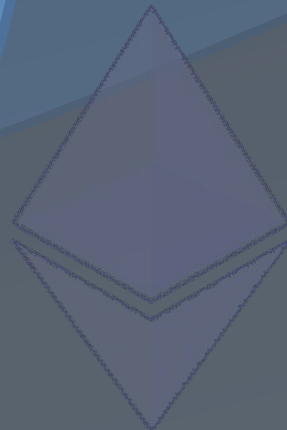
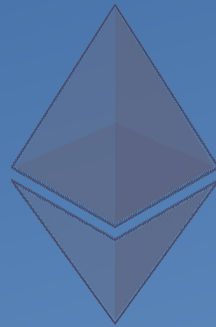


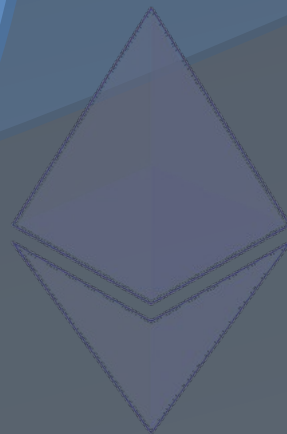
# Ethereum Vienna

## General Introduction



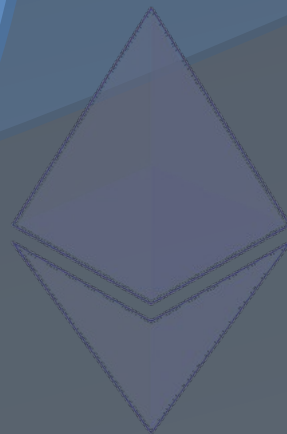
# Ethereum Project

- Decentralization of services
- Removing the role of centralized servers
- Control goes from server owner to users
  - Server can't disappear with your data
  - Server can't just randomly modify your data
  - Server can't just freeze your funds
  - Censorship-proof
  - DDOS-resistant



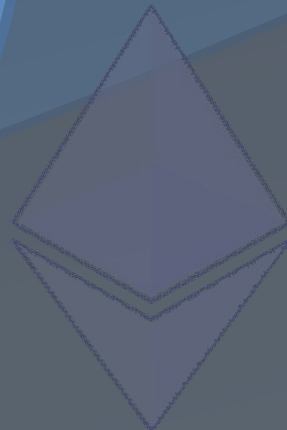
# Web 3.0

- Dapps (run in a Web 3.0 client)
  - **Ethereum** (Blockchain)
    - Agreements
    - Relationships
  - **Whisper**
    - Messaging
    - Bulletins
  - **Distributed Content System** (“Swarm”)
    - Data publication and distribution



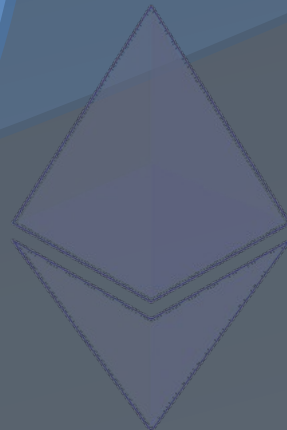
# Possible DApps

- Escrow (m-of-n transactions)
- Namecoin (decentralized dns)
- Subscription Service
- Crowdfunding
- Subcurrencies
- Decentralized Autonomous Organizations
- Marketplace



# Ethereum Blockchain

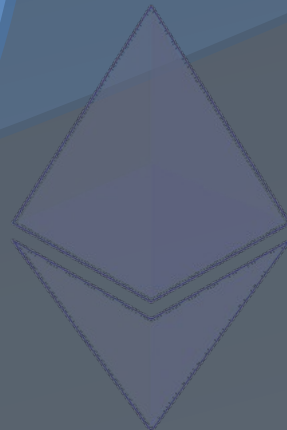
- Maintains Accounts with balances denominated in ether/wei
  - **Externally owned** (account)
    - Controlled by a private key
    - Owner can send ether to other accounts
    - Similar to normal bitcoin addresses



# Ethereum Blockchain

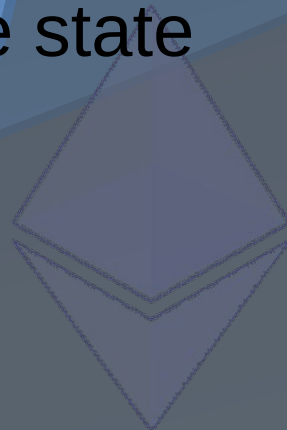
## – Internally owned (contract)

- Controlled by code
- Code is executed for each incoming transaction/message
- No private key, ether can only be sent by the code
- Has a 256 byte to 256 byte persistent storage
- Can call other contracts
- Code written in an ethereum-specific language:
  - **Solidity**: high-level, main language (still in development)
  - **Serpent**: python-like
  - **LLL**: low-level



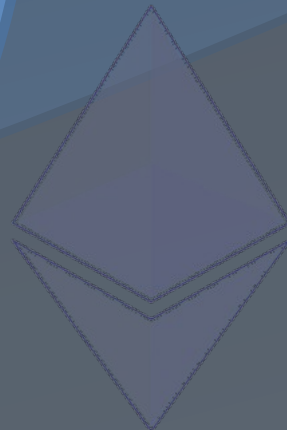
# Ethereum Blockchain

- Gas
  - Used for transaction fees
  - Sender “buys” necessary amount of gas at a specified **gasprice** (goes down as price goes up)
  - Every computational step has an associated gas cost
  - Remaining gas is returned to the sender
  - If the sender does not provide enough gas, the state reverts and the miner keeps the ether



# Ethereum Blockchain

- Gives messages an order
- Messages are grouped together in blocks
- Blocks are chained together
- Longest chain is considered valid
- 12s Block Time (made possible with uncle blocks)
- Hybrid PoW (ASIC-resistant) / PoS (planned)
- Constant Block Reward (dis-inflationary)

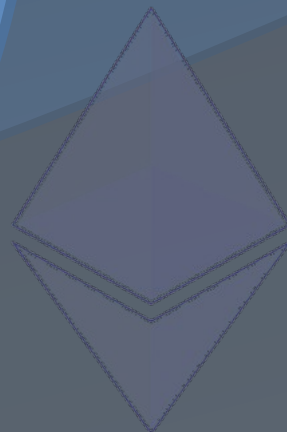




# Crowdfund

- Keeps track of crowdfunding campaigns
- Automatic payout if goal is reached
- Automatic payback if campaign fails
- 3 functions
  - **create\_campaign** *<id>* *<recipient>* *<goal>* *<timelimit>*
  - **contribute** *<id>*
  - **progress\_report** *<id>*

does not change state, only executed locally

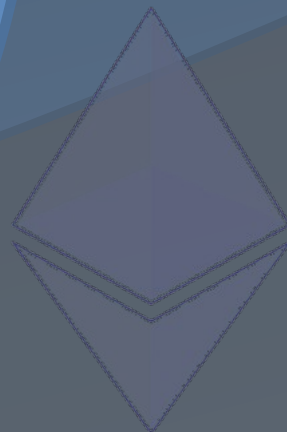


# Crowdfund

```
data campaigns[2^80](recipient, goal, deadline, contrib_total, contrib_count,  
contribs[2^50](sender, value))
```

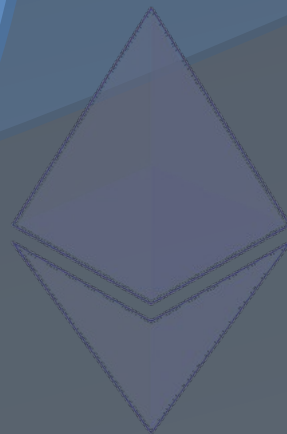
```
def create_campaign(id, recipient, goal, timelimit):  
    if self.campaigns[id].recipient:  
        return(0)  
    self.campaigns[id].recipient = recipient  
    self.campaigns[id].goal = goal  
    self.campaigns[id].deadline = block.timestamp + timelimit  
  
def contribute(id):  
    # Update contribution total  
    total_contributed = self.campaigns[id].contrib_total + msg.value  
    self.campaigns[id].contrib_total = total_contributed  
  
    # Record new contribution  
    sub_index = self.campaigns[id].contrib_count  
    self.campaigns[id].contribs[sub_index].sender = msg.sender  
    self.campaigns[id].contribs[sub_index].value = msg.value  
    self.campaigns[id].contrib_count = sub_index + 1  
  
    # Enough funding?  
    if total_contributed >= self.campaigns[id].goal:  
        send(self.campaigns[id].recipient, total_contributed)  
        self.clear(id)  
        Return(1)
```

```
...
```



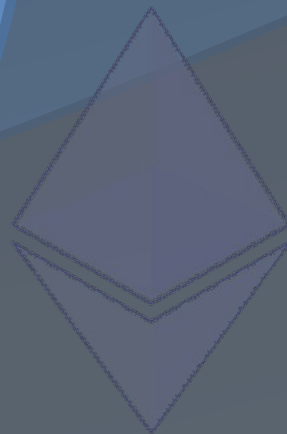
# Who?

- Ethereum Stiftung
  - Allocates resources
- ethereum Switzerland GmbH
  - Responsible for genesis-block-related tasks
- ÐΞV
  - Nonprofit
  - Building and promoting Ethereum 1.0



# Ether Sale

- Development funded via crowdfunding
- 31,529 BTC (~12.5m USD)
- Over 9000 transactions
- 2<sup>nd</sup> biggest crowdfunder



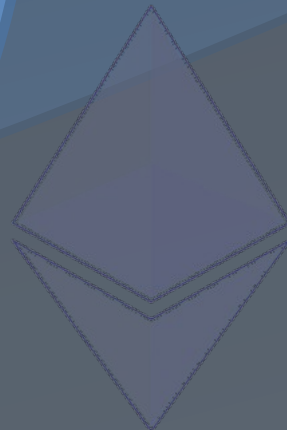
# Who?

- Vitalik Buterin
  - Invented the concept of ethereum
  - Co-Founder / Writer of Bitcoin Magazine in 2011
  - 2014 World Technology Award (IT Software)
  - Thiel Award



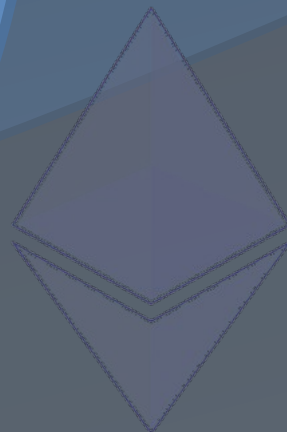
# Whisper

- Decentralized Messaging
- Messages are assigned a topic
- Private messages encrypted
- Public broadcasts
- Dark (no reliable tracing mechanism)
- Not designed for RTC



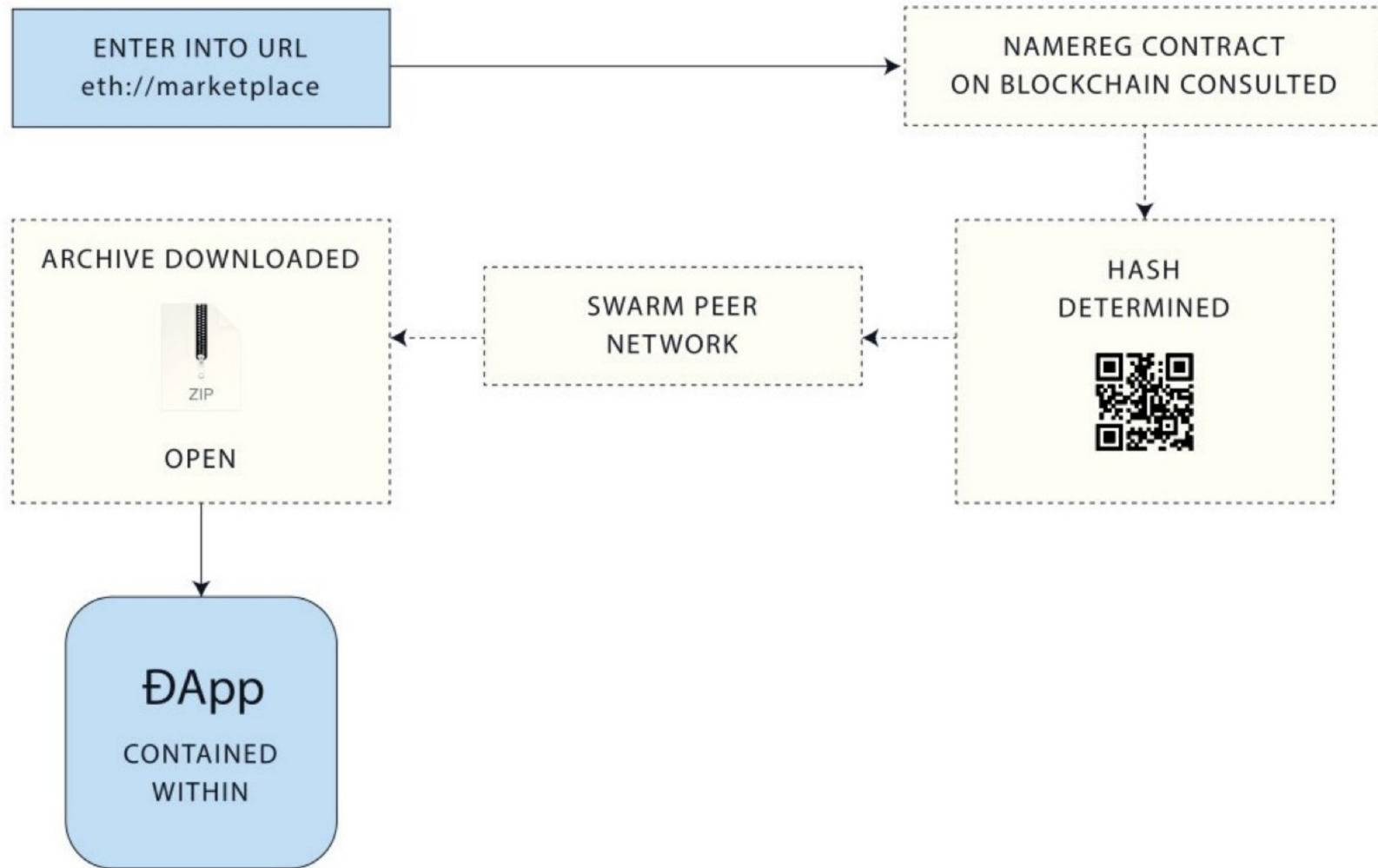
# Distributed Content System

- Not yet chosen. Needs those properties:
  - Reverse Hash Table
  - Like bittorrent with magnet links
  - Private
  - Low-latency
  - Incentivised (content can get lost if no one pays maintenance)



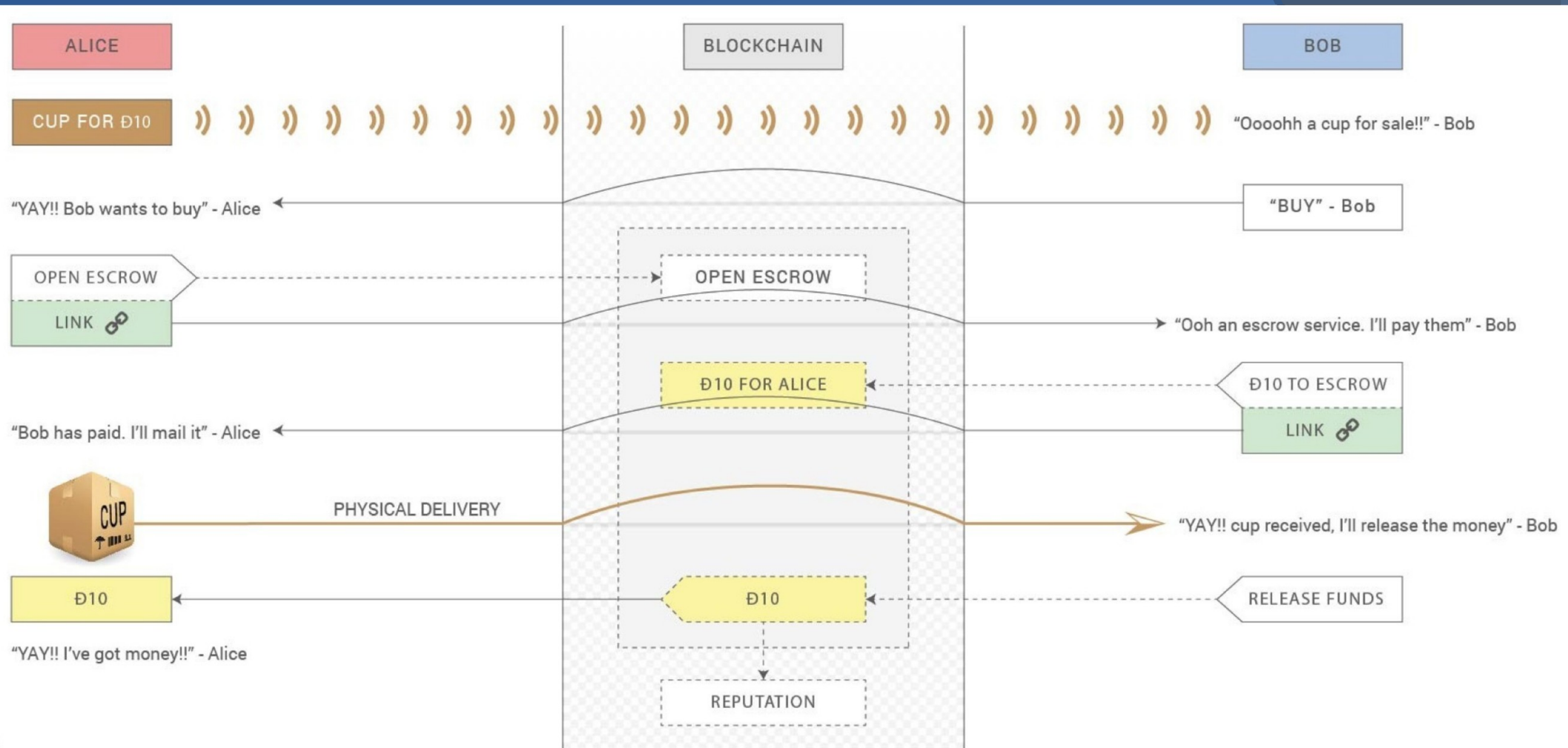


# Marketplace

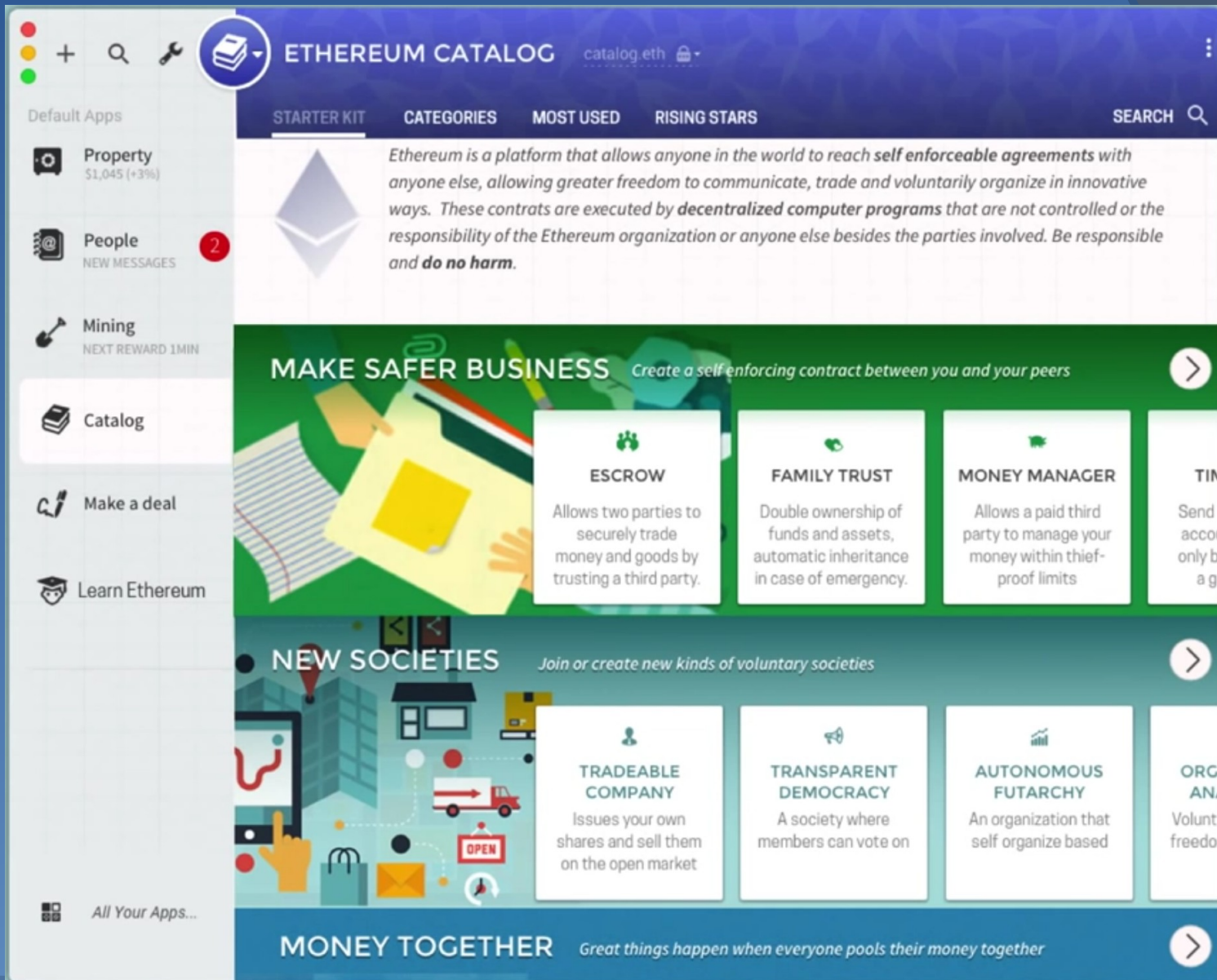




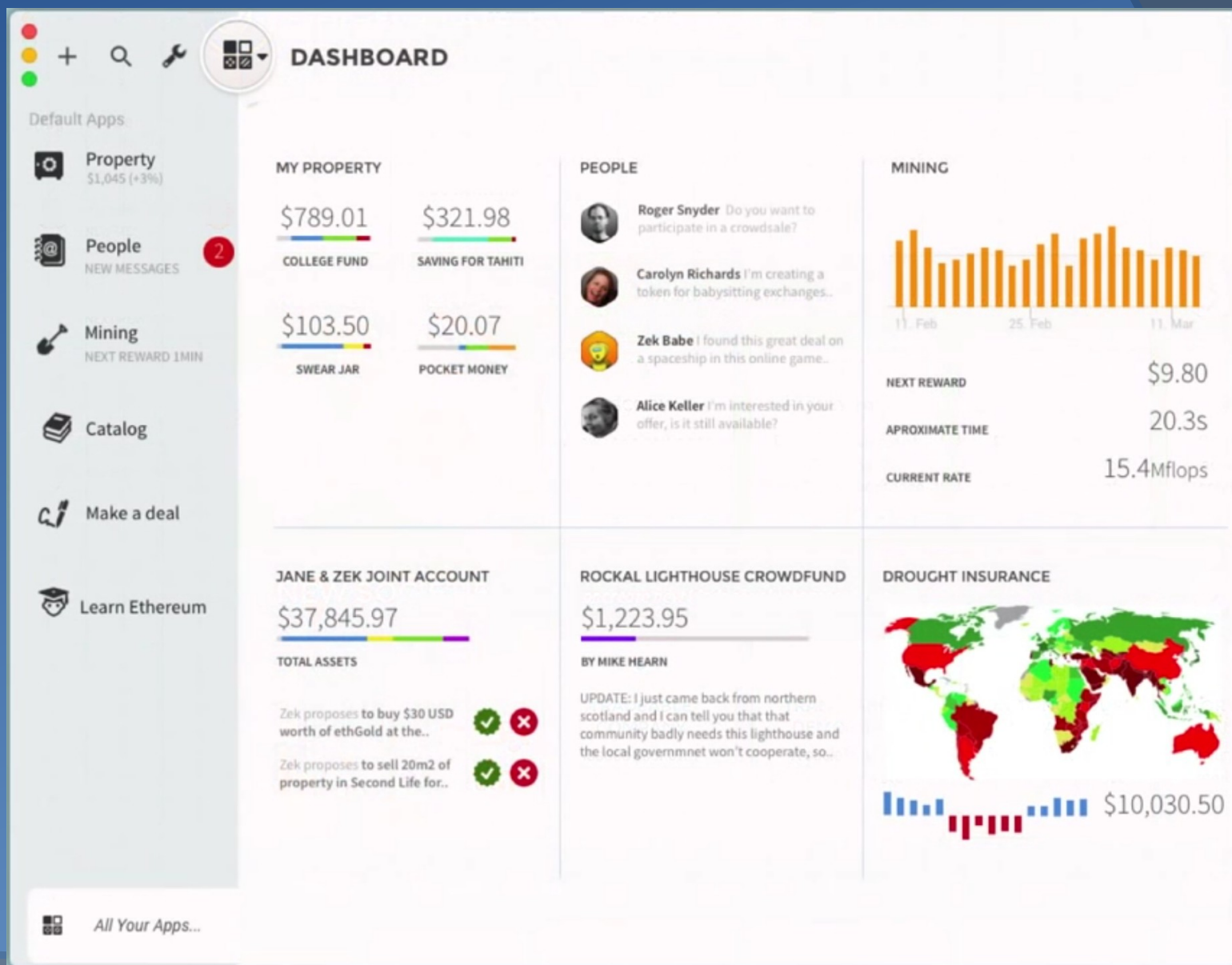
# Marketplace



# Mist (Web 3.0 Client)



# Mist (Web 3.0 Client)



# Mist (Web 3.0 Client)

The screenshot displays the Mist Web 3.0 Client interface for a joint account. The top header shows the account name 'JANE & ZEK JOINT ACCOUNT' and a public key '74be16979710d4c4e'. Below the header are three tabs: 'JOINT DECISIONS', 'TOTAL ASSETS', and 'ADVANCED OPTIONS'. The left sidebar contains a 'Default Apps' section with icons for Property, People, Mining, App Catalog, Make a deal, and Learn Ethereum. The main content area is titled 'PENDING DECISIONS' and contains a list of decisions made by the partner, each with a 'REFUSE' or 'ACCEPT' button. The decisions are:

- Buy 5 BTC at current market price**  
Initiated by Zek at [etherexchange.eth](#). This order will buy the bitcoins at etherexchange, if the price is between [see details](#). Buttons: REFUSE, ACCEPT.
- Buy an Eve Online Oracle Battlecruiser in for \$5,000 USD**  
Initiated by Zek at [spaceshipauctions.eth](#). "This is a great investment opportunity! Properly maintained this beauty could..." [see details](#). Buttons: REFUSE, ACCEPT.
- Buy house insurance for \$1,200 USD in 24 installments**  
Initiated by Jane at [decentralizedinsurance.com](#). Protection against accidental fire, earthquakes, flood and small leaks.. [see details](#). Button: REVOKE DECISION.
- Create an account at Couch Exchange**  
Initiated by Zek at [couchexchange.eth](#). This will share your public key, name, info and home address with the site. [see details](#). Buttons: REFUSE, ACCEPT.

Below the pending decisions is a section for 'PAST DECISIONS'.