**030719 - Meeting with Tassos**

1. We had problems using Composer as we found it to be too limited in certain ways. We have decided on using Hyperledger Fabric. Good Idea? Any tips to get started?

* Not using composer is the right decission
* Showing a demo is not a good idea
  + video or Mock-ups enough
  + UI does not have to be the same
  + Everything else must be the same
* Final presentation is not required to have code
* Start:
  + Use Fabric first network tutorial
  + Extend channels and users, etc.; first step only network
  + Later:
    - SHIM fabric API
    - Start creating basic UI -> extend it
  + Start simple!!!

2. Good idea to use CouchDB?

* CouchDB for storing on blockchain
* If we do not need to use bigchainDB we do not need to use it
* BigChainDB e.g. in addition to store the IPFS hashes

3. Get reviews on the transaction diagram.

* Tassos: Diagram “makes sense”
* Tasos is pretty sure that fabric also has events but the events take place outside of the blockchain in UI, etc.
  + Listener -> trigger transaction
    - Watch-statement in JS etc.
* Alternative: Split RegistrationRequest into the two types of registration
  + Everytime e.g.: the chief clicks approve a transaction is triggered
  + In chaincode count number of approves -> invoke transaction after a given number of approves
* Suggestion do it from the chaincode

4. Simplest version that we need to implement? Minimum required? When it comes to grades, what matters the most?

* We can tell more in the story then our prototype is really able to do
* Sequence diagram helpful
* Roleplay maybe in a video
* Not too many technical details

5. Do we need to have a Land Title Asset to keep track of Land Titles (Registrations) or are transactions alone sufficient? Can we go back and look at a transaction and query all details of it?

* In fabric you can get the history of transactions -> you can see the state of the assets
* If it is to complicated -> create new objects of land title..But other solution better

Other:

* Different rights of different users:
  + In Fabric you can enroll users
  + Users have different organisations -> user for chiefs, family members, etc…
* Generating IDs
  + Suggestion create Random ID
  + UUID possible
* Other channel -> new ledger
* 1 channel for our problem okay, if our actions are not privat
* More channely -> more complexity
* One organisation for each role => Chief, CLS, LC
* New organisation for each member -> dynamically add organisation
  + If not possible => specify attributes within one organisation
* Based Acces Control ABAC <https://openblockchain.readthedocs.io/en/latest/tech/attributes/>
* Fabric Example: [https://github.com/hyperledger/fabric/tree/release-1.4/examplesTassos: Two peers is fine](https://github.com/hyperledger/fabric/tree/release-1.4/examples)