Hyperledger Fabric Network Structure

To run a HL network, at least one of each of this entities is needed but there could be as many of them as wanted.

Certification Authority: Provide public and private keys to the **Clients**.

Orderer: They guaranteed the coherence of the DB participant in the ledger.

Couch DB: Database, the ledger.

Peer 1

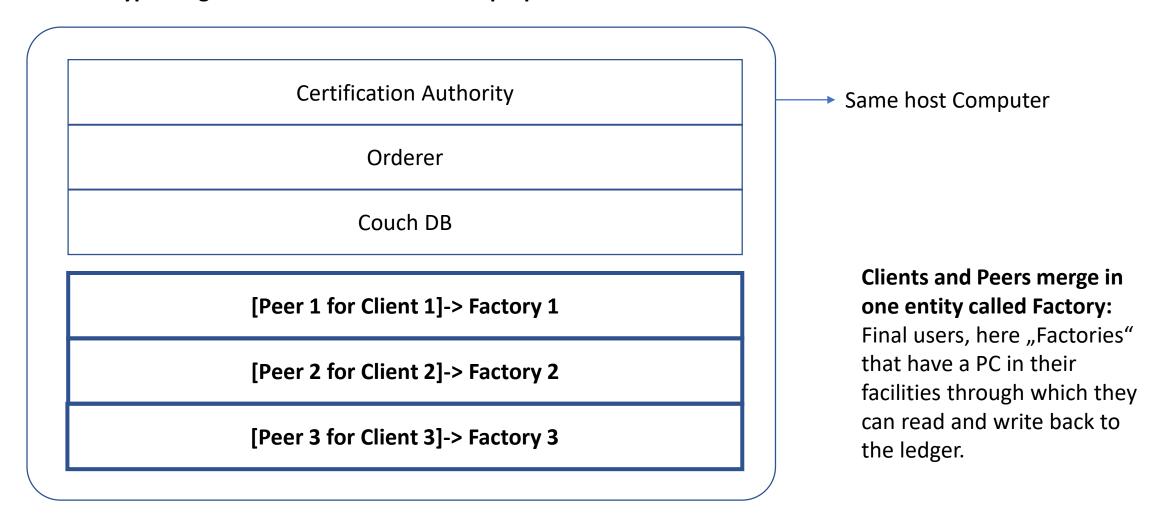
↓

Peer N

Participants of the network thus of the blockchain. **Clients** can interact to the ledger through a **Peer**. The aim of HL is to deploy this peers in different computers. Clients can use any peer to interact with the ledger through their public and private keys.

Clients: Final users, "Humans" who for their own benefit interact with the ledger.

Hyperledger Fabric Network Structure proposed as a solution.



SO FAR; THIS SOLUTION CAN NOT WORK AMONG DIFFERENT COMPUTERS BECAUSE OF DIFFICULTIES WHILE DEPLOYING PEERS.

Hyperledger Fabric Network Structure proposed as a solution walk around: HL Composer Factory 1 Through PYTHON can read Lambda and write back X2 Factory 2: Through **PYTHON** can read Lambda and write back X2 Certification Authority Orderer **LAN REST API** Hyperledger Composer (JSON) Factory 3: Couch DB Through **PYTHON** can read Lambda and write back X3 Peers - Structure of the database: participants, assets, comodities, etc. Inputs to the HL Composer - Logic of the transactions. as configuration files - Permisions of each participant defined in the structure.

HL Composer installed over the HL Fabric Network allows to deploy a REST API where "Fabrics" on the same LAN can access to

Observations:

- HL=hyperledger, HLF= HL Fabric, HLC= HL Composer
- In the HLC structure, the number of peers is not relevant. All the participants are definded in the structure of the HLC database.
- Currently, the HLC solution is deployed to one of the computers participant
 in the network and the others acces to the ledger through this computer.
 However, this behaviour is only because the HLF Network is embedeed in
 one computer. If we achieve to deploy the HLF peers in diferent computers
 and then install the current HLC solution in that network the expectation is
 that the REST API to be replicated in each machine that hosts a peer. Then,
 for each factory the interactions to the ledger is like to interacte to a local
 DB which in turns is replicated and consolidated across the HLF network.
- To achieve the previous point, there is no need to change anything in the HLC files, or add additional code. Only to deploy the HLF network to different computers.