Shares

1.0 total
Shares can be transferred and subdivided

Identity

Wallet addresses?

Treasury

Shareholders can vote to spend from

Payroll

Payments are sent to wallets (auto?)
May or may not require auth first

Roles

Shareholders can vote to add an individual to a roll

Secret Mgmt

Role membership grants access
to secrets

Shares

This is the most important piece of defining an organization.
Need to be able to create a smart contract without direct ownership.
The pub key of the original creator simply gets all 1.0 shares.
Then it goes to the share distribution contract.

So basically at the core, it's all just a share distribution contract.
This looks like the right place to start: MultiSig.hs

I guess this is the part where I need to learn Haskell/Plutus and try making such a contract.

https://prod.playground.plutus.iohkdev.io/

Shares

Init :: PubKey
Initializes the shares and gives them all to the owner of the public key

Give :: PubKey → Float
Give recipientKey amount = ...
Transfers a specific amount of share to a recipient

Honestly, this is seriously all that's necessary. Maybe we can bootstrap this thing on Ethereum.

MVP GUI

- This is where we make our money. A GUI that makes it easy to manage your org.
- Share (org) creation
- Share xfer
- Role mgmt
- Secret mgmt
- Treasury mgmt
- etc