

FELE

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, Sessa Kumar Reddy Nallamilli, Shiva Phaneendra Reddy Medapati

Overall, a good start. Good job.

Who wrote the minutes?

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 You may want to provide more technical details to guide your team.

- To come up with a basic design of MSP for FELE. Details?
 - 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.)
 - You will need to have a MSP for each channel.
 - The Fabric documentation is not very clear on this. Setting up a network also sets up at least one channel (such as Fabric sample).
 - Read:
 - MSP: https://hlf.readthedocs.io/en/latest/create_channel/create_channel_participat.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). You do not need to store the yaml file, but its contents give some ideas what information FELE needs to store.
- We will use only one CA in our project, but we will not authenticate the transactions using the certificates, instead we just store them.
 - We are not actually using any CA, but simulating their uses by stored related information.
 - There are at least one root CA per organization in a Fabric network. Thus, your FELE design needs to be able to handle this.
- Try to employ some restrictions so that admin or some users only have access to all features of FELE.

- Try to be more detailed. There are two kinds of users: (1) Local organization user (lorgUser), and (2) feleUser (which simulates Fabric's users). Thus, "some users" is an ambiguous term.
- Instead, what we have discussed is that there is a mapping between the two kinds of users. Here is how it works.
 - 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). Note that 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 ledger.
 - Do not try to use additional term. In Fabric, a ledger is a channel.
 - You probably need to have a full stack client application to use FELE-Client-API.
 - However, you will also need to have testing mechanism for CLI and RestFUL API. In the meetings, we have discussed using testing and documentation tools, such as Postman and Swagger for RestFul. Do not miss them in your minutes as they have been discussed, and will reappear in future meetings.
 - I have also asked the team to research on 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." This should not be hard to write.
- Added Fele-LocOrg API in the local organization to update simplified FELE environment.
 - Your team was tasked to clear up and refine the FELE architecture diagram. Note that the direction of arrow is the invocation (call) direction.
- CLI, RestfulAPI, JSONSchema as 3 tasks divided among the team members with two members assigned for each task. What exactly are the three tasks?

Items for next meeting

- Tools which are to be used for testing the CLI implementation. Good that you recorded this. Please see above.
- Can CLI directly interact with smart contract?
 - The question may be more clear stated.
 - Can CLI call smart contracts (by interacting with FELE-SC-API? The answer is surely yes, as CLI needs to call SC.
 - 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. See my notes above.
- We have also discussed the possibility of a security extension unit. Since you have six members, you should work on it.
- We also discussed the use of GitHub. You should follow up with it.