Voting

1 Organisational details

1.1 Prerequisites

Each voters receives (through mail) a **voting key** in the form of a 36-character alphanumerical string in the UUID4 format. Using these to identify voters should not be possible. Each helper at the polling station[1] needs to register their information in the database to receive an **identifier** (a UUID4 string). In addition, each polling station has a list of **one-time keys** that can be used to "activate" the voting machines.

1.2 Preparation

Each activation key is valid only for the corresponding polling-station, and it allows to trust requests coming from a machine for a set period of time (two hours), after which the machine needs to be activated with another key. The activation is performed by a voting-helper who need to enter their identifier when activating the machine. The server checks if the helper is registered for the same polling station as the key being used.

1.3 Process

After checking the identity of the voter, they are allowed to go to a voting machine to cast their vote. The machine must be activated, and the voters need to enter their voting key. The server checks that the machine's polling station and the voter's are the same, and if so, the voting key is marked as used and it cannot be used again to vote. After that, the voter leaves the polling station and the machine is ready to be used by another voter.

[1] Although the description uses polling stations "Wahllokale", the actual system operates on the level of regions "Wahlkreise". This is simply for simplicity purposes, as everything else is tracked at that level too.

2 Technical details

2.1 Authorization

In order to verify that the requests are coming from a trusted machine, JWTs (JSON Web Tokens) are used: after the server verifies that the helper provided a valid key to activate the machine, it returns a signed token containing the identifier of the polling station. When a request to cast a vote is received, it is only accepted if the signature is valid and is not expired.

2.2 Security

By using `postgREST`, the system can benefit from the already implemented security measures like checking and validating JWT tokens and sanitizing input to protect against SQL-injections. Moreover, database roles are used to define permissions. For example, the "voter" can execute the "vote" function, but does not have arbitrary write access to the underlying tables. Changing to the correct role is done through `postgREST` on the basis of the JWT-Token, which enables to write declarative permissions that are guaranteed on the database-level.