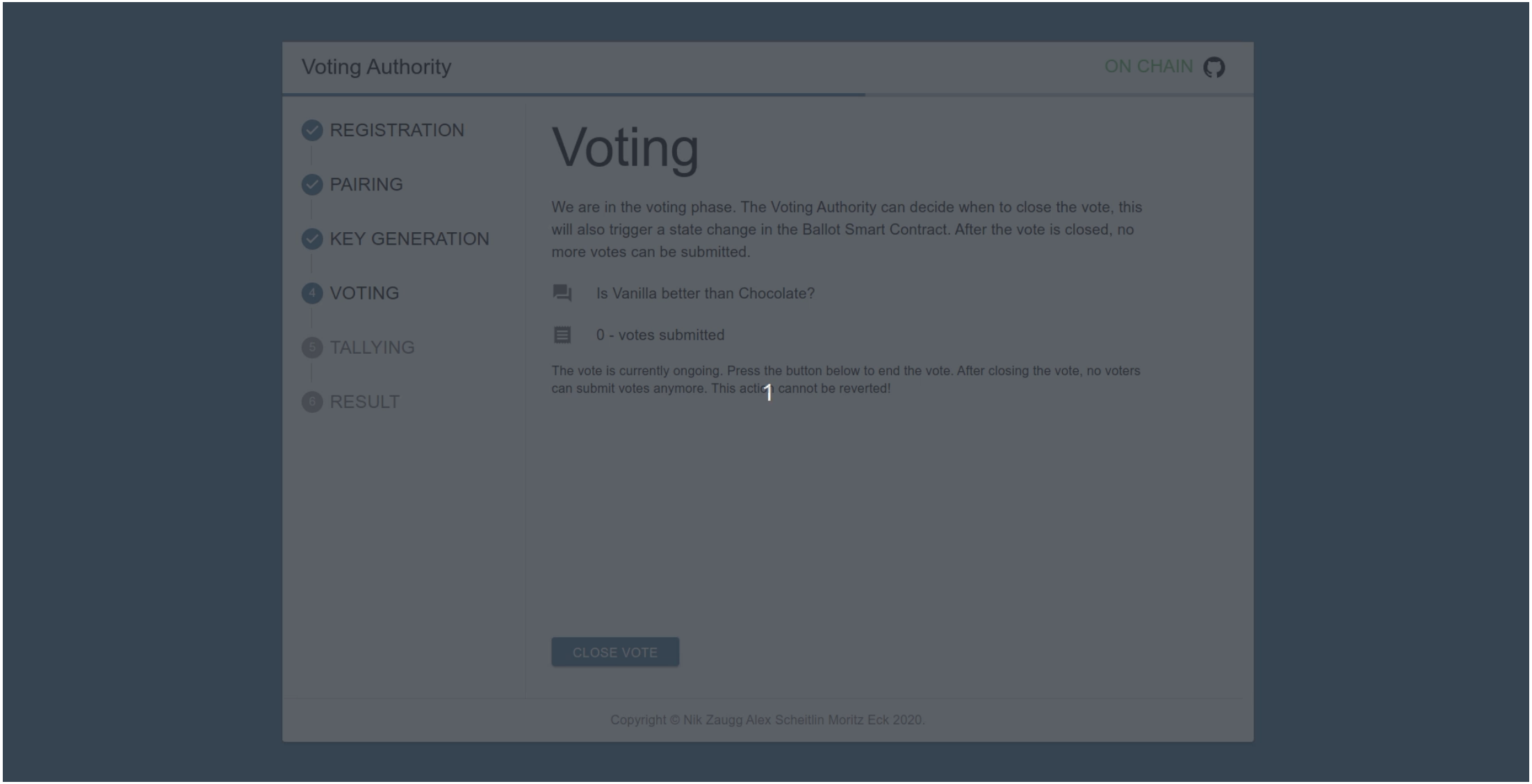
Voting Protocol: Key Generation (3)



Voting Protocol: Voting (4)



Voting Protocol: Voting (4)



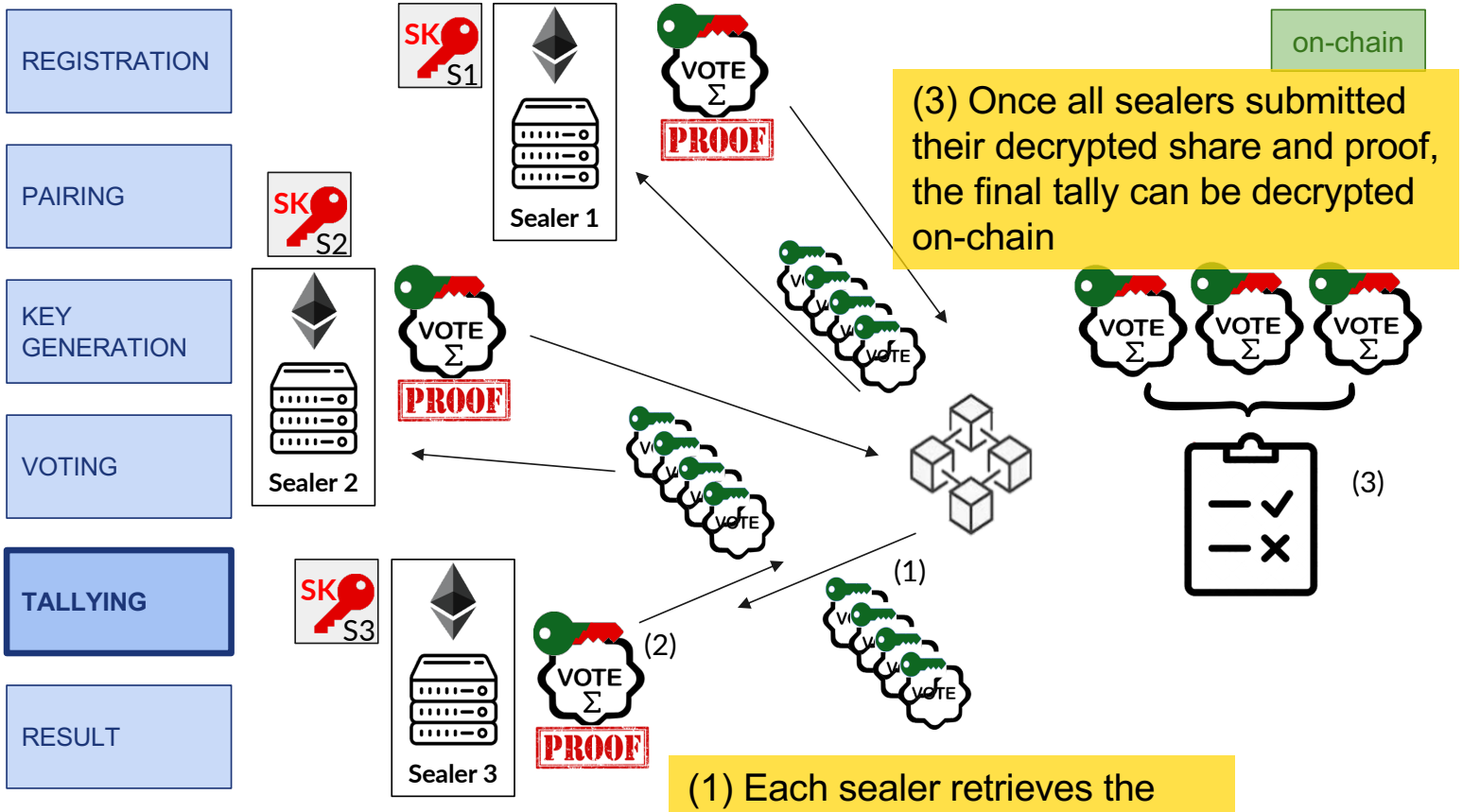
Voting Protocol: Voting (4)

Ballot.sol

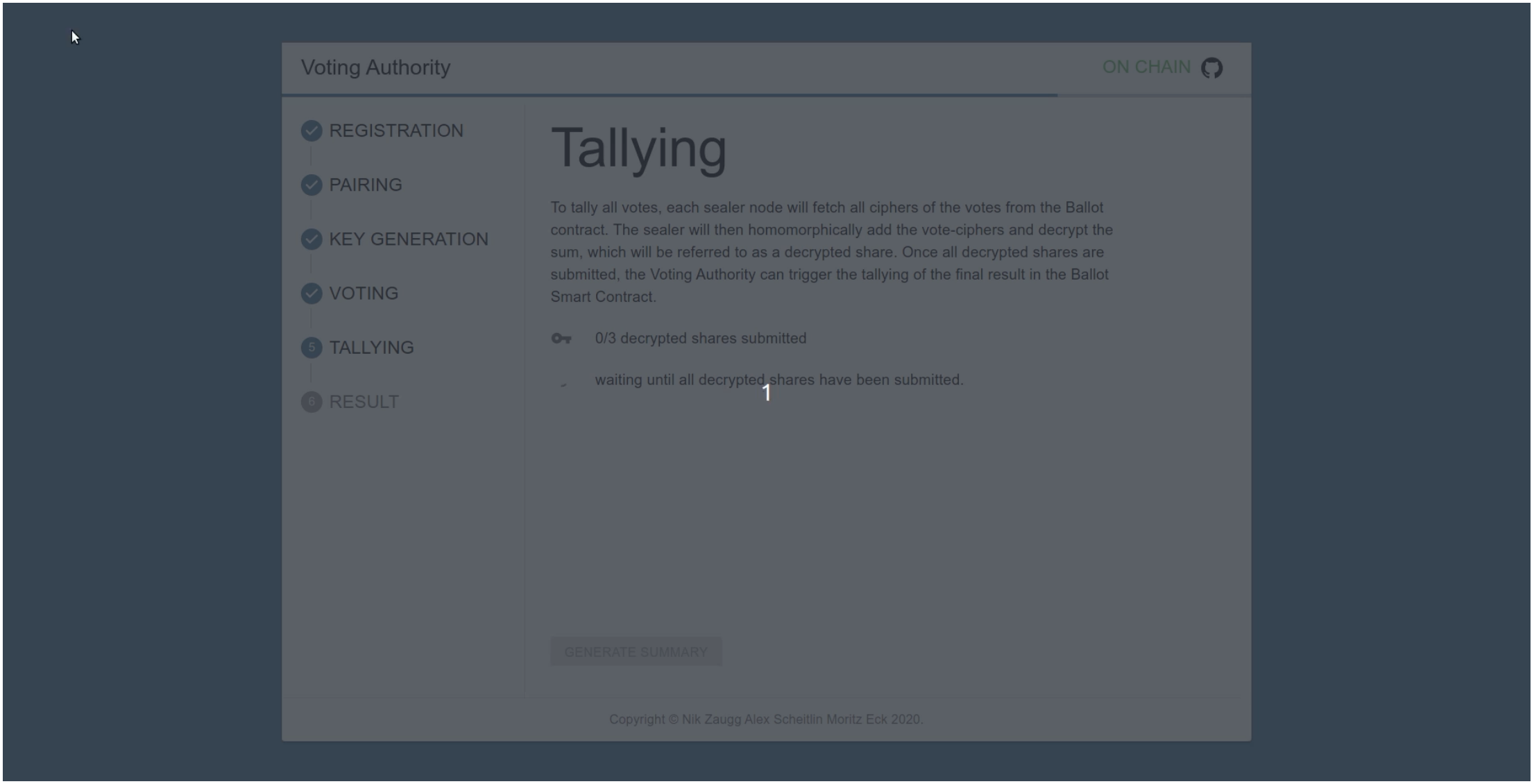
```
function vote(
    uint256[2] calldata cipher,
    uint256[2] calldata a,
    uint256[2] calldata b,
    uint256[2] calldata c,
    uint256[2] calldata f
) external returns (bool, string memory) {
    require(votingState == VotingState.VOTING, 'Vote not open.');
    require(isVoter(msg.sender), 'Address not allowed to vote.');
    require(!election.hasVoted[msg.sender], 'Voter already voted.');
    require(voteVerifier.verifyProof(cipher, a, b, c, f, msg.sender), 'Vote
    Proof not accepted.');

    VoteProof memory voteProof = VoteProof(a, b, c, f);
    Cipher memory _cipher = Cipher(cipher[0], cipher[1]);
    Voter memory voter = Voter(msg.sender, _cipher, voteProof);

    // add Voter struct
    election.voters.push(voter);
    election.nrOfVoters += 1;
    election.hasVoted[msg.sender] = true;

    emit VoteStatusEvent(msg.sender, true, 'Vote was accepted');
    return (true, 'Vote was accepted');
}
```

- Initial checks if
  - *Voting is open*
  - *Voter is eligible*
  - *Voter did not vote yet*
  - *Voter sent valid vote*
- If everything accepted, the vote containing the sender, ciphertext, and proof is pushed to the ballot box

Voting Protocol: Tallying (5)



Voting Protocol: Tallying (5)



Voting Protocol: Tallying (5)

TS voting.ts ●

```ts
import BN = require('bn.js')
import { GlobalHelper, Summary } from '../index'
import { Cipher, Encryption, SystemParameters } from './index'

export const generateYesVote = (sp: SystemParameters, pk: BN): Cipher =>
  Encryption.encrypt(1, sp, pk)
export const generateNoVote = (sp: SystemParameters, pk: BN): Cipher =>
  Encryption.encrypt(0, sp, pk)
export const generateBaseVote = (): Cipher => {
  return { a: GlobalHelper.newBN(1), b: GlobalHelper.newBN(1) }
} // encrypt with m=0, r=0

export const addVotes = (votes: Cipher[], sp: SystemParameters): Cipher => {
  return votes.reduce(
    (previous, current) => Encryption.add(previous, current, sp),
    generateBaseVote()
  )
}

export const tallyVotes = (sp: SystemParameters, sk: BN, votes: Cipher[]):
number => {
  return Encryption.decrypt1(addVotes(votes, sp), sk, sp).toNumber()
}

export const getSummary = (total: number, tallyResult: number): Summary => {
  const yes = tallyResult - 0
  const no = total - yes
  return { total, yes, no }
}
```

- On each sealer
  - Votes are first added
  - Then tallied (decrypted), and
  - Lastly a proof of correct decryption is generated as well

Takeaways Provotum

- REV systems require utmost scrutiny to achieve
  - Highly demanding properties of
    - Privacy, Verifiability, Software Independence, Fairness
    - But not limited to
  - Open aspects still include
    - Usability or legal compliance

- Leap from theory to practically deployable REV systems is significant and hard to achieve
  - More research is required to evaluate the viability of establishing such a Public Permissioned Blockchain for on-chain distribution of trust across the voting protocol

Evaluations Provotum 2.0

- E2E-Verifiability and Ballot Secrecy achieved ✓
- Client-Side Cryptography ✓
- On-Chain Proof Verification ✓
- Distributed Key Generation ✓
- Cooperative Decryption ✓

- Key-Size (256-bit) not secure ✗
- Mitigate Identity provisioning scheme threats ✗

- Receipt-Freeness and Coercion-Resistance ✗

Summary

Observations

- REVs are country-specific
  - Mainly due to local legislation
- REVs overall system requirements are comparable
  - Privacy and verifiability (cannot coexist in full at the same time)
  - Fairness, eligibility verifiability, and unforgeability
  - Software independence
- REVs building blocks: strong security mechanisms
  - Secret-Ballot Voting Schemes, *i.e.*, voting protocols
  - Cryptographic Primitives, *i.e.*, privacy and verifiability procedures
  - Public Bulletin Board (PBB), *i.e.*, ballot box for results/logs
- From theory to practical systems – a long way to go!

Considerations

- Remote Electronic Voting
  - If trust is distributed in REV, then REVs can offer better privacy and verifiability than traditional voting channels, under certain trust and computational hardness assumptions
    - As soon as a system is public, future adversaries may be able to decrypt cipher-text

- Achieving such trust distribution can be implemented by using a Public Permissioned Blockchain, where authorities collaborate with a Proof-of-Authority (PoA) consensus mechanism

Thanks for your attention

Questions

References

- **[KRM20]**[Peer-Reviewed, Full Paper] **C. Killer**, B. Rodrigues, R. Matile, E. Scheid, B. Stiller: *Design and Implementation of Cast-as-Intended Verifiability for a Blockchain-based Voting System*. In: Proceedings of the 2020 ACM Symposium on Applied Computing. SAC '20, Association for Computing Machinery, New York, NY, USA, 2020. Paper: http://bcbev.ch/uciv

- **[KS19]**[Peer-Reviewed, Full Paper] **C. Killer**, B. Stiller: *The Swiss Postal Voting Process and Its System and Security Analysis.* In: R. Krimmer, M. Volkamer, V. Cortier, B. Beckert, R. Küsters, U. Serdült, D.Duenas-Cid (eds.) Electronic Voting. Springer International Publishing, Cham, 2019, p. 134–149, Paper: http://pvpf.ch/paper

- **[MK18]**[Master's Project] R. Matile, **C. Killer:** *Privacy, Verifiability, and Auditability in Blockchain-based E-Voting*, 2018, Master's Project, University of Zurich, http://bcbev.ch/pv

Slides and Code

- To follow along the practical steps on your own system or virtual server, please refer to the guide provided at
  - http://bcbev.ch/icbc20
- The PDF slides are also provided at
  - http://bcbev.ch/icbc20slides
- The videos used in the presentation (thus, not embedded in the PDF, are provided at
  - http://bcbev.ch/icbc20videos