Smart Contract & Programming Solidity

Wanseob Lim

Applied zkp researcher @ Ethereum Foundation

Schedule

- Mar 29th (Wed) 4pm - 6:50 pm
  - Hands-on Programming of Ethereum
- May 1st (Mon) 7pm -
  - Applied Cryptography for Ethereum (1) - On-chain Treasure Hunt
- May 8th (Mon) 7pm -
  - Applied Cryptography for Ethereum (2)

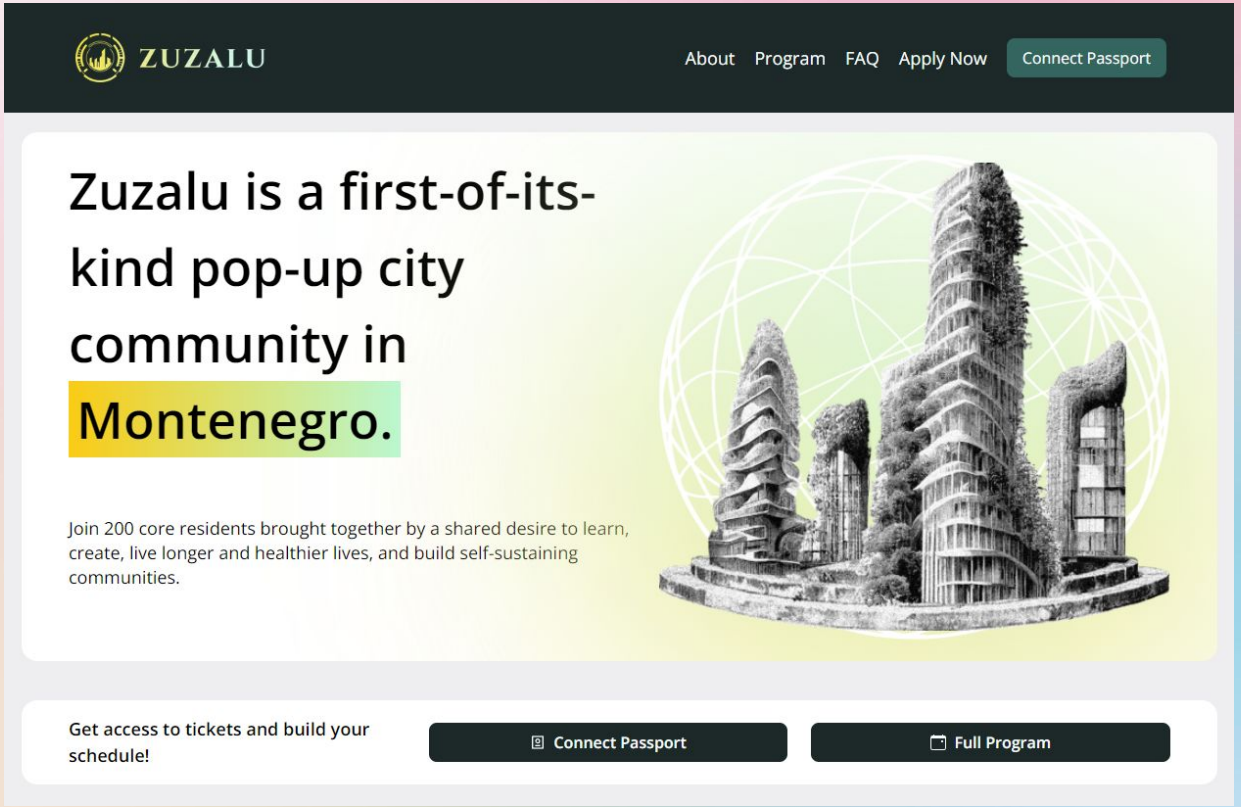
Wanseob

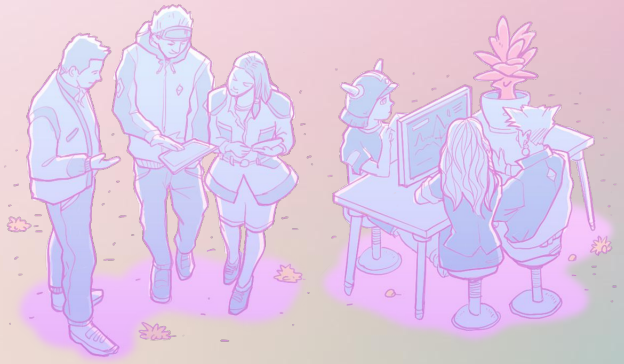
- ZK Researcher @ PSE team of the EF
- Steward of General EcoDev @ EF
- 09’ 선배 KAIST ME

What we're going to do today is A Hands-on programming

There’s one thing that you should keep in mind

Ethereum is not only a technology





Who are the Ethereum community people?

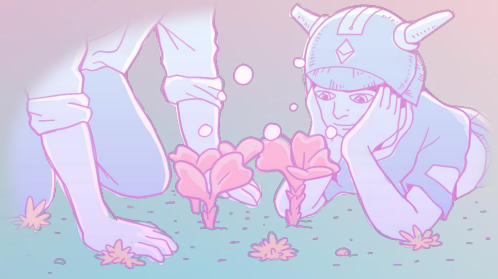
2012, my 1st start-up experience

- Founded a startup
- It’s get acquired by another startup in 2014
- I swapped all my equities to the new one

2014-2017, learned about capitalism

- Why others do not work very hard in this team?
- Incentive problems
  - Founder's share: 50%
  - 10th member's share: 0.5%
- Sweet-talks
- Founder is not greedy, because should take all responsibility.
- It also works well for rocket-speed companies

Opensource



Opensource

- Source code is one of the most powerful **means of production**.

Freedom

Run the software

Study the software

Modify the software

Share the software

by our own **freedom**

Value-driven Community

Ethereum
the world of freedom for Ethereans

Public Goods & Expeiments for the freedom

Which freedom..?

Universal Declaration of Human Rights by UN, Article 12.

No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

~~Trustworthy Party~~

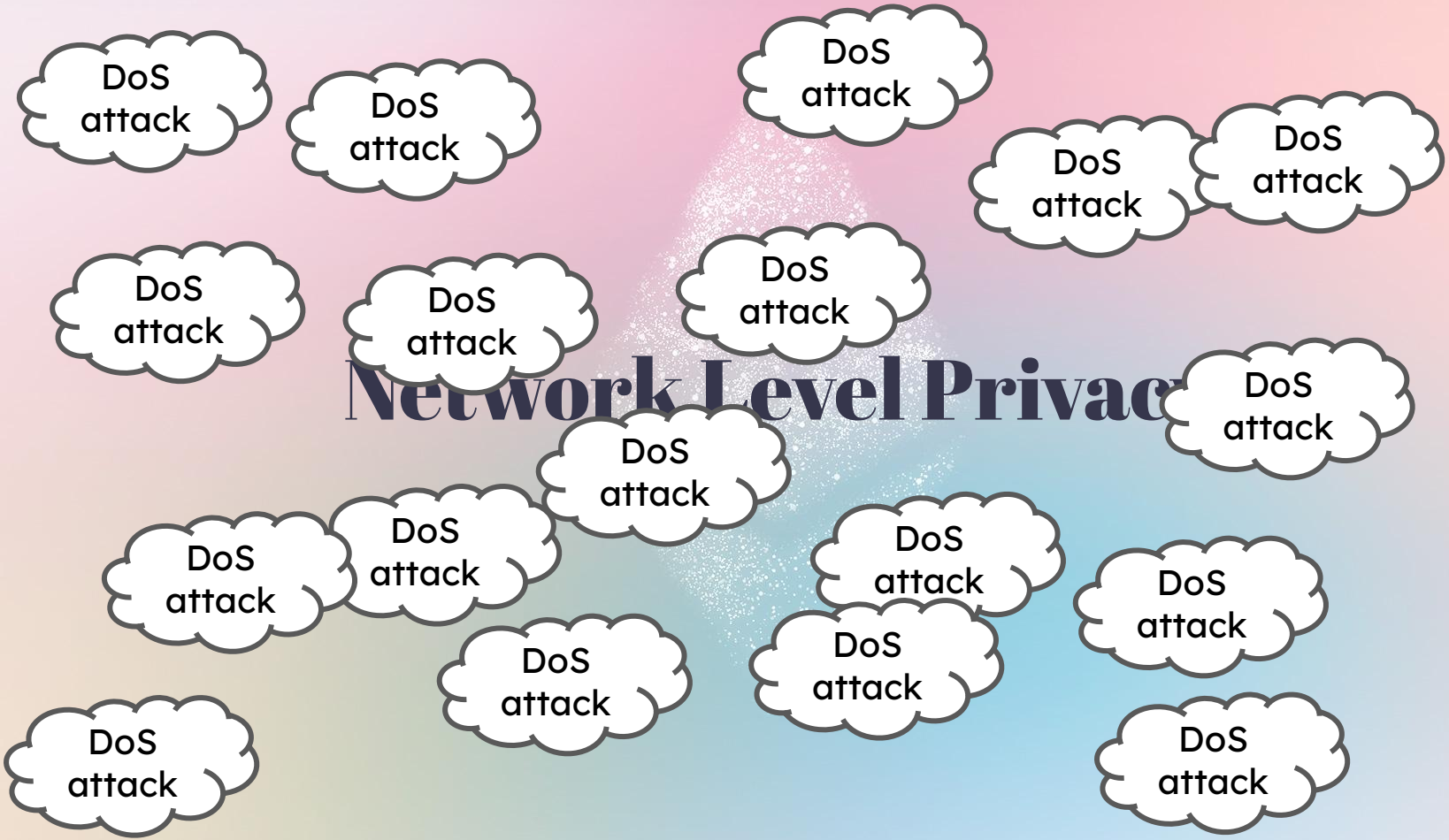
Trustless Privacy

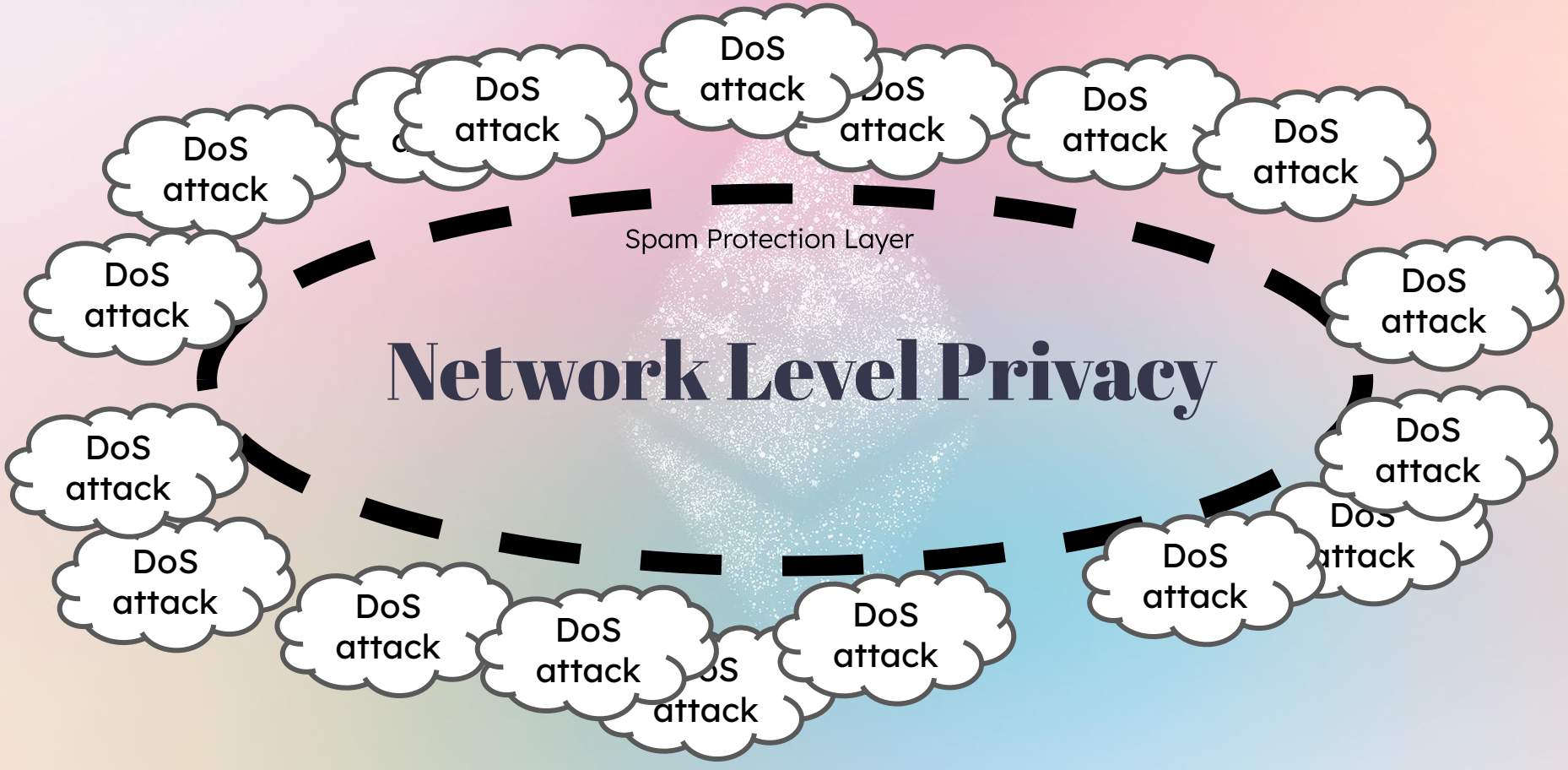
End-to-end Encryption is not enough

Privacy with the Knowledge of Arguments

Example: Ballot

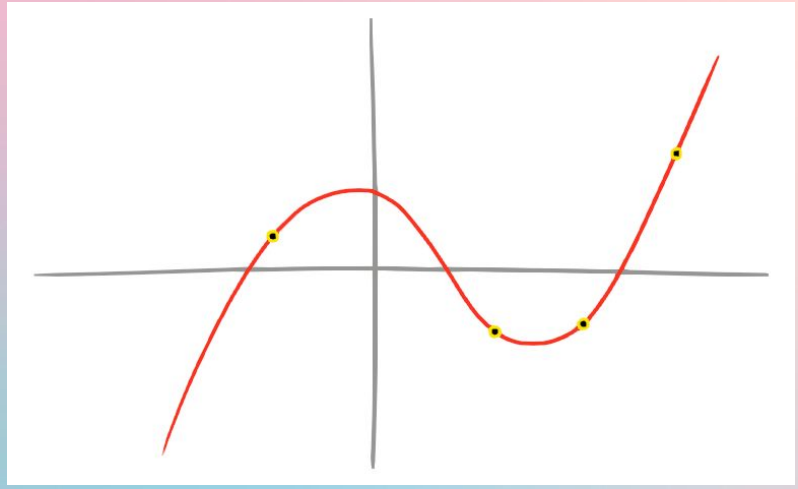
Network Level Privacy


DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
Network Level Privac
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack


DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
DoS attack
Spam Protection Layer
Network Level Privacy

Rate Limiting Nullifier

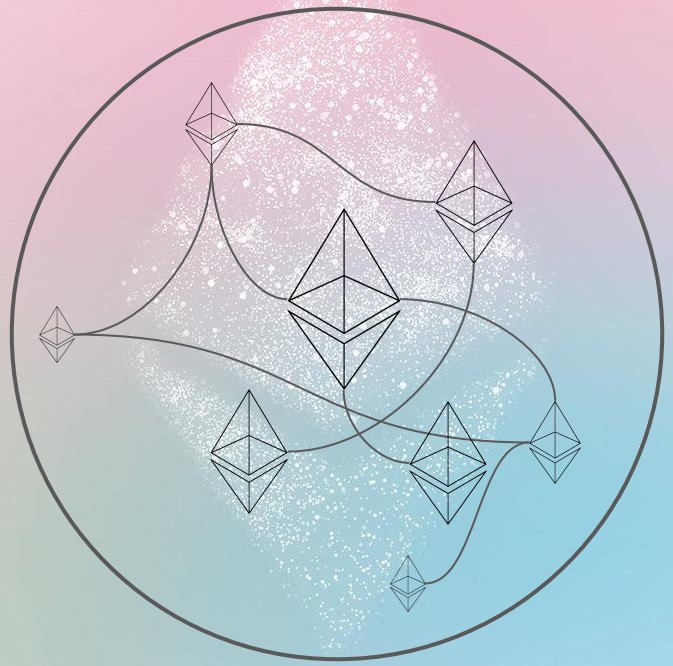
- Shamir Secret Sharing
- Polynomial Commitments
- ZKP



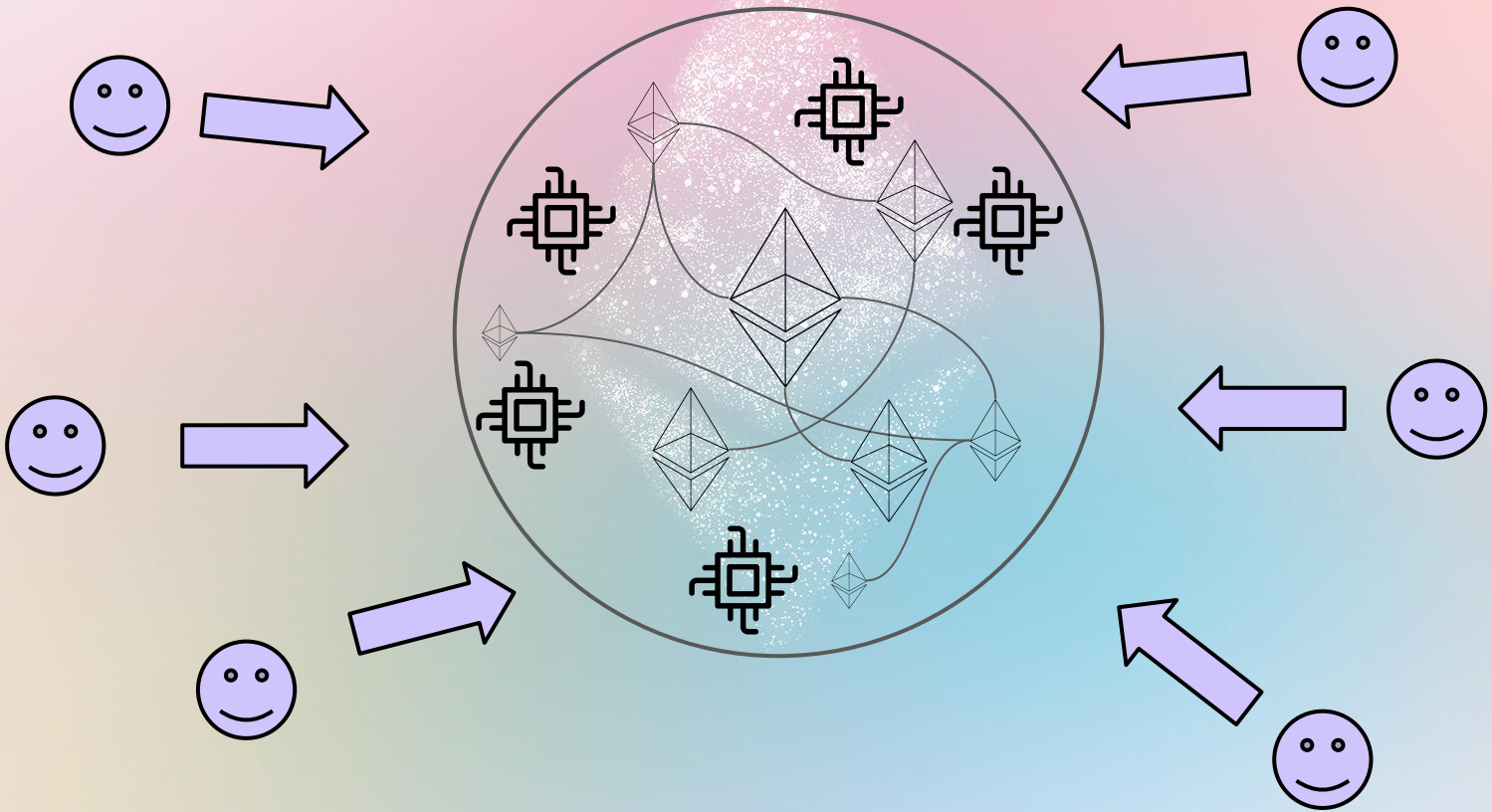
ZK for Scaling

- Computational Scaling
- Data Size

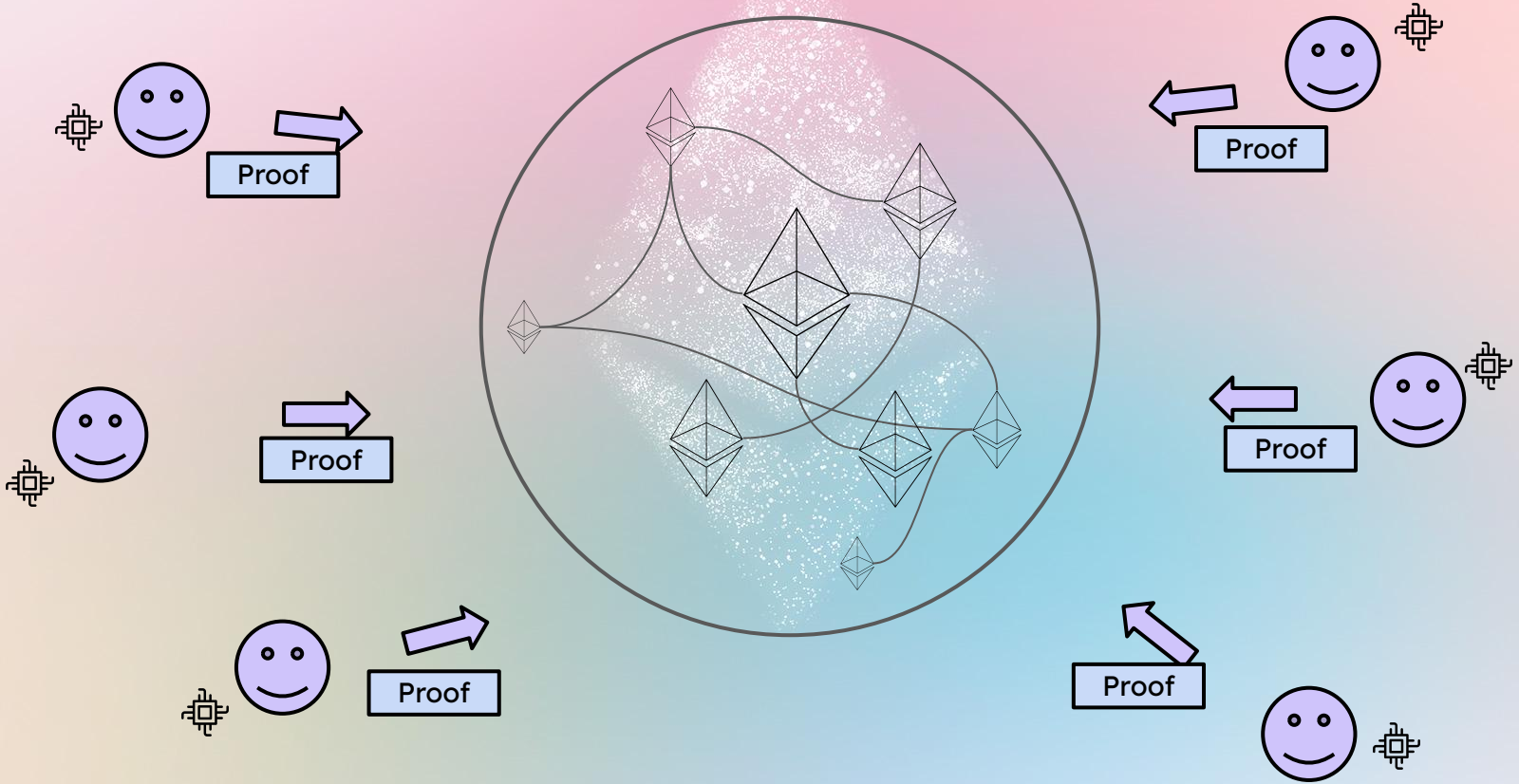
Ethereum is the world computer, or... an earth



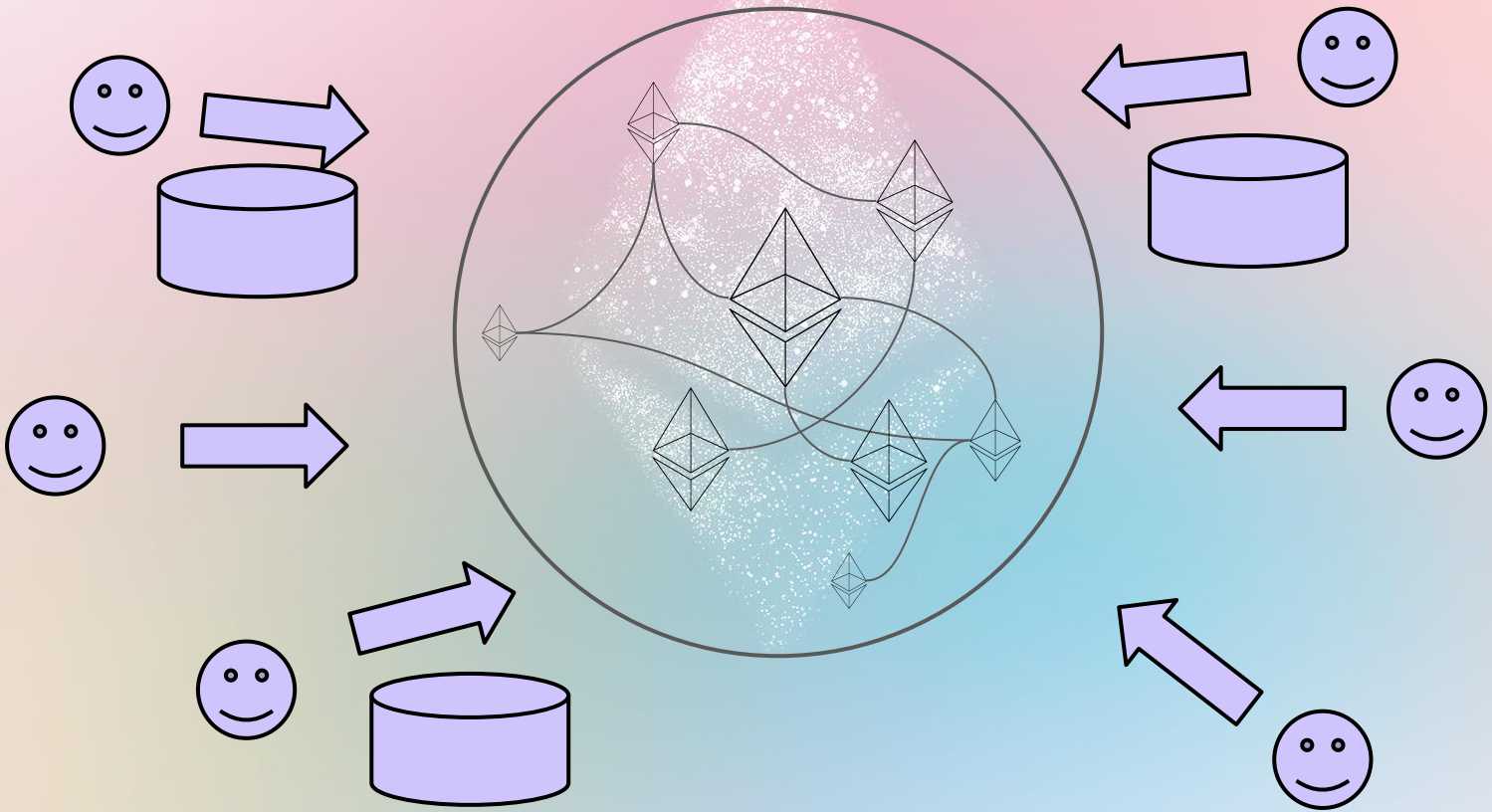
Instead of requesting all computations to the Ethereum



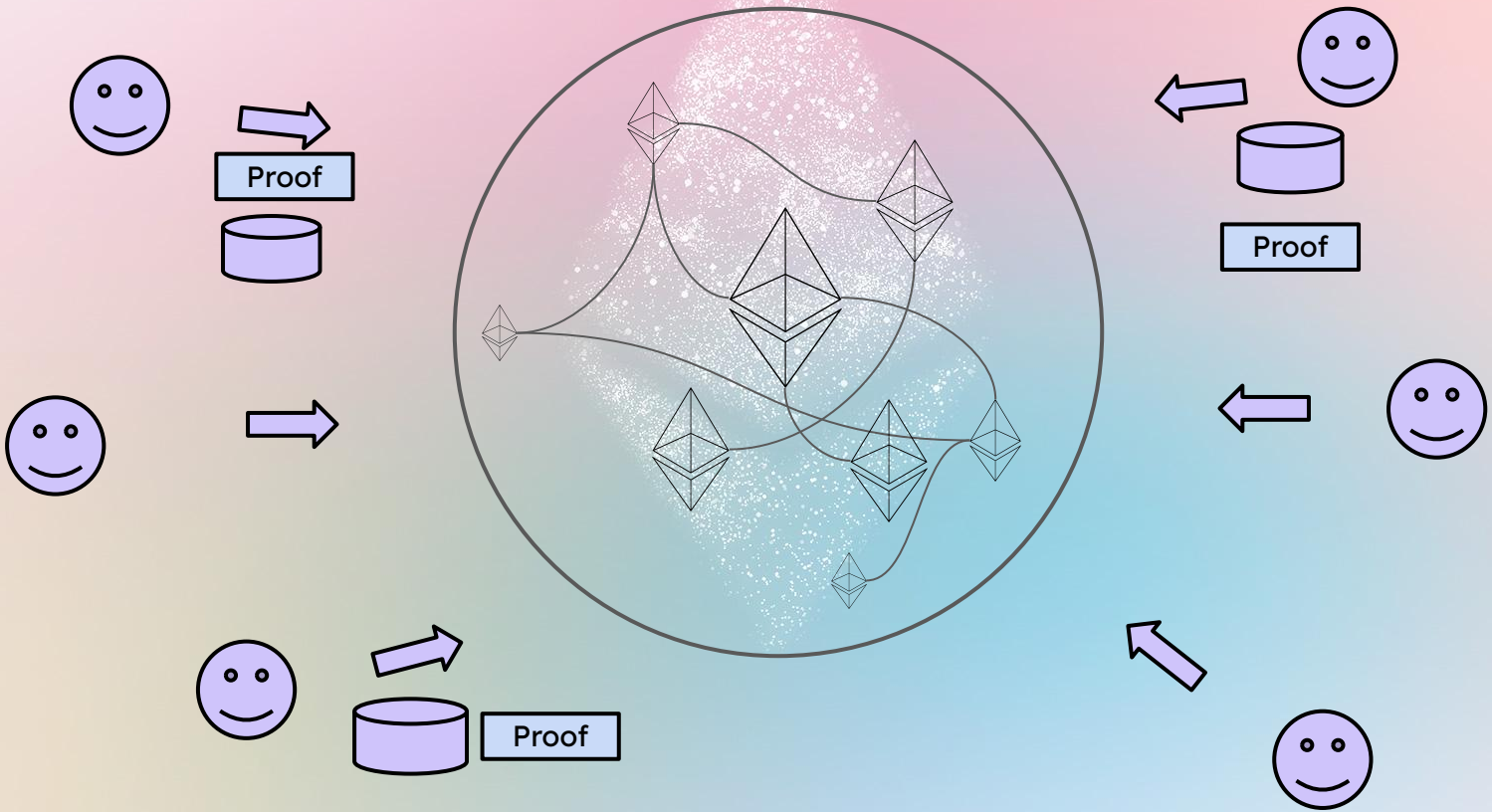
Each end can compute on behalf of the Ethereum



Instead of sending all intermediate data to the Ethereum



We can send only the result and the proof



PSE team's ethos

Public Goods

Experiments

Value Driven

Do what others never do

Maybe freedom(personally)

Basic of the Execution Layer

Bitcoin vs Ethereum

- Bitcoin: array
- Ethereum: Key-Value Dictionary

How to compute the balance in Bitcoin

```
const coinList = []
coinList.push({
 owner: 'alice',
 amount: 100,
})
coinList.push({
 owner: 'alice',
 amount: 200,
})
coinList.push({
 owner: 'bob',
 amount: 300,
})
console.log('all coins', coinList)
const alicesAmount = coinList.reduce((acc, coin) => (acc + coin.amount), 0)
console.log('alice\'s amount', alicesAmount)
```

How to compute the balance in Ethereum

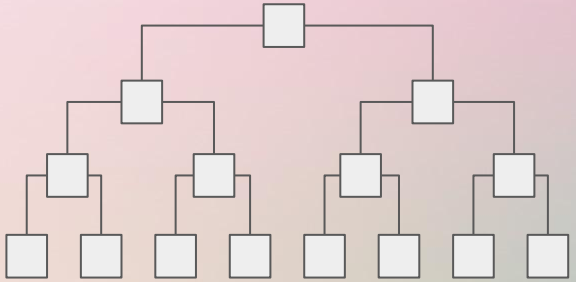
```
const accounts = {}

const stf = (state, tx ) => {
 const newState = Object.assign({}, state)
 if (tx.func ===
'send(ADDRESS,ADDRESS,UINT)') {
   const { from, to, amount } = tx.data
   newState[from] -= amount
   newState[to] += amount
 }
 return newState
}

const state0 = {
 'alice': 10000,
 'bob': 10000,
}
```

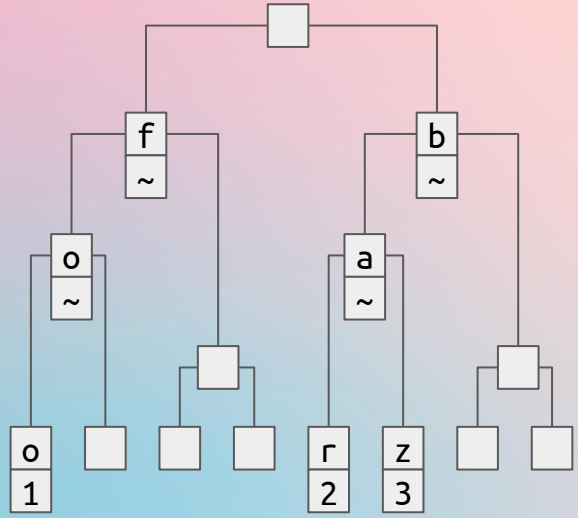
```
const tx0 = {
 func: 'send',
 data: {
   from: 'alice',
   to: 'bob',
   amount: 1000,
 }
}

const state1 = stf(state0, tx0)
```

```
const tx1 = {
 func: 'send',
 data: {
   from: 'alice',
   to: 'bob',
   amount: 2000,
 }
}

const state2 = stf(state1, tx1)

console.log('Alice\'s balance: ',
state2.alice)
console.log('Bob\'s balance: ',
state2.bob)
```

Merkle Tree -> List
Merkle Patricia Tree -> Dictionary

Merkle Tree vs Merkle Patricia Tree

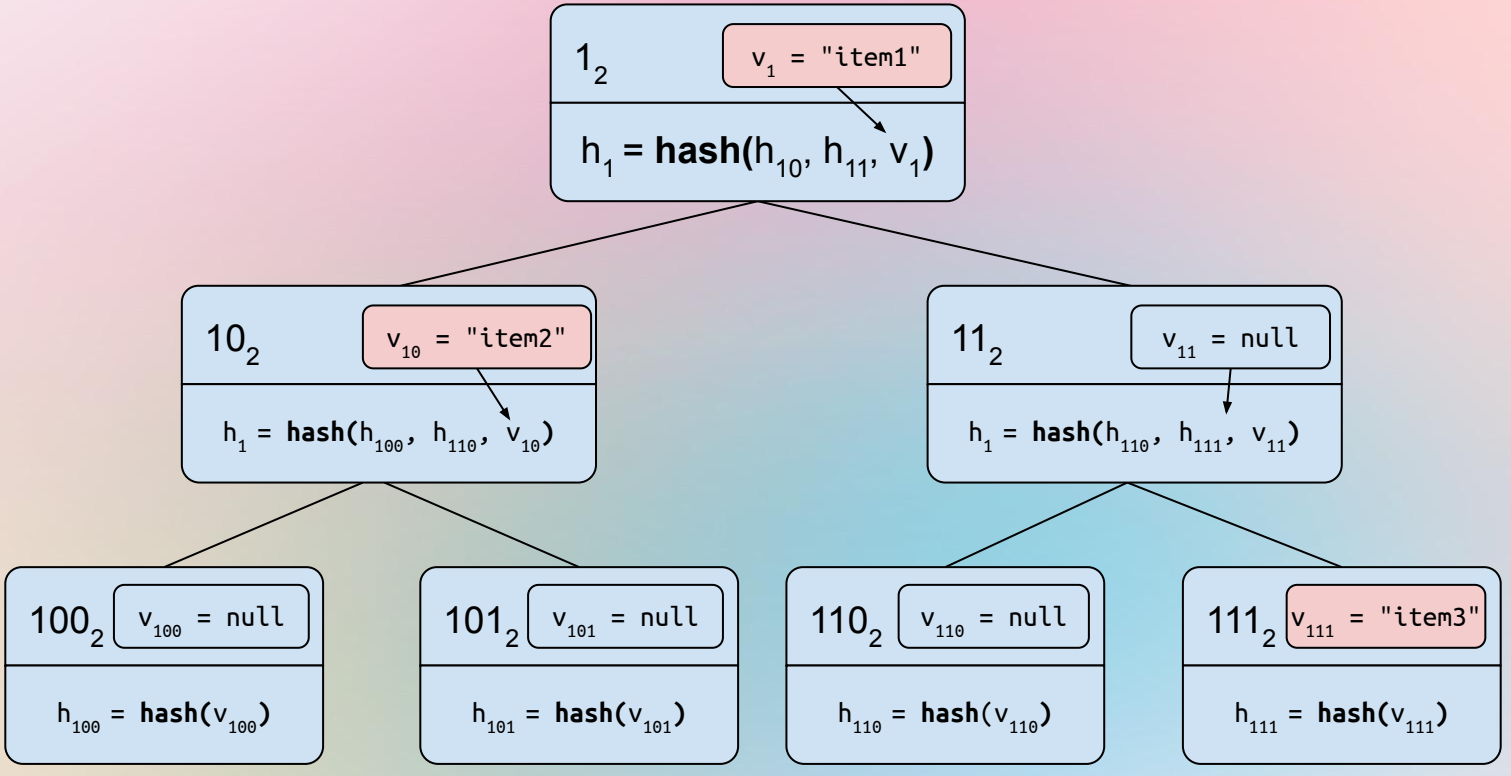




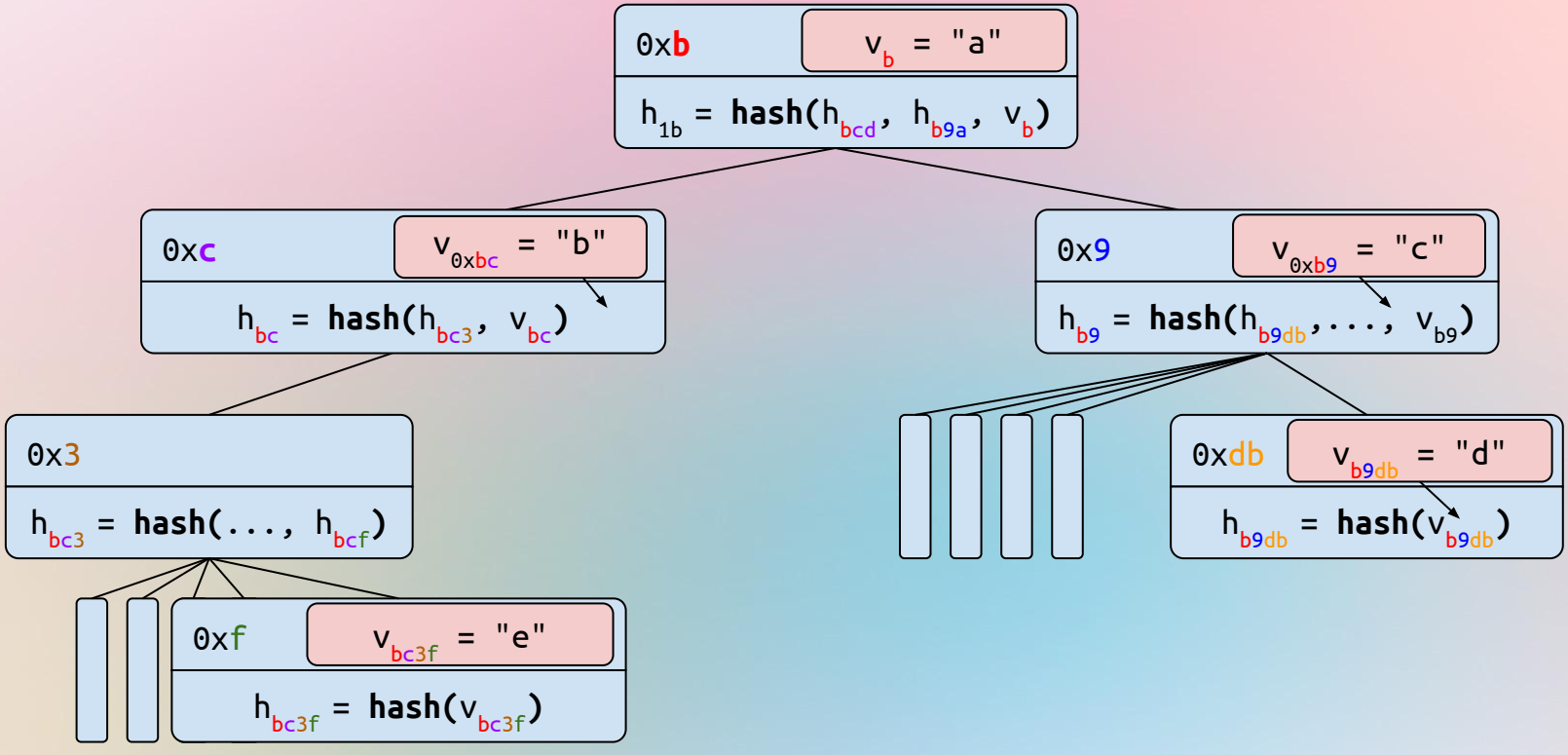
`[a, b, c, d, e, f, g, h]`

`{'foo': 1, 'bar': 2, 'baz': 3}`

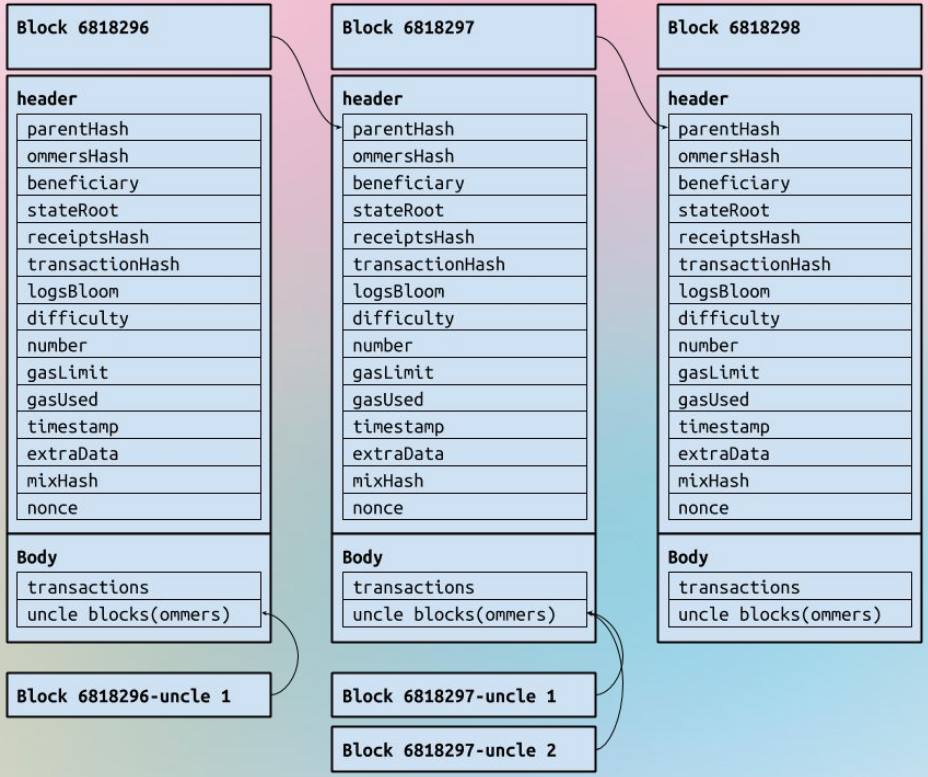
Binary version of Merkle Patricia Tree



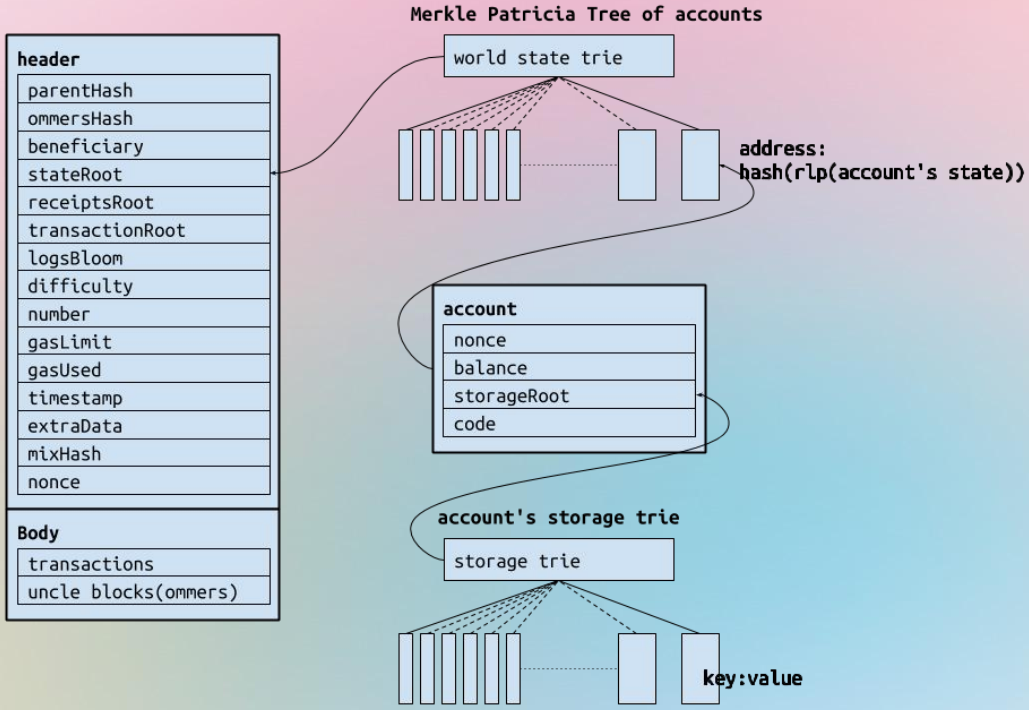
Hexary version of Merkle Patricia Tree



Ethereum Execution Layer's Block Structure

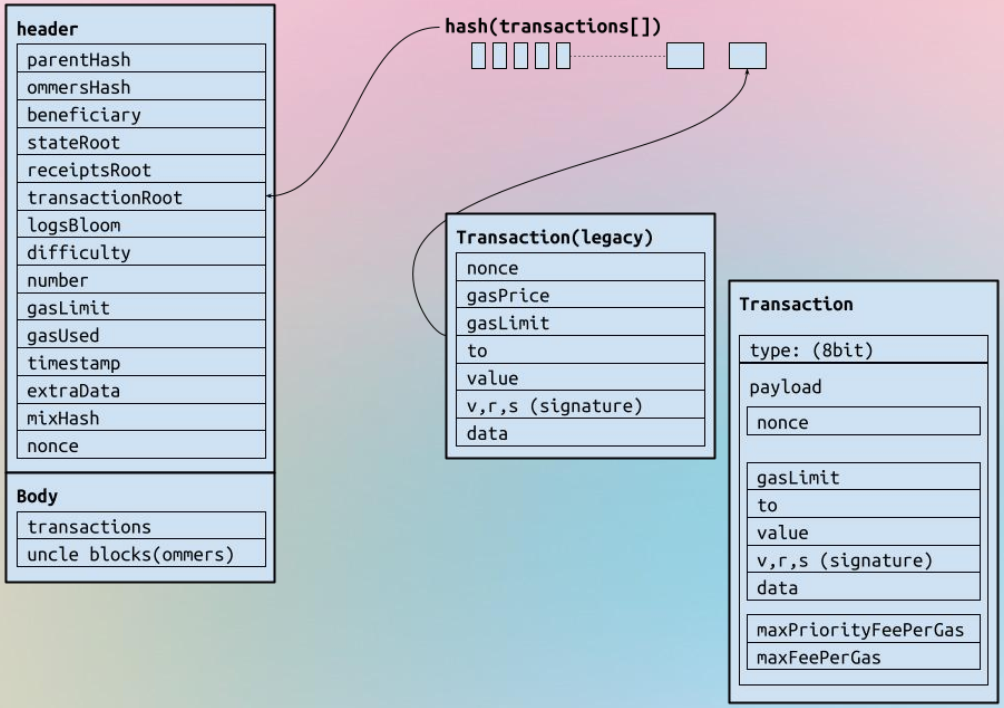


Ethereum's State Trie

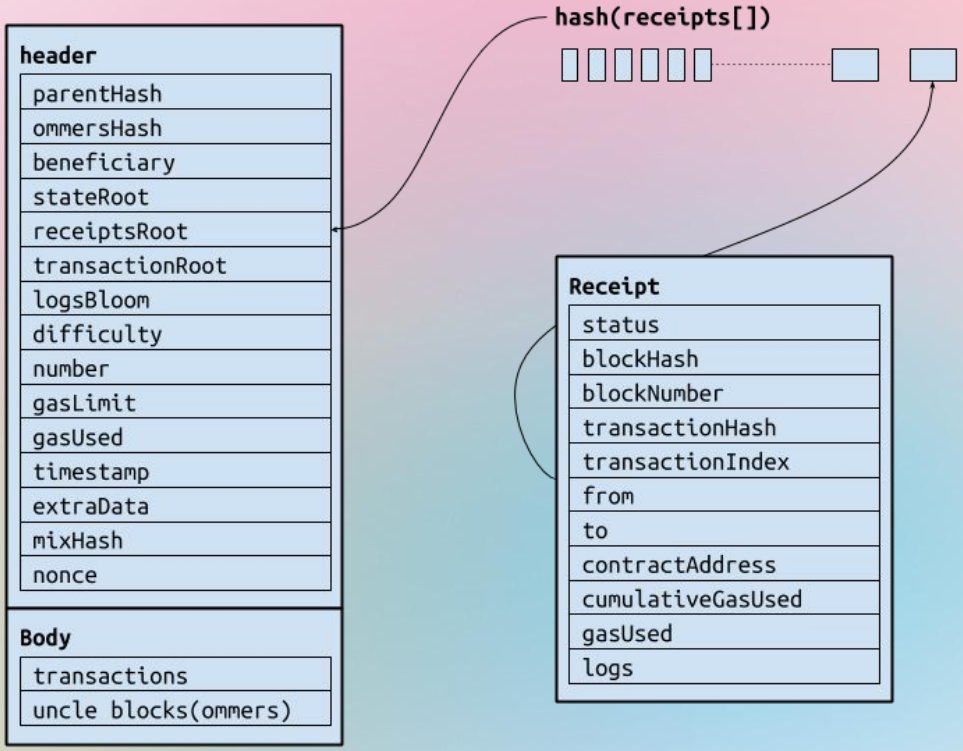


EOA
vs
Contract

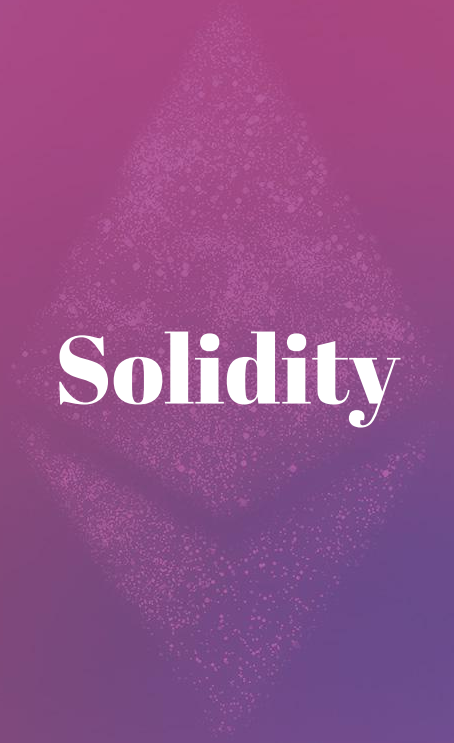
Ethereum L1 Tx



Receipt


hash(receipts[])
header
parentHash
ommersHash
beneficiary
stateRoot
receiptsRoot
transactionRoot
logsBloom
difficulty
number
gasLimit
gasUsed
timestamp
extraData
mixHash
nonce
Body
transactions
uncle blocks(ommers)
Receipt
status
blockHash
blockNumber
transactionHash
transactionIndex
from
to
contractAddress
cumulativeGasUsed
gasUsed
logs

Solidity



Solidity

- constructor
- Layout
- Types
- Event
- Gas
- Function modifier
- Library
- Fallback
- Function Sig

Dev toolings

- Remix https://remix.ethereum.org
- Foundry
- Hardhat
- Etc

Example: KAIST Bitcoin


github.com/wanseob/web3-kaist-bitcoin

Mission

- Implementing Bitcoin on top of Ethereum
- Requirements
  - Miner gets the mining reward by PoW
  - Implement using ERC20 reference

Step 1.

1. Validate KAISTBitcoinTx's signatures

Step 2.

- Implement the Proof of Work algorithm
- Validate the block header

Step 3.

- Validate the block body

Step 4.

- Execute the layer2 transactions and mine a new block

Real-world examples

Real world examples

- Uniswap
- Gnosis Safe
- MakerDAO
- Compound

Treasure Hunt!