Master in Computer Engineering (MEI)
Integrated Master in Informatics Engineering (MiEI)

Specialization Profile **CSI**: Cryptography and Information Security

Engenharia de Segurança

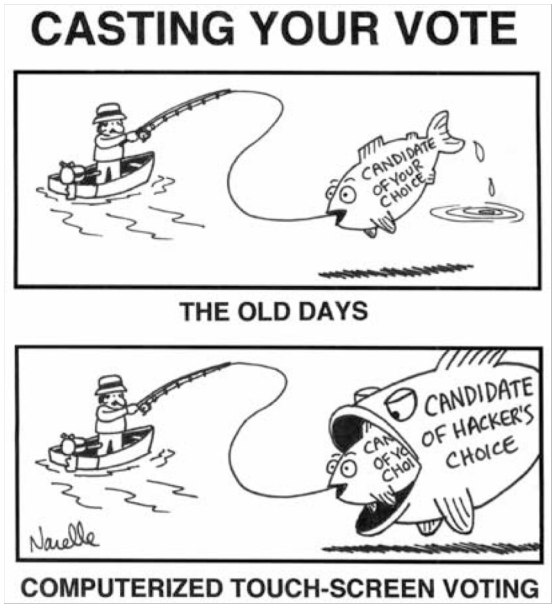
jose.miranda@devisefutures.com

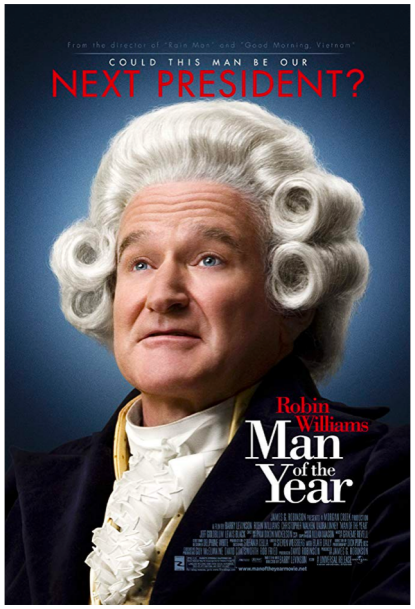
Tópicos

- Applied Cryptography
  - Cryptographic protocols / applications
    - Electronic Voting

Electronic Voting

- Network voting protocol, with the following **general safeguards**:
  - Voter Authentication
  - Voter Anonymity
  - Vote Confidentiality
  - Vote Integrity
  - Vote is not lost
  - Integrity of the voting system
  - Auditability of the voting system





- The examples presented in the following slides assume that voters do not use advanced means of authentication (such as personal digital certificate) nor are technological experts.

Electronic Voting – Example 1

- **Entities** involved in the Electronic Voting:
  - Promoter: The entity that promotes the voting;
  - SMV (Voting Broker System): a voting platform (frontend and backend) responsible for managing and processing electronic votes during the voting period, after which the electronic votes are sent to the Promoter;
  - Voter: The individual who wishes to exercise their voting rights, choosing the options available according to the ballot policy (for the purpose of the protocol, blank or spoiled ballot can be seen as additional options that the Promoter can activate);
  - Auditor: Individual/Entity that monitors the voting process, being able to prove its correction without knowing the voting behaviour of each of the voters.

Electronic Voting – Example 1

- With this electronic voting protocol:
  - The **Voter** can access the web interface or voting app that, after showing the options available and requesting the introduction of the necessary authentication elements, allows the voter to vote and get a proof that the ballot was casted within the voting period;
  - The Voter can prove the existence of situations of fraudulent rejection of votes casted within the established voting period;
  - The Voter may protect himself from the fact that, although the vote was casted within the voting period, the server only processed it after the voting period has ended;
  - **SMV** does not become aware of the content of the votes;
  - SMV cannot change the content of the votes without being detected by the Promoter;
  - SMV cannot change the number of votes without being detected by the Promoter;
  - The **Promoter** does not become aware of the content of the votes until the end of the election;
  - The Promoter cannot change the content of the votes without being detected by the SMV;
  - The Promoter cannot change the number of votes without being detected by the SMV;
  - The Promoter receives, securely, the votes (encrypted with the Promoter's public key) casted during the voting period, together with a proof of the time they were saved in the SMV;
  - The Promoter can defend itself from unfounded suspicions of rejection of votes casred within the voting period;
  - The Promoter can be sure that during the voting period no one becomes aware of the content of the votes;
  - **No actor** in the voting process can know who voted what.

Electronic Voting – Example 1

- Preparation of electronic voting (**pre-voting phase**):
  - The **Promoter** informs the SMV:
    - Voter groups, voting period, ballot papers and voting options of each ballot (ballot policy).
    - Identification of voters and contacts so that the SMV can send voting authentication credentials (for example, the voter identification number known by the Promoter, and the PIN generated by SMV);
  - The **Promoter** informs the SMV what type of additional authentication credential should be requested from the Voter so that the Promoter is able to validate it (eg, voter identification number and date of birth);
  - The Promoter generates a key pair and the respective voting certificate, providing the certificate to the SMV;
  - The Promoter generates a key pair and the respective signature certificate, providing the certificate to the SMV;
  - The Promoter generates a key pair and the respective certificate for ciphering the votes, providing the certificate to the SMV;
  - The Promoter provides an HTTPS Web service that, receives a public key together with the Voter's credentials (known to the Promoter), and returns a token signed by the Promoter (using his signature certificate) to authorize the vote.

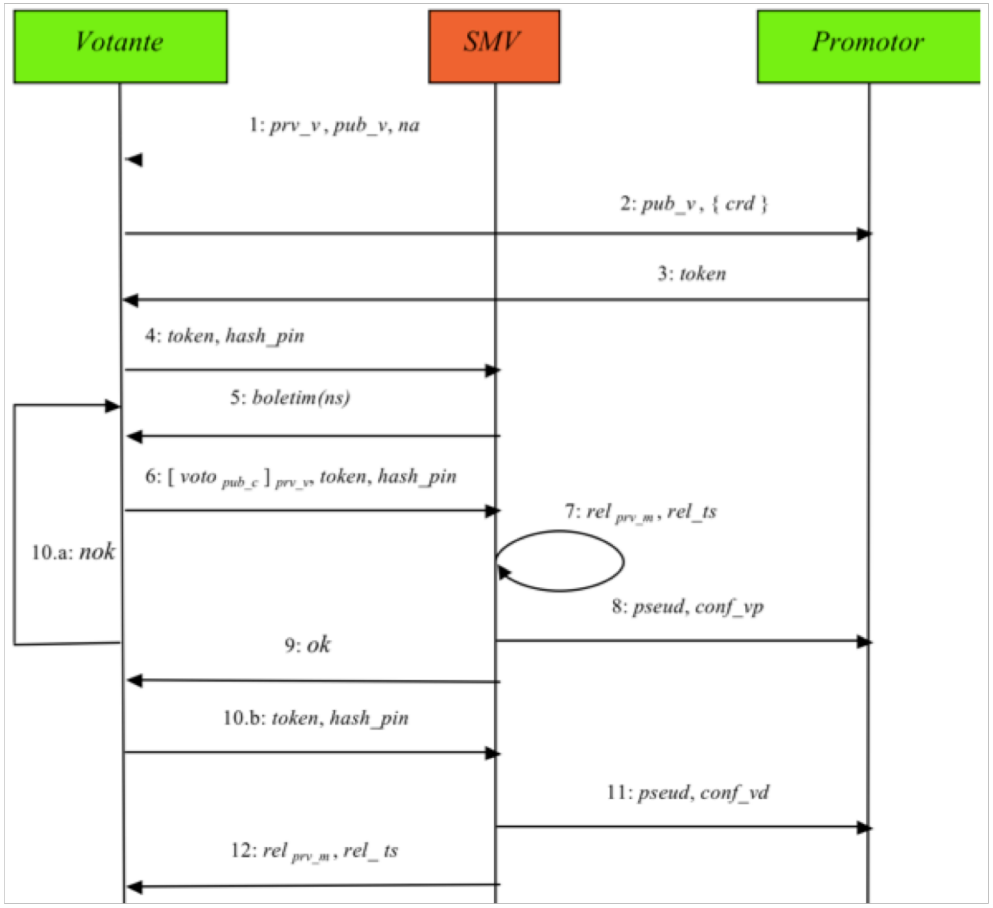
Electronic Voting 1 – voting period

Step 1:

a. The Voter accesses the web interface (or voting app) whose URI address has been communicated by the Promoter;

b. The Voter enters the credentials requested (one known by the voter and the SMV, and other known by the voter and the Promoter);

c. A key pair (*prv_v*, *pub_v*) is generated and a random number *na* is generated for this Voter.



Step 2:

a. The web interface or voting app starts an HTTPS POST to the Web service provided by the Promoter (validating the service by the Web server certificate), and sends the generated public key (*pub_v*) and the credentials (known by the Voter and Promoter) introduced in the previous step.

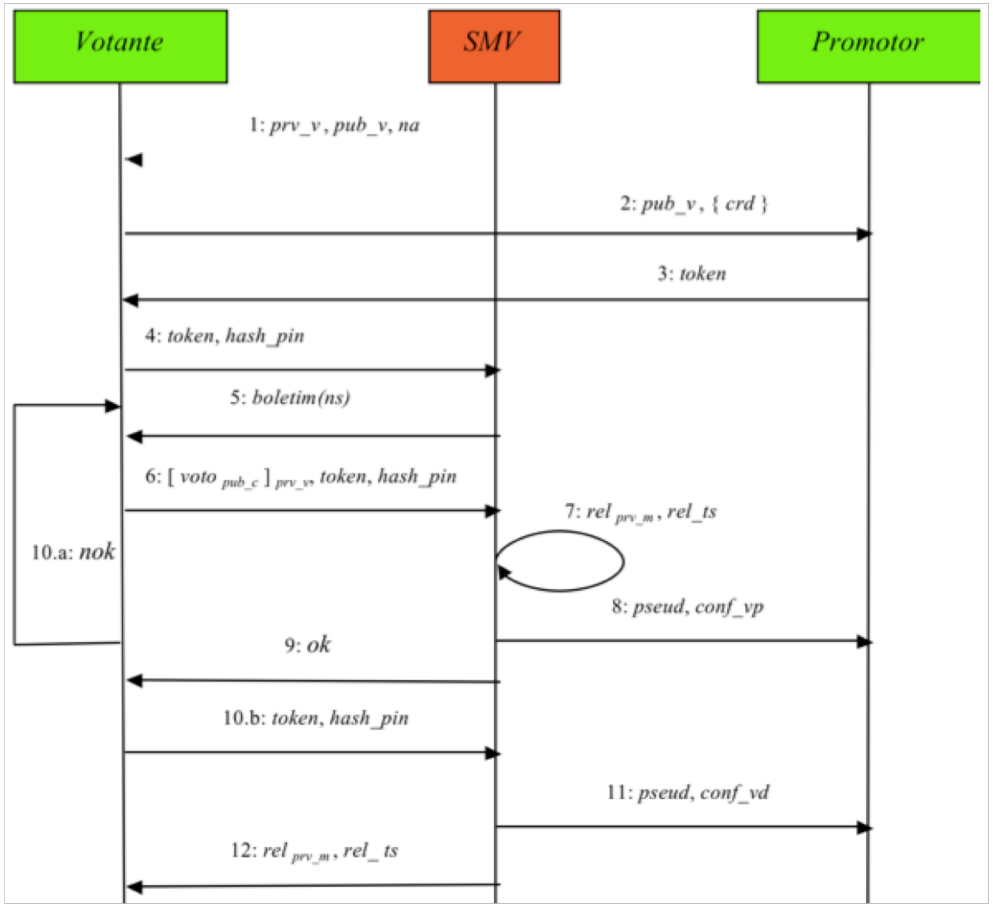
Electronic Voting 1 – voting period

Step 3:

a. If the credentials received are correct, the Promoter returns a signed authorization token (signed with its private signature key) that authorizes the Voter to participate in the election. From this token it is possible to extract the pseudonym of the Voter, the group to which the Voter belongs, and which ballot papers he should have access to, and the number of votes his vote will represent, as well as the voter's public key (*pub_v*).

Step 4:

a. The web interface or voting app sends (without the intervention of the Voter) the signed authorization token (received from the Promoter) to the SMV, along with the hash of the credentials known between the Voter and SMV (*hash_pin*).



Electronic Voting 1 – voting period

Step 5:

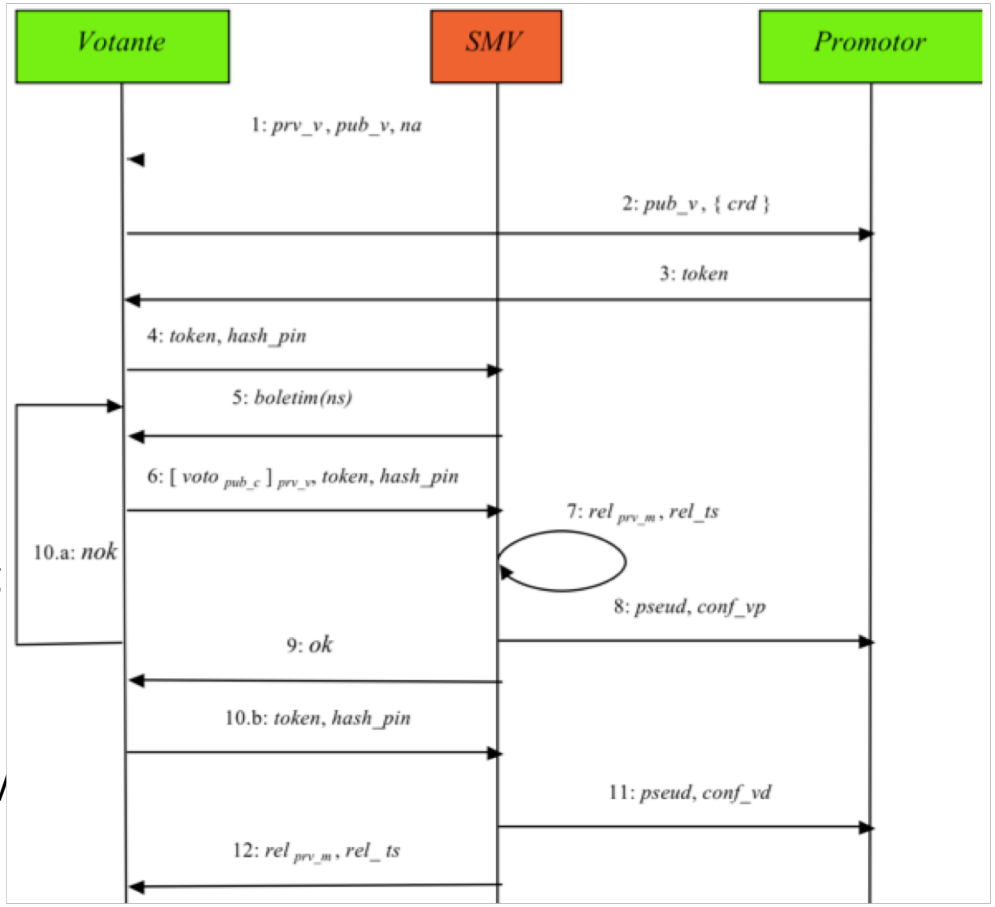
a. SMV "informs" the web interface / voting app what ballot papers the Voter is authorized to view;

b. The Voter cast its vote in accordance with the policy of each voting Ballot.

Step 6:

a. The voting web interface / app "creates" the vote that includes also the random number *na* generated in step 1, and encrypts the vote with the public key of the Promoter (*pub_c*, obtained from the promoter's voting certificate) and signs the vote with the private key *prv_v* of the Voter (generated in step 1).

b. The generated signed vote is sent to the SMV along with the authorization token and *hash_pin*.



Electronic Voting 1 – voting period

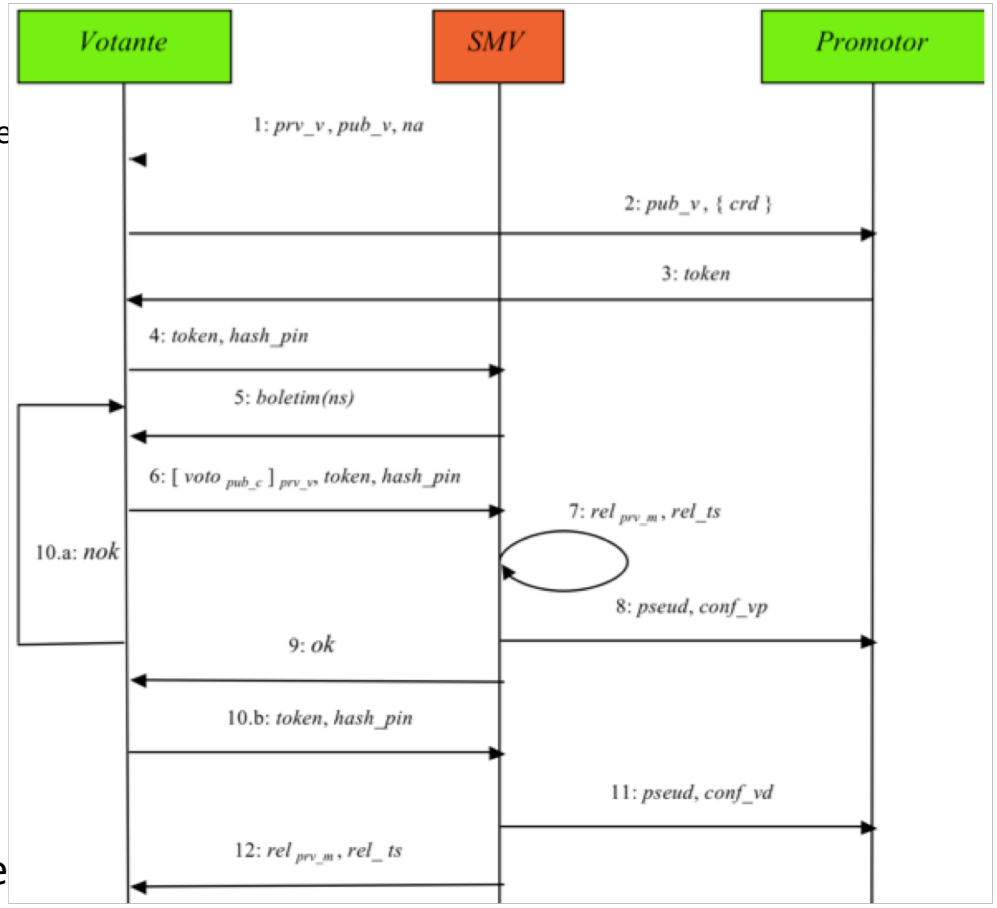
Step 7:

a. SMV verifies:
   - If it is a known *hash_pin* (each *hash_pin* is unique)
   - If the authorization token is valid and,
   - If the vote was signed with the private key (of the voter) corresponding to the public key contained in the token.

b. SMV generates a reception report (*relprv_m*) which includes the pseudonym of the Voter *pseud*, and signs it with the SMV private key *prv_m*.

c. SMV timestamps the signed report (*rel_ts*), in order to have proof of the date / time that the processing of the vote finished.

d. If the date / time is out of the voting period, the vote is discarded and an error message is returned in step 9. Otherwise, the SMV considers the vote valid and stores it in its database , together with the reception report and its timestamp.



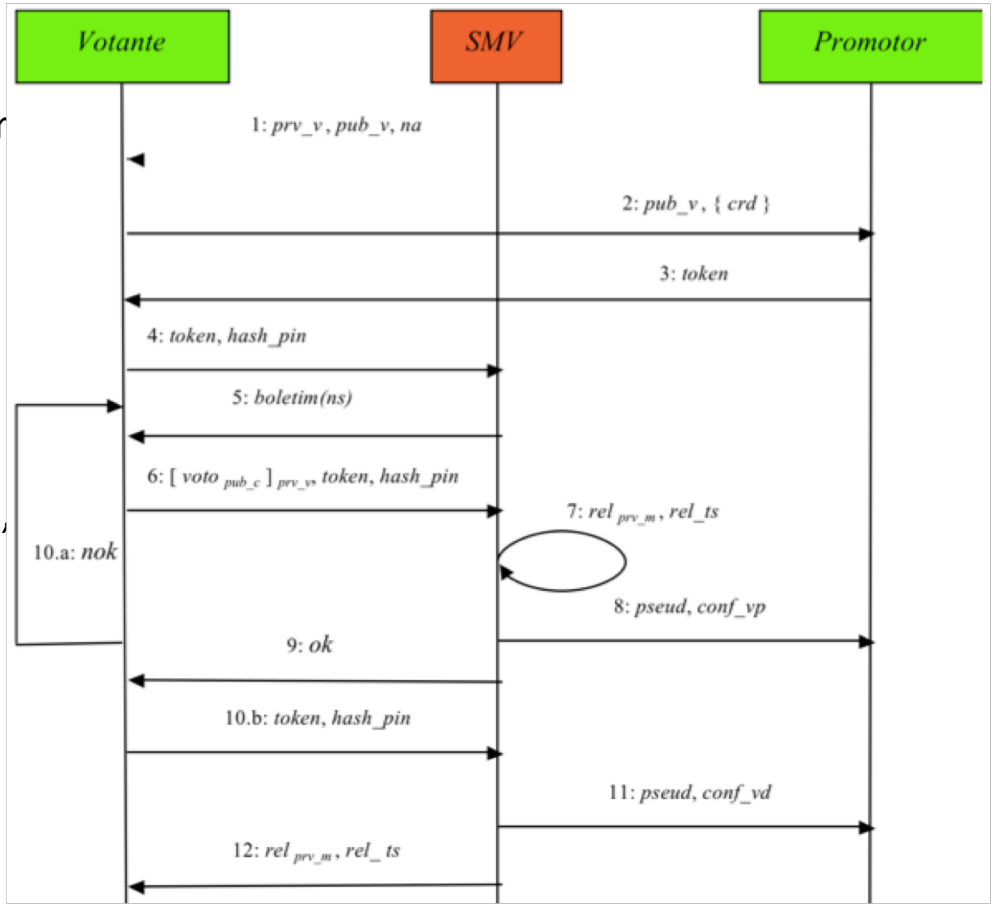
Electronic Voting 1 – voting period

Step 8:

a. If the checks carried out in the previous step are successful, a provisional confirmation (*conf_vp*) that the Voter with the pseudonym *pseud* has just exercised his / her voting rights is sent, via POST HTTPS, to the Promoter Web service.

Step 9:

a. If the checks performed in step 7 have not been successful, an explanatory error message is displayed to the Voter. Otherwise, the ballot papers with the voter choices are displayed to the voter, asking for confirmation.



Electronic Voting 1 – voting period

Step 10a:

- If the Voter wishes to change his / her voting choices, the voting process returns to step 5.

Step 10b:

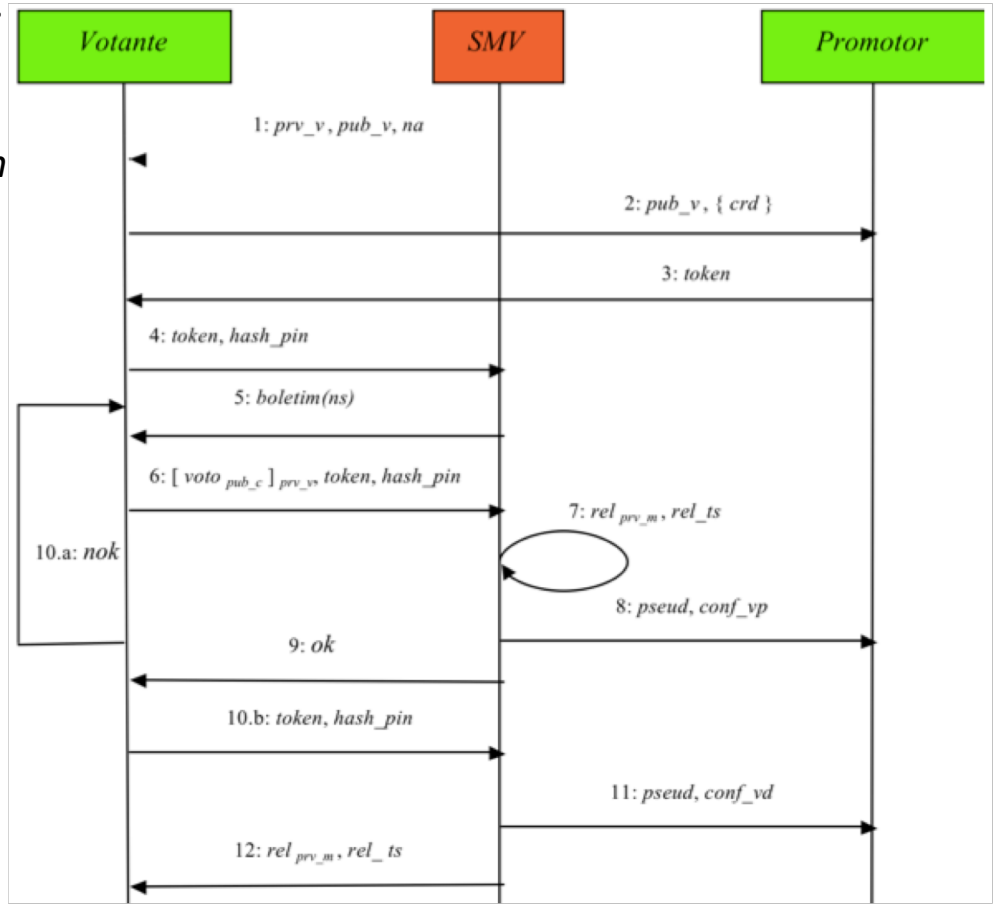
- If the Voter wishes to confirm his / her vote, the voter's authorization token and *hash_pin* are returned to the SMV, and the vote is final.

Step 11:

a. SMV repeats step 8, but in this step a final confirmation (*conf_vd*) of the vote made by the Voter *pseud* is sent to the Promoter.

Step 12:

a. The signed reception report (*relprv_m*) and the respective time stamp (*rel_ts*) generated in step 7 are delivered to the Voter so that he/she can store them as proof (with legal validity) that he/she exercised his right to vote within the voting period, without breaking the confidentiality of the vote.



Electronic Voting – Example 1

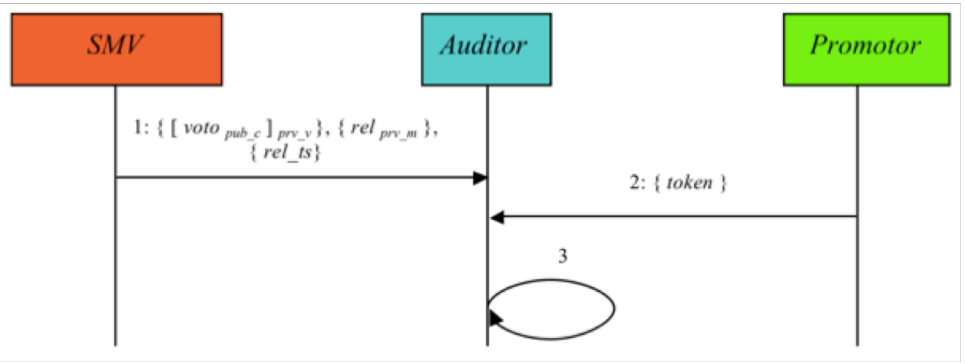
- Electronic vote (**post-vote phase**):
  - At the end of the voting period, SMV sends to the Promoter:
    - set of votes (encrypted but not signed by the Voters) signed by SMV - ensuring that the Promoter does not know who voted what;
    - signed reception reports and the respective timestamps.
  - The Promoter, with its private voting key deciphers the votes and automatically counts the votes.
- With the data provided by SMV, the Promoter can:
  - check whether the number of votes delivered by the SMV is correct (comparing with the number of authorization tokens issued),
  - obtain the result of the election (deciphering the votes with its private key), and
  - be assured that the votes received were entered during the voting period.

Electronic Voting – Example 1

- **Auditability**
  - In order to preserve the confidentiality of the voting process, the SMV cannot deliver the votes to the Promoter in the format signed by the Voter (since the Promoter would know who had voted what, compromising the secrecy of the vote),
  - The Promoter may raise suspicions about the veracity and originality of the votes delivered by the SMV;
  - It is necessary to have an **Auditor** who, in case of doubt, can prove the honesty of the SMV using a process that also does not allow him to know who voted what.

Electronic Voting – Example 1

- Auditability – operating scheme



Step 1:

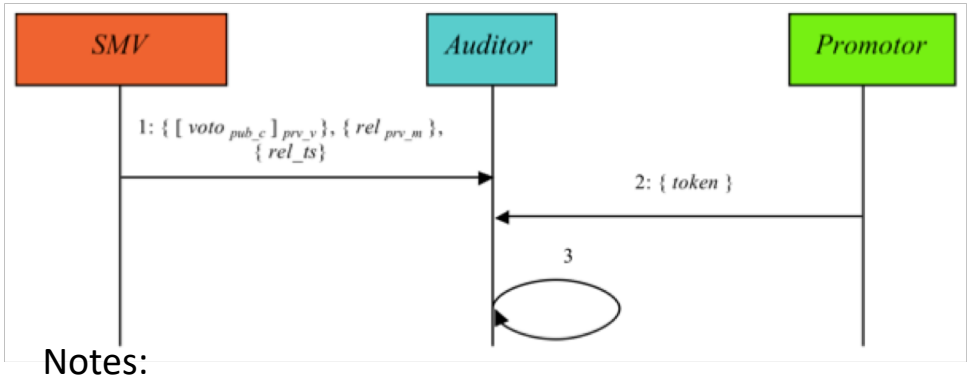
SMV delivers to the Auditor the signed votes by the Voters, the receipt reports signed by the SMV and the respective timestamps.

Step 2:

The Promoter delivers to the Auditor the list of authorization tokens issued (by the Promoter) during the voting process.

Electronic Voting – Example 1

- Auditability – operating scheme



Notes:

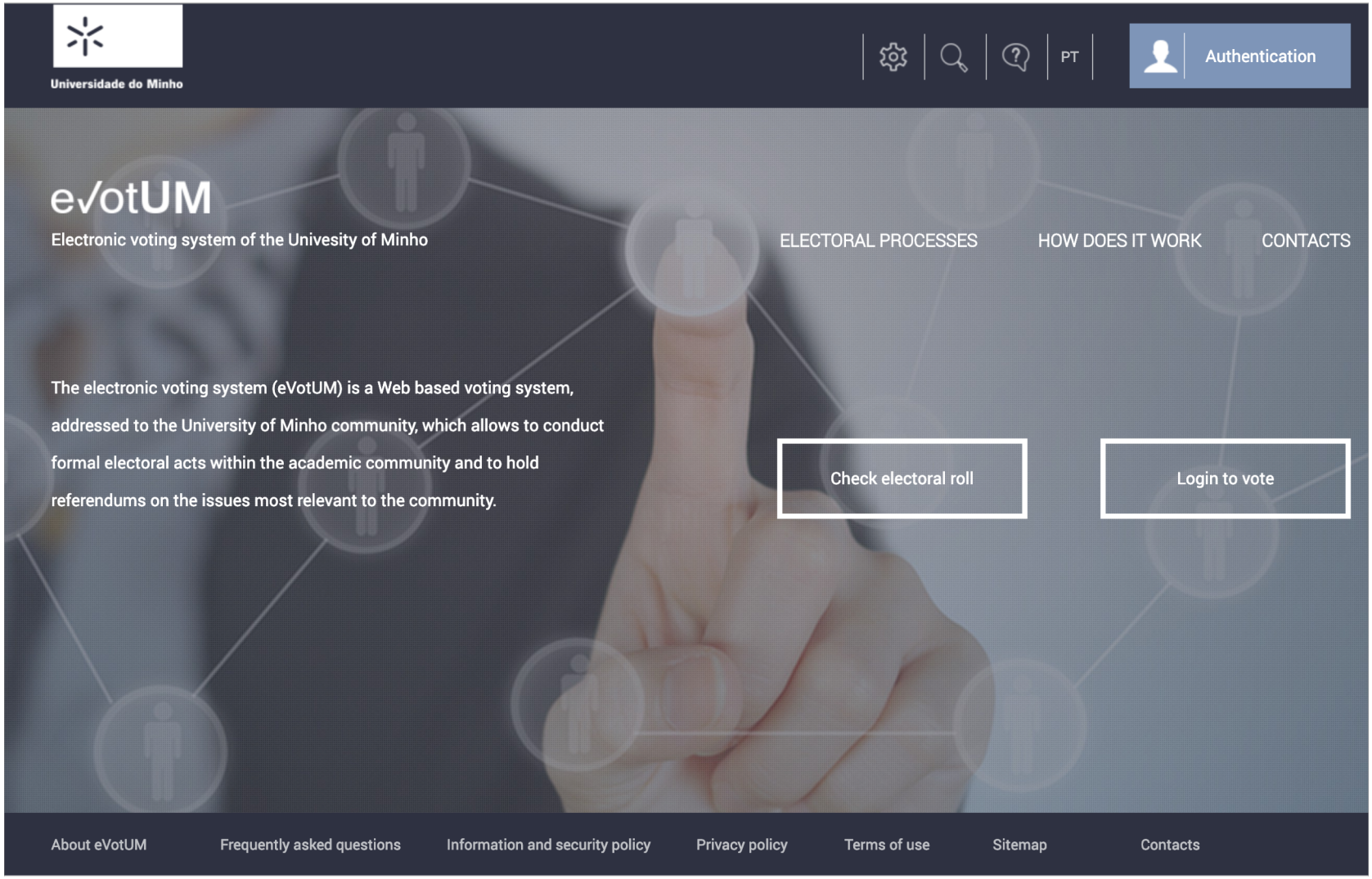
- The formats of the public keys, certificates, digital signatures, encrypted messages and time stamps used by the voting system are standard, so the development of the applicatio (s) to be used to audit the data provided may be the responsibility of the Audito , thus ensuring full independence of the audit and transparency of the whole audit process
- During this process, the Auditor could not know more than she/he should, since the contents of the signed votes delivered by the SMV are encrypted, and the Auditor does not have the necessary private key to open them.

Step 3:

- The Auditor verifies that each signed vote delivered by the SMV is signed with a private key corresponding to a public key contained in one of the tokens delivered by the Promoter.
- Additionally, the Auditor can verify that all the votes delivered by the SMV have a corresponding receipt report and timestamp within the voting period.

If this happens, it will be proved that the SMV has not adulterated any of the votes, and the Promoter does not have reasons to doubt the result of the election.

Electronic Voting – Example 2



Electronic Voting – Example 2

Features

AUTHENTICITY
Only people with voting rights can vote.

UNITY
Each voter votes only once.

ANONYMITY
It is not possible to associate a vote with a voter, nor vice versa.

INTEGRITY
Votes can not be modified nor destroyed

Unreleased
No voter can prove his vote.

VERIFICABILITY
It is possible to independently verify that all votes were counted correctly.

Electronic Voting – Example 2

Features

AUDITABILITY

The eVotUM e-voting system can be tested and audited by independent entities.

MOBILITY

The eVotUM e-voting system does not restrict where you vote.

TRANSPARENCY

The eVotUM e-voting system is clear, accurate, precise and secure.

AVAILABILITY

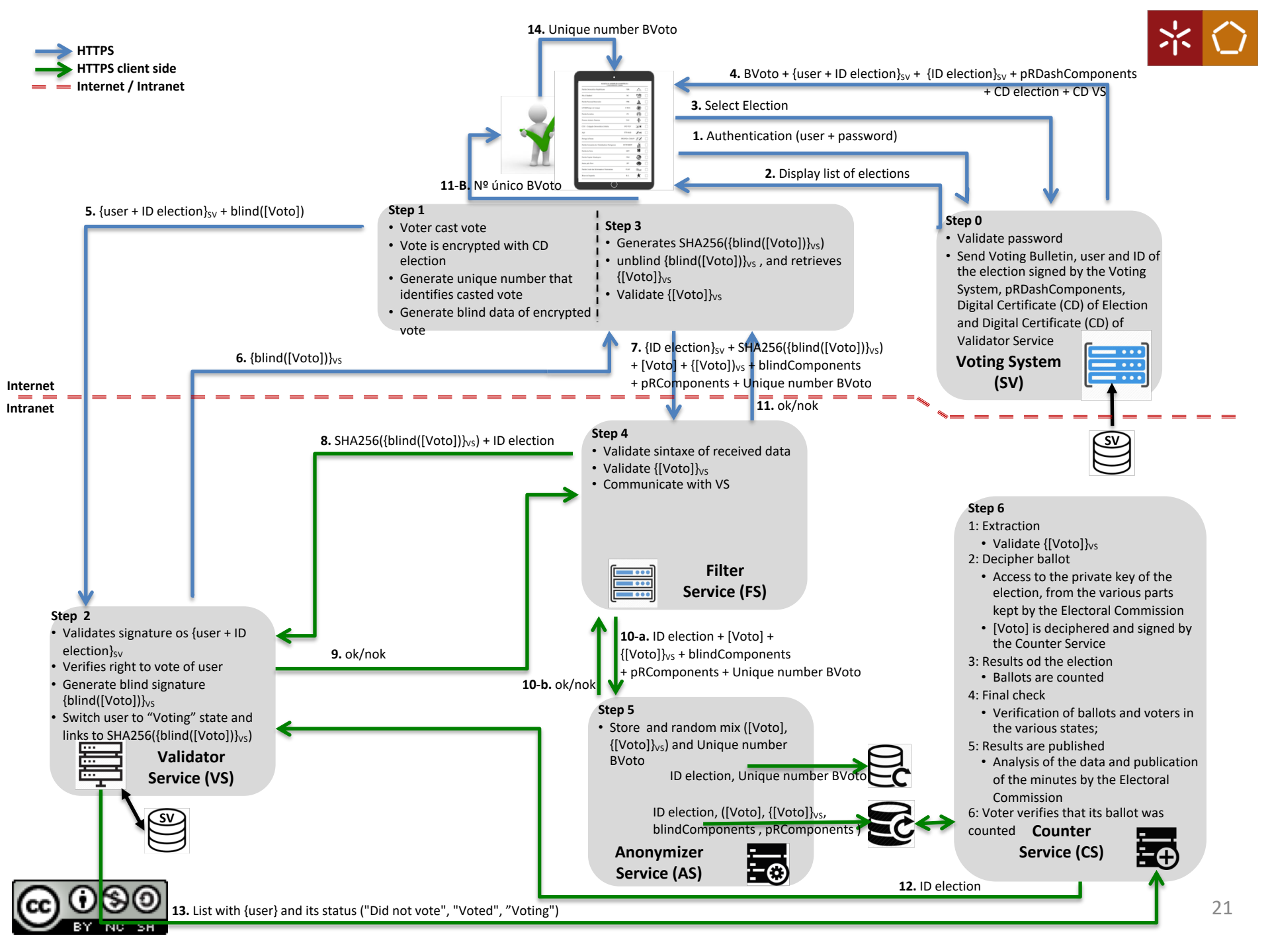
The eVotUM e-voting system is always available during the voting period.

DETECTION AND RECOVERY

The eVotUM electronic voting system detects errors, failures and attacks and recovers information to the point of failure.

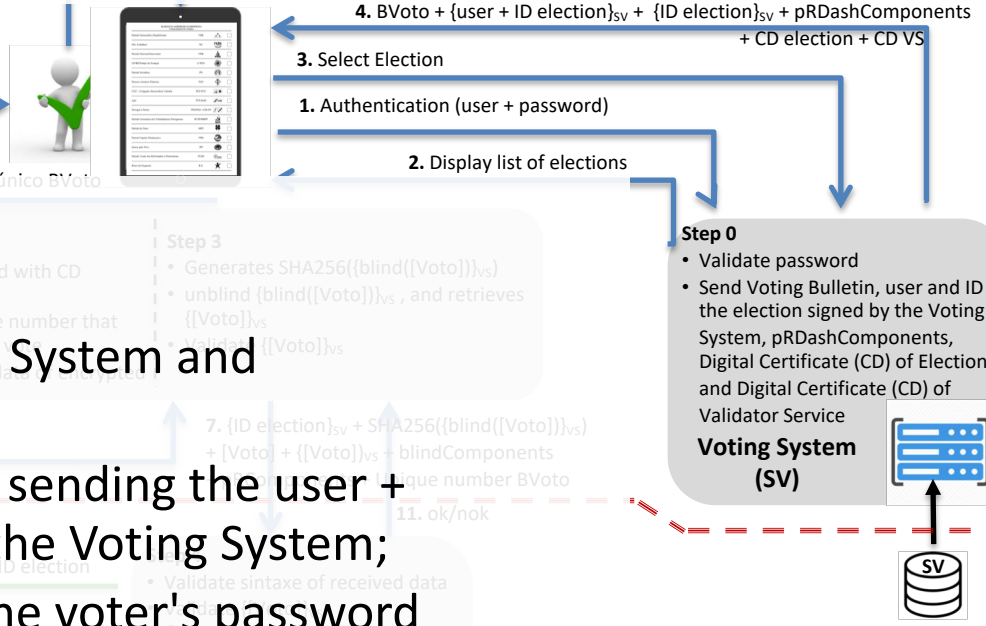
Electronic Voting – Example 2

- Steps and Flow of communication / messages

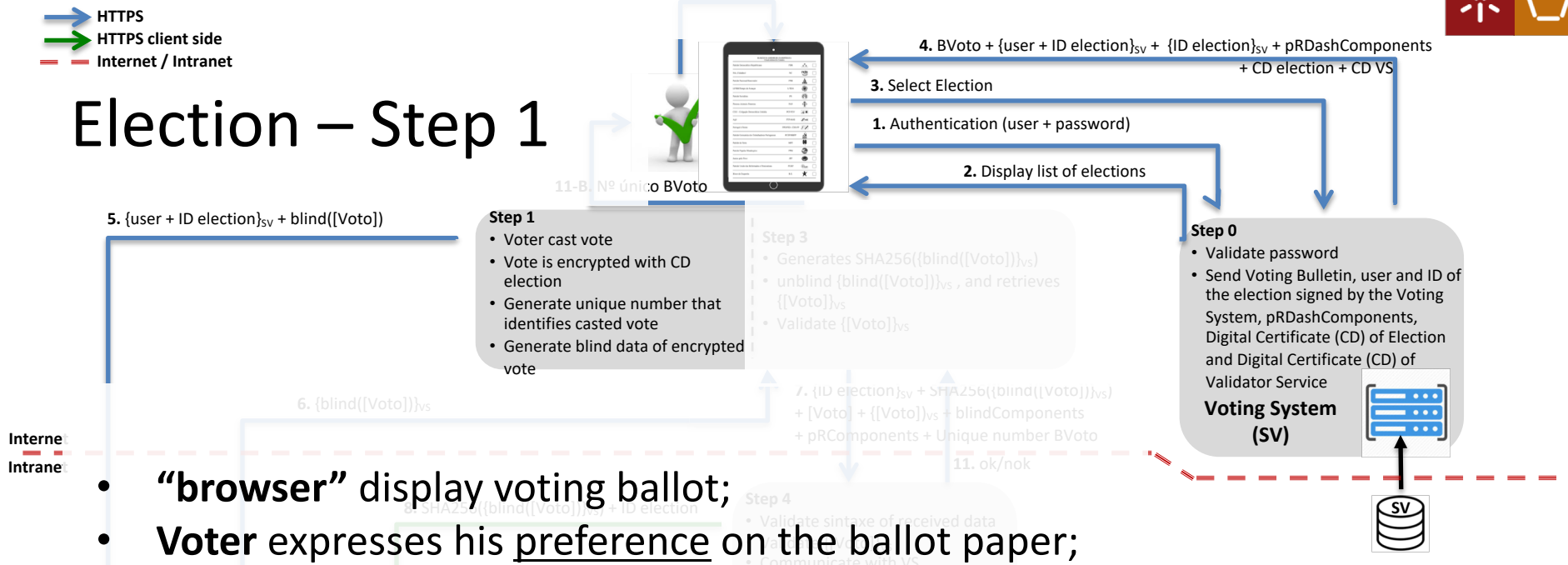

HTTPS
HTTPS client side
Internet / Intranet
14. Unique number BVoto
4. BVoto + {user + ID election}sv + {ID election}sv + pRDashComponents + CD election + CD VS
3. Select Election
1. Authentication (user + password)
2. Display list of elections
11-B. Nº único BVoto
5. {user + ID election}sv + blind([Voto])
Step 1
• Voter cast vote
• Vote is encrypted with CD election
• Generate unique number that identifies casted vote
• Generate blind data of encrypted vote
Step 3
• Generates SHA256({blind([Voto])}vs)
• unblind {blind([Voto])}vs , and retrieves {[Voto]}vs
• Validate {[Voto]}vs
Step 0
• Validate password
• Send Voting Bulletin, user and ID of the election signed by the Voting System, pRDashComponents, Digital Certificate (CD) of Election and Digital Certificate (CD) of Validator Service
Voting System (SV)
SV
6. {blind([Voto])}vs
7. {ID election}sv + SHA256({blind([Voto])}vs) + [Voto] + {[Voto]}vs + blindComponents + pRComponents + Unique number BVoto
11. ok/nok
Internet
Intranet
8. SHA256({blind([Voto])}vs) + ID election
Step 4
• Validate sintaxe of received data
• Validate {[Voto]}vs
• Communicate with VS
Filter Service (FS)
Step 2
• Validates signature os {user + ID election}sv
• Verifies right to vote of user
• Generate blind signature {blind([Voto])}vs
• Switch user to "Voting" state and links to SHA256({blind([Voto])}vs)
Validator Service (VS)
SV
9. ok/nok
10-a. ID election + [Voto] + {[Voto]}vs + blindComponents + pRComponents + Unique number BVoto
10-b. ok/nok
Step 5
• Store and random mix ([Voto], {[Voto]}vs) and Unique number BVoto
ID election, Unique number BVoto
ID election, ([Voto], {[Voto]}vs, blindComponents , pRComponents)
Anonymizer Service (AS)
Step 6
1: Extraction
• Validate {[Voto]}vs
2: Decipher ballot
• Access to the private key of the election, from the various parts kept by the Electoral Commission
• [Voto] is deciphered and signed by the Counter Service
3: Results od the election
• Ballots are counted
4: Final check
• Verification of ballots and voters in the various states;
5: Results are published
• Analysis of the data and publication of the minutes by the Electoral Commission
6: Voter verifies that its ballot was counted
Counter Service (CS)
12. ID election
13. List with {user} and its status ("Did not vote", "Voted", "Voting")

HTTPS
HTTPS client side
Internet / Intranet

Election – Step 0

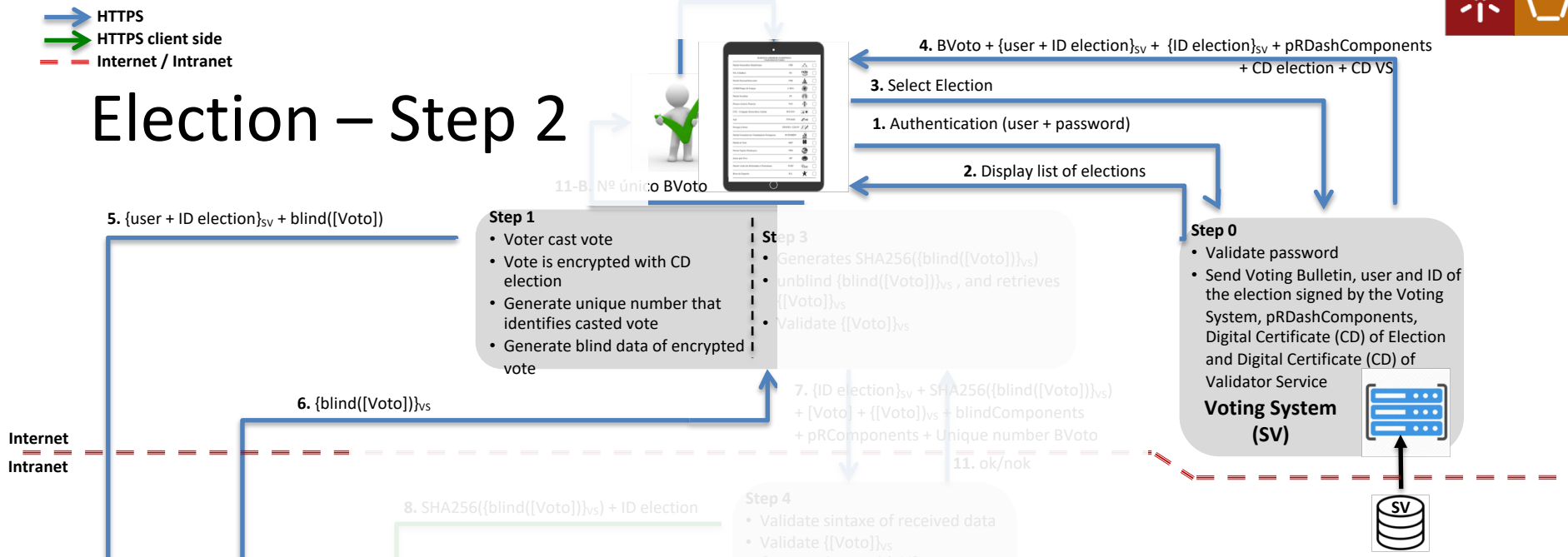


- **Voter** accesses the Voting System and authenticates;
- Authentication is done by sending the user + password of the voter to the Voting System;
- **Voting System** validates the voter's password (PBKDF2);
- **Voting System** lists the elections in which the voter can participate (elections that are in the "open to vote" state);
- The **voter** selects the election;
- **Voting System** sends ID of the election and (voter user + election ID) signed with the Voting System private key, and pRDashComponents, digital certificate of the election and digital certificate of the Validator Service.



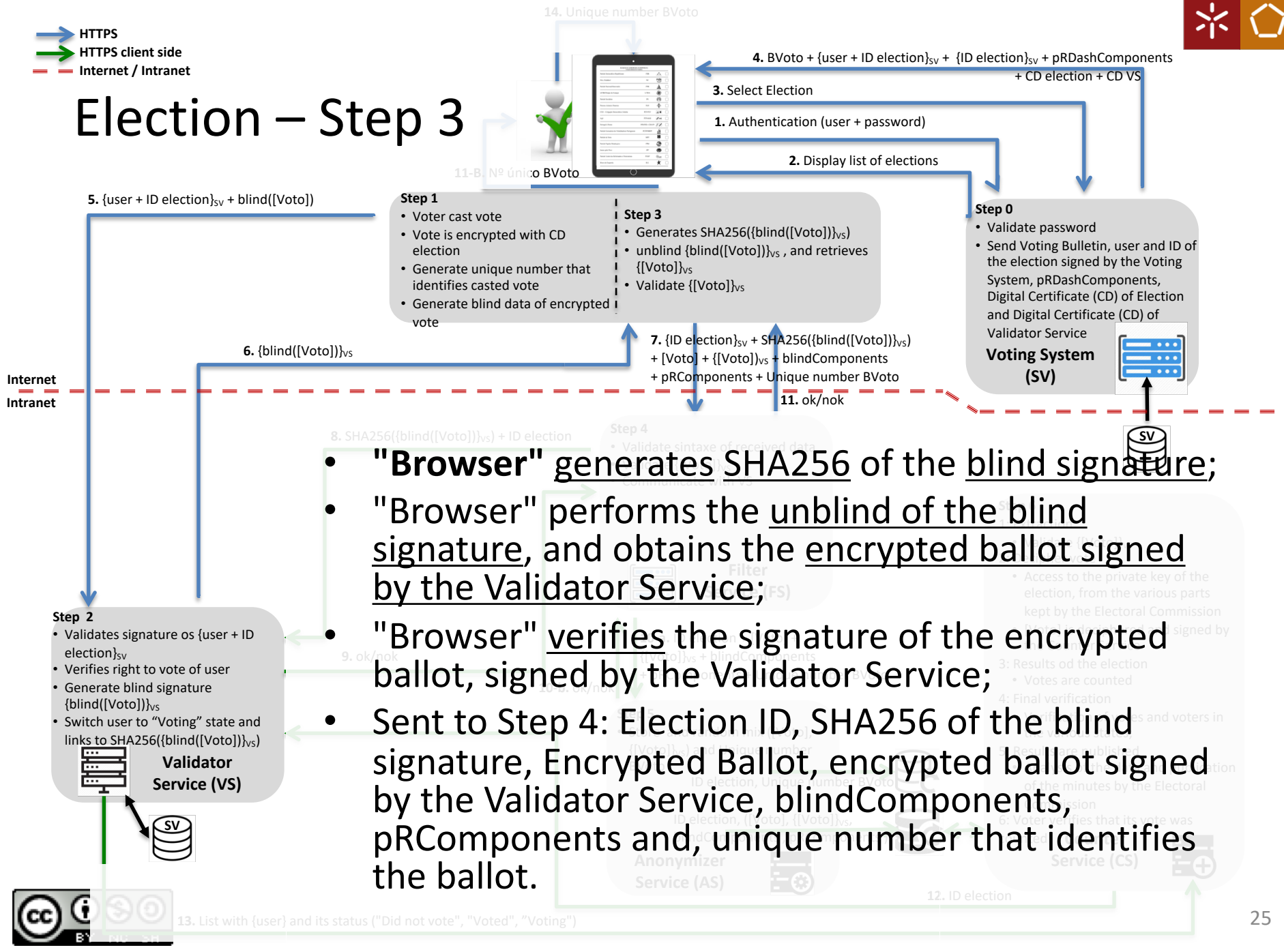
- **“browser”** display voting ballot;
- **Voter** expresses his preference on the ballot paper;
- **Voter** signals that he has voted;
- **"Browser"** transforms the **ballot** into **json format**, adding a field with SHA256(user + ID election);
- "Browser" interprets vote in json, and diaplays it to the voter and asks for confirmation;
- Ballot is encrypted with the public key of the election;
- SHA256 (SHA256 (encrypted vote) + signature (user + ID election)) will serve as the unique number that identifies the ballot;
- Generate blind data on the encrypted vote;
- Send to Step 2: (user + ID election) signed by Voting System, Blind data

Election – Step 2



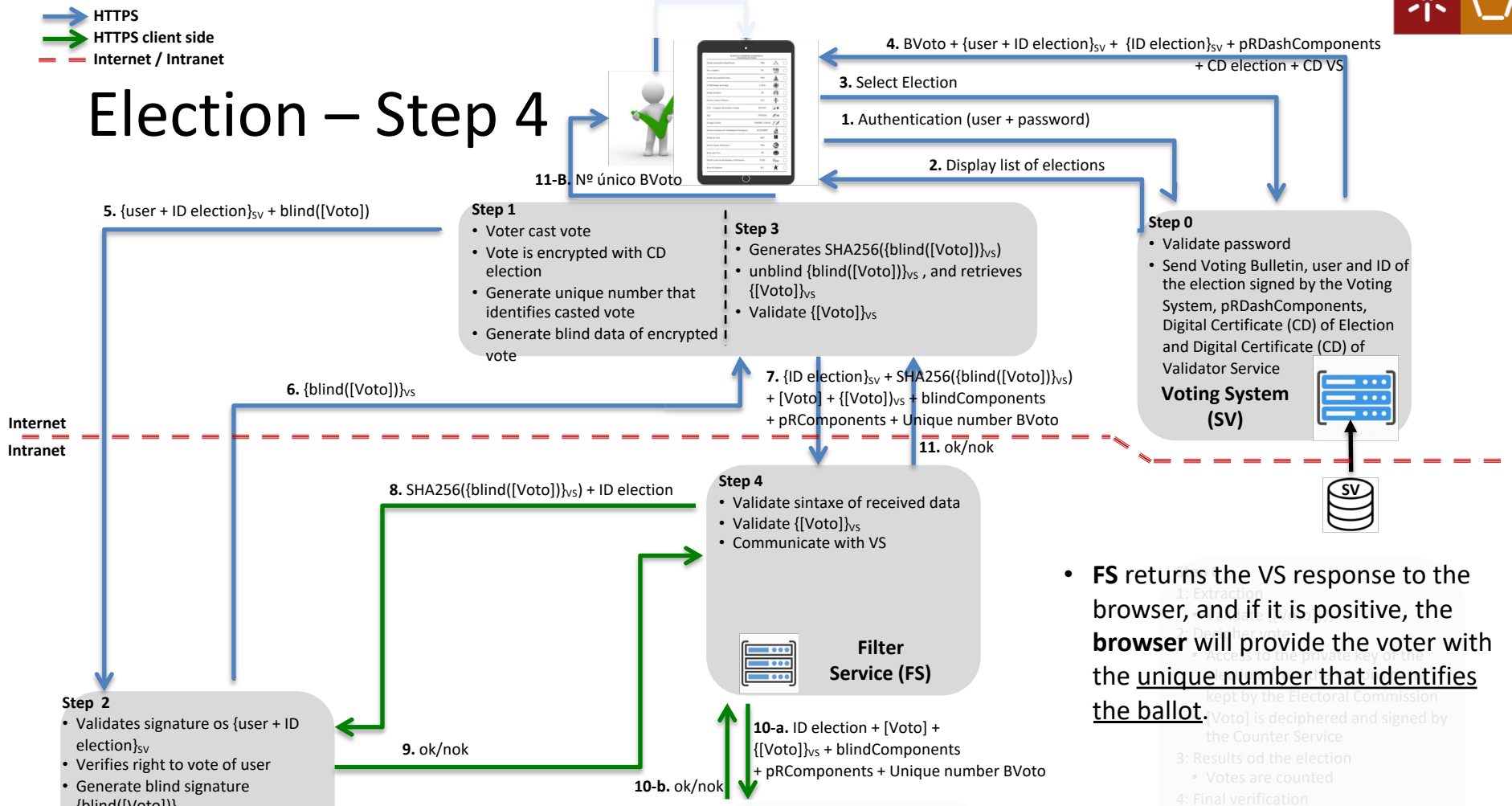
- **Validator Service** validates the signature of the (user + ID election), as well as validates if the signing certificate was issued in the eVotUM hierarchy and if it is within validity;
- VS validates if the ID election is within the voting period;
- VS verifies the voting rights of the user in the ID election;
- VS verifies if the user state in the election ID is "Not voted" (or if it is already in the state "Voting" for more than two minutes), and in that case places the state in "Voting" and adds the time at which the state was changed;
- VS signs the blind data (i.e., it generates the blind signature);
- VS generates the SHA256 of the blind signature and associates it to the user in the ID election (in Step 4, the filter service will use the same SHA256 to inform that the user voted in the ID election)
- The blind signature is sent to Step 3.

Election – Step 3



- **"Browser"** generates SHA256 of the blind signature;
- "Browser" performs the unblind of the blind signature, and obtains the encrypted ballot signed by the Validator Service;
- "Browser" verifies the signature of the encrypted ballot, signed by the Validator Service;
- Sent to Step 4: Election ID, SHA256 of the blind signature, Encrypted Ballot, encrypted ballot signed by the Validator Service, blindComponents, pRComponents and, unique number that identifies the ballot.

Election – Step 4



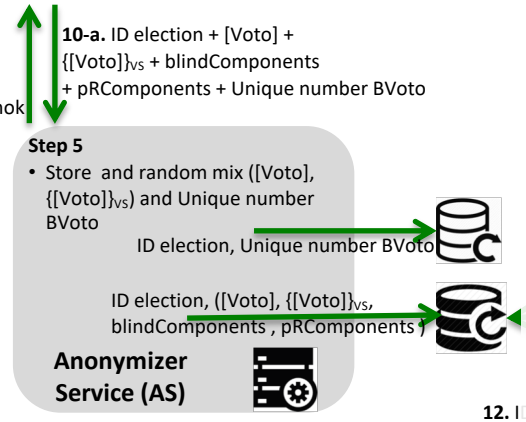
- **FS** returns the VS response to the browser, and if it is positive, the **browser** will provide the voter with the unique number that identifies the ballot.

- **Filter service (FS)** verifies signature of the encrypted ballot signed by the Validator Service, validating also if the signing certificate was issued in the eVotUM hierarchy and if it is within validity;
- **FS** sends election ID and SHA256 of the blind signature to VS. **VS** validates if a SHA256 of the blind signature is in the "voting" state for less than two minutes in the ID of the election. If it is, the state is changed to "Voted" and answers ok to FS (otherwise replies nok);
- Sent to Step 5: Election ID, encrypted ballot, Signature of Encrypted ballot Signed by Validator Service, blindComponents, prComponents, and unique number that identifies the ballot.

HTTPS
HTTPS client side
Internet / Intranet

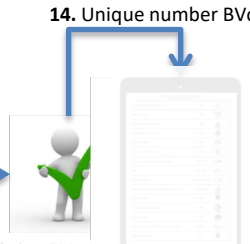
Election – Step 5

- **Anonymizer** stores (and randomly mix) the encrypted ballot, the signature of the encrypted ballot signed by the Validator Service, blindComponents and pRComponents on a table (of the Electoral ballot box) and, stores (and randomly mix) the unique number that identifies the ballot in another table (of the Electoral ballot box);
- Note - Random mix can be processes as follows:
  - Before the election starts, a table is created in the ballot box with three times the entries in relation to the potential voters of the election;
  - Each time a ”packet" to store in the ballot box arrives, a random number between 1 and the number of entries in the table is generated – this random number is the line of the table where the ”packet" is stored;
  - If a ”packet" has already been placed on that line, return to to the previous step until a line that does not have any ”packet” is found;
  - A diferente random numbers is generated for each table, so that the ebcrypted ballot is placed on one random number line, and the unique number is placed on the another random number line.



HTTPS
HTTPS client side
Internet / Intranet

Eleição – Etapa 6



- 5: Election results are published
- 6: **Voter** can verify it its ballot was counte (through the unique number that identifies the ballot).

- 1: Extraction
  - **Counter service** verifies the signature of encrypted ballot signed by Validator Service, validating also if the signing certificate was issued in the eVotUM hierarchy and if it is within validity;
- 2: Decipher ballot
  - The password for access to the private key of the election is generated, from the various parts kept by the Electoral Commission;
  - The ballot is decipheres, and the json structure of the ballot is verified;
  - The ballot (with *timestamp*) is signed by the Counter Service, and stored in the DB;
- 3: Results
  - All the signed ballots (by the Counter Service) are counted;
- 4: Final check
  - The VS is asked for the number of voters in the various states (”Did not vote", "Voted", "Voting") for the election ID;

Internet
Intranet

Step 6
1: Extraction
- Validate $\{[Voto]\}_{VS}$

2: Decipher ballot
- Access to the private key of the election, from the various parts kept by the Electoral Commission
- [Voto] is deciphered and signed by the Counter Service

3: Results od the election
- Ballots are counted

4: Final check
- Verification of ballots and voters in the various states;

5: Results are published
- Analysis of the data and publication of the minutes by the Electoral Commission

6: Voter verifies that its ballot was counted

Counter Service (CS)

Switch user to "Voting" state and links to SHA256({blind([Voto])}$_{VS}$)

Validator Service (VS)

SV

12. ID election

13. List with {user} and its status ("Did not vote", "Voted", ”Voting")