Blockchain

Part III: Smart Contracts

Credits: Several of the slides borrowed from Prof. Ari Juels, Jacobs Technion-Cornell Institute and Prof. Aggelos Kiayias, The University of Edinburgh

Motivation

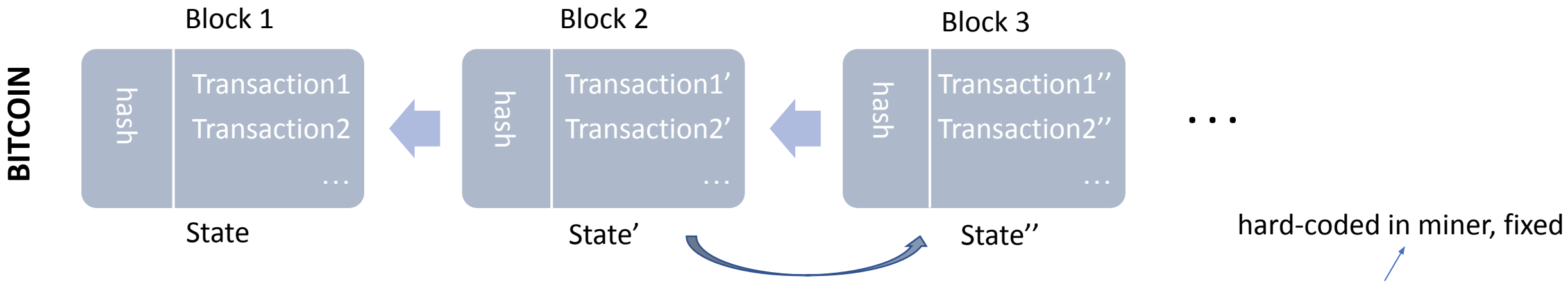
People that use fiat currency as a store of value, there is a name for them, we call them poor.

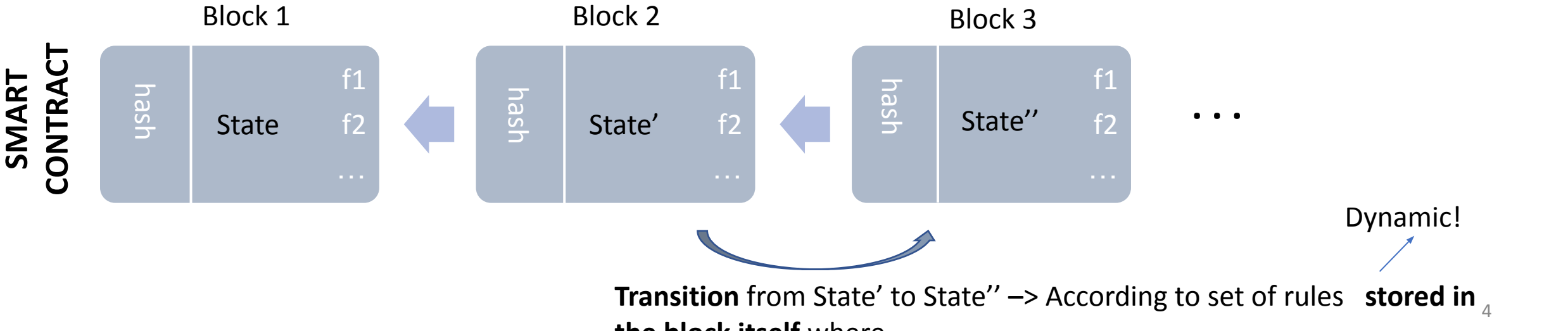
Micheal Saylor, Former CEO of MicroStrategy

The Idea of Smart Contracts

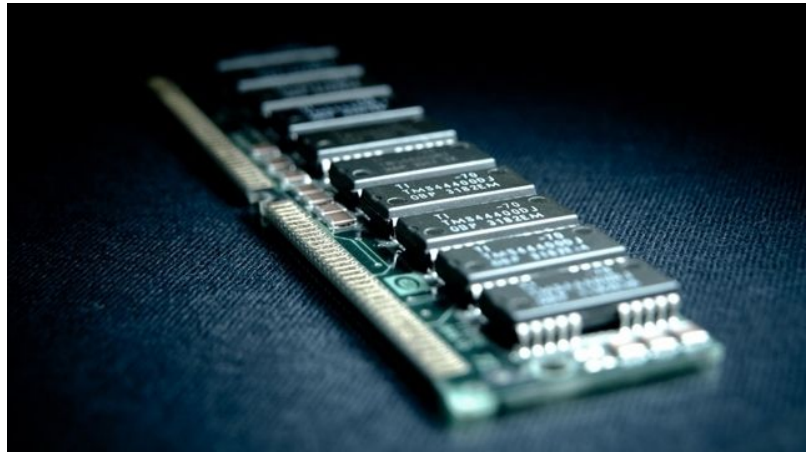
- Contract formalizes a relationship between parties and contains a set of promises made between them
- Smart Contract (1994, Nick Szabo) provides new ways to formalize and secure ‘digital’ relationships
  - can reduce costs
  - uses cryptography in order to secure algorithmically specifiable relationships from being breached and ensure the agreed upon terms are satisfied

Bitcoin vs. Smart Contracts





From Monetary Transactions to Smart Contracts



- As the *set of rules itself is stored in the blocks* in smart contract it can be seen as a memory including 'the code' and 'the data' at the same hardware
- Transition from 'Bitcoin' to 'Smart Contracts' can be seen as transition from fixed rule computer to general-purpose computers

1961: FIRST ALL-ELECTRONIC DESKTOP CALCULATOR

Ethereum

Vitalik Buterin (born 1994, founded Ethereum in 2013)

- Ethereum is an open blockchain platform on which user defined decentralized applications can be built and run
- Ethereum, similar to Bitcoin, is based on a distributed public blockchain, peer to peer network and consensus
- … a programmable blockchain
- In order for users to build an application on Ethereum, they design smart contract(s)
- Each smart contract is stored in the Ethereum blockchain and run by the network nodes when it is invoked

Ethereum

- "Ether" is the internal currency of Ethereum

- Ether is used to pay transaction and computation fees

- Ethereum has a list of denominations with different names and the smallest denomination is called Wei

Unit	Wei
Wei	1
Kwei	1'000
Mwei	1'000'000
Gwei	1'000'000'000
Szabo	1'000'000'000'000
Finney	1000,000,000,000,000
Ether	1000,000,000,000,000,000

Ethereum and Accounts

- Two types of account in Ethereum:
  - Externally owned account
  - Smart contract account

- Each account has an ether balance and stores the number of ether it possesses in Wei.

Ethereum and Accounts

	Externally owned account	Contract account
Identified by an address	✓	✓
Holds the account's Ether balance	✓	✓
Hold contract code	✗	✓
Holds the account's storage	✓	✓
Associated private-public key	✓	✗
Can send signed transactions	✓	✗
Can create contracts	✓	✓
Can send unsigned transactions	✗	✓
Holds a nonce	✓	✓

- Note that contract can send messages and create another contract(s)!

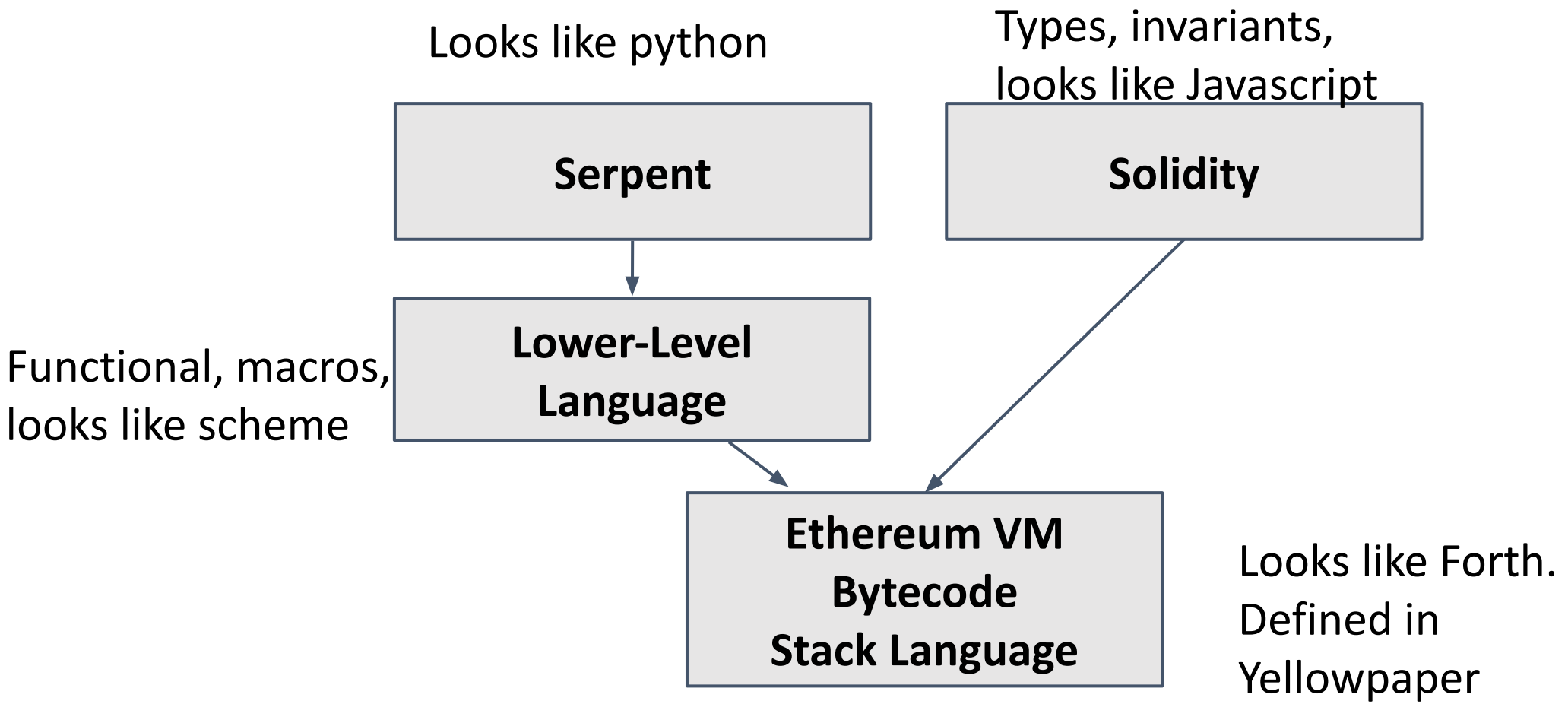
- A signed transaction is sent only by externally owned account and sending it will incur a transaction cost.

Ethereum & Gas

- Gas is a unit that measures the amount of computational effort that it will take to execute certain operations
- Each operation is tagged with an explicit cost (Gas)
- Each transaction incurs a cost to the sender
- Gas is the unit of all computational tasks in Ethereum

https://www.ethos.io/what-is-ethereum-gas/

Ethereum Languages



Basics

- Submit a transaction to the blockchain in order to create a new contract
  - transaction contains the *code* as data
  - contracts have an “account balance” denominated in Ether
  - contracts have a persistent “storage” file
- Submit transactions to the blockchain to interact with the contract
  - transactions can contain monetary “value,” added to the account
  - procedure calls

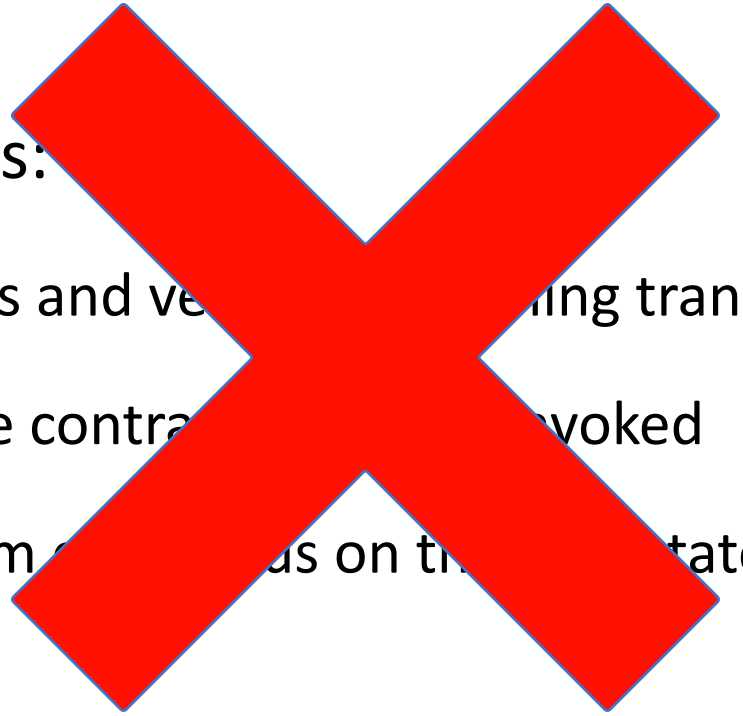
Ethereum Smart Contract

- An Ethereum smart contract is an account with code
- It cannot invoke itself
- It must be invoked by another account

Ethereum Structure

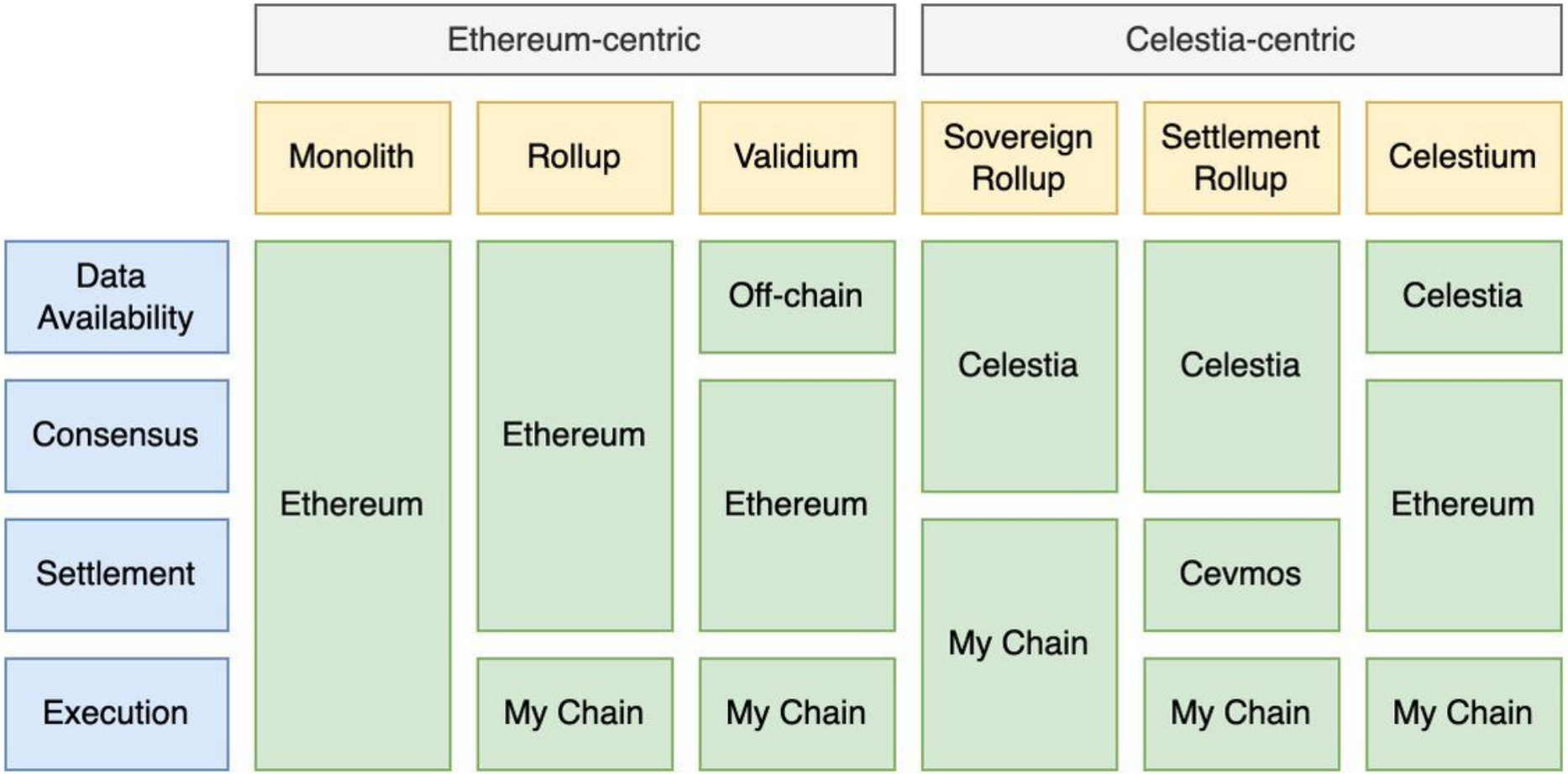
- All miners:
  - process and verify all incoming transactions
  - run the contracts that are invoked
  - perform consensus on the new state using the blockchain

Ethereum Structure



- All miners:
  - process and ve ing transactions
  - run the contra voked
  - perform us on t tate using the blockchain

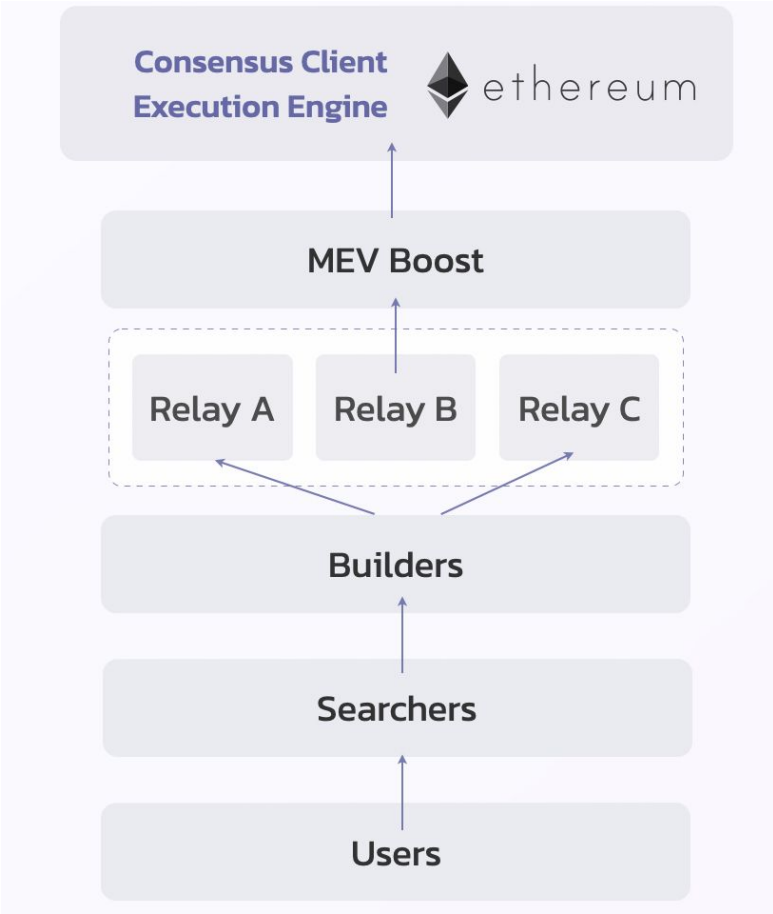
Ethereum Structure



https://twitter.com/ptrwtts/status/1509869606906650626

Ethereum Structure

“MEV-Boost is an implementation of proposer-builder separation (PBS) built by Flashbots for proof of stake Ethereum. Validators running MEV-Boost maximize their staking reward by selling blockspace to an open market of builders.”



https://mirror.xyz/ohotties.eth/kw_7qbkOl4NV1pmpRgVwtsS-7TZff_zTmmNEOm2BbmU

https://boost.flashbots.net/

Smart Contract – Example 1 (Dummy Example)

- How does a smart contract look like?
- Let’s design a contract that:
  - Holds its owner/creator address
  - Assigns an arbitrary value to its state
  - After it is deployed to the blockchain, each time a function of it is called and a value is passed to it, it doubles the value and returns the result

Smart Contract Example 1

We have used Solidity compiler version: “v0.4.21+commit” on remix

```
1  pragma solidity ^0.4.4;
2
3  contract Example1 {
4
5      uint public variable;
6      address public owner;
7      function Example1 () {
8          variable = 30;
9          owner = msg.sender;
10     }
11     function double_it (uint value) returns (uint){
12      var temp = value * 2;
13      return temp;
14     }
15 }
```

This function is called constructor. It is run only once when the contract is sent to the blockchain. It is not considered as a contract function and cannot be called, after the contract is created.

We can call double_it() from outside as: instance.double_it.call (5, {from:”0x11”}); “instance” is just an instance of the contract and is defined (using web3.eth) when we want to interact with.

This function doubles the value it receives and returns the result.

msg.sender here is the address of the account that sends the contract to the blockchain.

In this example, the contract function does not change the contract state, i.e. the value of “variable” is not changed by function: double_it ().

Smart Contract – Example 2 Flight delay insurance

- An example of aviation delay insurance
- Purchasing the delay insurance, the aircraft is often delayed until the passengers get on the plane before the deadline of **2 hours**
    - This avoids the insurance company paying the compensation fee

https://medium.com/@cypherium/three-application-examples-of-cypherium-smart-contract-1e14eeaa2bcc

Smart Contract – Example 2 Flight delay insurance

1. “Automated underwriting claims”
    - Policy is recorded on the blockchain
    - If the aircraft is delayed, the smart contract can automatically transfer the claim amount from the insurance company’s account to the customer’s account
2. “Fairness”
    - With the insurance companies the formulation and review of rules can not be completely fair, and the corresponding smart contracts on the public chain can solve the fairness problems.

https://medium.com/@cypherium/three-application-examples-of-cypherium-smart-contract-1e14eeaa2bcc

Smart Contract – Example 2 Flight delay insurance

- The smart contract example for this problem from **Pacific Insurance Company:**

Protection Program

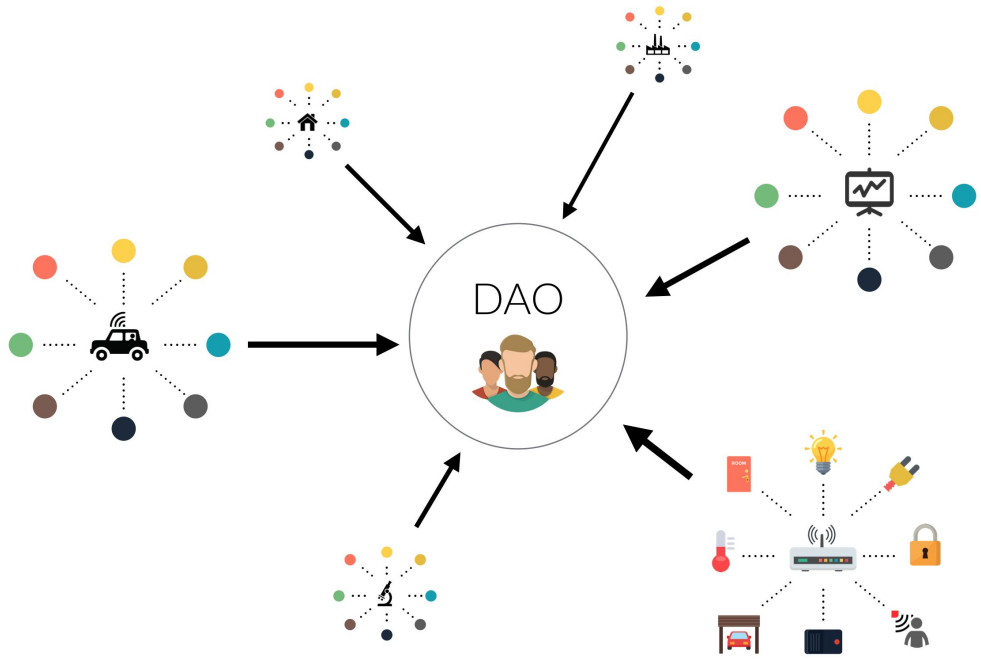
Flight-delay insurance(Takeoff delay)		
Insurance period		Designated flight
Insurance cost		10RMB
Insurance amount	A	100RMB indemnity if flight delay 2.5H or more
	B	200RMB indemnity if flight delay 3.0H or more
	C	400RMB indemnity if flight delay 4.0H or more

- You can see the code using Cypherium Java smart contract here: https://github.com/cypherium/ContractExample/blob/master/src/use_case/Insurance/Fligt.java

https://medium.com/@cypherium/three-application-examples-of-cypherium-smart-contract-1e14eeaa2bcc

Smart Contract Applications

- Distributed Autonomous Organisations (DAO)
  - Smart contract acts as a virtual organisation with a predefined set of rules and actions/functions
  - If the majority of it's members/stakeholders decide (via voting) to take certain action, the contract automatically does it and delivers the result



Smart Contract Applications

- Decentralized Crowd Funding
  - Central authority who receives the funding is substituted by a smart contract
  - Donors pay smart contract and when the funding reaches a certain value, the funding is automatically delivered to the funding recipient

Smart Contract Applications

- Robust and Fair Multi-party Computation
  - Allows all parties to engage in a multi-party computation to get the output of computation; in case of an abort, they will be monetarily compensated

- Efficient Verifiable Computation
  - To incentivize certain computations of high cost and monetarily compensate them

What’s the deal with The DAO?

- Crowdfunding instrument that raised $150+ million dollars of ETH
  - Initially, the goal was $10,000 to fund a Bike Lock company
- TheDAO contract is ambitious! Lets users pool their investments and vote
- A subtle bug in TheDAO led to the loss of $50 million worth of tokens
- The Ethereum community developed a “Hard Fork” to cancel the theft
- A faction of dissenters are maintaining “Eth Classic”, also traded on exchanges