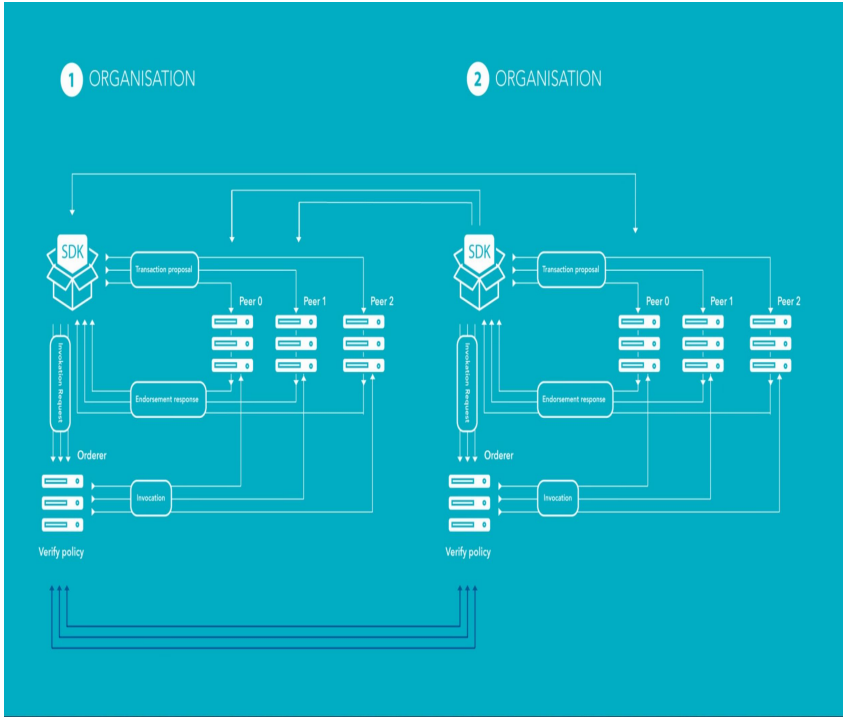
DoxChain - Week 3

Hyperledger dive

Last meeting: Architecture



- Read Alberta's regulation.
- Overview of technologies used for the web app.
- First approach of the blockchain architecture.
- Implementation decisions (geofences as channels, regulator node, distribution/transference/accounting chaincodes, yearly issuance of tokens).
- Team division for upcoming weeks (application/blockchain infrastructure).

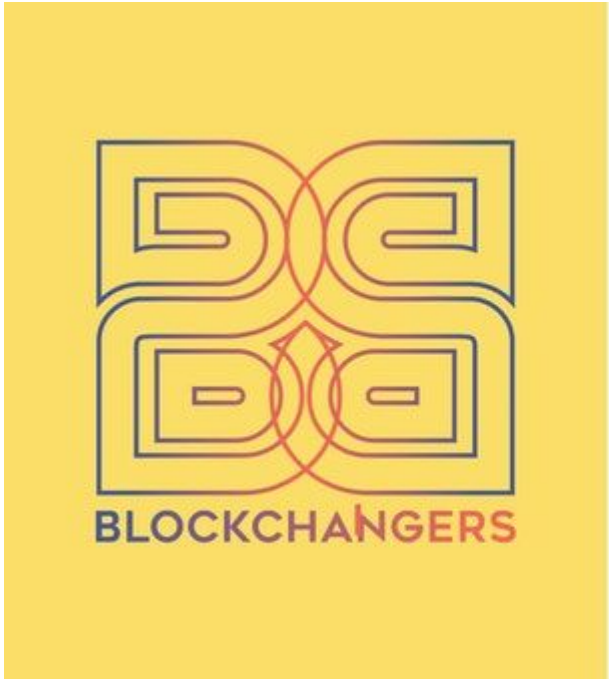
Last meeting: Discussion

- Privacy requirement -> sufficient with public key encryption provided in current version.
- Temperature factor -> model should be able to handle both spot and demand market, therefore one chaincode for transference is not enough. We discussed the possibility of having two different tokens or time constraint tokens. The companies that get demand tokens and don't want to use them can trade them to other companies. We need to make sure a company that issues a demand token does not double spend and can meet demand.
- We can assume that data given by companies is correct.
- Supervisor concerned about hyper ledgers structure, an improper architecture can lead to improper use of the blockchain.

1. Emission trading regulation 2 (parts 5 to 8)

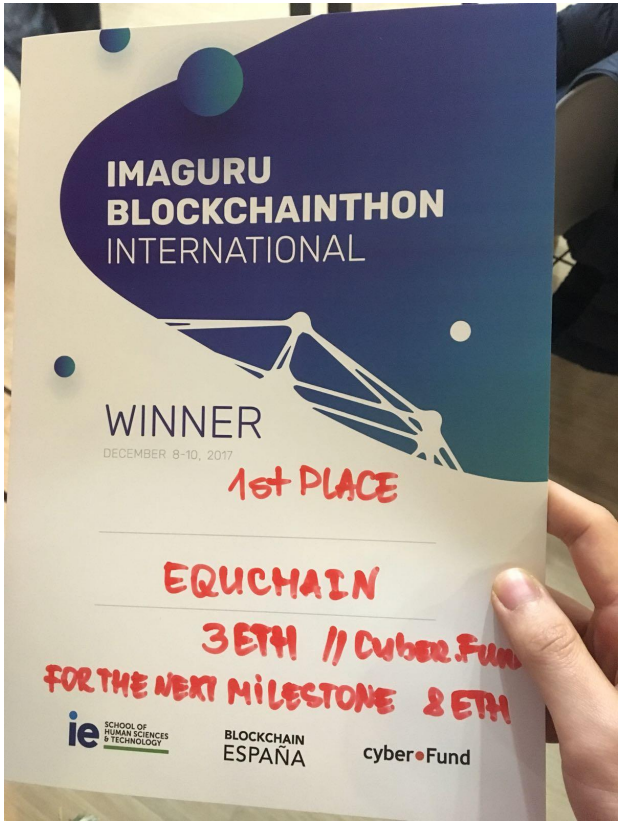
- Emission credit -> unique, licence revocable by law
- Transfer not effective until it is recorded in the registry
- The transfer information must consist of: the transfer and the parties involved, the serial number of the emission credit transferred, the emissions trading accounts affected by the transfer.
- The registry operator: may require a transfer of emission credits to be authenticated, must confirm the transaction has been recorded in the registry, must issue a notice of the transfer.
- Regulator may extinguish an emission credit.

2. Blockchaingers Tech Deep Dive (5/5)

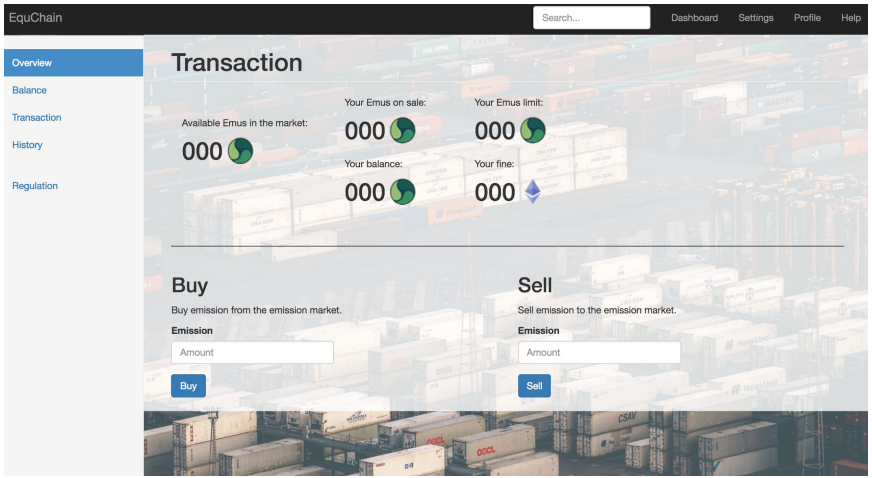


- Attended all the talks
- Discovered Hyperledger Composer
- Attended the Hyperledger Q&A where we solved some doubts
- Connected with Rick Reesen from IBM

3. Blockchain Hackathon (3/5)



- Reusable front end, but ethereum backend



4. Fabric and Composer tutorials -> documentation

“Hyperledger Composer provides an abstraction layer away from Hyperledger Fabric, making it MUCH easier to model and develop Business Applications against a Hyperledger Fabric. The examples use a simple minimal 'Development Fabric' so that developers can concentrate on the Model and the Application and not Focus on the Fabric. **Creating a Multi-Org Multi-Peer Fabric is a significant task in its own right**. Very sophisticated applications can be developed using Composer against a simple development fabric”

5. Market modelling proposal

Asset type 1:

Owned by company

Read by owner, regulator

Defines amount of NoX to spend for each month (future trades)

Jan2018 : 5000

Feb2018 : 4500

etc...

Owner : Company1

Asset type 2: Currency exchange

Owned by regulator

Read by everyone

Written by regulator

Defines how much one unit of tokens is worth in NoX emmision

jan2018 : 10

Feb2018 : 9

etc...

6. Types of contracts we need

1. Regulator creates assets for NoX emissions and transfers the ownership to companies
2. Regulator creates and updates currency exchange asset
3. Companies trade assets (Endorsers are the peers in the company and the regulator)
4. Contract for queries for display and audit purposes

7. IBM expert meeting?

It would be helpful, to validate the fabric design since it not well documented.

Next week proposal

- First proposal of the application side and transaction logic with Composer, on top of basic fabric.
- Advance on the network infrastructure for our case (deep dive fabric documentation).