# 22.06.2017 Meeting summary

We have presented and discussed our approaches and solutions to exercise 5. We have come to the conclusion that it would be the best to reuse and upgrade some of our previous work incuding

* Reuse and upgrade the contract from group S
  + Add withdraw money
  + Add token creation and distribution
* Reuse and upgrade API controllers from group R
  + Add list projects API
  + Add list project per backer API
  + Add withdraw funds API
  + Add login API
  + Get share API

In terms of project structure and architecture we have decided to go with the following architecture

* Frontend build with Meteor framework
* Contracts deployed on Etherium testnet
* Additional Relational / Document database as helper for listing all projects and project per backer and user management
* Backend which exposes APIs to the frontend to retreive and display relevant data by communicating with the contracts or database according to the request type

Discussed token creation approaches

1. Return TX address + token.
   1. Issue: If project is overfunded, how to calculate the exact ratio of shares the backer gets.
2. Distrubute tokens after the funding goal is reached
   1. Issue: one more transaction needed in similar fashion as when project is killed and creating one more check shares API to communicate shares to the frontend
3. Allow use to claim token only after the project is funded (as a button in the frontend)
   1. Issue: Create get token API and additional logic in the frontend