Fellow Chain

Specification

# Idea Overview

1. I do not like long introductions so I will be as brief as I can. Idea is to enable communities for easily building of cooperatives in order to provide certain kind of insurance for all participants of cooperative.

# Why?

1. **Why Ethereum blockchain?** Because it is first tool that allows for uninterrupted execution of designed algorithms where many different parties are involved and trust among them is limited. Ethereum blockchain is also a platform which enforce full transparency of all participants actions. That allows to build tools that will help all the users to make wise decision about their participation or exclusion both from the system or particular parts of it. Also designing solution on blockchain allows for easy build in governance of all parts of the system.   
   **Why low exposure insurance?** First to provide insurance from moderately probable not very expensive events there is no need for huge, highly regulated companies or governments. While insuring a house or life is a big thing and usually requires complicated regulations and big scale of institution that guaranties execution of an insurance, insurance from for example broken car window, home pet, bicycle, laptop do not require neither scale nor level of complexity. Big corporations usually provide that kind of low exposure insurance for relatively high price, due to management inefficiency.   
   **What are other external benefits of the project?** Main reason to build a project like that and to share it in public domain is to show practical use case of blockchain for ordinary non technical people. Also project in that form encourage people to gather and solve their problems independently from government or corporations serving as shiny example of thesis that people left free can govern themselves very effectively.

# How?

1. The goal of the project is to create a set of smart contracts with together with web UI that will allow for following operations:
2. To Authenticate the user (real identity)
3. To Create new cooperative or join already existing
4. To configure newly created cooperative – add statute (which hash will be on blockchain to provide immutability ), define initial share distribution and price and initial capital
5. To choose way new users are added to cooperative (capital criteria, voting, recommendation etc)
6. To allow users to Exit contract in case of fraudulent behavior of cooperative (return their initial deposit and most of premiums minus taken claim payouts)
7. To allow users to claim insurance payout in case of insurance event
8. To allow cooperative to reject incorrect claim.
9. To allow users and cooperative to go for arbitrage (with additional financial bond to encourage self settlements)
10. Allow users to view performance of all cooperatives and compare them.
11. System should assign reputation based on rate of conflicts and end results of them

# Recognized Entities

Analysis recognize following Entities:

* CooperativeFactory – smart contract responsible for registering and processing of new cooperatives
* CooperativeFactoryRescueFund – smart contract responsible for 2 layer guarantee of claims from broken Cooperatives (funds comes from some constant share )
* Cooperative – smart contract
* CooperativeGovernor – address or smart contract (voting?) rejecting/accepting claims, users, changing insurance rates etc.
* FactoryGovernor - address or smart contract (voting?) rejecting/accepting new Cooperatives
* FundsVault – smart contract which secures funds of cooperative and allows payouts only under strictly defined conditions
* InsuranceFund - smart contract responsible for management of claims
* Claim – action of registration of new insurance event which then need to be analyzed and rejected or approved for payouts, requires deposit from client which must be lost in case of rejection
* Arbitrage – situation when claim is rejected and there is no settlement and client disagree with Claim, require additional deposit from both parties
* Settlement – action after claim is rejected when cooperative offer different sum for insurance event and client agrees

# Further Analysis

There are many topics that are not addressed yet and need further investigation namely:

* Store of personal data (if required)
* Data Availability problem (for instance availability of off-chain documents confirming or denying rightfulness of a claim
* Procedure of analysis if claim is right or wrong (and costs connected to it)
* Bonds for each operation (like claim, rejection, arbitrage)
* Nature of arbitrage
* Exit procedure for users of a cooperative who are not satisfied with how cooperative works