

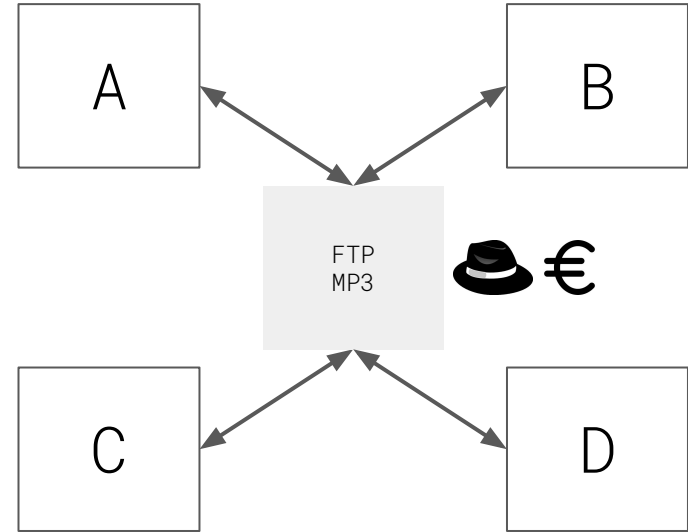
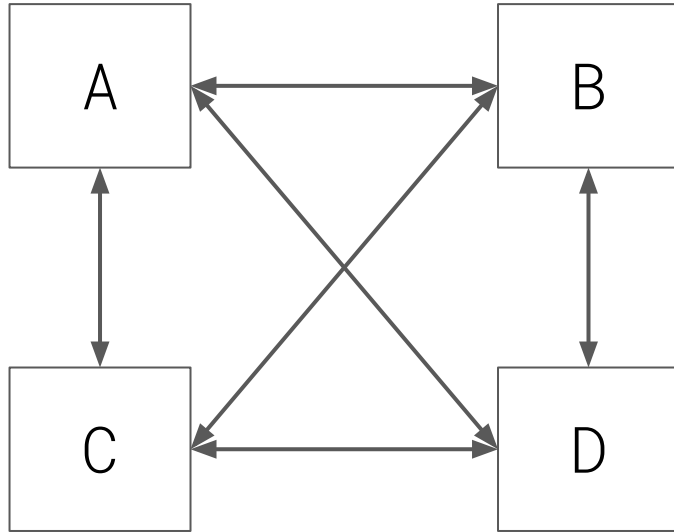
# ERC20 // VAULT

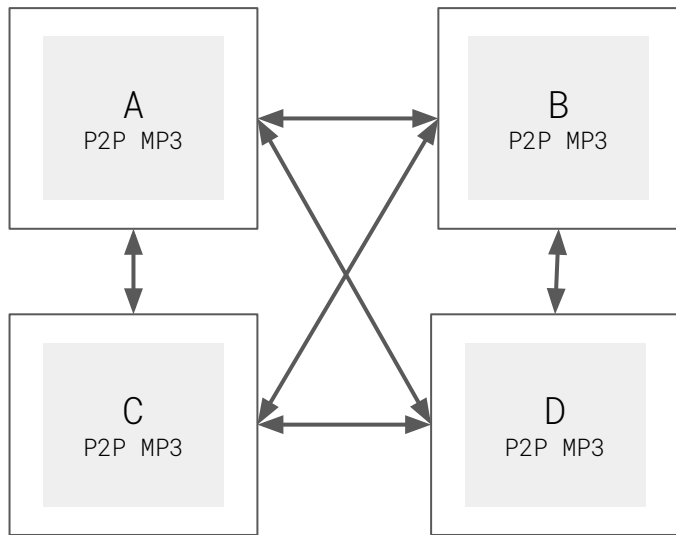
`adria@codecontext.io`

- I. Ethereum VM
- II. ERC20, Tokens & ICOs
- III. Vault + controller
- IV. Q&A

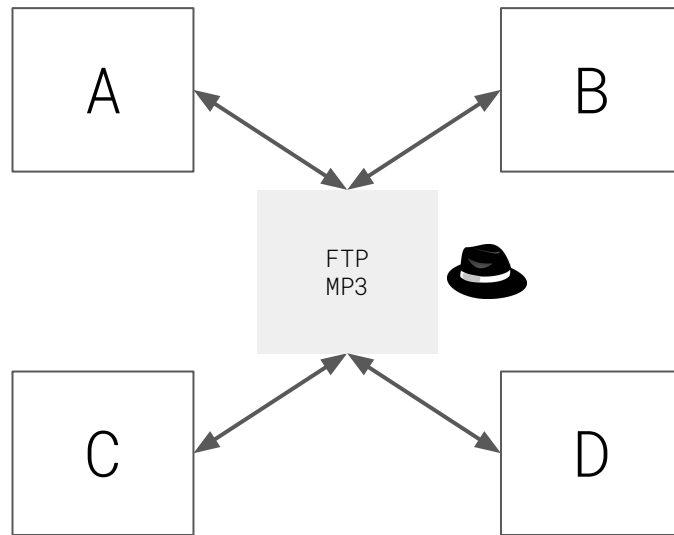
from  
centralized information  
to  
decentralized applications

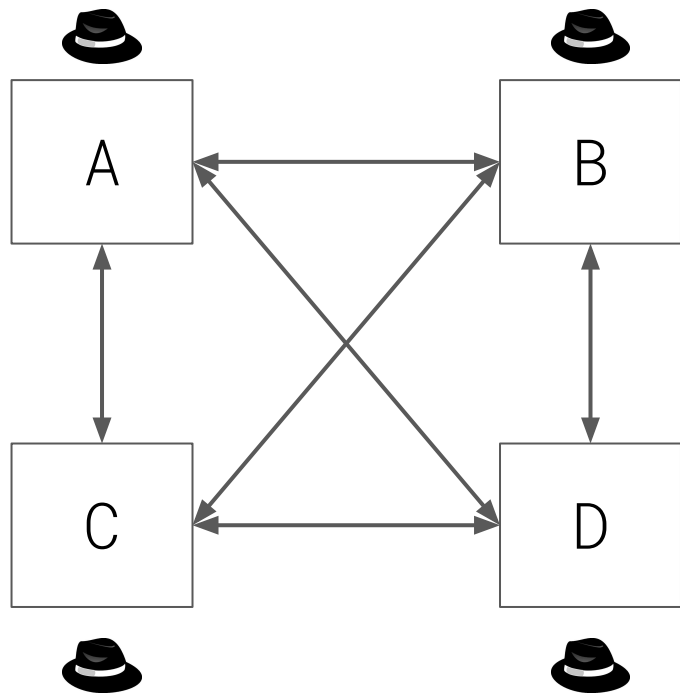
Who owns computer systems?



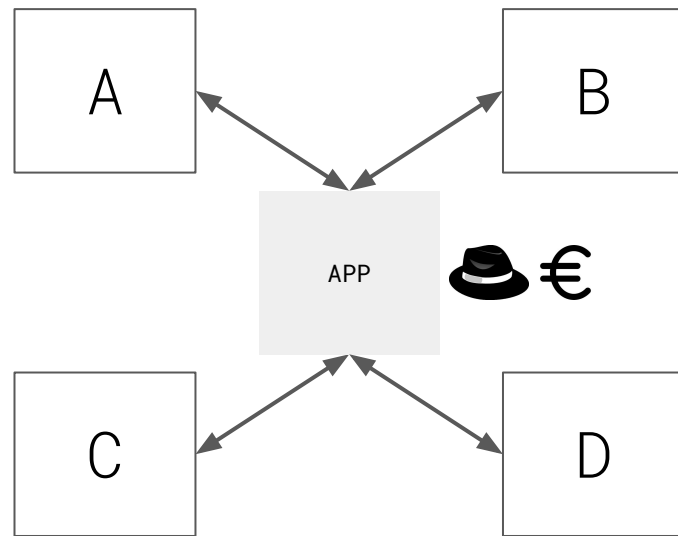


Who owns computer systems?

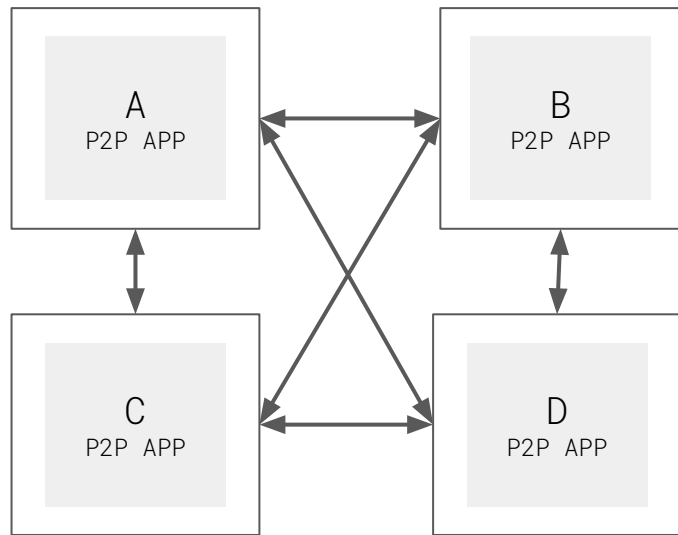




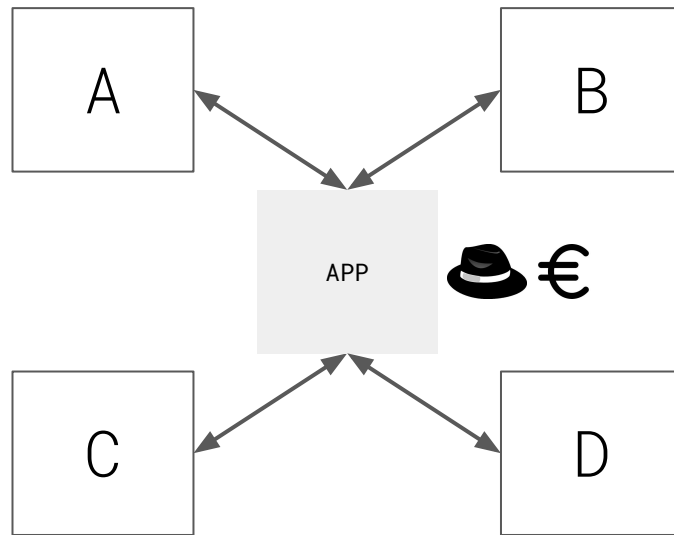
Who owns computer systems?

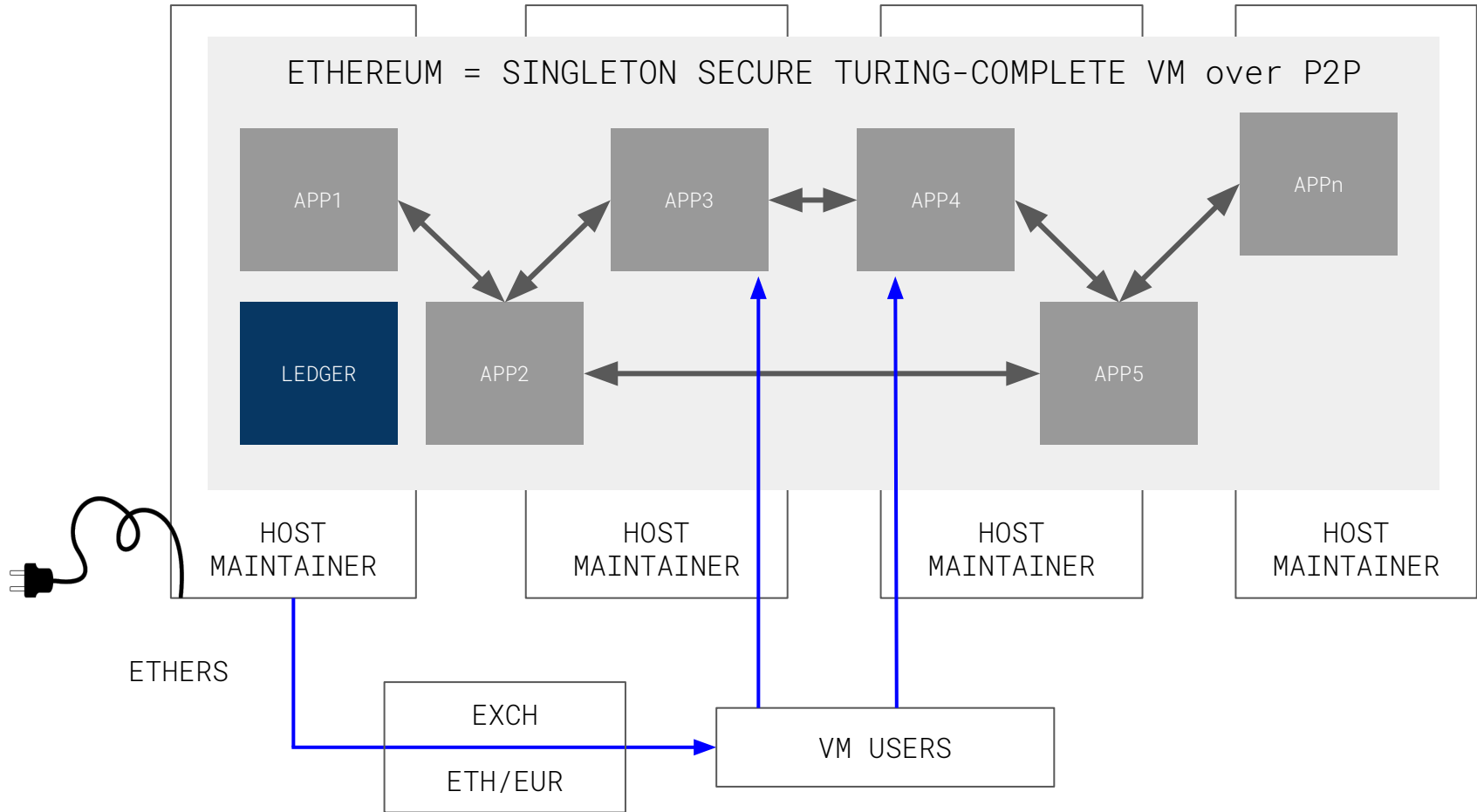


Amazon  
AirBnb, Uber, etc...  
Swift  
State



Who owns computer systems?







ethereum is a technology

<http://gavwood.com/Paper.pdf>

geth / parity / cpp-ethereum / pyethereum

and a global implementation

<https://ethstats.net/> <https://etherscan.io/>

with an market valued

embedded virtual coin (IoI vs IoE)

<https://coinmarketcap.com/currencies/ethereum/>

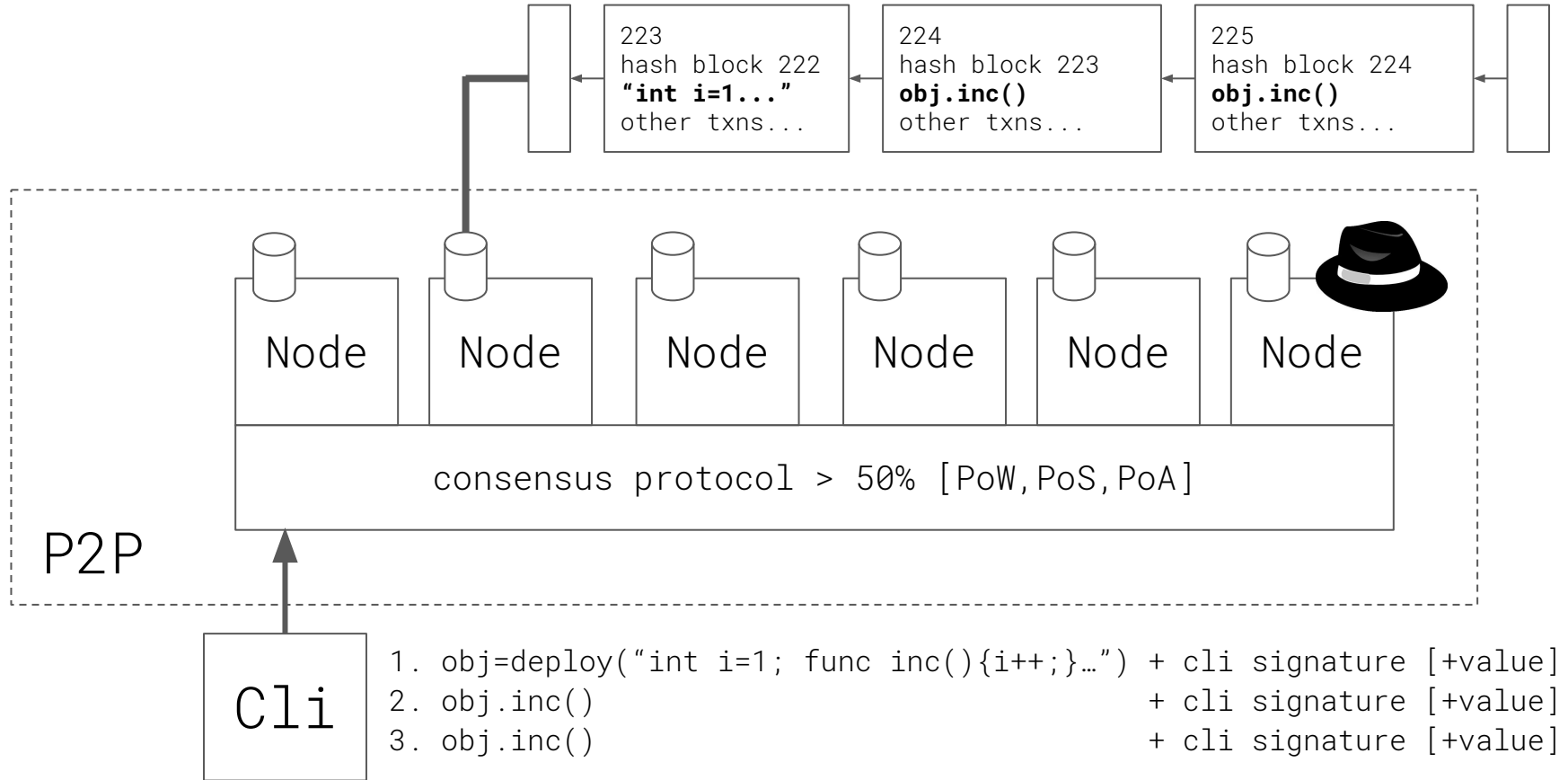
Ethereum is an architecture to execute code using a P2P network.

```
function Counter {  
    uint i=1;  
    function inc() {  
        i++;  
    }  
    function get() returns (uint) {  
        return i;  
    }  
}
```

What is execute code?

- deploy code to compiled be executed
- execute method inc()
- execute method inc()
- execute method inc()

blockchain is a secure, append-only transaction/state database  
where a block is sealed each 14s with accepted transactions by consensus



<https://ropsten.etherscan.io/address/0xa127362d46356632b523cf81fddec1e948d711b0#code>

## Example apps

- DAO
- ENS
- Gnosys
- Aragon
- Chronobank

## Metropolis

- Nicer user experience
- Current way to create ethers is too expansive for earth and centralizes too much the power (PoW), moving to another way (PoS)
- By definition all data is public, introduce a new way to exchange ciphered data between users in a ciphered way (ZKP, zk-snarks)
- RSA signatures

# ERC20

standard interface for tokens

<https://github.com/ethereum/EIPs/issues/20>

```
mapping(address => uint) balances;
```

```
function token(address _faucet, uint _amount) {  
    balances[_faucet] = _amount;  
}
```

```
function transfer(address _from, address _to, uint _value){  
    balances[_from] -= _value;  
    balances[_to]    += _value;  
}
```

```
function balanceOf(address _owner) returns (uint) {  
    return balances[_owner];  
}
```

```
function totalSupply() constant returns (uint256 totalSupply)
```

```
function balanceOf(address _owner) constant returns (uint256 balance)
```

```
function transfer(address _to, uint256 _value) returns (bool success)
```

```
function transferFrom(address _from, address _to, uint256 _value) returns (bool success)
```

```
function approve(address _spender, uint256 _value) returns (bool success)
```

```
function allowance(address _owner, address _spender) constant returns (uint256 remaining)
```

```
Contract A {
```

```
    function transferAndAuthorize() {
```

```
        ERC20 token = 0xdEe06DaC1106C6c6f043783EBD8Fbe6dAd303522
```

```
        token.transfer(B.address, token.balanceOf(A.address)-10)
```

```
        token.approve(B.address,10)
```

```
    }
```

```
}
```

```
Contract B {
```

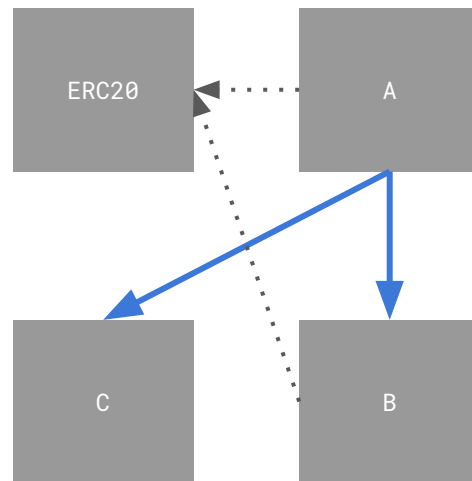
```
    function sendback() {
```

```
        ERC20 token = 0xdEe06DaC1106C6c6f043783EBD8Fbe6dAd303522
```

```
        token.transferFrom(A.address,C.address, 10)
```

```
    }
```

```
}
```



## Security, security, security

- code is immutable, so bugs => M\$ disaster
- stoppable pattern : if there's any error in the code, activate a flag that "stops" the contract (all functions throws)
- bug bounties
- code review:  
<https://etherscan.io/address/0xa74476443119A942dE498590Fe1f2454d7D4aC0d#code>



crowdfund your company with those tokens to use the platform later\*

<https://etherscan.io/tokens>

Aragon ICO raised 25M\$ in 25m

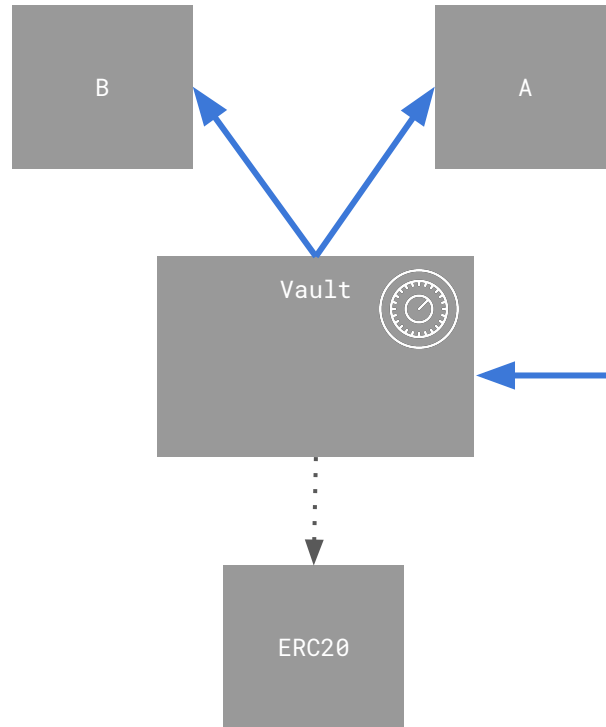
# vault

to hold tokens safely and automate payments  
to a pre-approved white list of recipients

<https://github.com/Giveth/vaultcontract>

@jbaylina



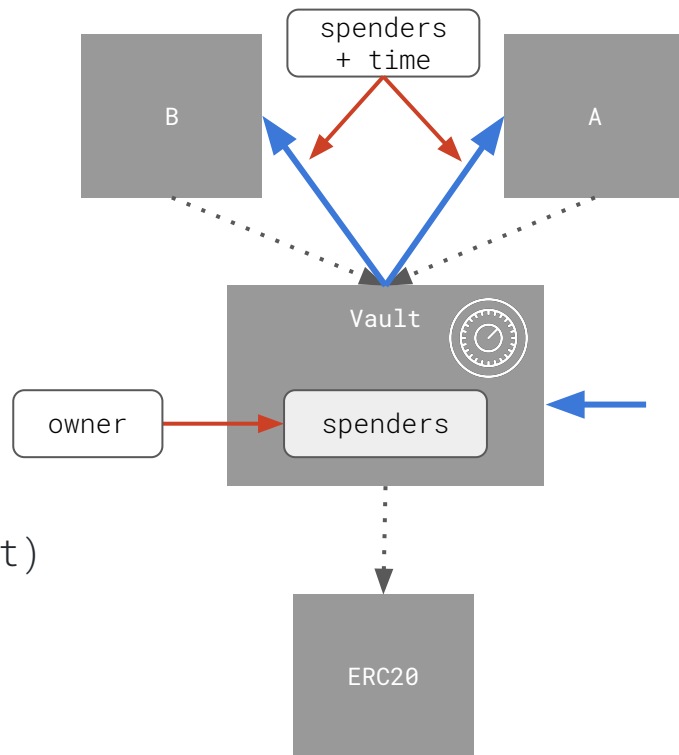


```
function Vault(address _baseToken,...)
```

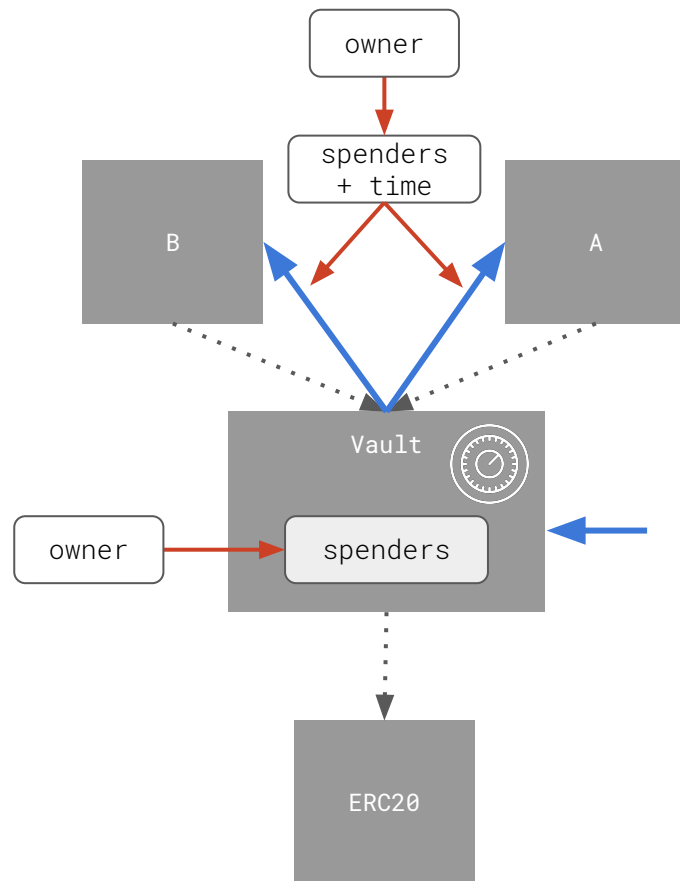
```
function authorizeSpender(address _spender, ...)
```

```
function authorizePayment (  
    ...,  
    address _recipient, uint _amount,  
    uint _paymentDelay  
) returns (uint paymentId)
```

```
function collectAuthorizedPayment(uint _idPayment)
```



```
function cancelPayment(uint _idPayment)
```



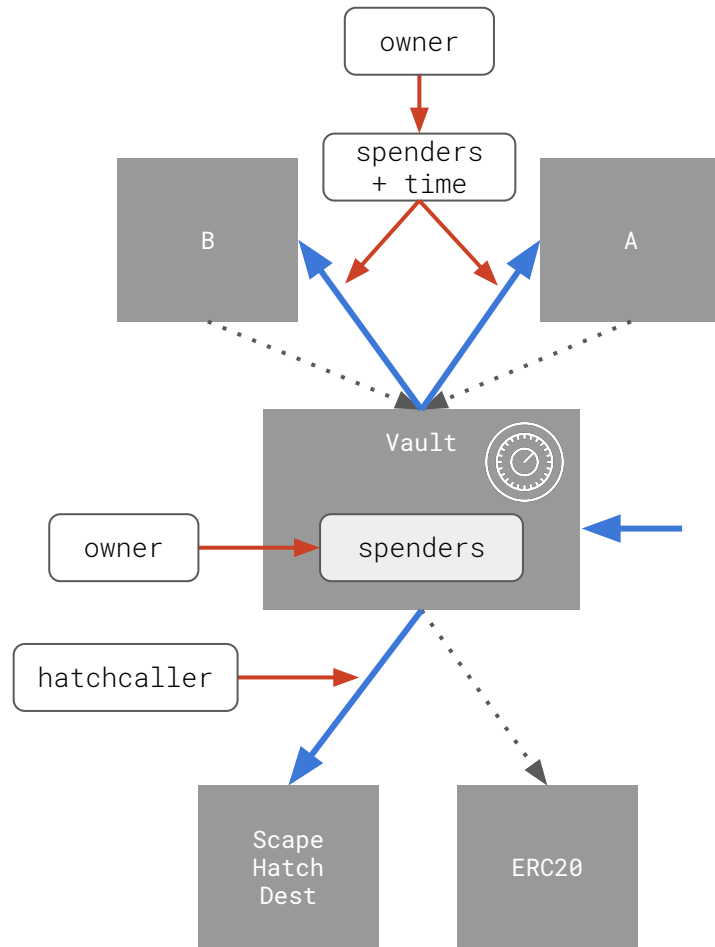
```

function Vault(
    ...
    address scapeHatchCaller,
    address scapeHatchDestination
)

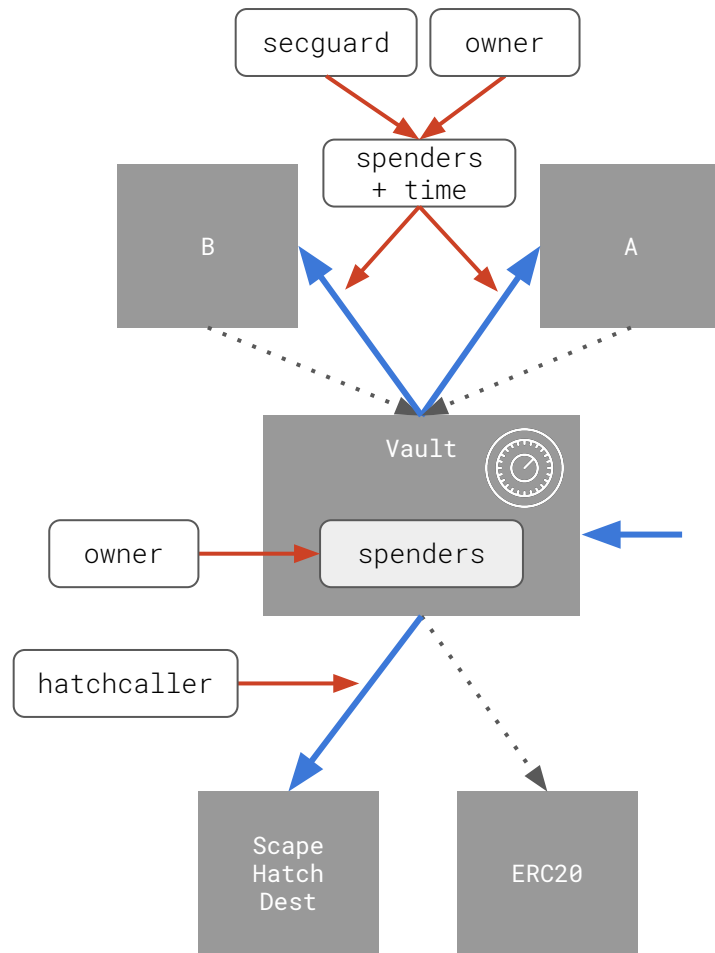
function escapeHatch()

function changeEscapeHatchCaller(
    address _newEscapeHatchCaller
)

```



```
function Vault(...  
    address _securityGuard,  
)  
  
function delayPayment(  
    uint _idPayment,  
    uint _delay  
)
```



# vaultcontroller (WIP)

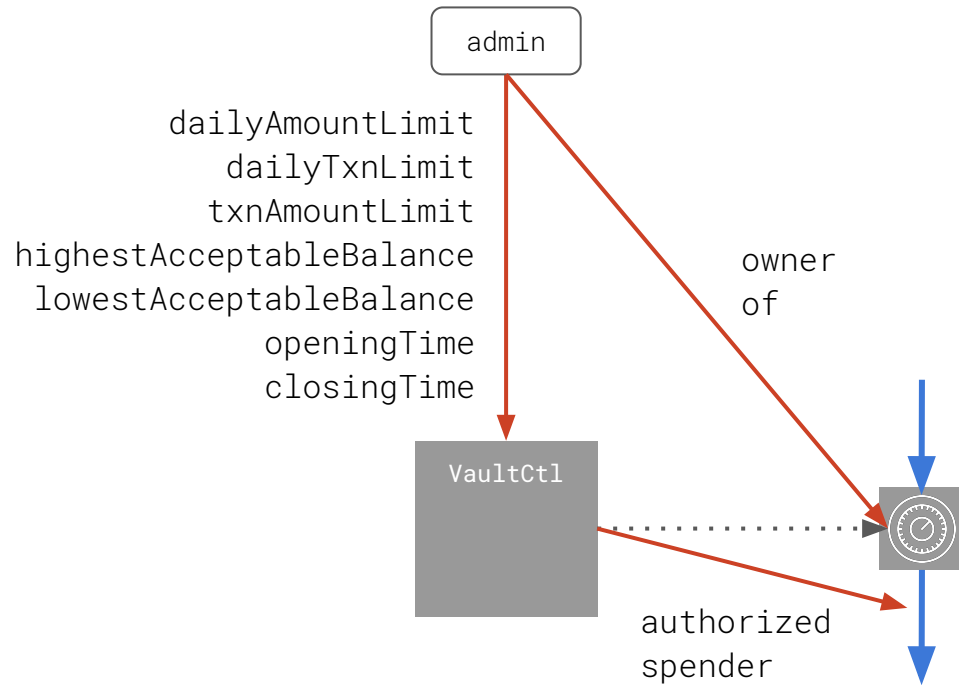
cascade-pouring vaults with limits

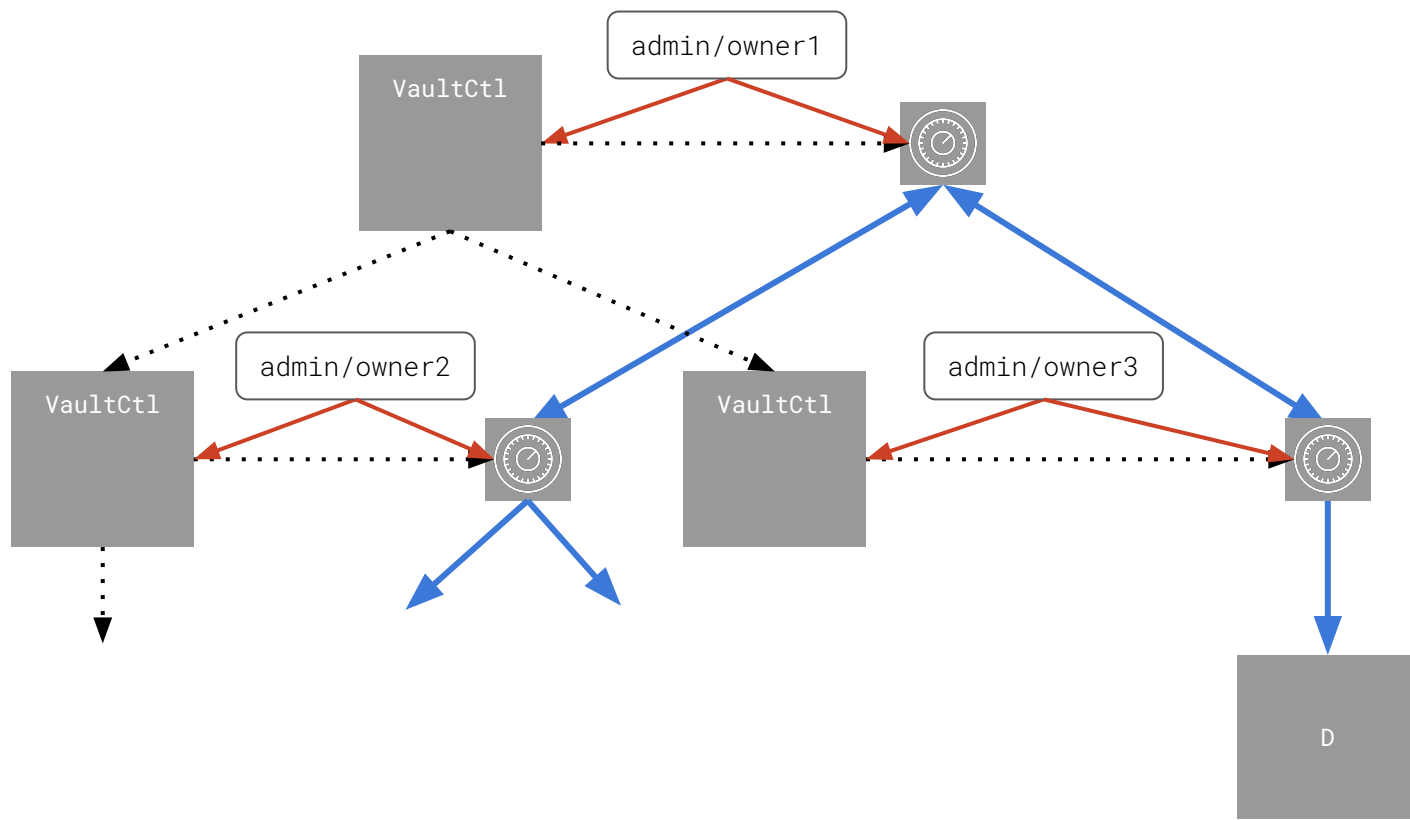
<https://github.com/Giveth/vaultcontroller>

@jbaylina











ETHEREUM DEV  
BARCELONA  
*Meetup*

<https://www.meetup.com/ethereumbcn/>