INFO8002 Large-scale data systems
Exercise session IV

Instructions

For the main questions, be ready to:

- Describe the architecture of your solution and its operations.
- Define all its components and their interactions.
- Motivate your design decisions.
- Make diagrams whenever necessary.
- State clearly your choices and assumptions

For feedback, feel free to e-mail joeri.hermans@doct.uliege.be

Question 1.

The Belgian government decided to modernize the centralized vote registration system. The architects (you) of that system are given the following list of requirements:

- Only the vote of Belgian citizens should be registered (no e-id = no vote),
- The voting process needs to be transparent, i.e., votes should be publicly verifiable,
- As the law requires that all citizens have to vote, we also have to identify the citizens that did not cast a vote,
- Non-governmental parties should not be able to identify a voter,
- The voter should be notified whenever his vote has been verified by the system.

How will you tackle these requirements? How will you design your datastructure? What rules will you impose on the system to satisfy these requirements. Be aware that the government has access to a list of all eID's and their corresponding public keys. Provide a description of a sample execution for 'casting a vote', and 'registration of the vote in the system'.

Guidelines

It is recommended to design your system in the following steps:

1. What is the distributed system model you are assuming?
2. Identify all actors in your system. Provide a short description.

3. What is the abstract problem you are trying to solve? What kind of methodologies are available in your toolbox given the distributed system model you've assumed.
4. Sketch the implementation of the interfaces specified above while not considering errors.
5. What happens if a voter casts several votes? How is your system able to handle that?
6. Describe the reliability of your system? Is there an upper limit until which the correctness is guaranteed?
7. **Depending on your implementation:**
8. What kind of broadcast primitives will you use to broadcast the vote of a user? Is it reasonable, and why? Would it scale to millions of (concurrent) users?
9. Draw sketches of your execution (it can be in MS Paint or something, no problem).

Question 2.

Describe *probabilistic finality*. How does it relate to blockchain-protocols and consensus?