FELE MEETING MINUTES

Meeting Date: January 31, 2023

Start Time: 09:30 AM

End Time: 10:40 AM

Attendance: Dr. Kewei Sha, Dr. Kwok-Bun Yue, Preethi Vuchuru, Ramachandra Petla, Sooraj Sivadasan Nair, Thomas Pulickal Joseph, Sesha Kumar Reddy Nallamilli, Shiva Phaneendra Reddy Medapati

Authors: Sesha Kumar Reddy Nallamilli, Shiva Phaneendra Reddy Medapati, Ramachandra Petla

Items Discussed

- Successfully setup the Fabric and FELE environment in all our machines.
- Discussed whether to implement MSP or not.
- Discussed about number of Certificate Authorities (CA) to be used and whether to implement it or not.
- Discussed about restricting users from executing certain commands (Role based Access).
- Discussed about using different client applications for accessing ledgers at nodes of different organizations.
- Discussed on refinement of existing simplified FELE environment.
- Talked about creating a Git repository for this project and adding team members and mentors as collaborators for monitoring and working together.
- Discussion on division of labor among the team members.

Decisions Reached

- To come up with a basic design of MSP for FELE. In Fabric, there are channel MSP (sometimes refer to 'global' MSP) and node MSP (sometimes refer to as 'local' MSP). Since FELE does not simulate the physical Fabric network, the current focus will be in channel MSP.) Need to have a MSP for each channel.

MSP: https://hlf.readthedocs.io/en/latest/create_channel/create_channel_participation.html

Network: https://hyperledger-fabric.readthedocs.io/en/latest/network/network.html (Here, it mentions networks and channels are synonymous, which are not.)

Fabric sample configtx.yaml: https://github.com/hyperledger/fabric-samples/blob/main/test-network/configtx/configtx.yaml (Each channel creation needs a yaml).

- We store the certificates for simulation but we don't use them for authentication and dont use CA in our project. There is at least one root CA per organization in a Fabric network. Thus, FELE design needs to be able to handle this.

- Try to employ some restrictions so that admin or some fele users only have access to all features of FELE. Local user, LU logg on the FELE system, which authenticates it. LU1 selects the feleNetwork FN (e.g. Artemis). LU1 wants to call the Smart Contract SC. FELEs find out the mapped FeleUserId, FU, from LU in FN by calling a mapping function to look up CouchDB: i.e. UserMapping(FU, SC). It is a function of FU and SC as LU may have a different mapping for a different smart contract. Permissions are granted by the security policies stored in FN: grant(FU, SC). FN is likely to look up the role of FU in the invoking organization on SC.
- For now, we have to design only one client application for accessing the channel.
- You probably need to have a full stack client application to use FELE-Client-API.
- Research on a testing tool for CLI. A simple technique is that once you are in the FELE interpreter, the user can run a command like "execute -f testScript.fele" in which the file testScript.fele contains fele commands, one line per command."
- Added Fele-LocOrg API in the local organization to update simplified FELE environment.

Items for next meeting

- Tools which are to be used for testing the CLI implementation.
- Can CLI directly interact with smart contracts?
- The question may be more clearly stated.
- Call smart contracts (by interacting with FELE-SC-API)
- Can CLI call FELE-Client-API? We need to think through it. No decision yet.
- Learn channel and network configuration and come up with design to implement channel and network commands functionality.
- We have also discussed the possibility of a security extension unit.
- We also discussed the use of GitHub.