ethereum vienna

General Introduction

Goals

Decentralisation of the web

Removing the role and power of central points

Take away control from service operators

Reduce trust requirements between parties

Why decentralise?

Data cannot just disappear

Data can only be modified by certain rules*

- provides audit trail
- protects system state from manipulation

Censorship resistant

Server cannot freeze funds

* most of the time

ethereum

Project

Platform for decentralised applications (**DApps**)

Ethereum (Blockchain)

Consensus Layer

Whisper

Messaging and Broadcasting

Swarm / IPFS (Content System)

Data publication and distribution

DApps

Escrow Standard UI Wallet

Crowdfunding Weifund

Insurance etherisc

Prediction Markets Augur / Gnosis

Registries ENS

Marketplace Safemarket

Decentralised Autonomous Organisations (DAO)

Stablecoins MakerDAO

ethereum

blockchain

ethereum

Blockchain

Public record of all transactions

Stored and processed by all full nodes

Determines order of transactions

Necessary to compute the state of the system

This enables **global consensus** over the current state

Enterprise

Public Blockchain

- Public Ledger
- Anyone can participate
- Proof of Work
- Expensive
- Global consensus
- Rollbacks by mining majority

Enterprise Blockchain

- Private Ledger
- Access restricted
- PBFT
- Cheaper
- Local consensus
- Rollbacks by node majority

ethereum

Enterprise Alliance

Blockchain

Account based System

- identified by a 160 bit address
- has a balance of Ether / Wei

2 types of accounts

- "Accounts" (external)
- Contracts (internal)

Blockchain

Account (external)

user controlled account

controlled by a private key

can send and receiver ether

0x1350cf34d093953ce0d2803648da8f3b6a84de77	100
0xd5f9d8d94886e70b06e474c3fb14fd43e2f23970	2500
0xd2963cd505c94dbf3bc663bdd2321bd3000204bb	23290
0xd2963cd505c94dbf3bc663bdd2321bd3000204bb	123809
...	...

Blockchain

Contract (internal)

Controlled by code (EVM byte-code)

Gets executed whenever it receives a message (e.g. ether transfer, function call)

Ether can only be sent out by the code

Persistent storage to preserve state across transactions

Can also call other contracts during its execution

```
DUP2 SWAP1 SSTORE POP DUP5 DUP5 POP PUSH1 0x6 ADD
PUSH1 0x0 SWAP1 SLOAD SWAP1 PUSH2 0x1 0x0 EXP SWAP1
DIV PUSH1 0xff AND PUSH2 0x6 0x88 JUMPI DUP5 DUP5
POP PUSH1 0x1 ADD PUSH1 0x0 POP SLOAD DUP4 LT
ISZERO PUSH2 0x5 0x8e JUMPI PUSH2 0x6 0x83 JUMP
JUMPDEST DUP5 DUP5 POP PUSH1 0x0 ADD PUSH1 0x0
```

ethereum

Blockchain

Code written in an ethereum specific language

- Solidity

high level

official language

- lll

  lisp-like (low level)

- EVM Assembly

```
contract Coin {

    event Transfer(address indexed from, address indexed to);

    mapping (address => uint) public balances;

    function() {
        balances[msg.sender] = 10;
    }

    function Send(address to, uint amount) {
        if(balances[msg.sender] >= amount) {
            balances[msg.sender] -= amount;
            balances[to] += amount;
        }
    }
}
```

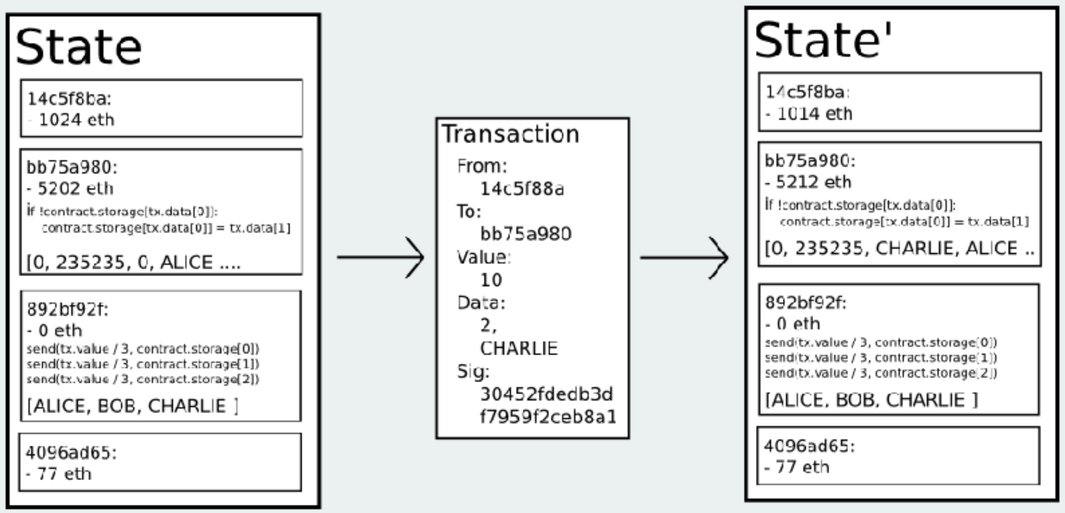
Blockchain

Transaction

Signed by a private key (external account)

Transitions from one state to the next

Can transfer ether, call contract functions, etc.



ethereum

Blockchain

Gas

- Used for transaction fees
- Sender “buys” gas at a **sender-specified gasprice**
- Every computational step has a fixed gas cost
- Remaining gas sent back to sender
- If gas runs out
  - the state reverts (including any ether transfers)
  - but miner keeps ether

ethereum

Blockchain

Gasprice

Associated gas cost for some action is constant

But the price of ether is not

Gasprice can be a scale factor against ether price

=> but there is also a lower bound due to block reward

Ether goes up -> Gasprice goes down

Ether goes down -> Gasprice goes up

ethereum

Blockchain

Example

Bob sends a transaction to contract C

He provides 100000 gas at a gasprice of 0.00001 eth

Minimal transaction cost: 0.21 eth (=21000*gas price)

Maximal transaction cost: 1 eth (=100000*gas price)

Exact cost only certain when included in a block

Blockchain

Example

Actual gas usage: 30k

Transaction cost: 0.3 eth (=30k*0.00001)

Bob gets 0.7 eth back (=> Bob does not overpay)

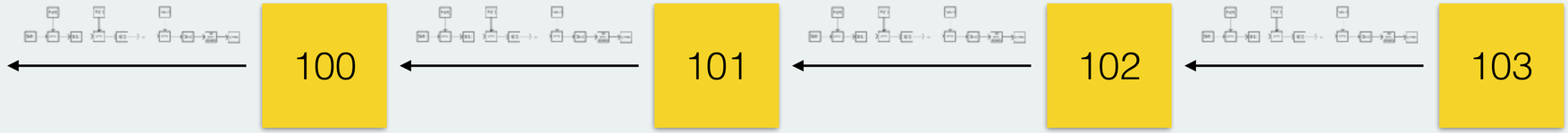
Transaction ends with an error (=100000 gas usage)

Transaction cost: 1.0 eth (=100k*0.00001)

Blockchain

Blockchain gives transactions an order

Transactions are grouped together into blocks (~15s apart in time)



Order is important:

- Double spend (no unspent outputs, but balance might become 0)
- 2 transactions interacting with the same contract
- Different order -> Potentially different outcome

Blockchain

Proof of Work (Ethereum 1.0)

- EthHash
  - asic-resistant (high memory, io bandwidth)
  - targets gpu mining (2GB+ GRAM)

To be succeeded by Casper (PoS)

Constant Block Reward during PoW Phase

ethereum Yellow Paper

ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER
HOMESTEAD DRAFT

0xf1 CALL 7 1 Message-call into an account.

$\mathbf{i} \equiv \boldsymbol{\mu}_\mathbf{m}[\boldsymbol{\mu}_\mathbf{s}[3] \dots (\boldsymbol{\mu}_\mathbf{s}[3] + \boldsymbol{\mu}_\mathbf{s}[4] - 1)]$

$$(\boldsymbol{\sigma}', g', A^+, \mathbf{o}) \equiv \begin{cases} \Theta(\boldsymbol{\sigma}, I_a, I_o, t, t, & \text{if} \quad \boldsymbol{\mu}_\mathbf{s}[2] \leqslant \boldsymbol{\sigma}[I_a]_b \wedge \\ \quad C_{\text{CALLGAS}}(\boldsymbol{\mu}), I_p, \boldsymbol{\mu}_\mathbf{s}[2], \boldsymbol{\mu}_\mathbf{s}[2], \mathbf{i}, I_e + 1) & \quad I_e < 1024 \\ (\boldsymbol{\sigma}, g, \varnothing, \mathbf{o}) & \text{otherwise} \end{cases}$$

$n \equiv \min(\{\boldsymbol{\mu}_\mathbf{s}[6], |\mathbf{o}|\})$

$\boldsymbol{\mu}'_\mathbf{m}[\boldsymbol{\mu}_\mathbf{s}[5] \dots (\boldsymbol{\mu}_\mathbf{s}[5] + n - 1)] = \mathbf{o}[0 \dots (n-1)]$

$\boldsymbol{\mu}'_g \equiv \boldsymbol{\mu}_g + g'$

$\boldsymbol{\mu}'_\mathbf{s}[0] \equiv x$

$A' \equiv A \uplus A^+$

$t \equiv \boldsymbol{\mu}_\mathbf{s}[1] \mod 2^{160}$

where $x = 0$ if the code execution for this operation failed due to an exceptional halting $Z(\boldsymbol{\sigma}, \boldsymbol{\mu}, I) = \top$ or if $\boldsymbol{\mu}_\mathbf{s}[2] > \boldsymbol{\sigma}[I_a]_b$ (not enough funds) or $I_e = 1024$ (call depth limit reached); $x = 1$ otherwise.

$\boldsymbol{\mu}'_i \equiv M(M(\boldsymbol{\mu}_i, \boldsymbol{\mu}_\mathbf{s}[3], \boldsymbol{\mu}_\mathbf{s}[4]), \boldsymbol{\mu}_\mathbf{s}[5], \boldsymbol{\mu}_\mathbf{s}[6])$

Thus the operand order is: gas, to, value, in offset, in size, out offset, out size.

$C_{\text{CALL}}(\boldsymbol{\sigma}, \boldsymbol{\mu}) \equiv G_{call} + \boldsymbol{\mu}_\mathbf{s}[0] + C_{\text{CALLXFER}}(\boldsymbol{\mu}) + C_{\text{CALLNEW}}(\boldsymbol{\sigma}, \boldsymbol{\mu})$

$$C_{\text{CALLXFER}}(\boldsymbol{\mu}) \equiv \begin{cases} G_{callvalue} & \text{if} \quad \boldsymbol{\mu}_\mathbf{s}[2] \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

ethereum

Whisper / Swarm
Mist

ethereum

Whisper

Decentralised Messaging

Messages can be filtered by topics

Very flexible

- Messages can be encrypted
- Messages can be signed
- Broadcast

PoW for spam protection and priority

Not designed for real time communication

ethereum

Swarm

Swarm

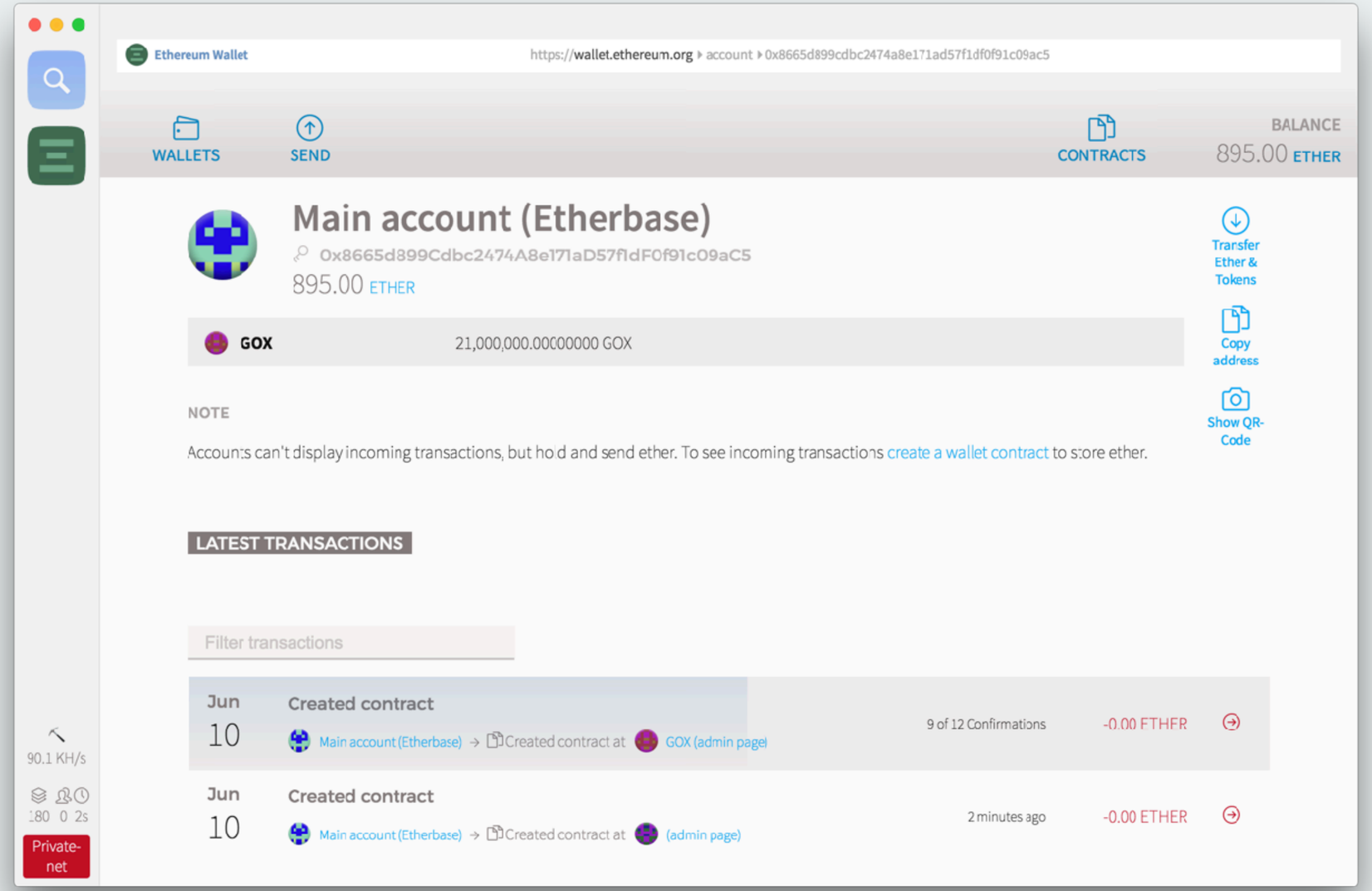
- Reverse Hash-table
- Distributed chunk store
- Low-latency
- Incentivation model for storage

part of go-ethereum

Orange Papers

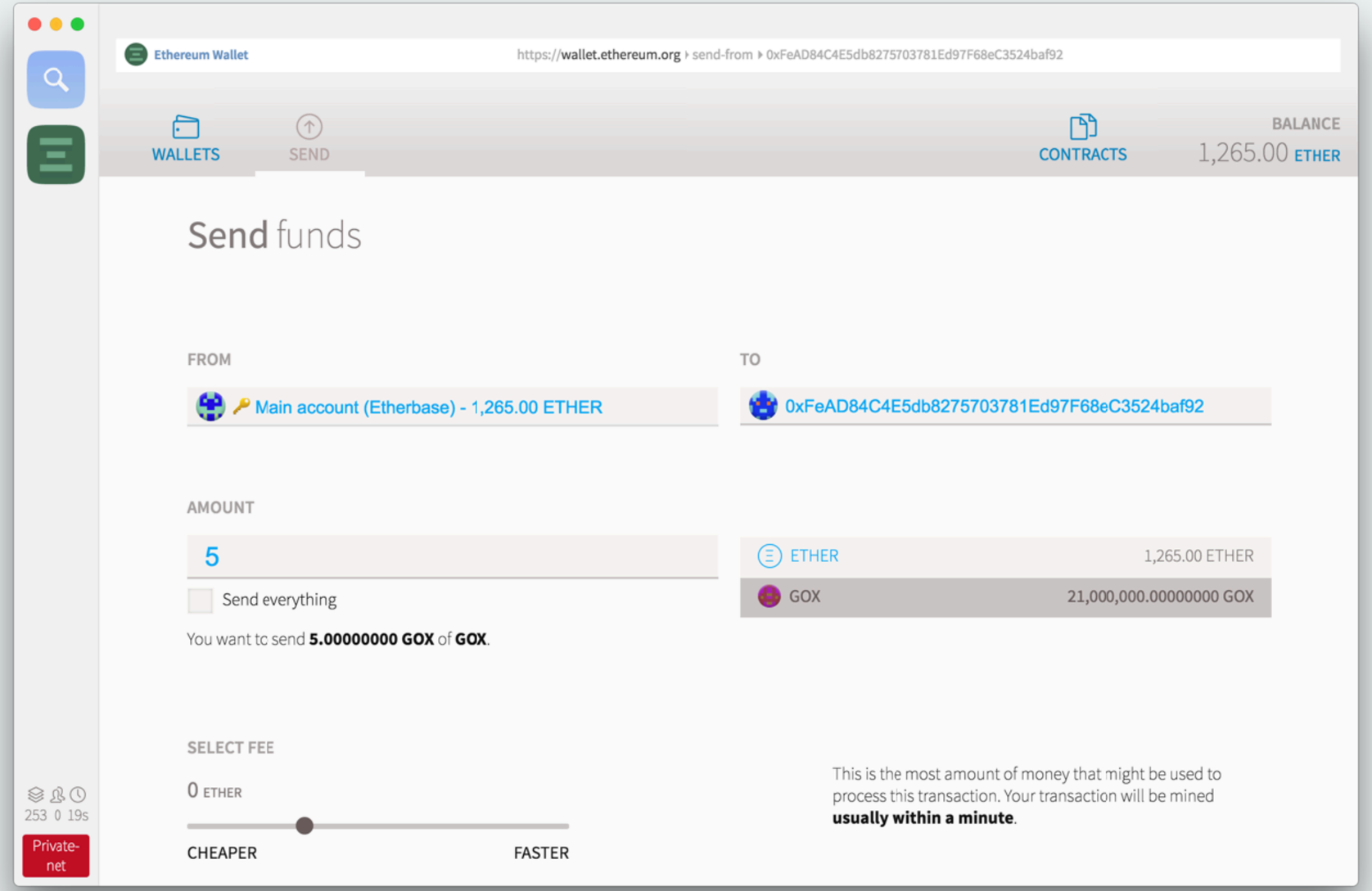
ethereum

Mist / Wallet



ethereum

Mist

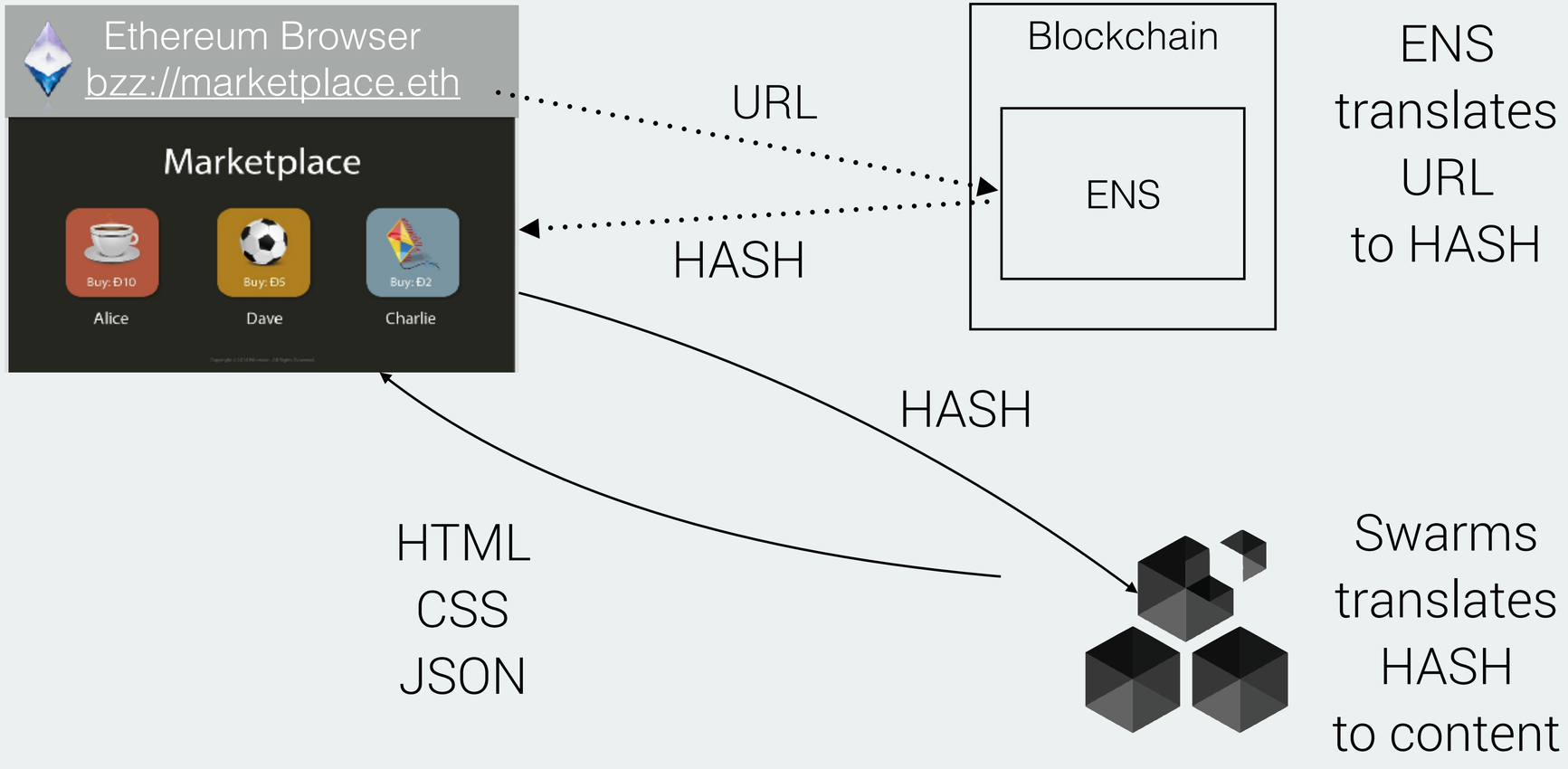


Marketplace DApp

(Badly designed) Example

ethereum Marketplace

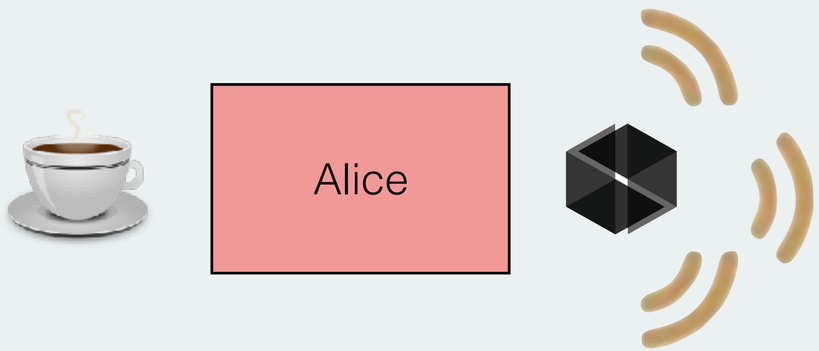
User enters URL



ethereum Marketplace

Alice wants to sell a cup for 10 ETH

Whisper Broadcast
"I want to sell a cup for 10 ETH"

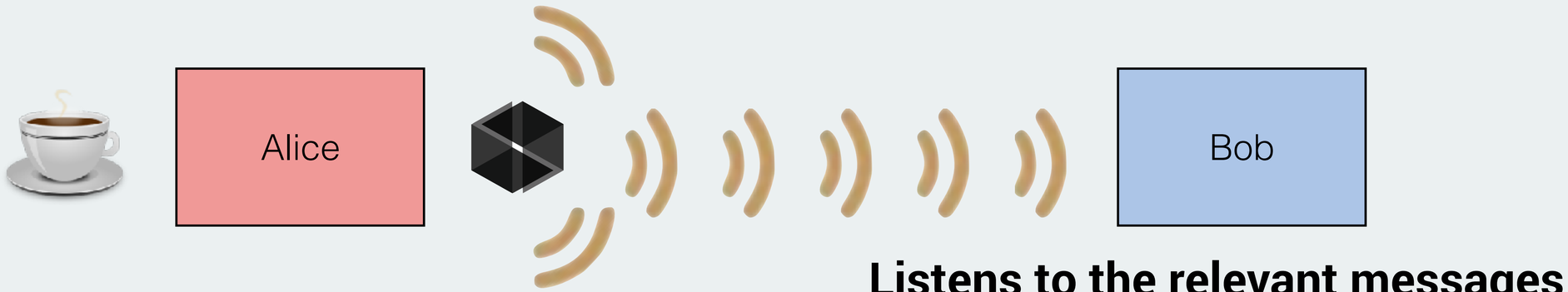


Broadcasts a Whisper message

ethereum Marketplace

Bob wants to buy cups

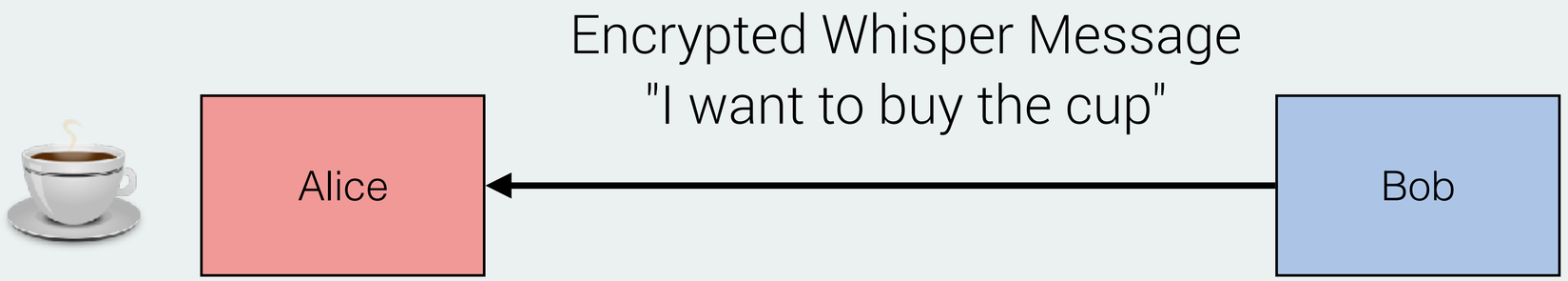
10 ETH



Listens to the relevant messages

ethereum Marketplace

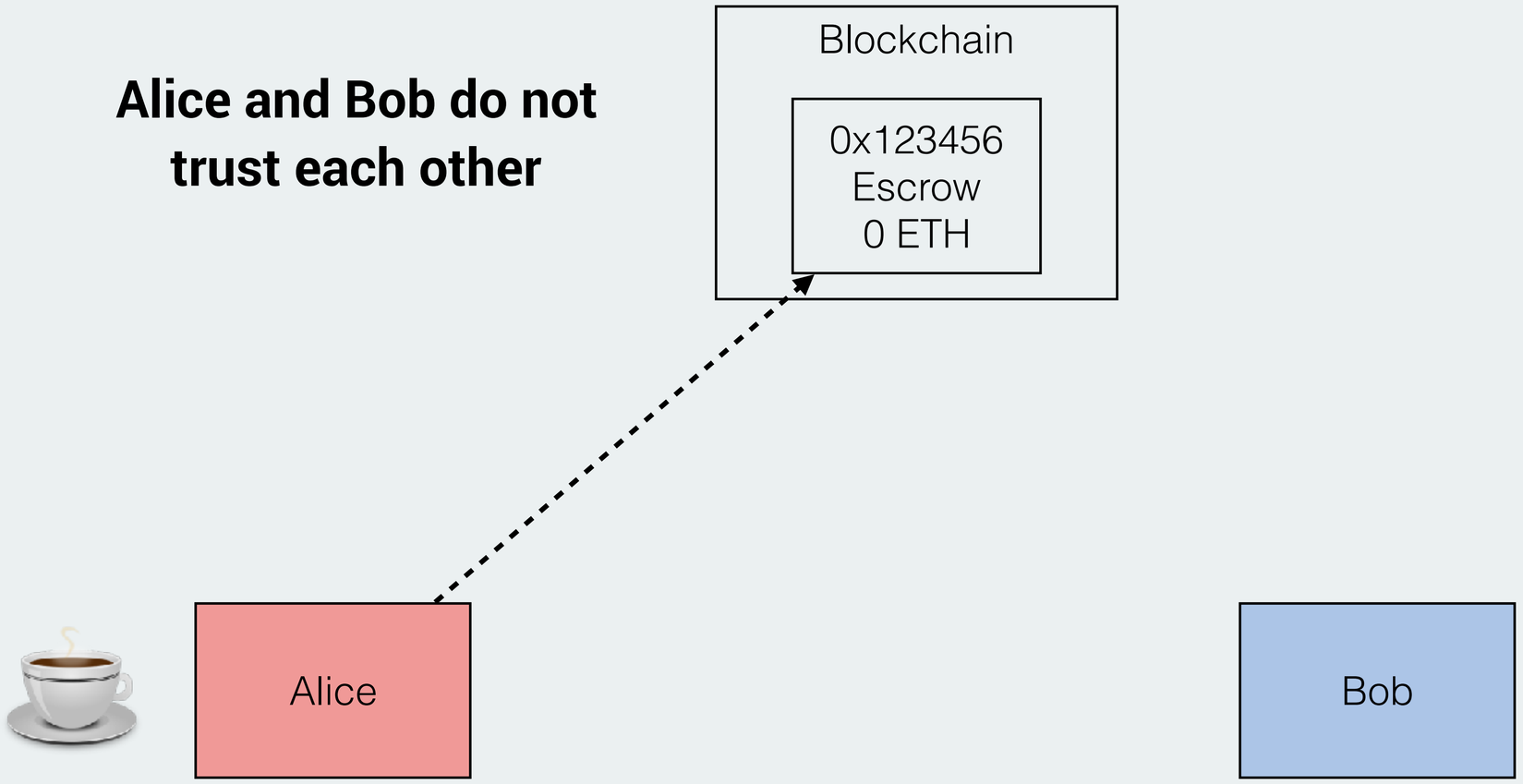
Bob sees Alice's offer and wants to buy



Sends a private message to Alice

ethereum Marketplace

Alice and Bob do not trust each other

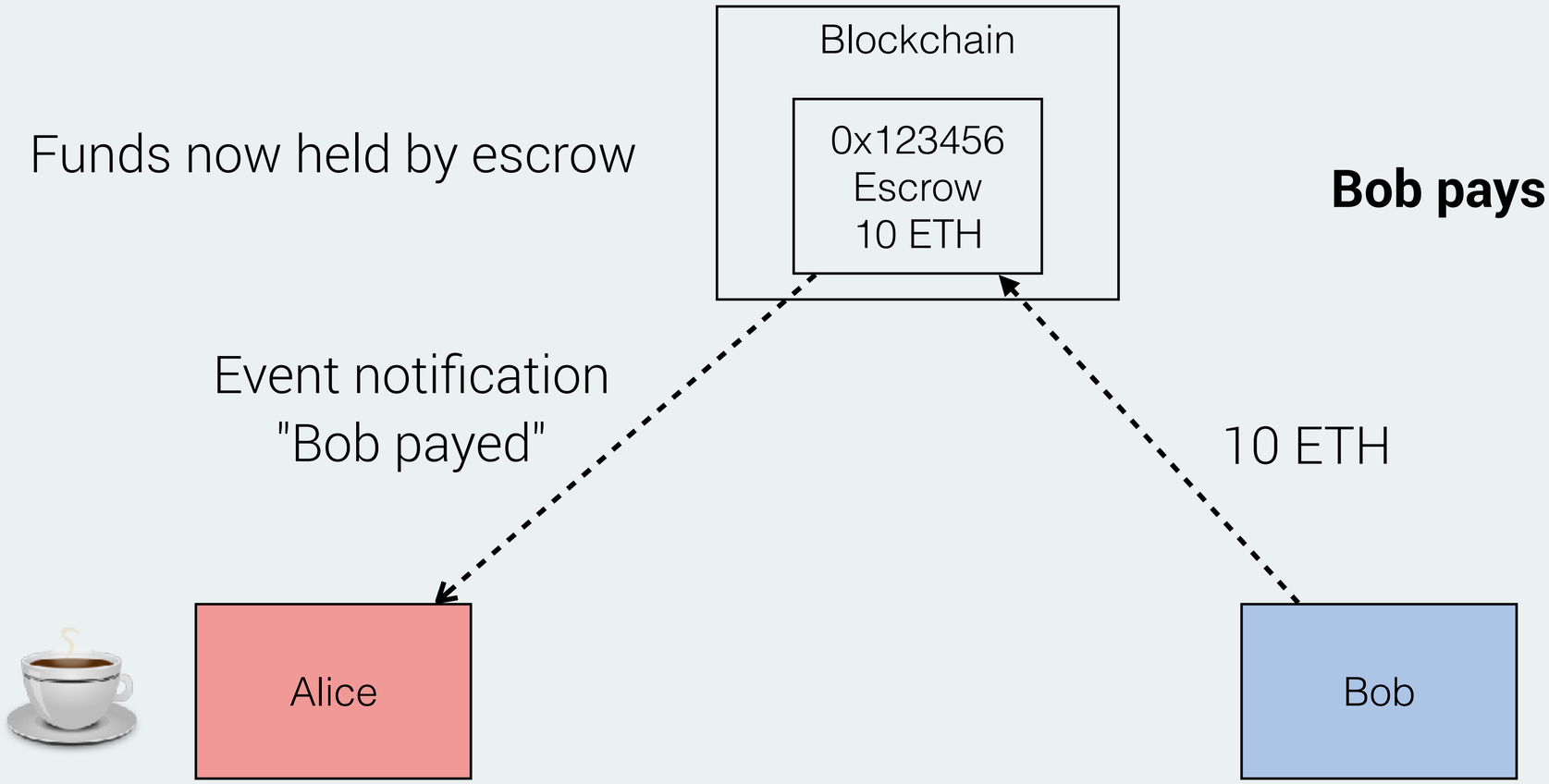


Alice creates an escrow contract

ethereum Marketplace

Alice informs Bob about the escrow

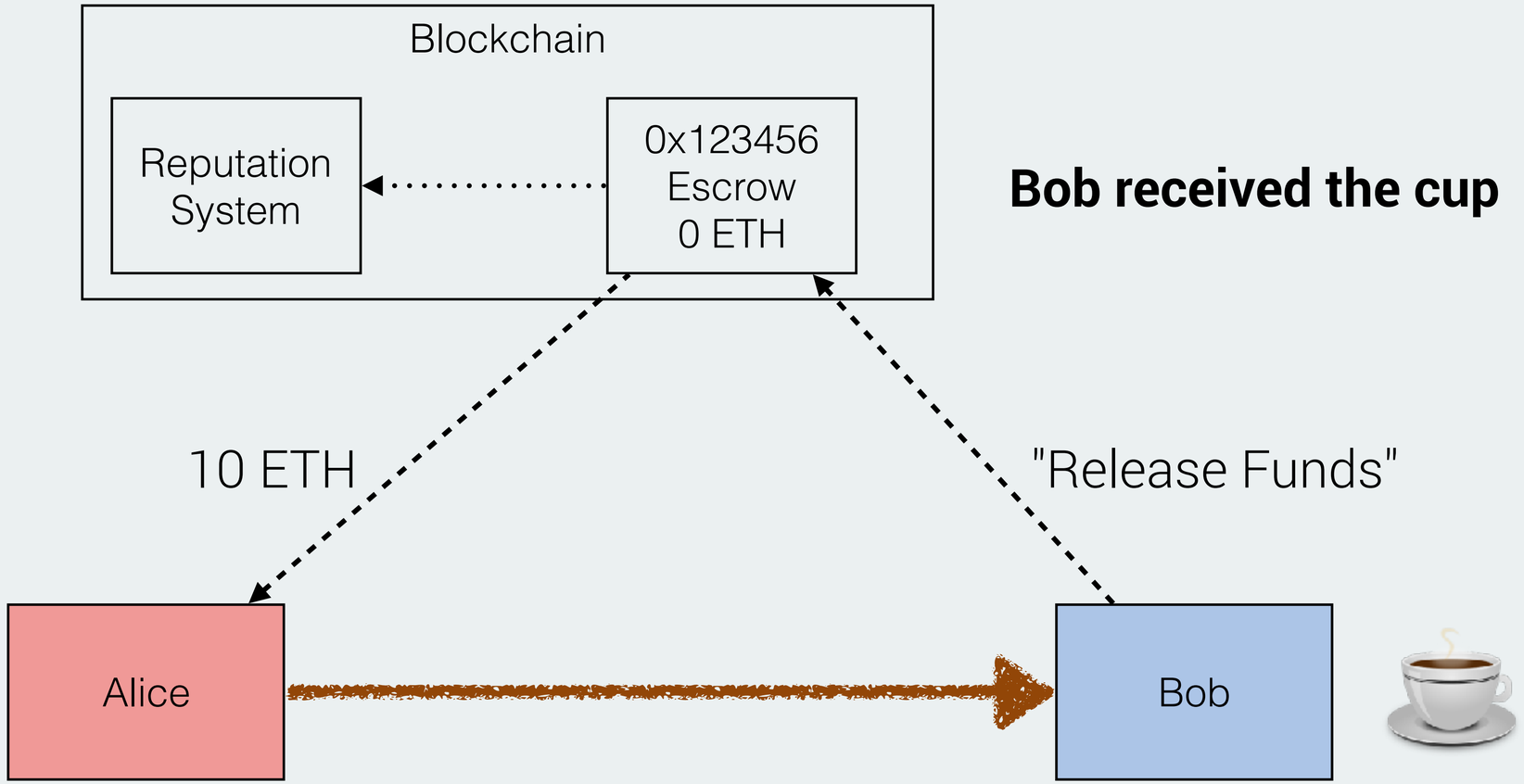
ethereum Marketplace



Alice watches the blockchain

Marketplace

ethereum Marketplace



Bob received the cup

Funding

Funded by crowdfunding

31.529 BTC raised (~18.5m USD at the time)

Over 9000 Transactions

half of that value lost due to bitcoin price decline

(**but** rise in ether price secured funding for 4 years)

recent rise made eth foundation rich (~200m$)

2.0 and beyond

Abstraction

- Contract pays fee
- Other signing mechanisms

Casper

- Proof of Stake with finality
- Prediction market for blocks

Scalability

- Sharding (also offchain solutions like Raiden)

ethereum Release Process