ETHEREUM DEVELOPERS BARCELONA MEETUP

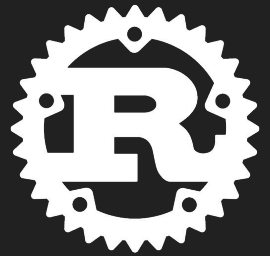
6th edition µWorkshop RUSTFEST EDITION

STARTS AT 18:10

next meetup Wed 20th

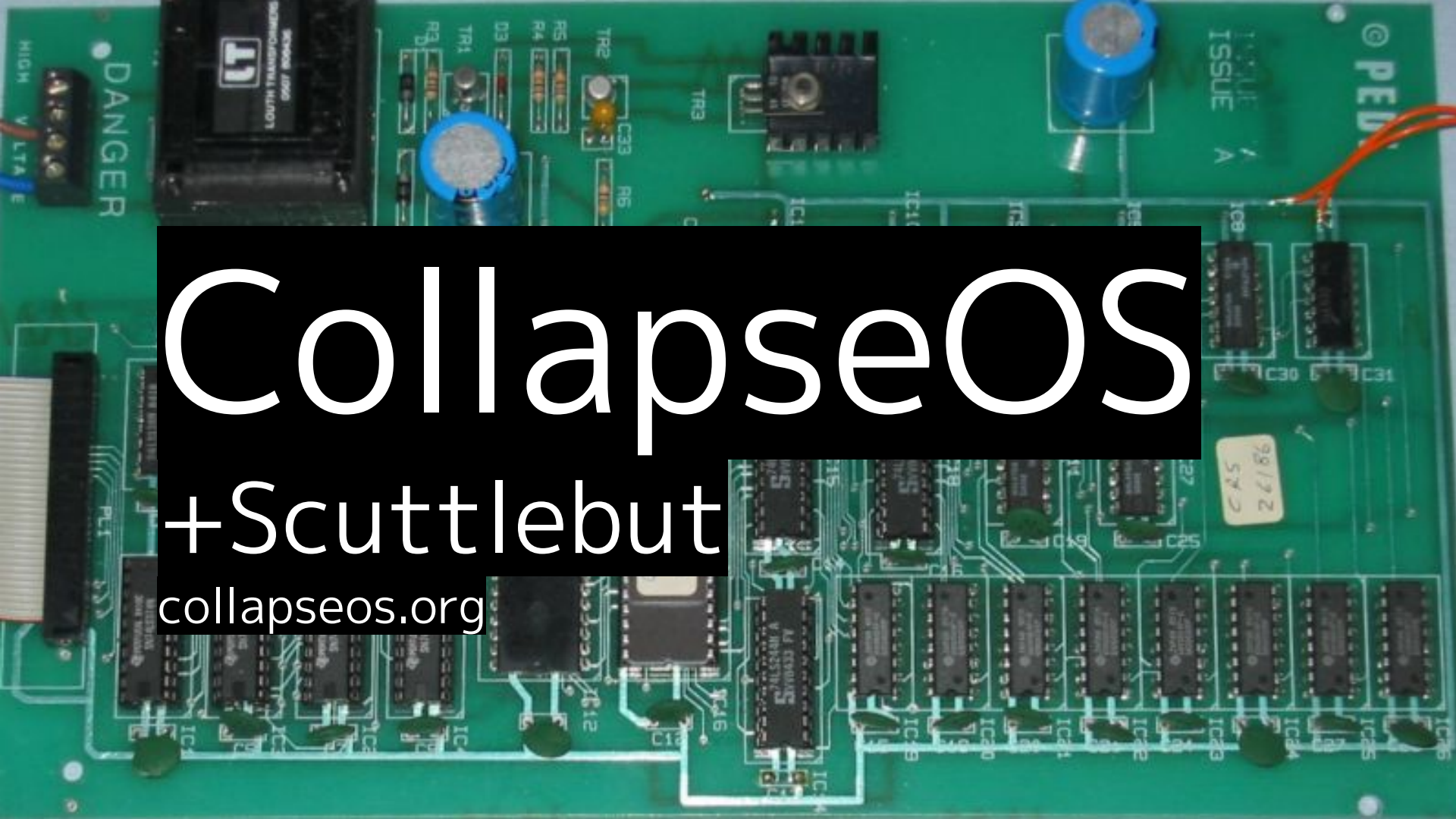
gnosys

underscore protocol



2047


www.electricBricks.com


CollapseOS
+Scuttlebut
collapseos.org

we need
trusted consensus engine

identity
economy
democracy
ownership

THEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LED

BYZANTIUM VERSION 7e819ec - 2019-10-20

DR. GAVIN WOOD
FOUNDER, ETHEREUM & PARITY
GAVIN@PARITY.IO

Abstract. The blockchain paradigm when coupled with cryptographically-secured transactions has demonstrated i utility through a number of projects, with Bitcoin being one of the most notable ones. Each such project can be seen a a simple application on a decentralised, but singleton, compute resource. We can call this paradigm a transactions singleton machine with shared-state.

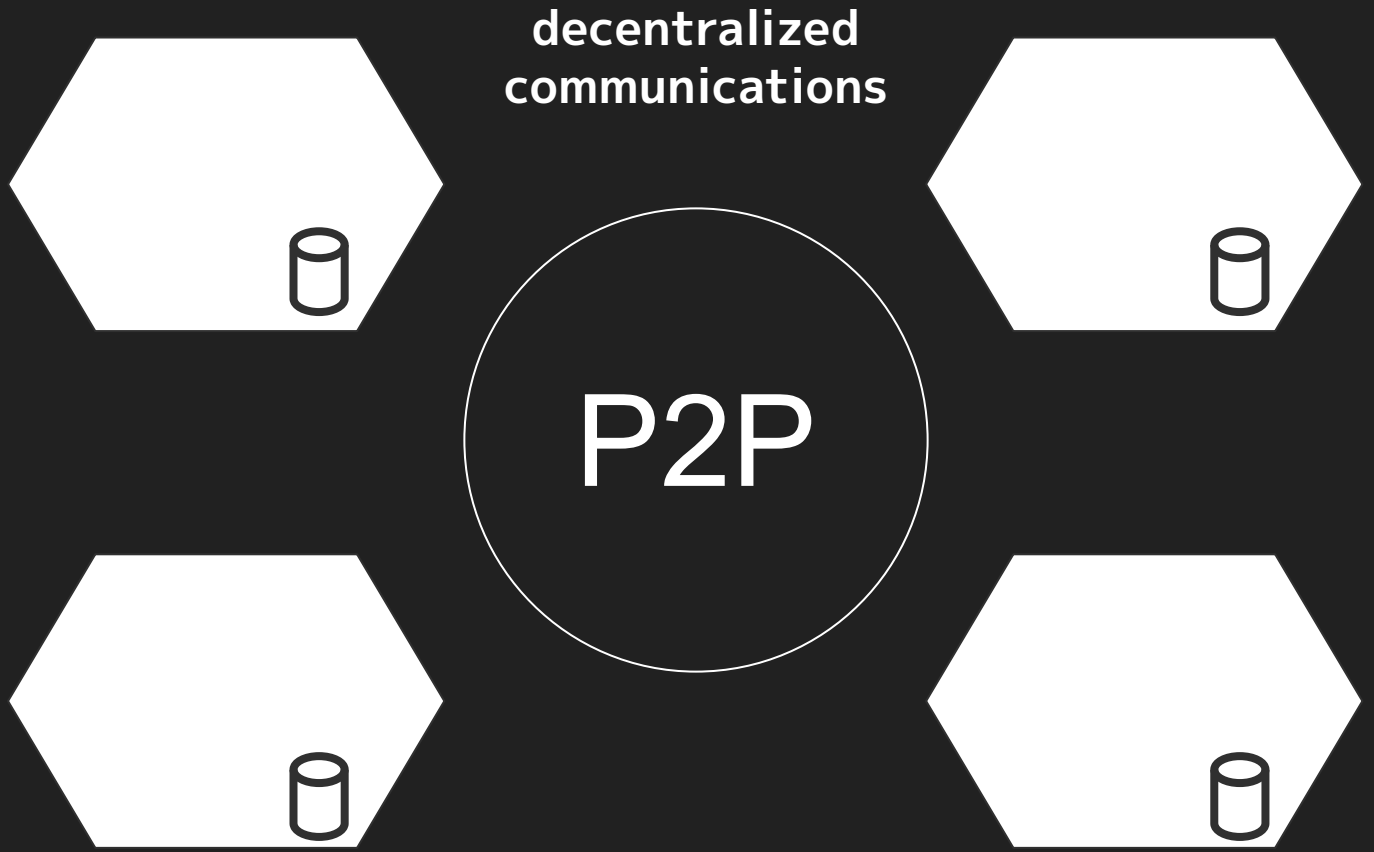
Ethereum implements this paradigm in a generalised manner. Furthermore it provides a plurality of such resour each with a distinct state and operating code but able to interact through a message-passing framework with othe We discuss its design, implementation issues, the opportunities it provides and the future hurdles we envisage.

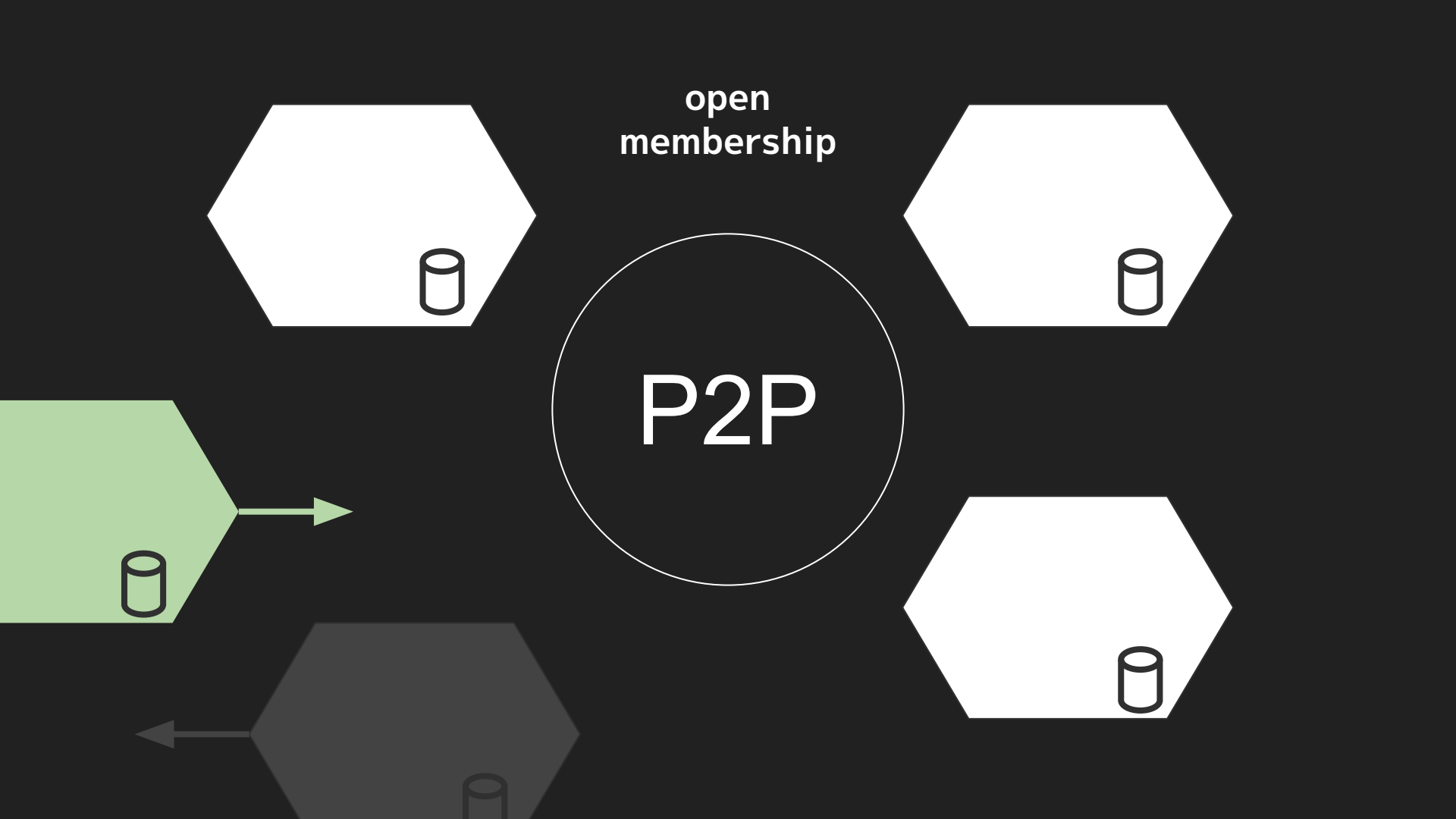
1. Introduction

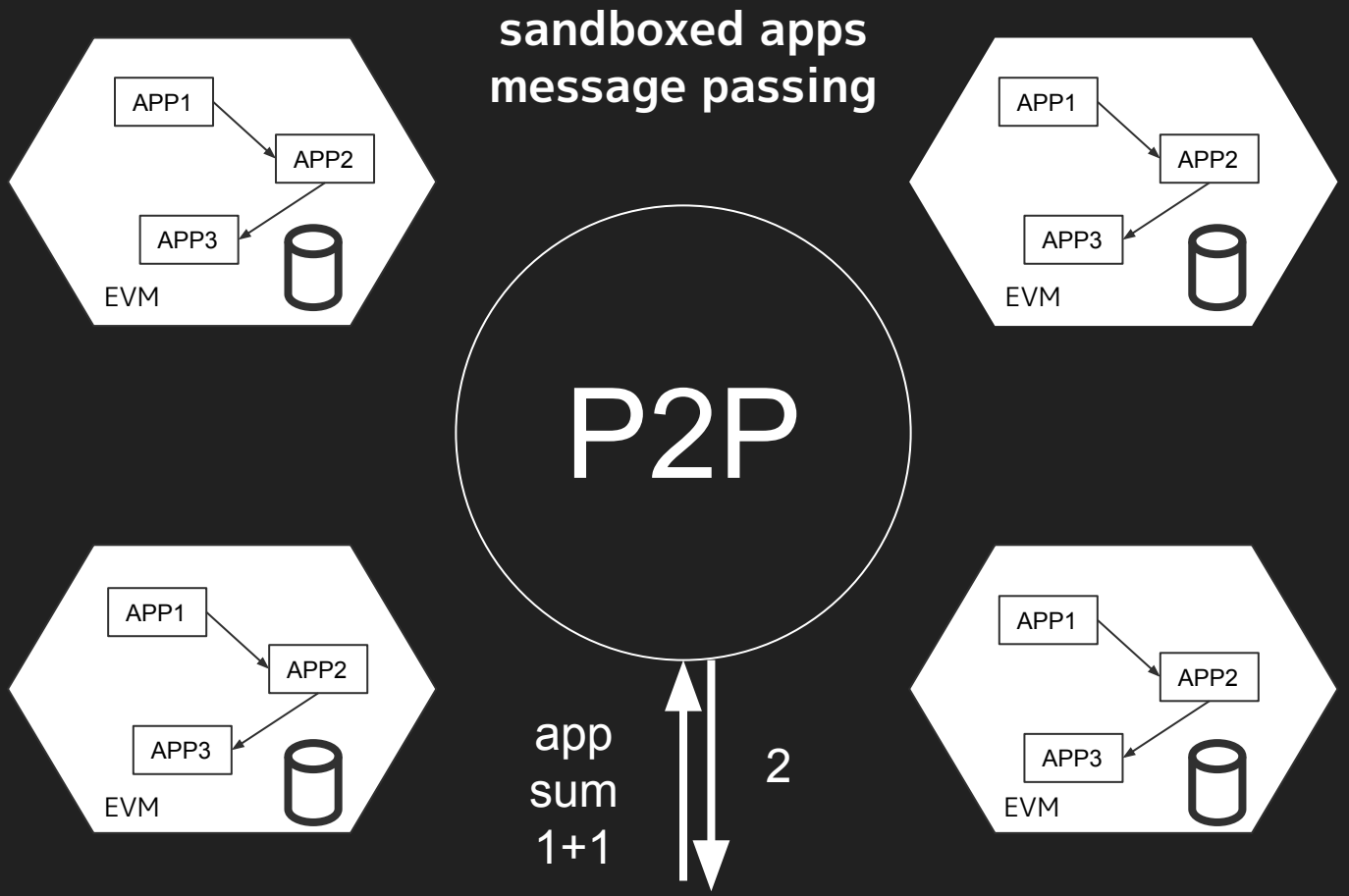
biquitous internet connections in most places global information transmission has become n. Technology-rooted movements like Bit-

is often lacking, and plain old prejudices are shake.

Overall, we wish to provide a system such can be guaranteed that no matter with wh


decentralized
communications
P2P

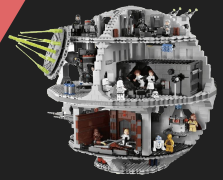

open
membership
P2P


sandboxed apps
message passing
APP1
APP2
APP3
EVM
APP1
APP2
APP3
EVM
P2P
APP1
APP2
APP3
EVM
APP1
APP2
APP3
EVM
app
sum
1+1
2

secure

app sum
1+1=2

app sum
1+1=5



P2P

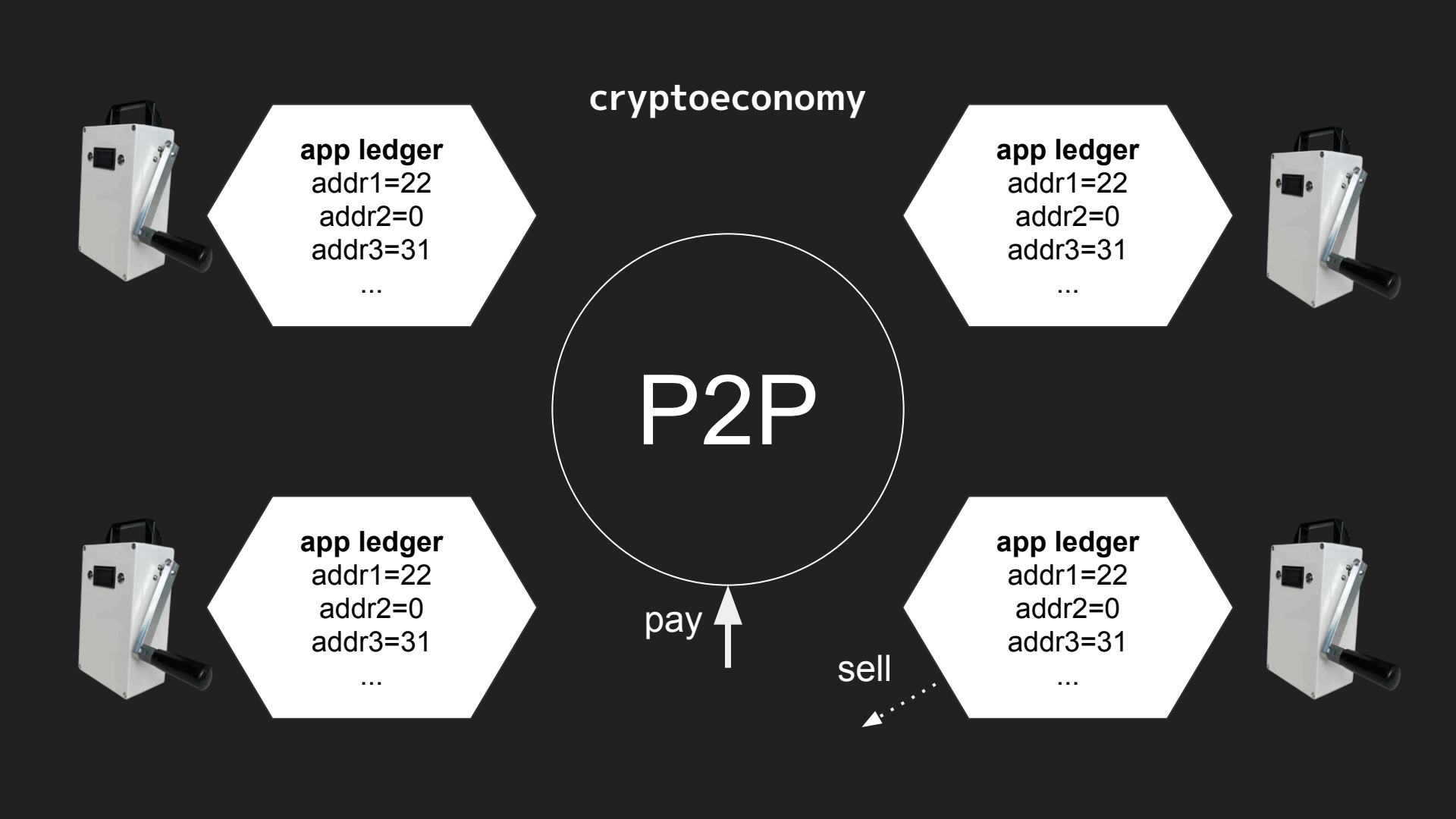
app sum
1+1=2

app sum
1+1=2

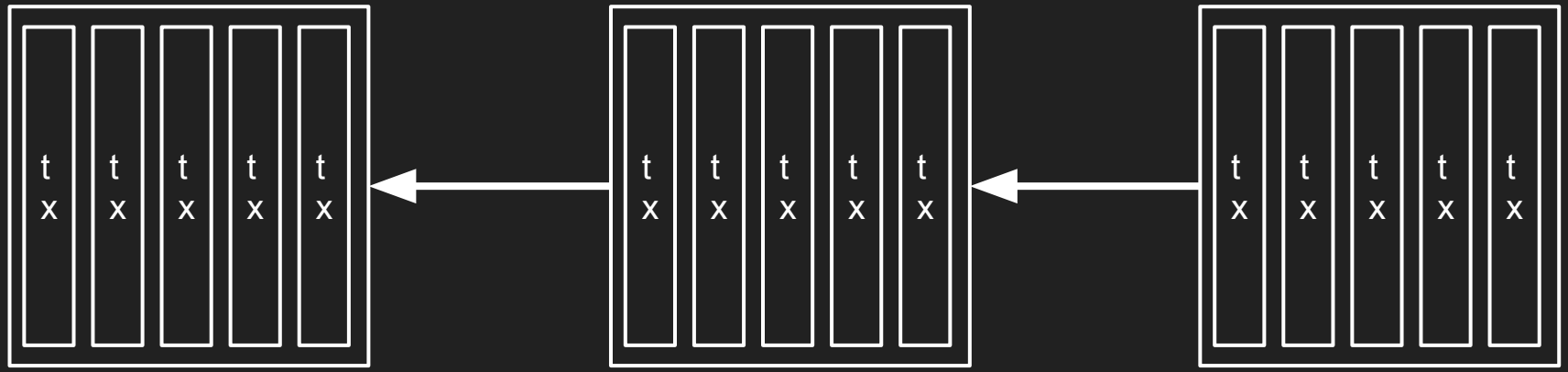
app
sum
1+1

2

digital signature


cryptoeconomy
app ledger
addr1=22
addr2=0
addr3=31
...
app ledger
addr1=22
addr2=0
addr3=31
...
P2P
app ledger
addr1=22
addr2=0
addr3=31
...
app ledger
addr1=22
addr2=0
addr3=31
...
pay
sell

how data is stored

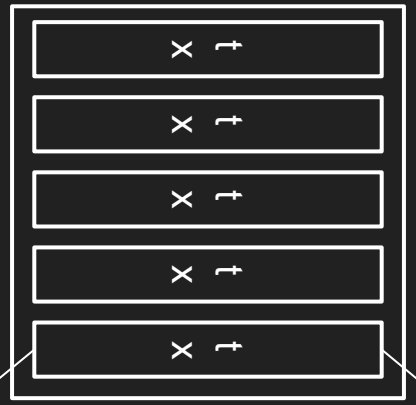


tx

transfer eth

install app

call app



from
to
value
data
gas / gasprice
signature of above

FROM	TO	VALUE	DATA
0x5A0b54D5dc17e0AadC383d2db43B0a0D3E029c4c			bytecode
0x5A0b54D5dc17e0AadC383d2db43B0a0D3E029c4c	0x8fD00f170FDf3772C5ebdCD90bF257316c69BA45		0x18271627 + P
0x5A0b54D5dc17e0AadC383d2db43B0a0D3E029c4c	0x7400c18e24cf5fA59FA9Fe7A91dfc2d4Bd87fDB7	1000	
0x5A0b54D5dc17e0AadC383d2db43B0a0D3E029c4c	0x8fD00f170FDf3772C5ebdCD90bF257316c69BA45	1000	0x18271627 + P

new era bootstrap apps

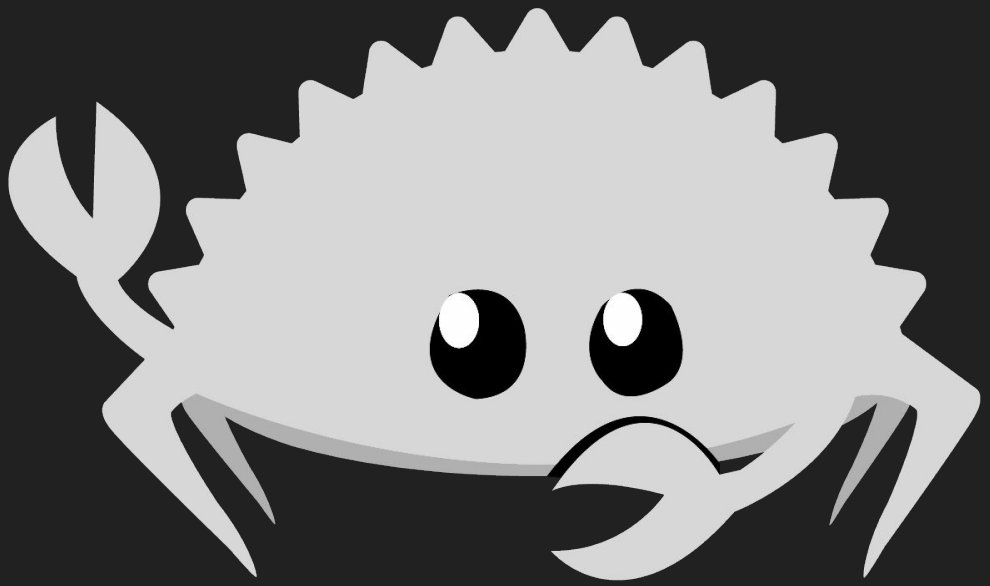
```
contract Yeah {
  uint public yeahs;
  function yeah() public {
     yeahs++;
  }
}
```

```
contract Economy {
  mapping(address=>uint) public balance;
  constructor() public {
    balance[msg.sender] = 1000000;
  }
  function transfer(address to, uint amount) public {
     require(balance[msg.sender]>=amount);
     balance[msg.sender]-=amount;
     balance[to]+=amount;
  }
}
```

```
contract Property {
  mapping(bytes32=>address) public owner;
  function transfer(bytes32 assetid, address newowner) public {
     if (msg.sender==newowner && owner[assetid]==address(0)) {
        owner[assetid]=newowner;
     } else {
        require(owner[assetid]==msg.sender);
        owner[assetid] = newowner;
     }
  }
}
```

```
contract Identity {
  struct identity {
    string name;
    mapping (address=>bool) approvals;
  }
  mapping(address=>identity) public ids;
  function claim(string calldata name) external {
    require(bytes(ids[msg.sender].name).length==0);
    ids[msg.sender].name=name;
  }
  function trust(address who) external {
    ids[who].approvals[msg.sender]=true;
  }
}
```

```
contract Democracy {
  struct poll {
    mapping(uint=>uint) result;
    mapping(address=>bool) voted;
  }
  mapping(uint=>poll) polls;
  function vote(uint pollno, uint option) public {
    require(polls[pollno].voted[msg.sender]==false);
    polls[pollno].voted[msg.sender]=true;
    polls[pollno].result[option]+=1;
  }
  function votes(uint pollno, uint option) public view returns (uint) {
    return polls[pollno].result[option];
  }
}
```



https://github.com/adria0/ethworkshop-rustfest

KATA1 - query the blockchain

KATA2 - deploy minimum smart contract and interact with it

KATA3 - deploy economy smartcontract interact together

KATA4.1 - write your own smartcontract

KATA4.2 - check internal library rust source code