COMP4137 Blockchain Technology and Applications
COMP7200 Blockchain Technology

Lecturer: Dr. Hong-Ning Dai (Henry)

Lecture 6

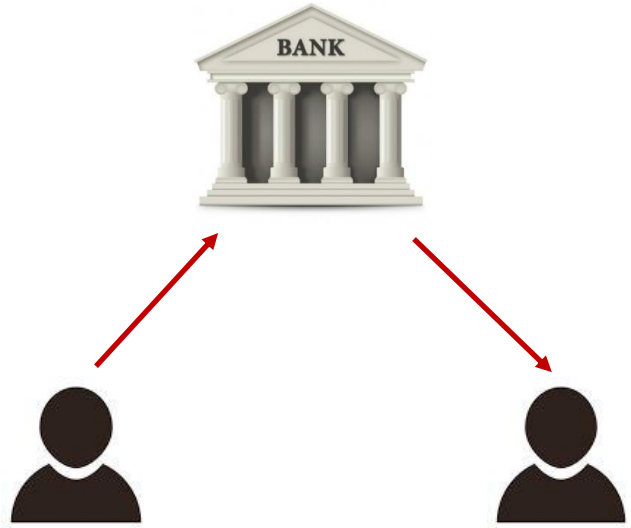
Permissionless blockchain 1

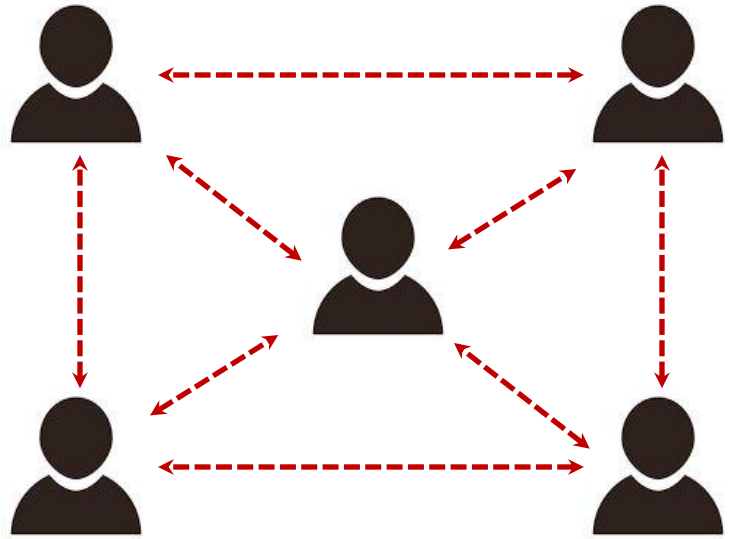
Outline

- Permissionless Blockchain
- Ethereum
  - Account-based Model
  - External Account and Contract Account
  - Gas
  - Transactions
  - Ethereum Virtual Machine
  - Smart Contract (Solidity)
- Coins and Tokens

Bitcoin

- Decentralized Money





- Use money digitally through an intermediary, such as a bank or Paypal.
- The money used was still a government issued and controlled currency.

- Bitcoin changed all that by creating a decentralized form of currency that individuals can trade directly without an intermediary.
- Bitcoin transactions are validated and confirmed by entire Bitcoin network.
- No single point of failure so the system is virtually impossible to shut down, manipulate or control.

What else can we decentralize?

- Voting
  - a central authority to count and validate votes
- Real estate transfer records
  - centralized property registration authorities
- Social networks
  - (e.g., Facebook) based on centralized servers that control all of the data we upload to them
- …

Blockchain Technology

Cryptography

+

Consensus Algorithm

+

Decentralized Networks

A system that can reach decisions without a central authority

Blockchain is to Bitcoin what the Internet is to email.

The Bitcoin Network

- Bitcoin script - Turing incomplete language
  - No loops or complex flow control capabilities
  - We need new system for decentralized applications

Blockchain Architectures

- Permissionless (public) Blockchains
  - Bitcoin
  - Ethereum
  - Zcash
  - …
- Permissioned (private/consortium) Blockchains
  - Hyperledger Fabric
  - Quorum

Permissionless Blockchains

- Characteristics:
  - Participation open to the **public**
  - **Peer-to-peer** transactions
  - Typically tied to **Cryptocurrency**
  - Fully **decentralized**
- Challenges:
  - **Privacy** and **scaling**

Permissionless blockchain is a disruptive technology that can dramatically change how we conduct business activities.

Permissioned Blockchains

- Characteristics:
  - Participation can be **private** and/or **controlled**
  - **Trusted** participants
  - More **efficient** than many public blockchains
  - Can support **privacy** and **confidentiality** in transaction
- Challenges:
  - Some level of **centralized trust** through governing authority

Permissioned blockchains may lead to cost-savings, workflow improvements, automation and improved auditing with current business processes.

Outline

- Permissionless Blockchain
- Ethereum
  - Account-based Model
  - External Account and Contract Account
  - Gas
  - Transactions
  - Ethereum Virtual Machine
  - Smart Contract (Solidity)
- Coins and Tokens

Limitation of Bitcoin

- Recall: UTXO contains (hash of) public key scripts

  (simple) script: indicate conditions when UTXO can be spent

- **Lack of Turing-completeness**

  script does not nearly support everything

  Lack of loop instructions

- **Value-blindness**

  UTXO is all-or-nothing – it must be spent completely as a whole

  Cannot provide fine-grained control over the amount that can be withdrawn

  Example – Hedging contract: A and B put in $1,000 worth BTC; after 30 days sends $1,000 worth of BTC to A and the rest ($2,000) to B

Limitation of Bitcoin

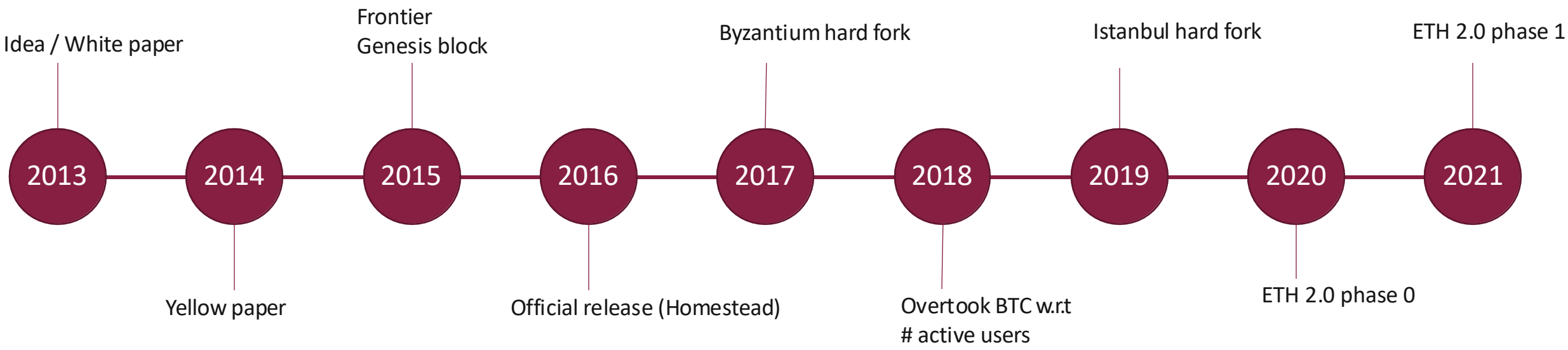
- **Lack of state**
  - UTXO can be either spent or unspent
  - Script does not have their own internal persistent memory
    - Impossible for multi-stage contracts or enforce global rules on assets
    - Difficult to implement complex stateful contracts

- **Blockchain-blindness**
  - Scripts cannot access some blockchain data such as nonce, timestamp – all are valuable sources of randomness
  - Limit applications in gambling

Ethereum

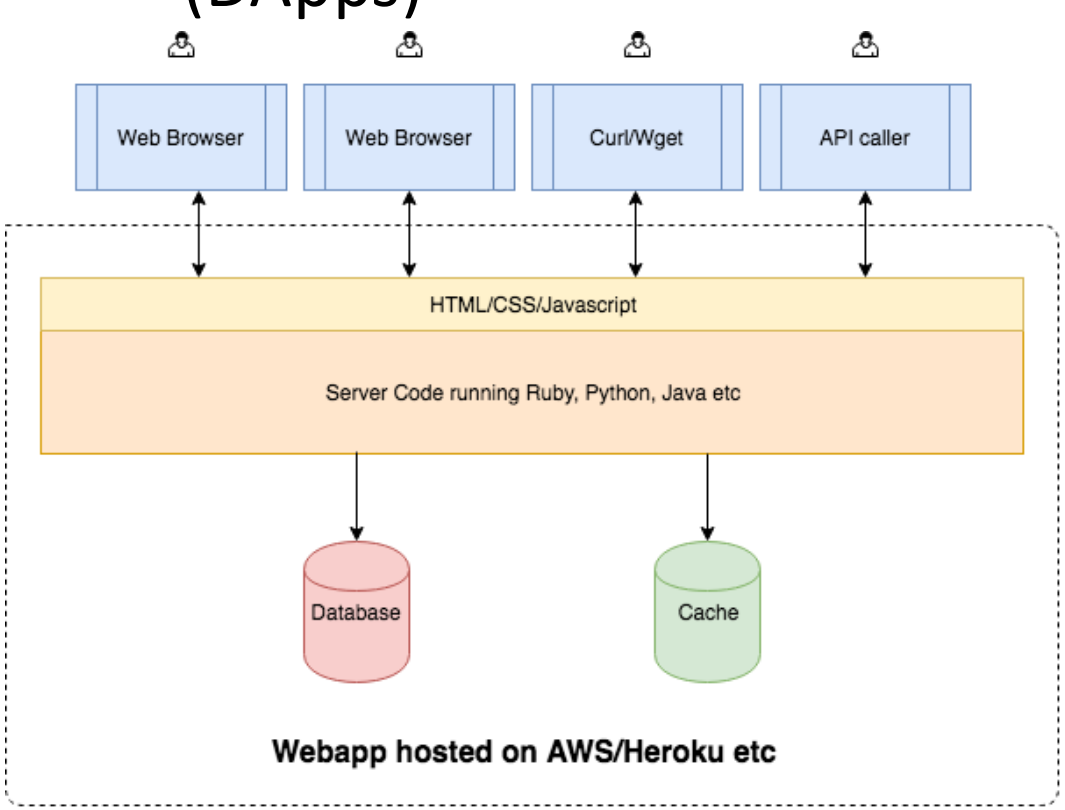
- Founded by Vitalik Buterin, Gavin Wood and Jeffery Wilcke in 2014
- Support “Turing complete” programming language, Solidity, as the smart contract

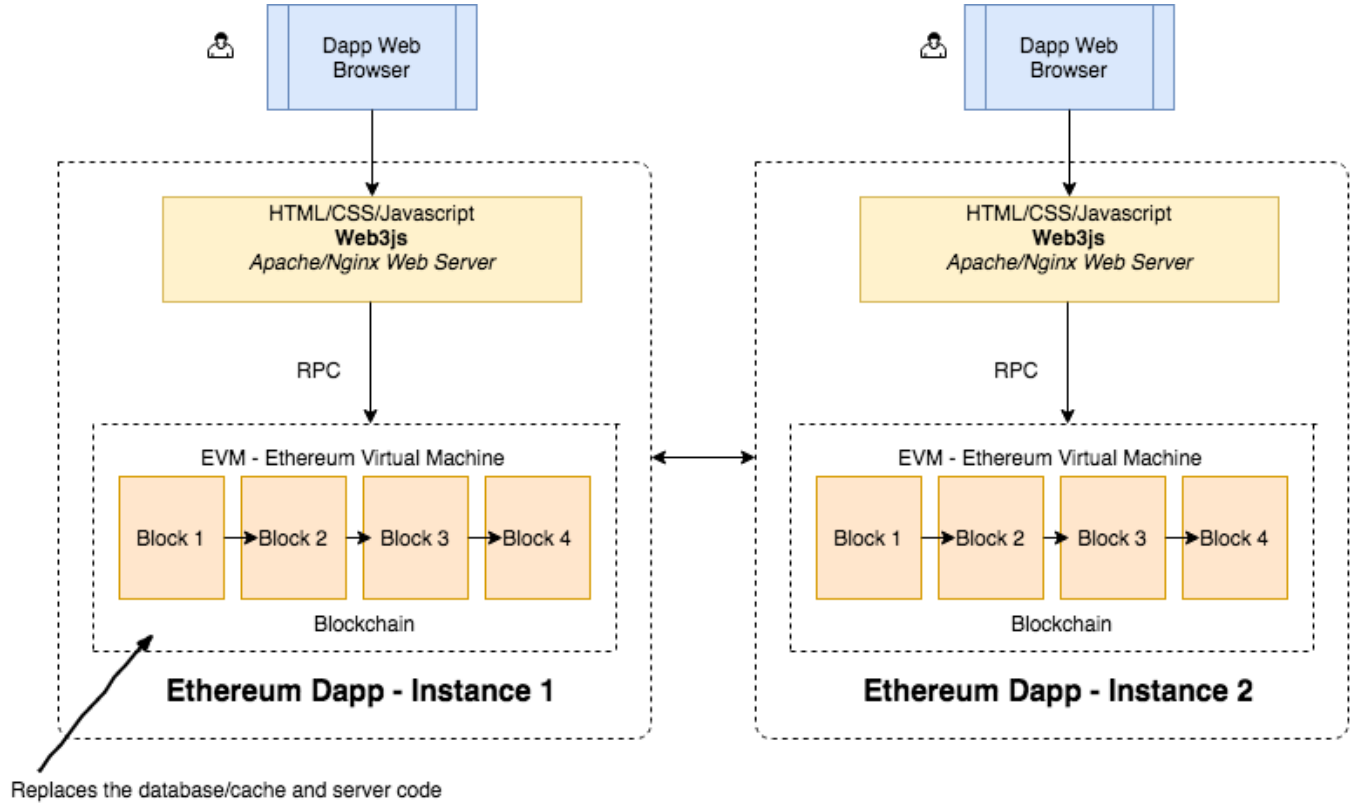
Timelines



Ethereum

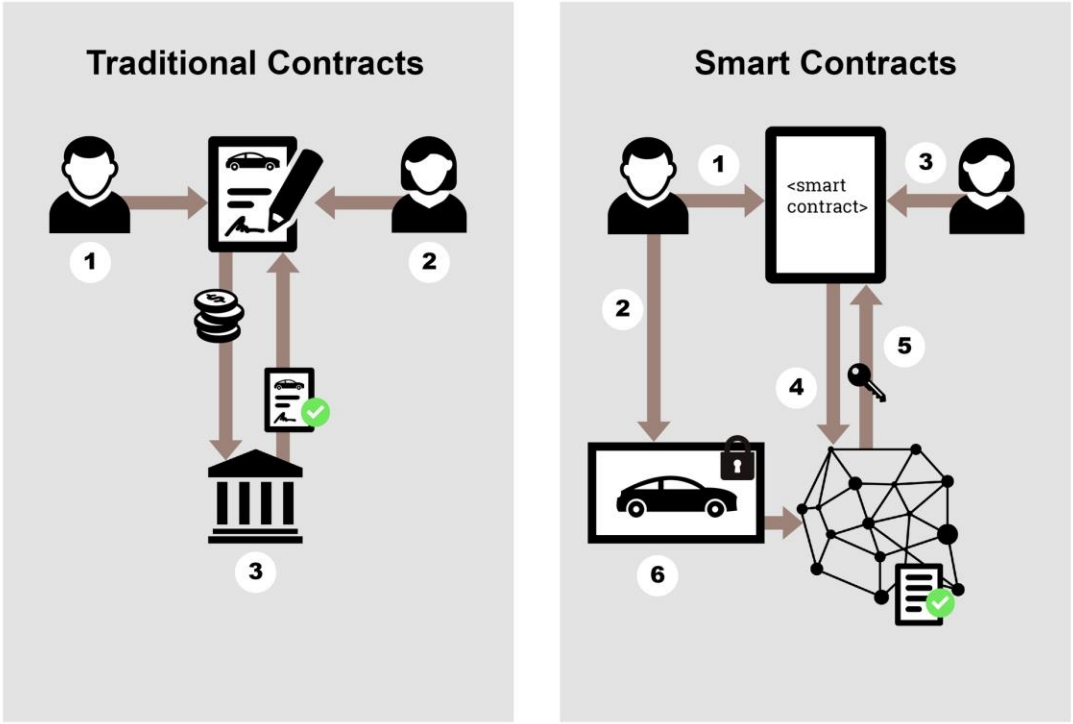
- Ethereum is an open source, distributed software platform based on blockchain technology.
- It enables anyone to build and deploy decentralized applications (DApps)





Ethereum

- Every node of the network runs the Ethereum Virtual Machine (EVM) and execute the same instructions on the blockchain
- Smart contracts are executed on EVM when pre-specified conditions are met



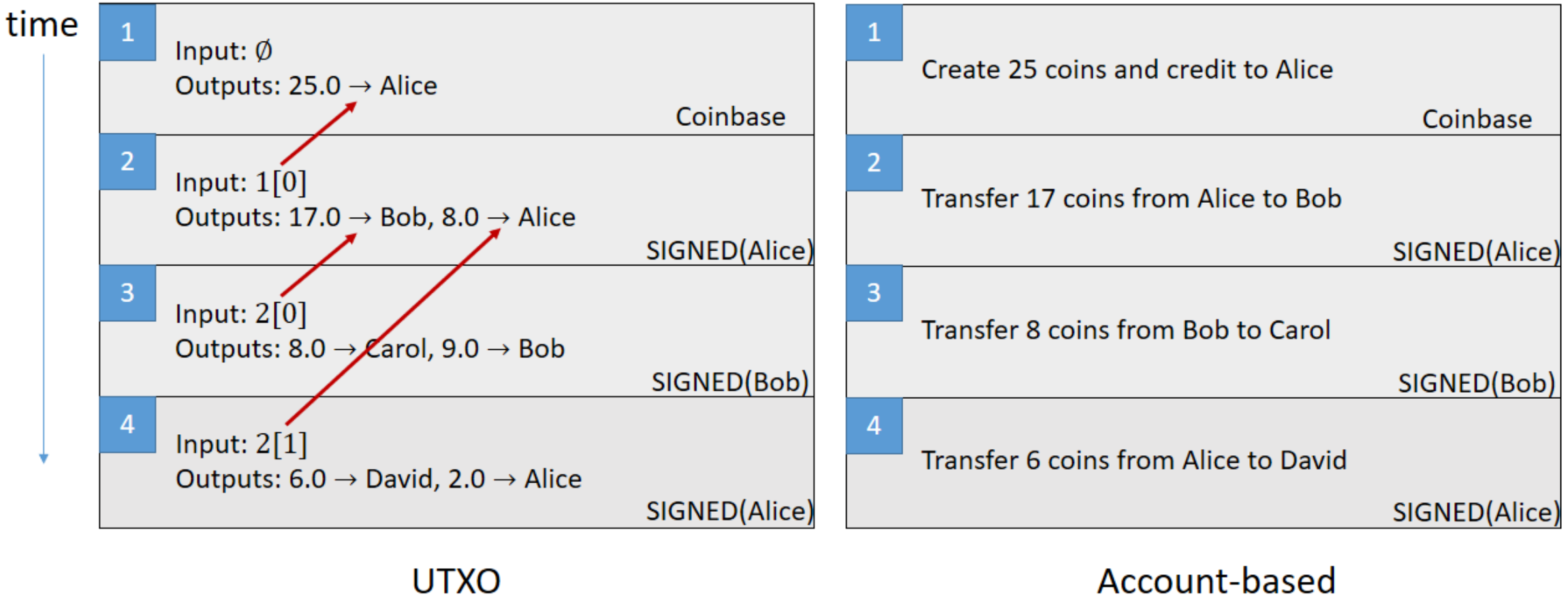
Ethereum

- Ethereum provides a universal, programmable blockchain which anyone can use

Ethereum

- Ether
  - Mined in a way similar to Bitcoin
  - Block reward: 2 ETH
  - 1 Ether = 10^{18} Wei
  - Currently $ 1,500+ per Ether
- Ether is the crypto-fuel for the Ethereum
  - Used for payment to compensate the machines for executing the smart contract
  - Used as an incentive ensuring that developers write better quality applications (wasteful code costs more)
- Ether is maintained by the account-based model
  - Unlike the Bitcoin's UTXO

UTXO vs. Account-Based Model



UTXO

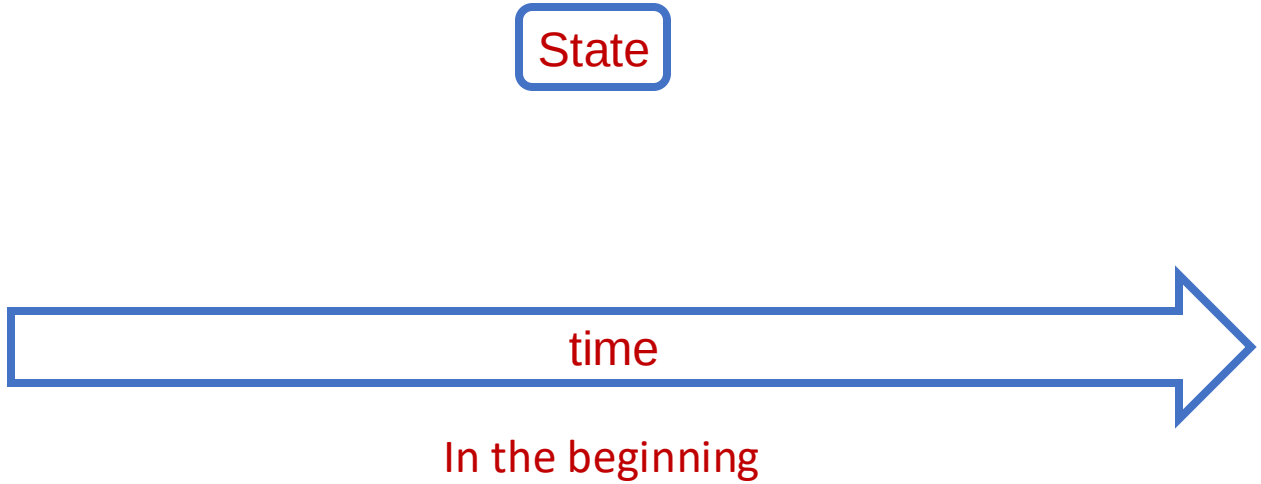
Account-based

UTXO vs. Account-Based Model

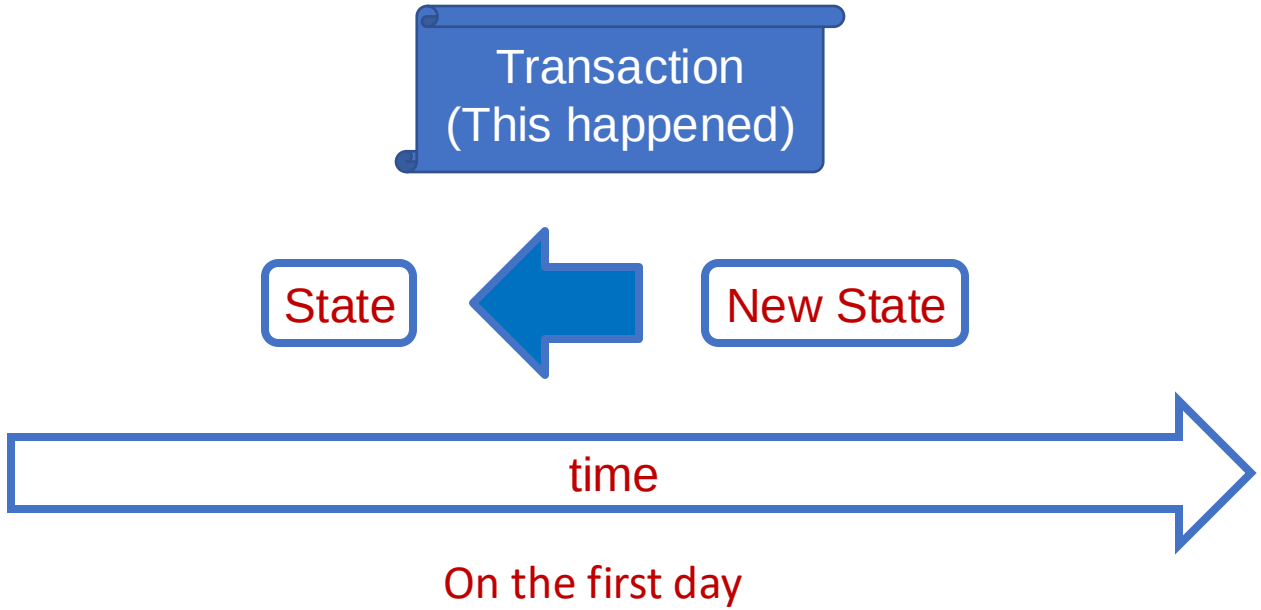
- UTXO: unspent or spent
- Record receipts of transactions
- Record balances on the client-side by adding up the available unspent transaction outputs
- Used for Bitcoin

- Account-based
- Keep track of the balance of each account globally
- Check whether the balance is no less than the spending transaction amount
- Used for Ethereum (also banks)

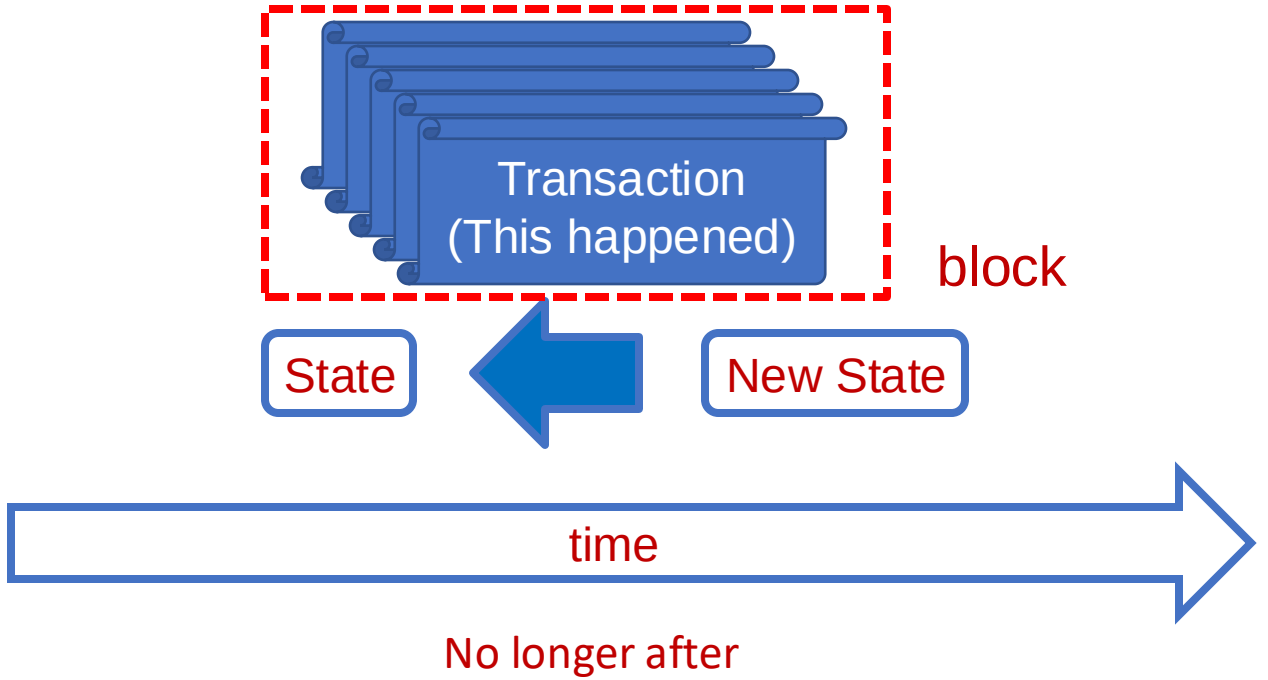
Account-Based Model



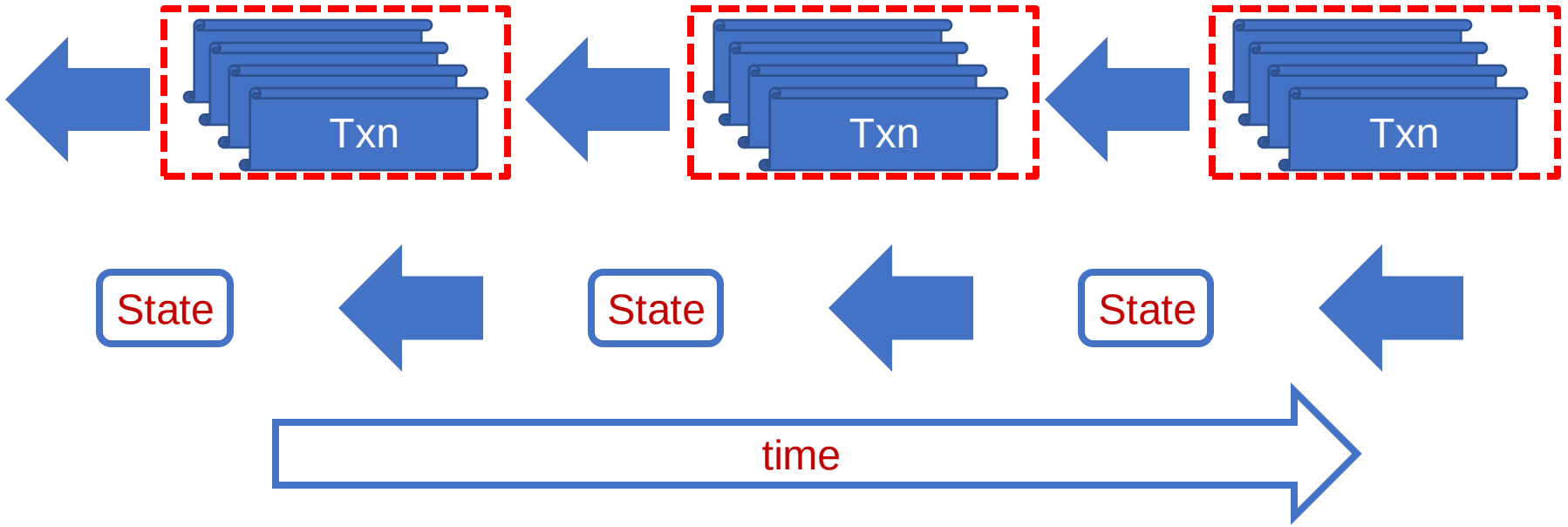
Account-Based Model



Account-Based Model



Block-State Duality

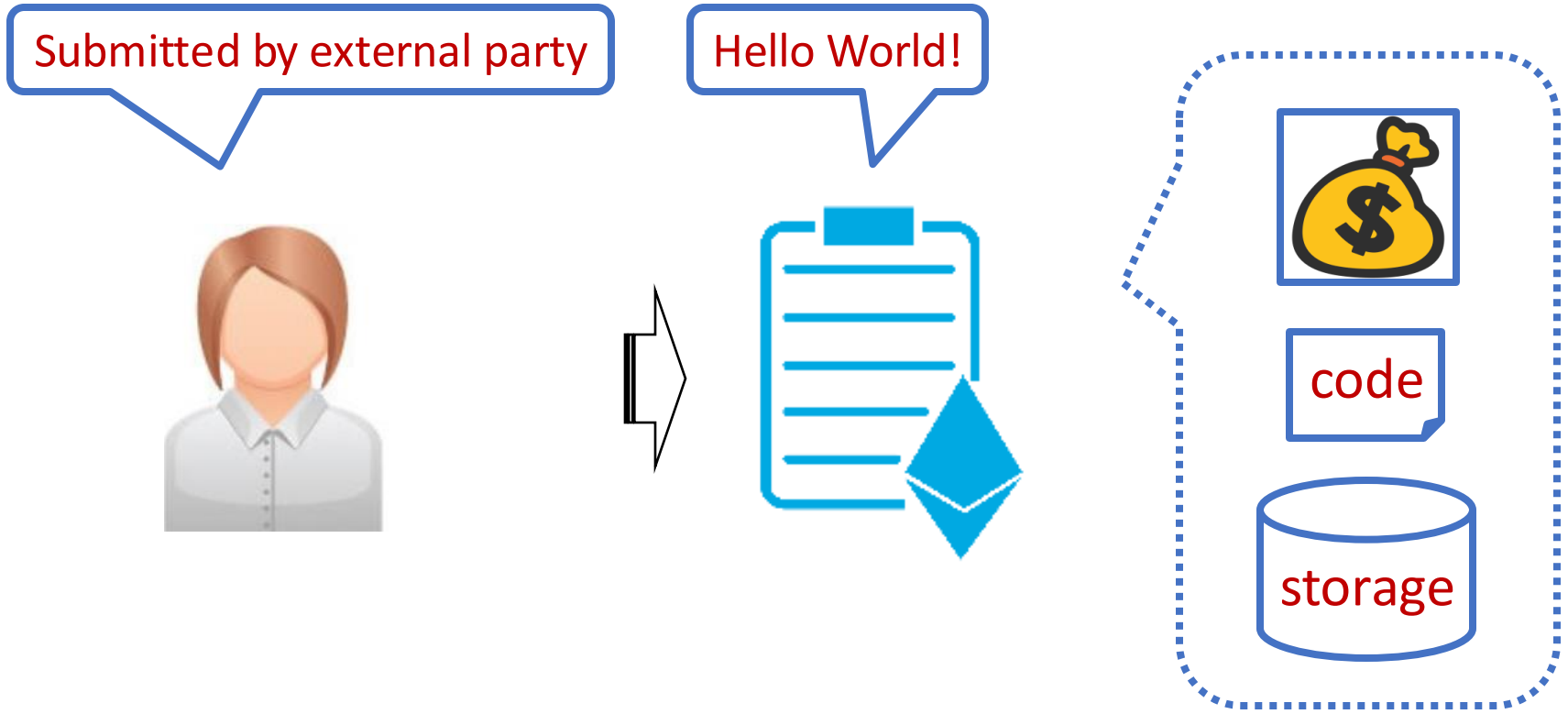


Ethereum Accounts

- Externally Owned Accounts (EOA):
  - Has an Ether balance
  - Can send transactions (Ether transfer or trigger contract code)
  - controlled by ECDSA signing key pair (pk,sk)
  - Has no associated code
- Contract Accounts
  - Has an Ether balance
  - Has associated code (smart contract)
  - Code execution is triggered by transactions or messages received from other contracts
  - Has its own permanent state

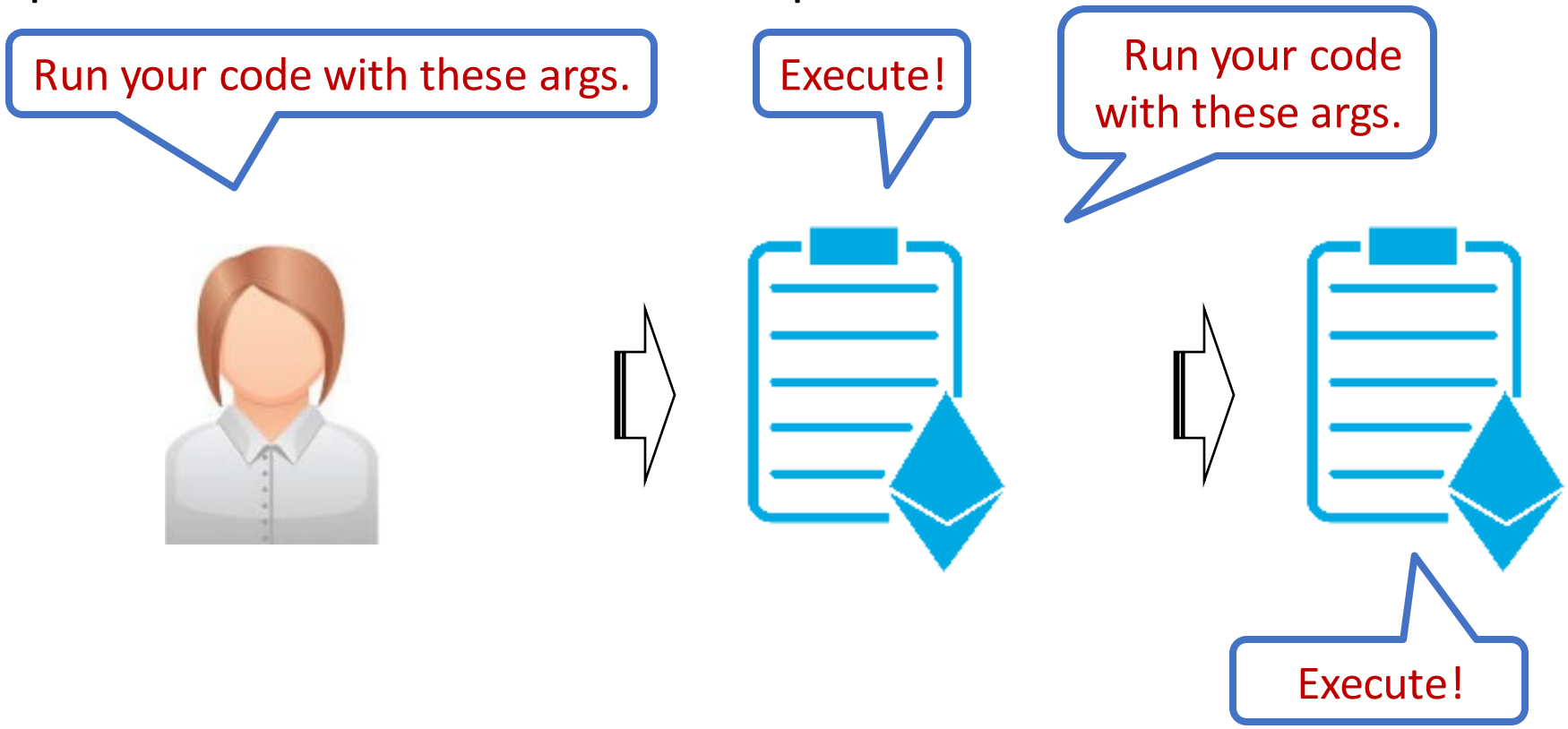
Ethereum Transactions

- **Contract creation transaction**
  - Create new contracts on blockchain
  - EVM code for account initialization is specified



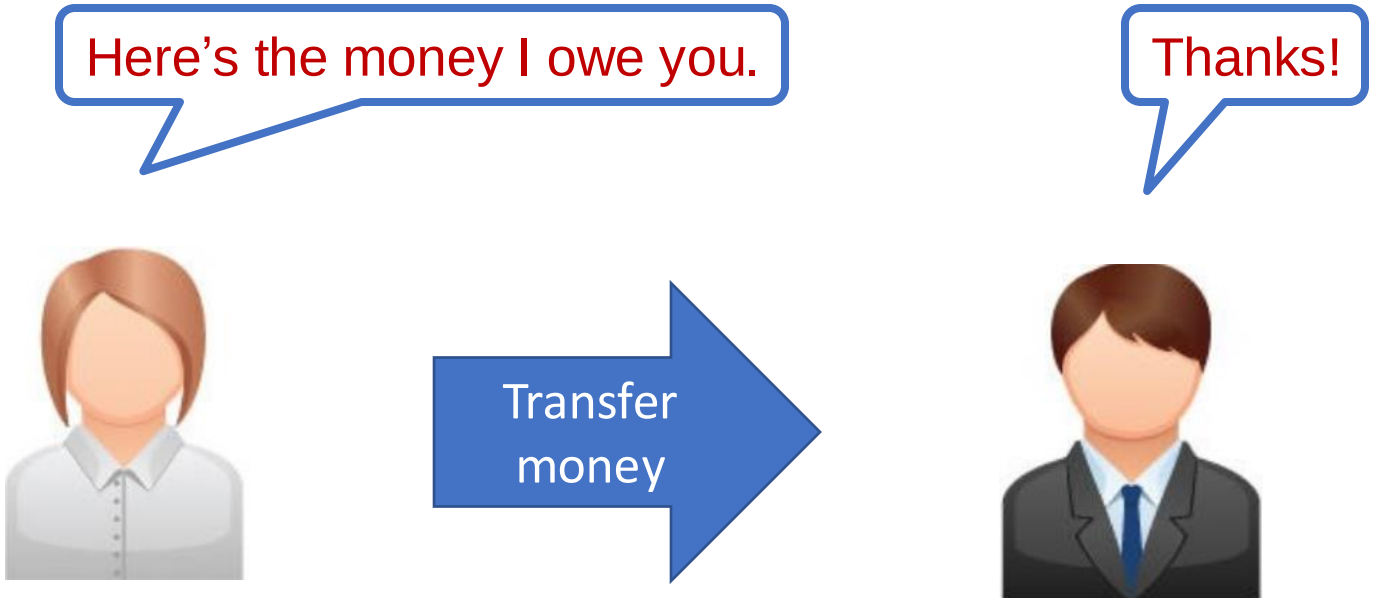
Ethereum Transactions

- **Message call transaction**
  - Call methods in an existing contract
  - Input data to contract methods is specified



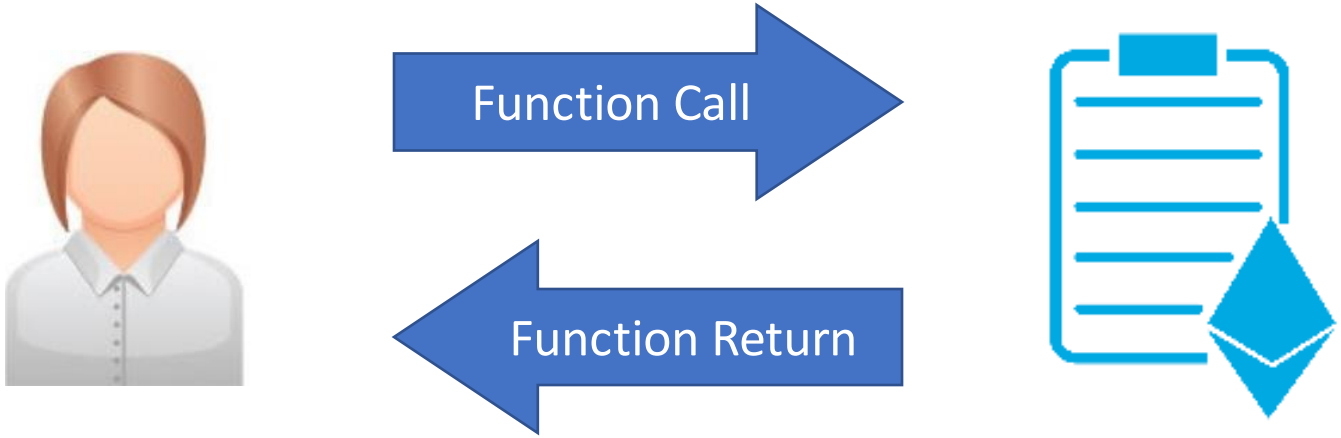
Ethereum Transactions

- EOA to EOA Message



Ethereum Transactions

- EOA to Contract & Vice-Versa



Gas

- Contract execution requires miners' computation and storage resource!
  - Require a mechanism to keep the system healthy
- Gas Mechanism
  - Caller pays fee for each transaction step
  - Each step during the execution has fixed "gas" fee
  - But gas price in Ether is up to the caller
  - Transaction fee = consumed gas * gas price
  - **Lower price** means low priority

  - The fee makes Denial of Service (DoS) attacks more expensive

Gas

- **Gas** is a unit to measure the amount of computational effort in the execution of an operation.
- Gas costs of different operations differ dramatically!
  - Storing a word to storage is the most expensive one
  - In memory operations are much cheaper

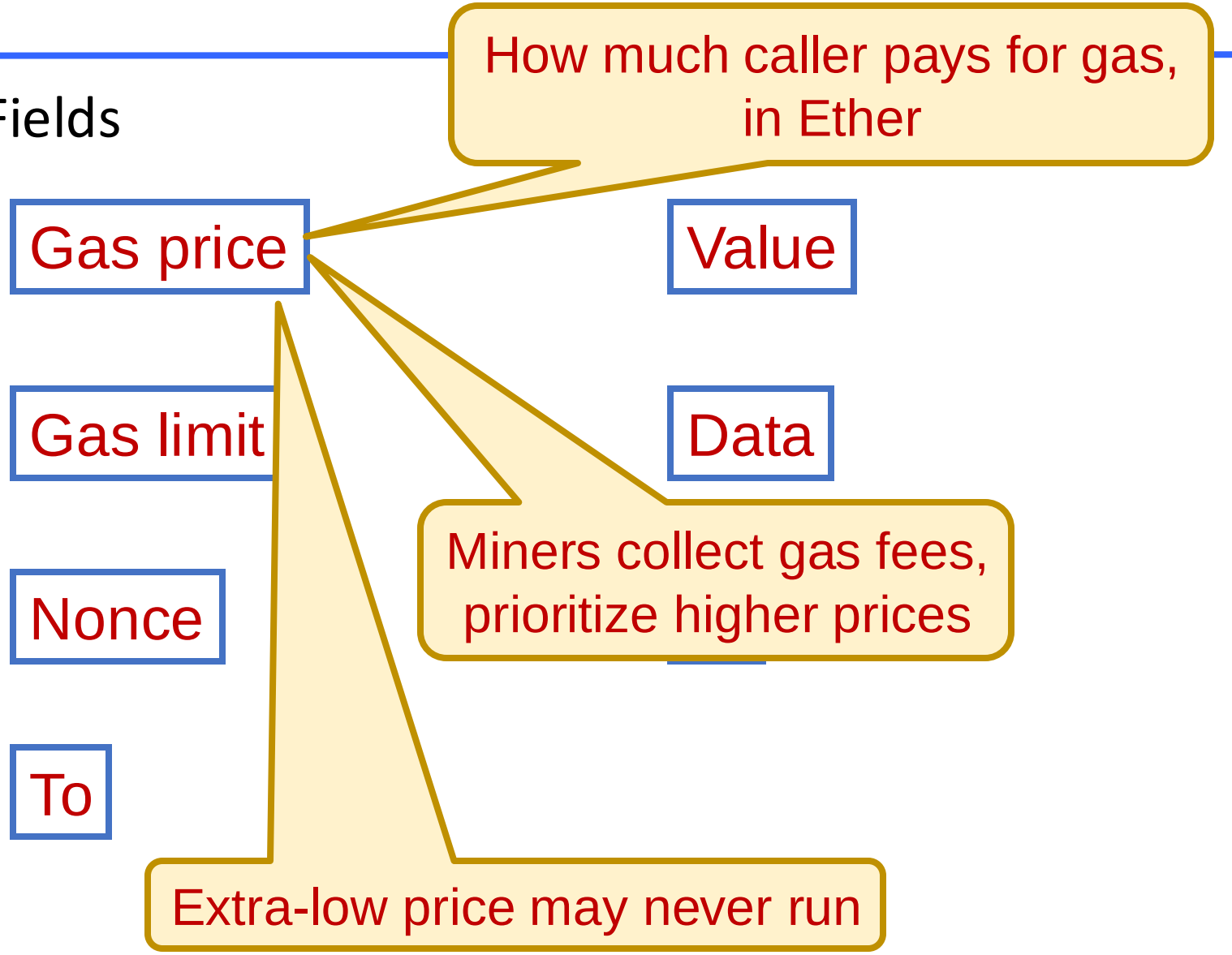
Operation	Gas Used	Explanation
C_{sload}	200	load a word from storage
C_{sstore}	20,000	save a word to storage
$C_{supdate}$	5,000	update a word to storage
C_{mem}	3	access a word in memory
C_{hash}	$30 + 6x$	hash a x-word message
C_{tx}	21,000	execute a transaction
C_{txdata}	68	Transact a byte of data

Gas

- If a call runs out of gas (Out of gas exception)
  - Effects will be discarded
  - Gas will not be refunded
- If a call has leftover gas
  - Unused gas refunded
- Block Gas Limit
  - Bitcoin has limit on block size (1MB)
  - Ethereum has limit on block gas (30 Million)

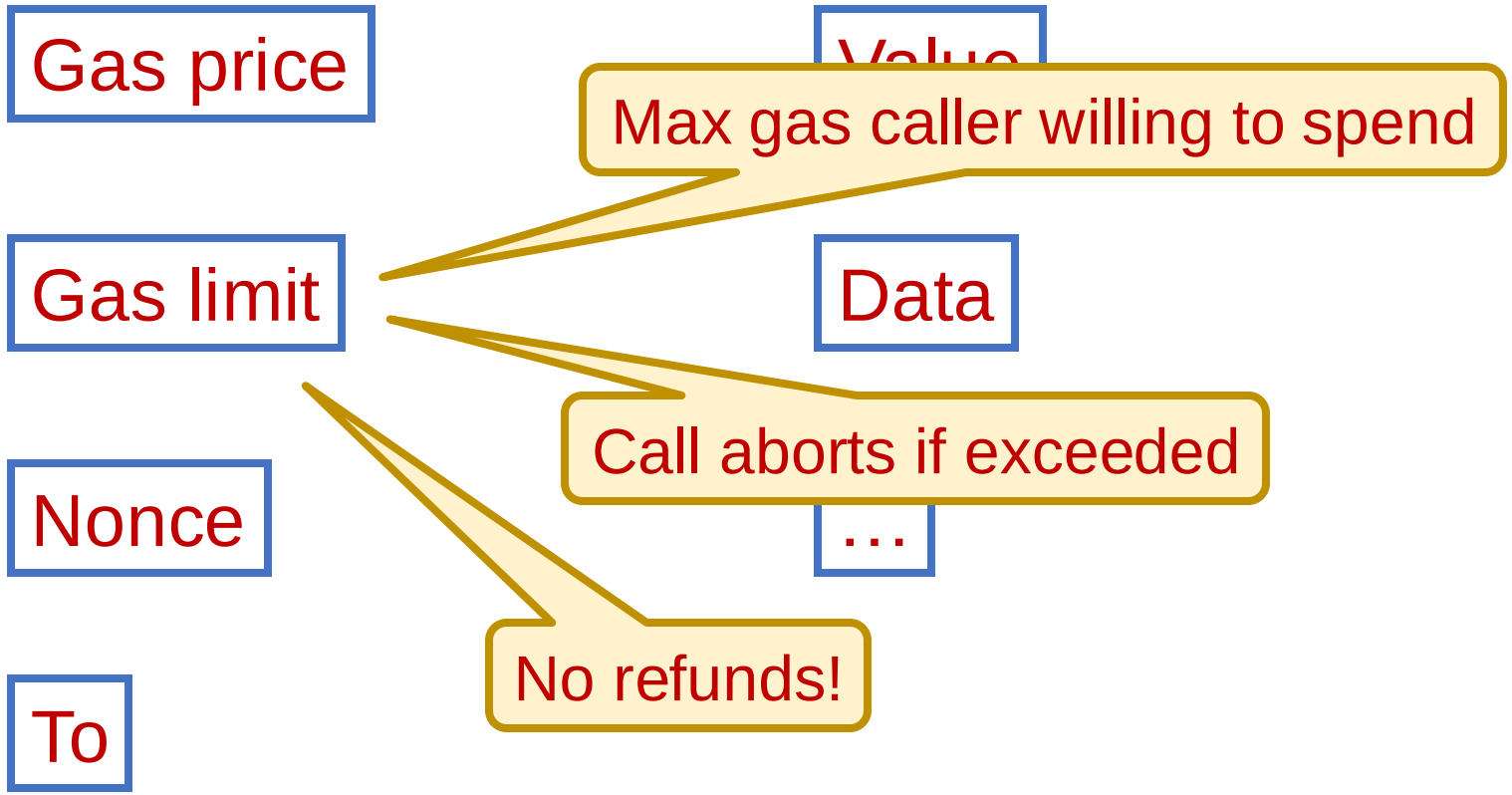
Ethereum Transaction

- Transaction Fields



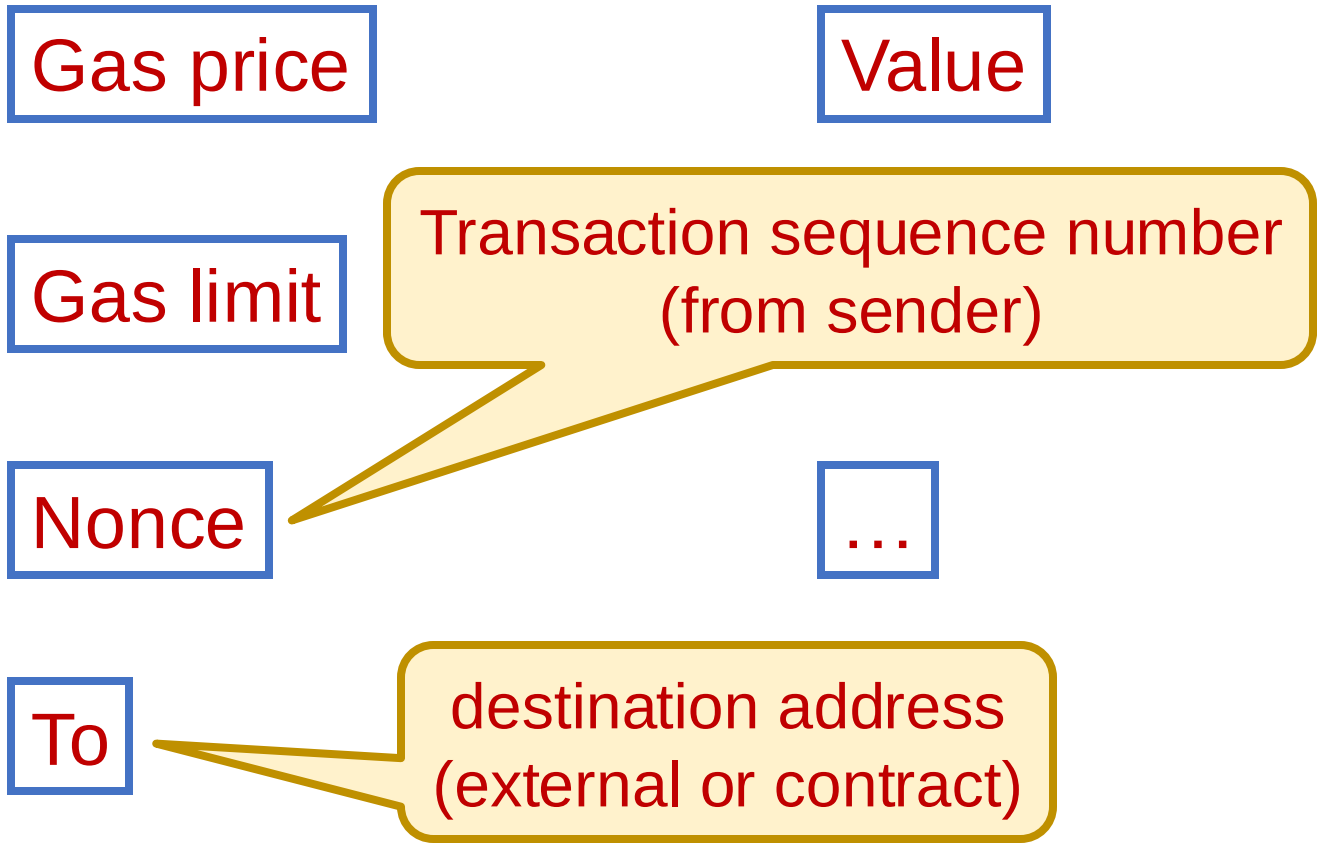
Ethereum Transaction

- Transaction Fields



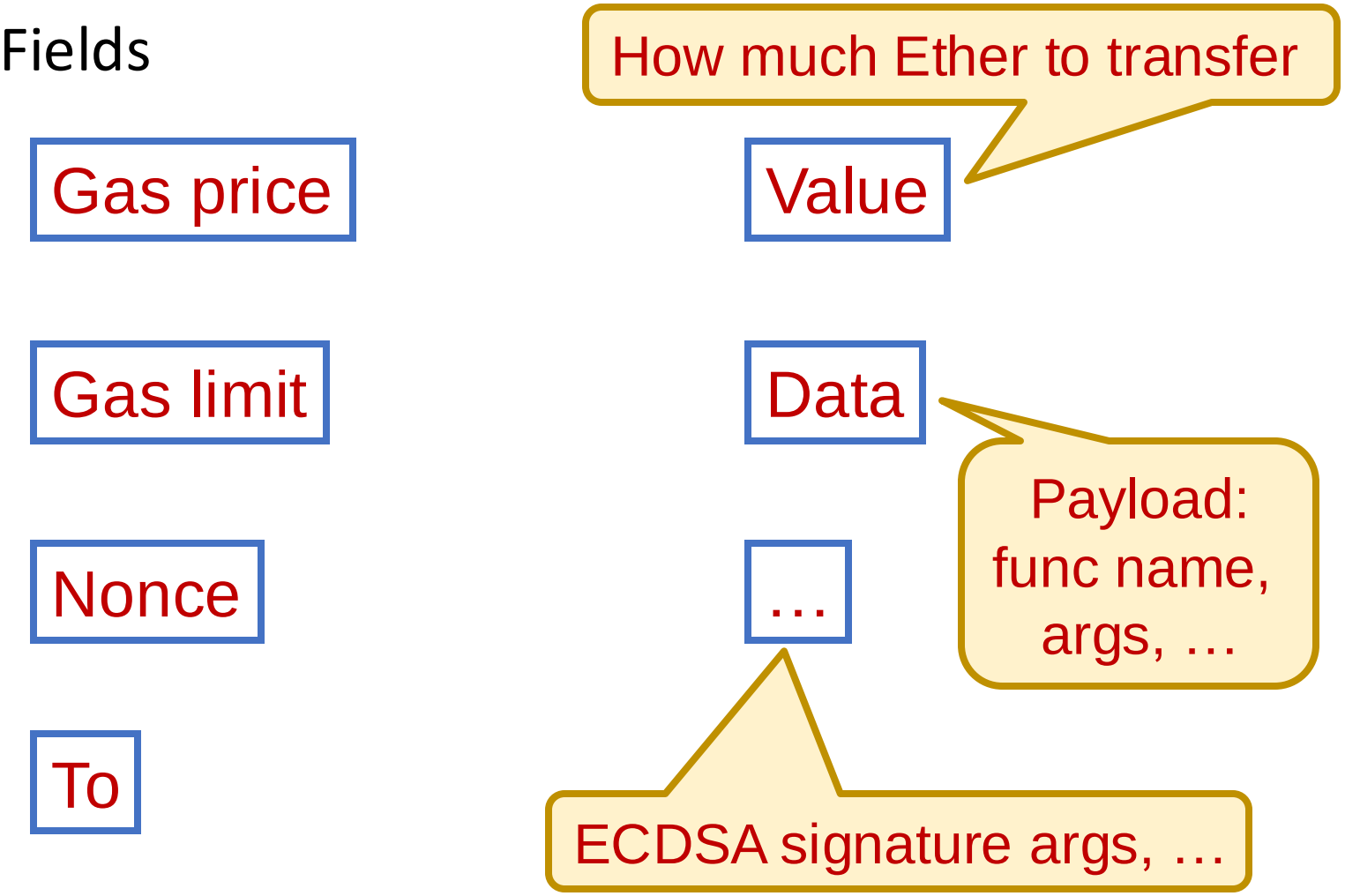
Ethereum Transaction

- Transaction Fields



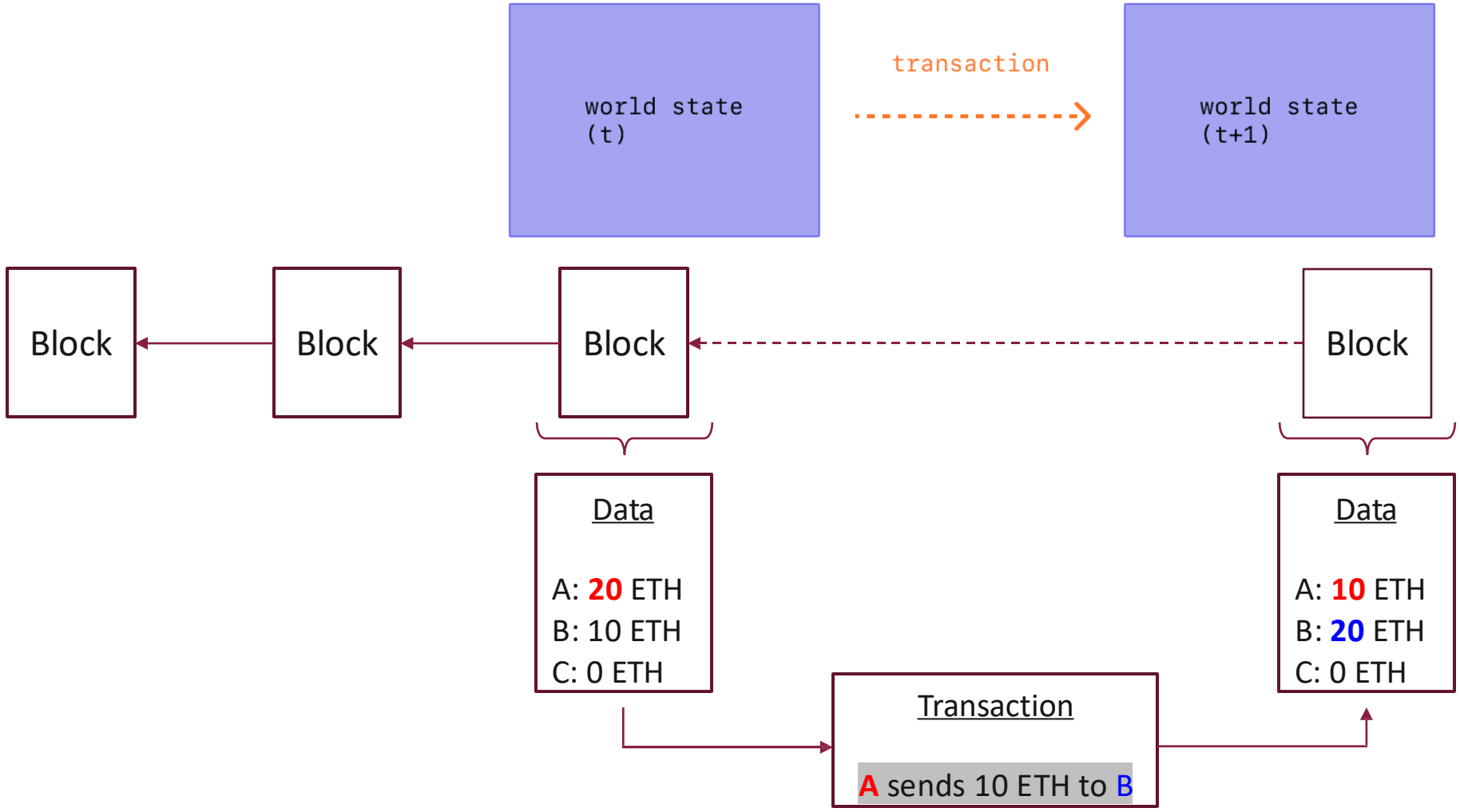
Ethereum Transaction

- Transaction Fields



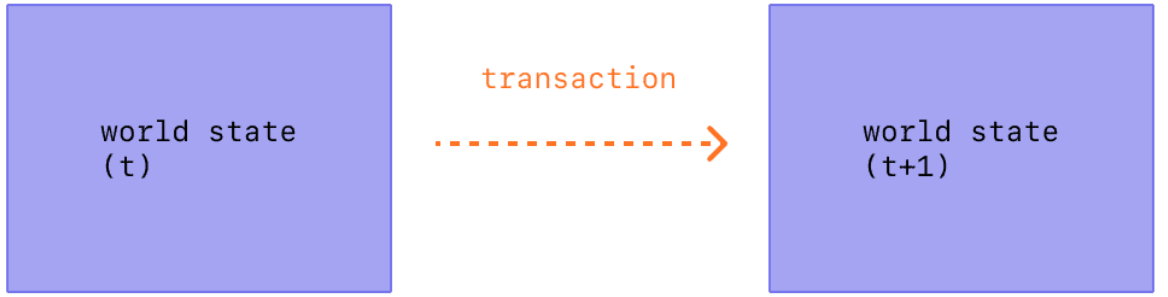
Ethereum Transaction

- Ethereum can be considered as a transaction-based state machine



Ethereum Transaction

- A request (initiated by EOA) to modify the state of the blockchain can run code (contracts) to change global world state
- Cryptographically signed by originating EOA
- Transaction Types
  - Send value from one account to another account
  - Create smart contract
  - Execute smart contract code



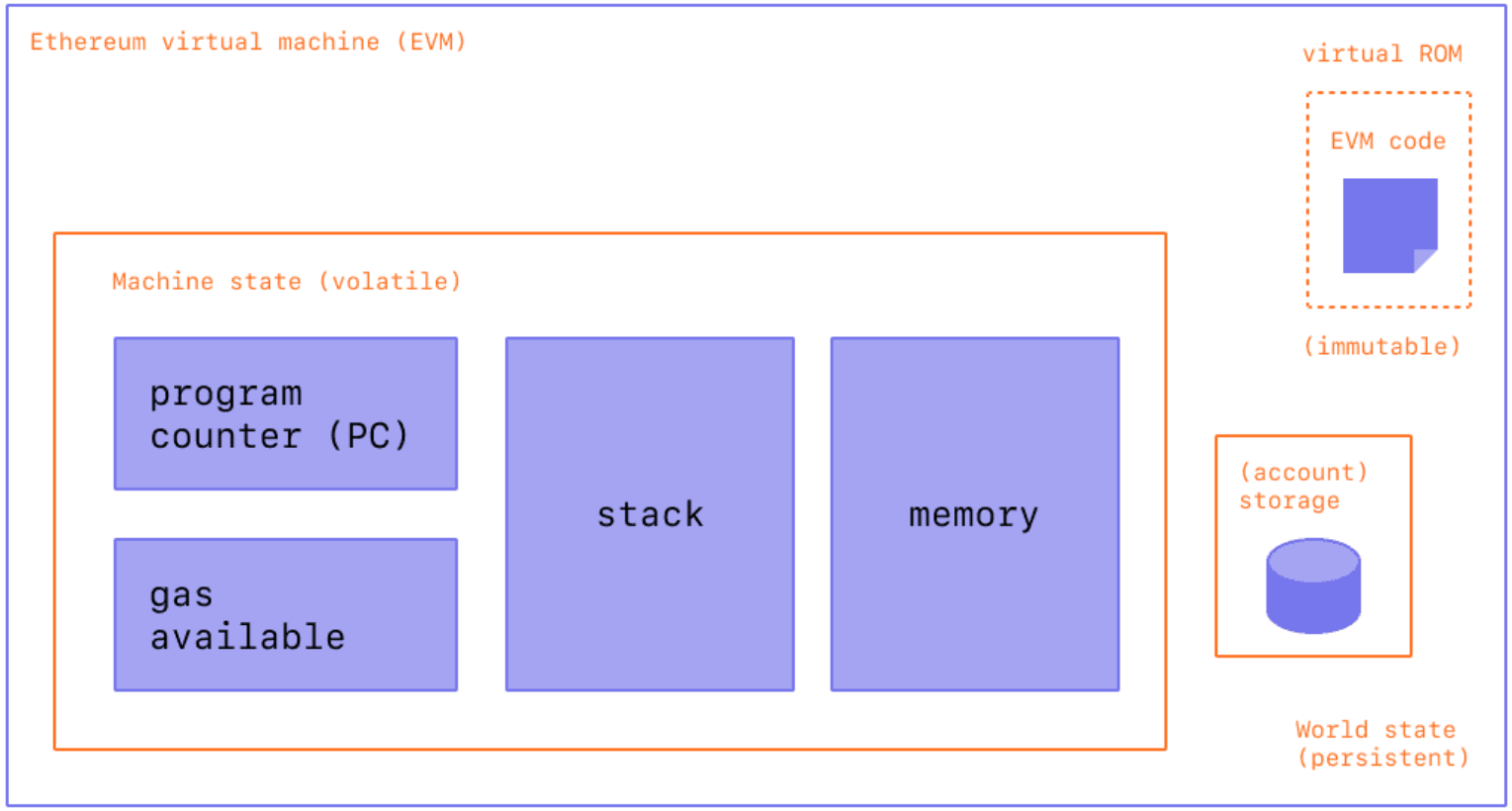
Ethereum Transaction

- A submitted transaction includes the following information
- Recipient: Receiving address
  - If EOA, will transfer value.
  - If contract account, will execute contract code
- Signature: Sender identifier
- Value: Amount of ETH to transfer from sender to recipient (in WEI)
- Data: optional field to include arbitrary data
- gasLimit: Maximum amount of gas units consumed by transaction where Units of gas represent computational steps
- gasPrice: The fee sender pays per unit of gas

```
{
    from:
    "0xEA674fdDe714fd979de3EdF0F56AA9716B898ec8",
    to:
    "0xac03bb73b6a9e108530aff4df5077c2b3d481e5a",
    gasLimit: "21000",
    gasPrice: "200",
    nonce: "0",
    value: "10000000000",
}
```

Ethereum EVM

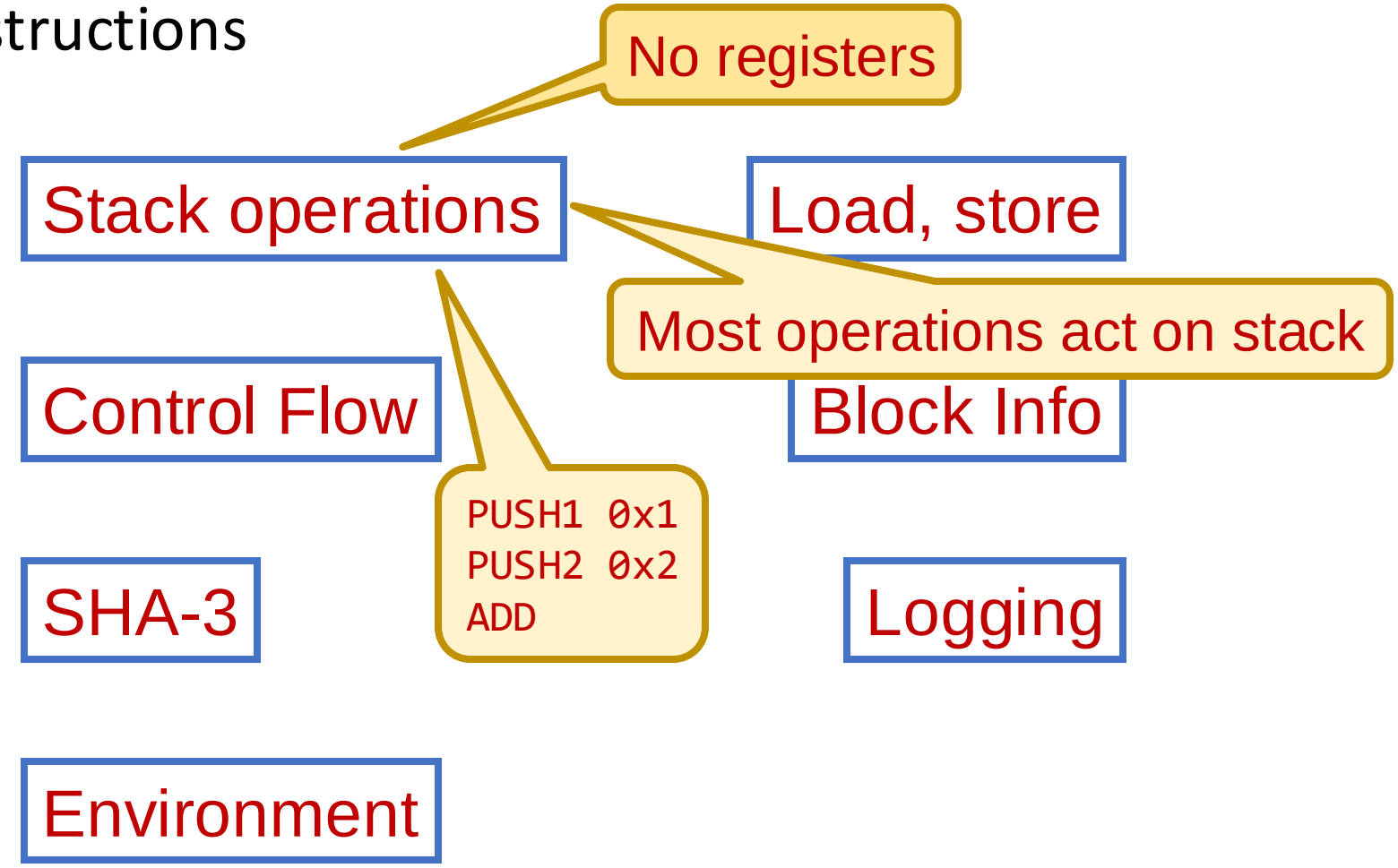
- Ethereum Virtual Machine



https://ethereum.org/en/developers/docs/evm/

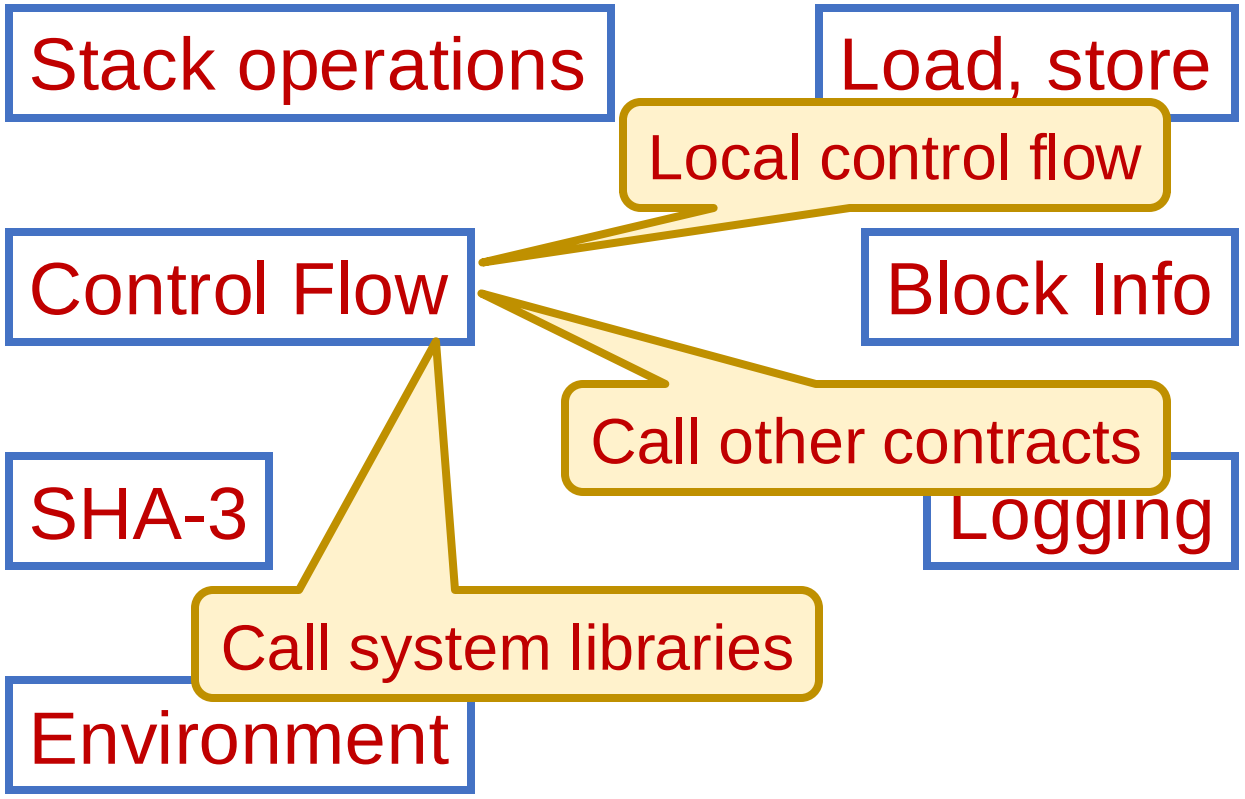
Ethereum EVM

- Types of Instructions



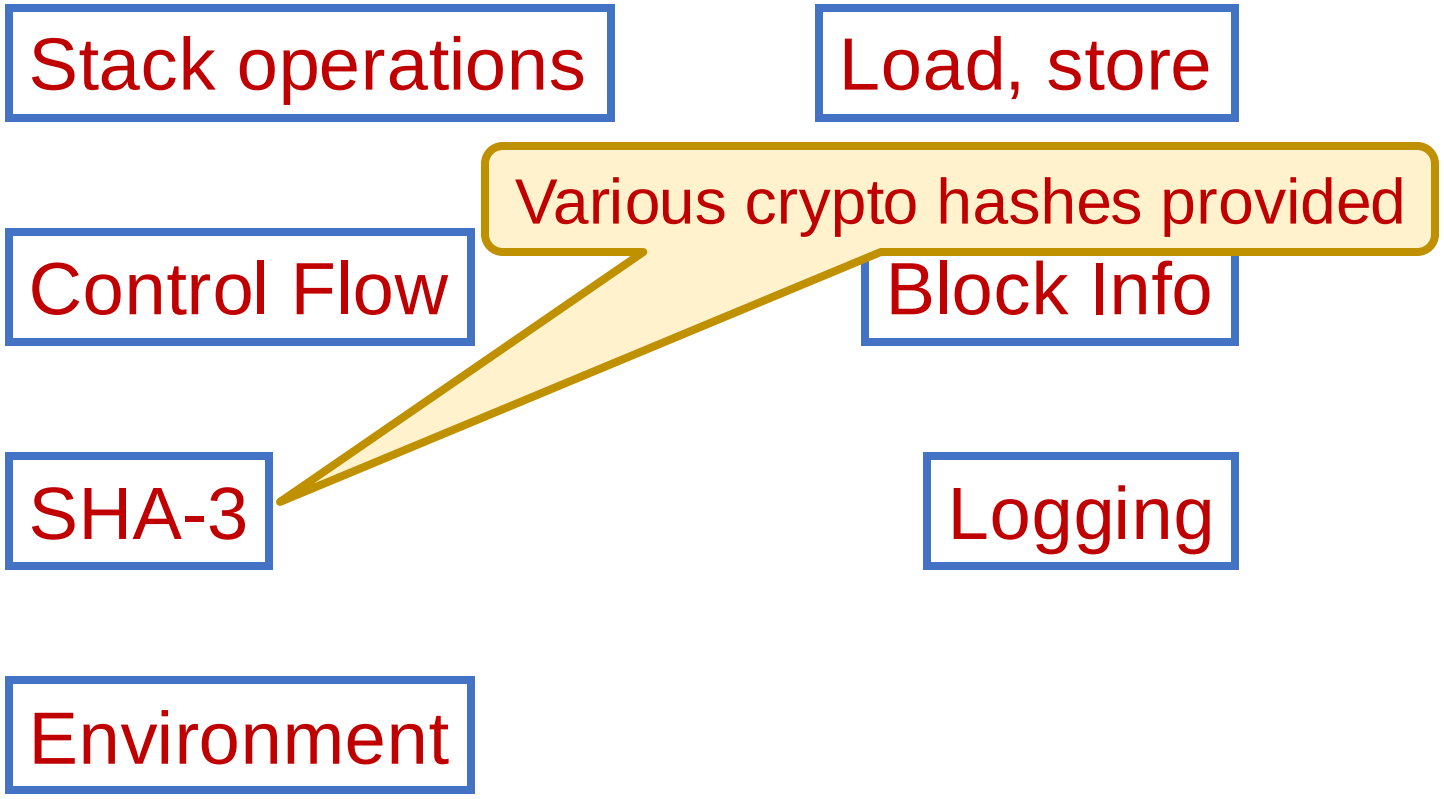
Ethereum EVM

- Types of Instructions



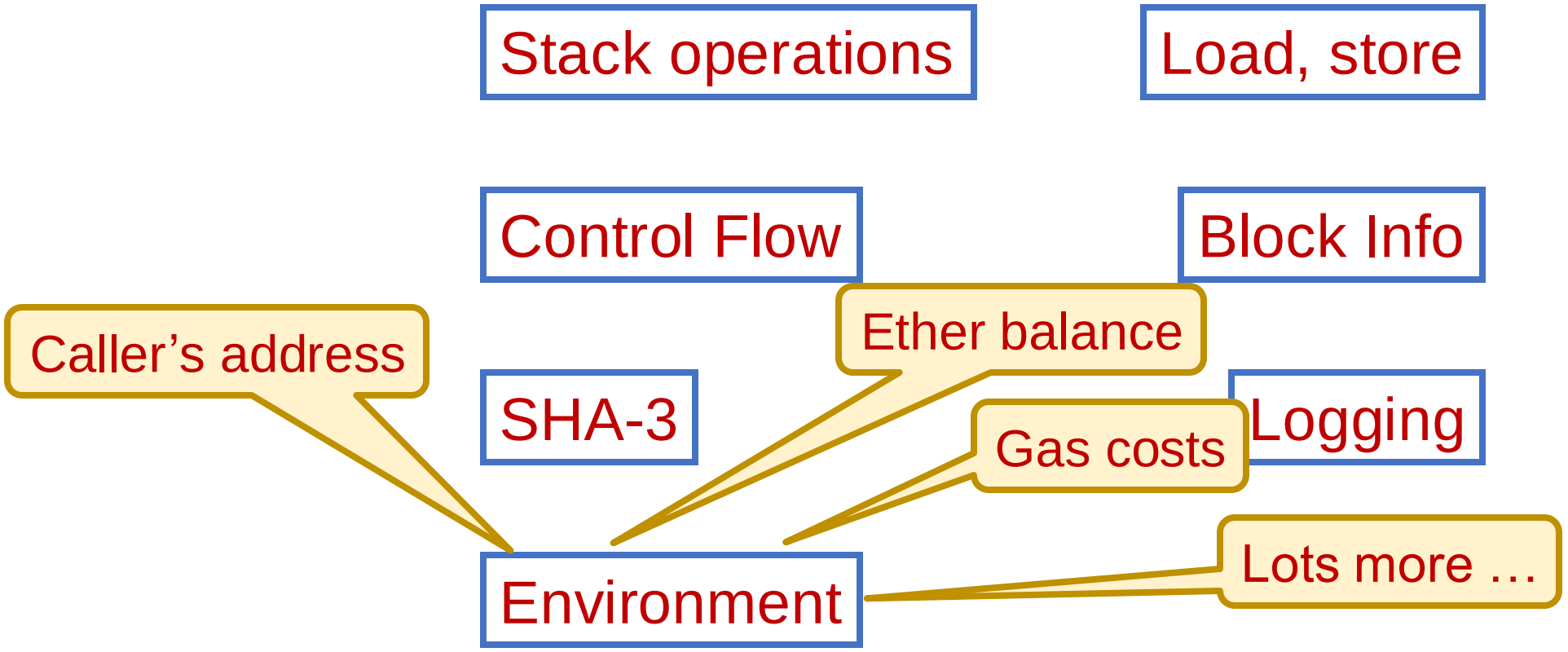
Ethereum EVM

- Types of Instructions



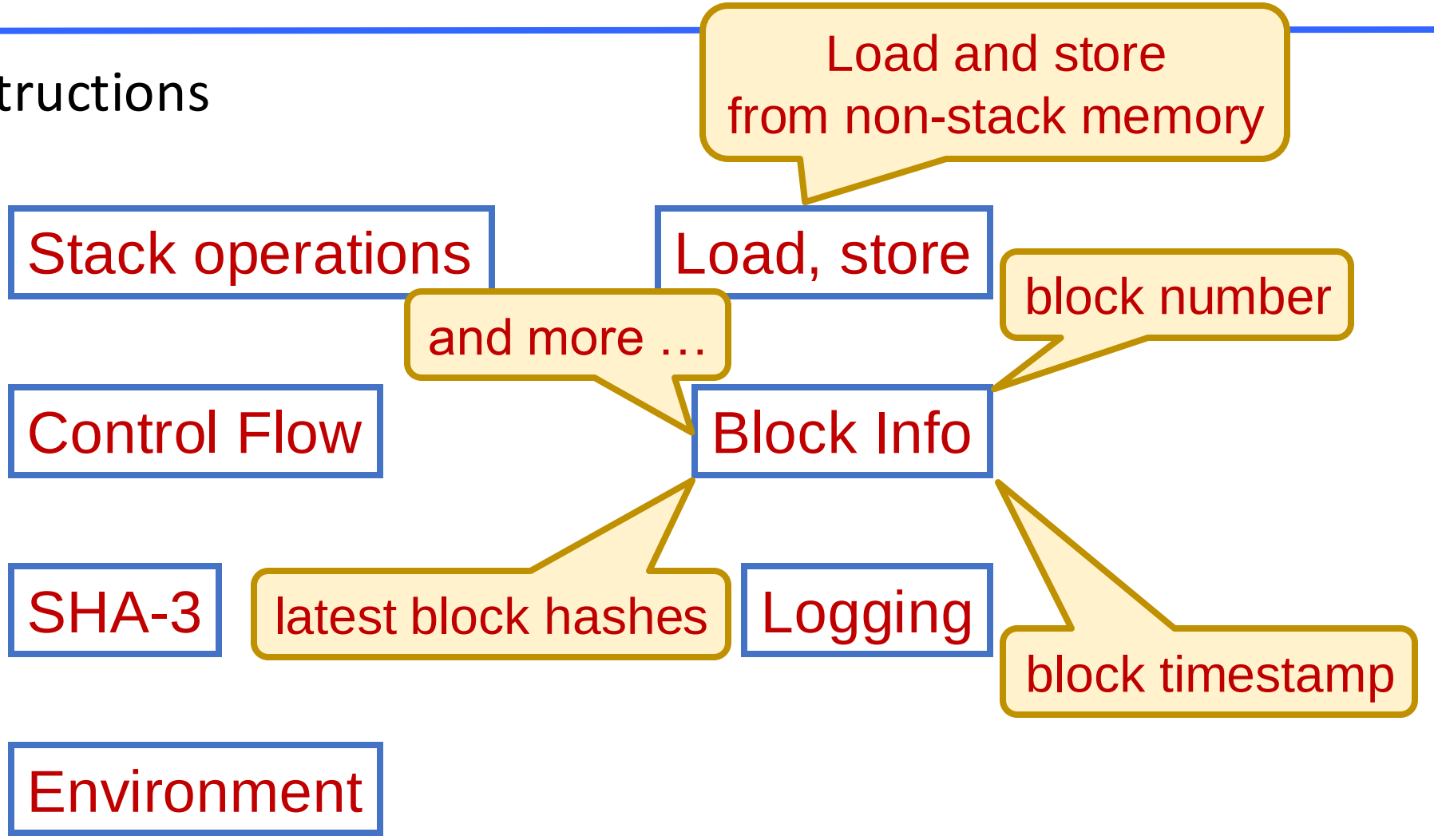
Ethereum EVM

- Types of Instructions



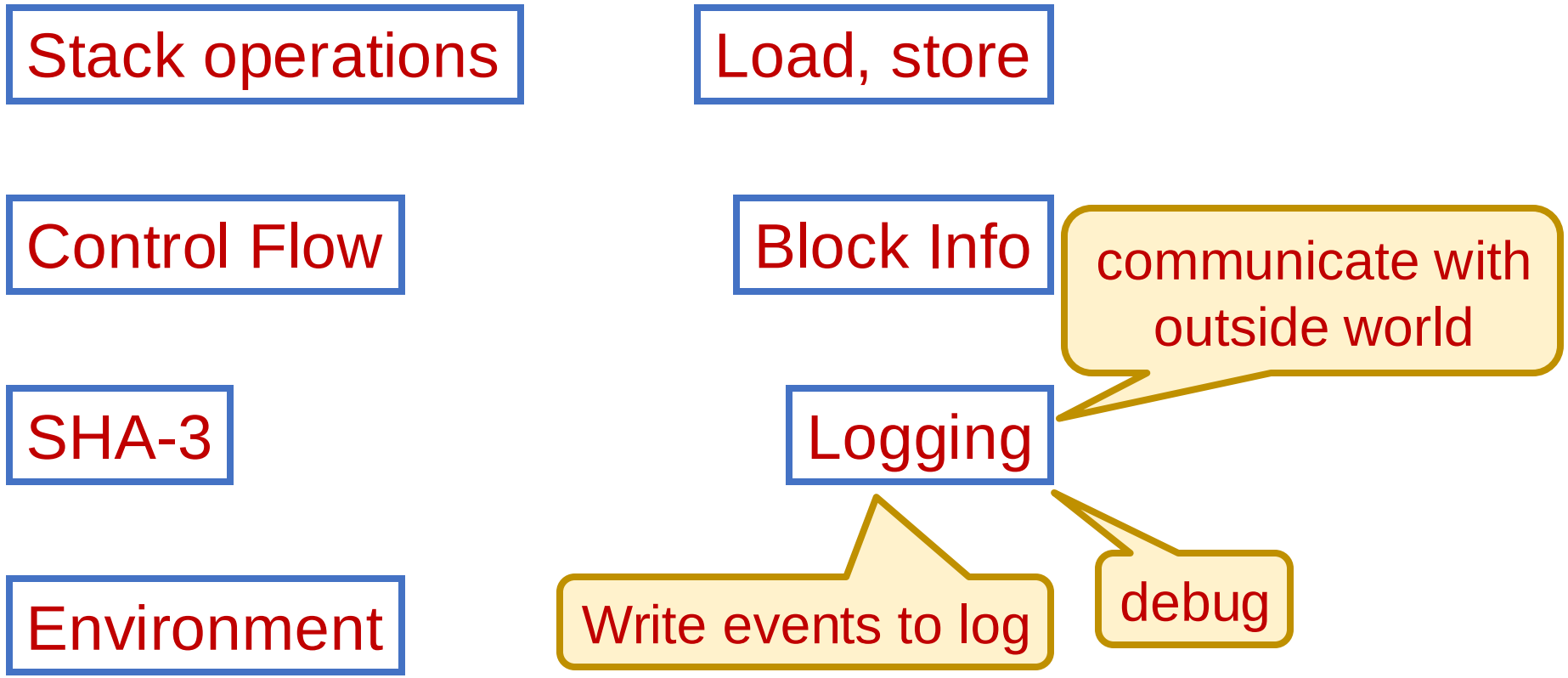
Ethereum EVM

- Types of Instructions



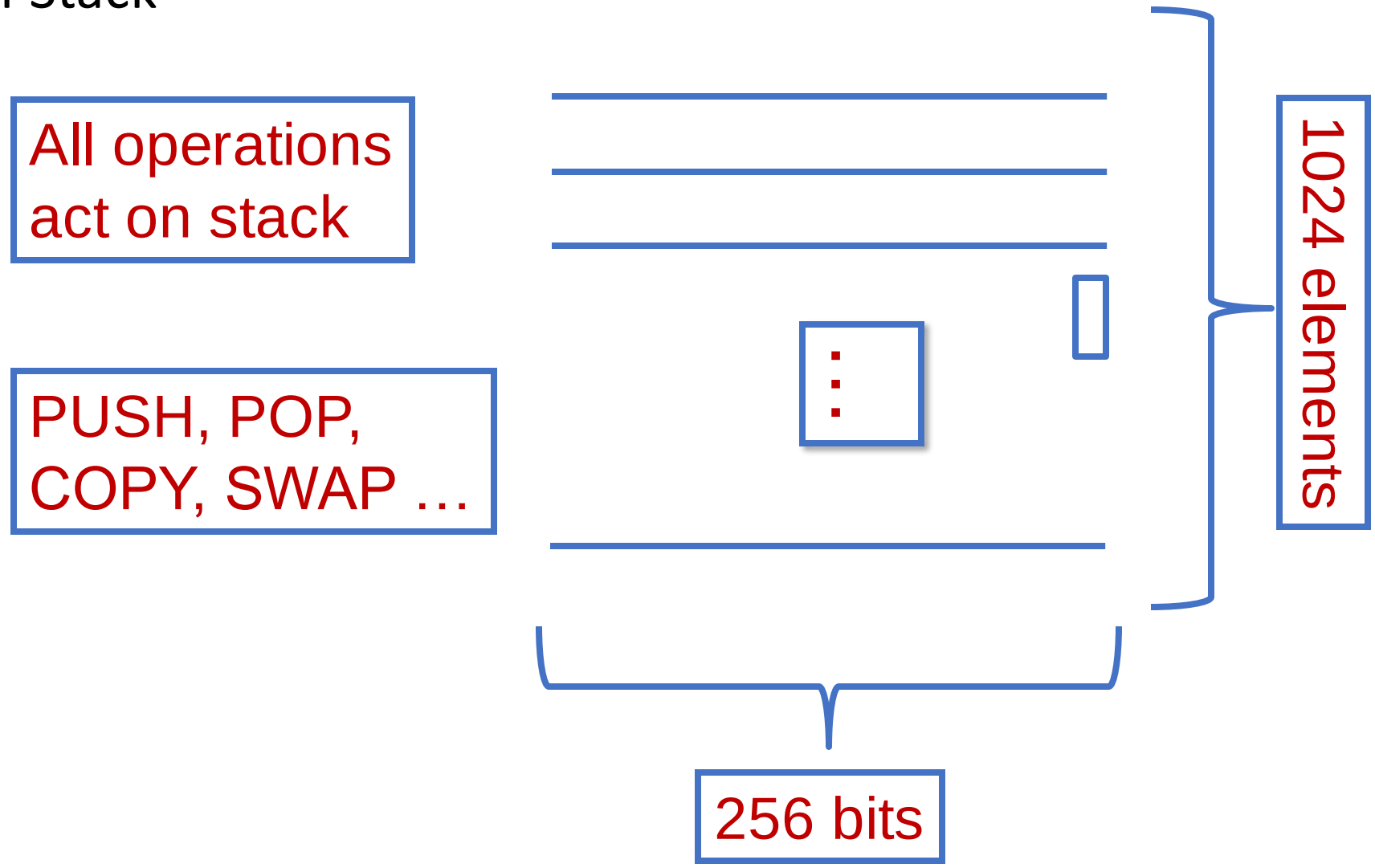
Ethereum EVM

- Types of Instructions

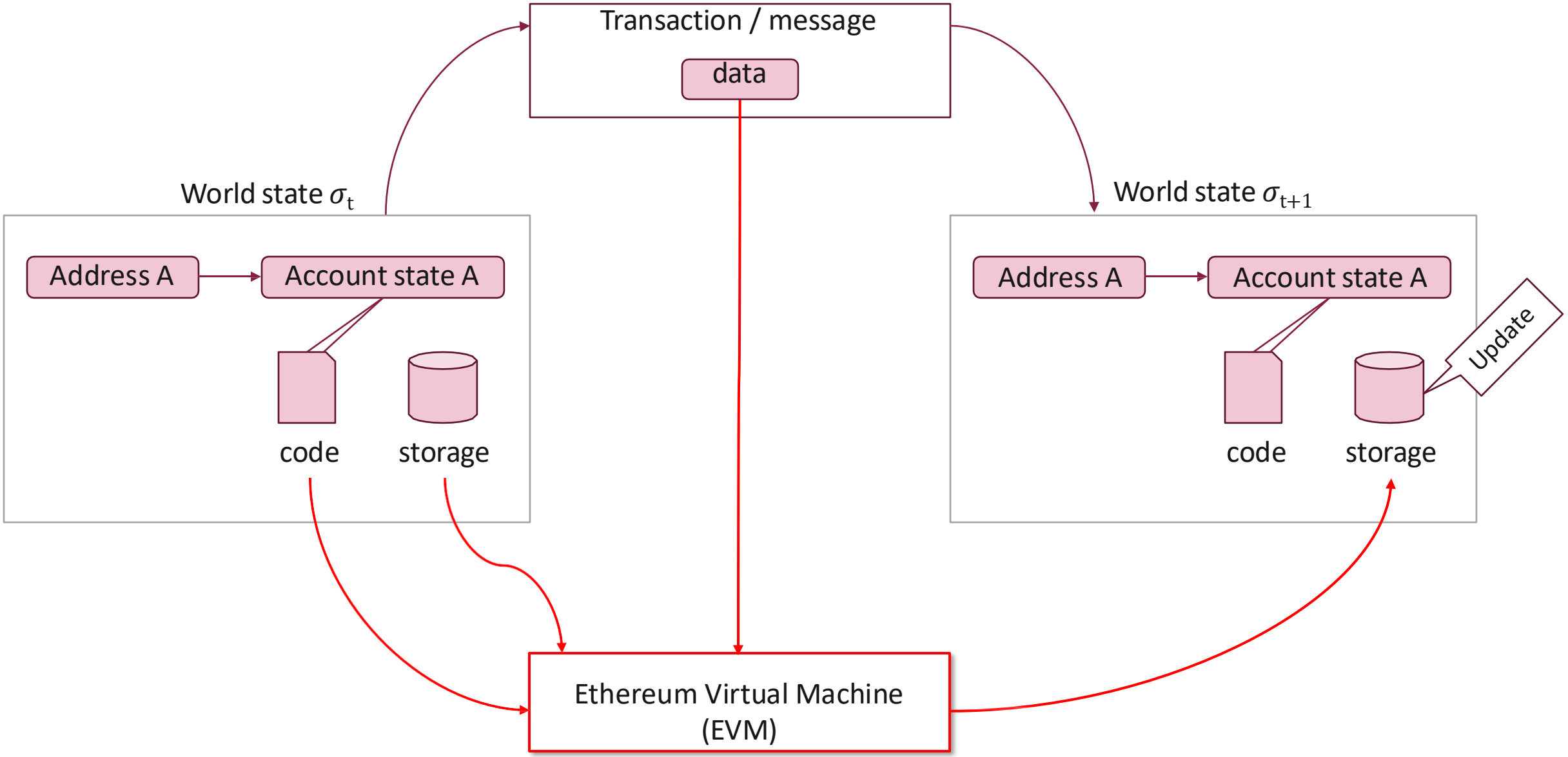


Ethereum EVM

- EVM Stack

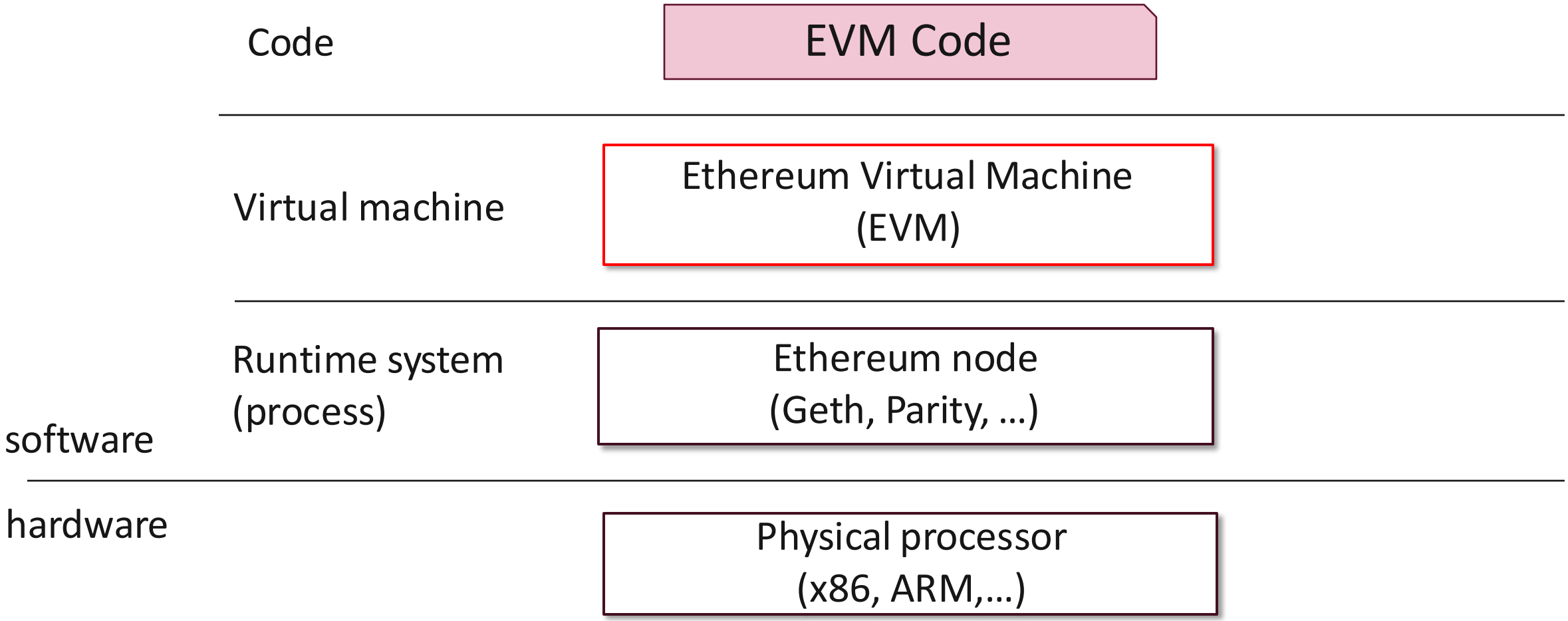


Ethereum Virtual Machine

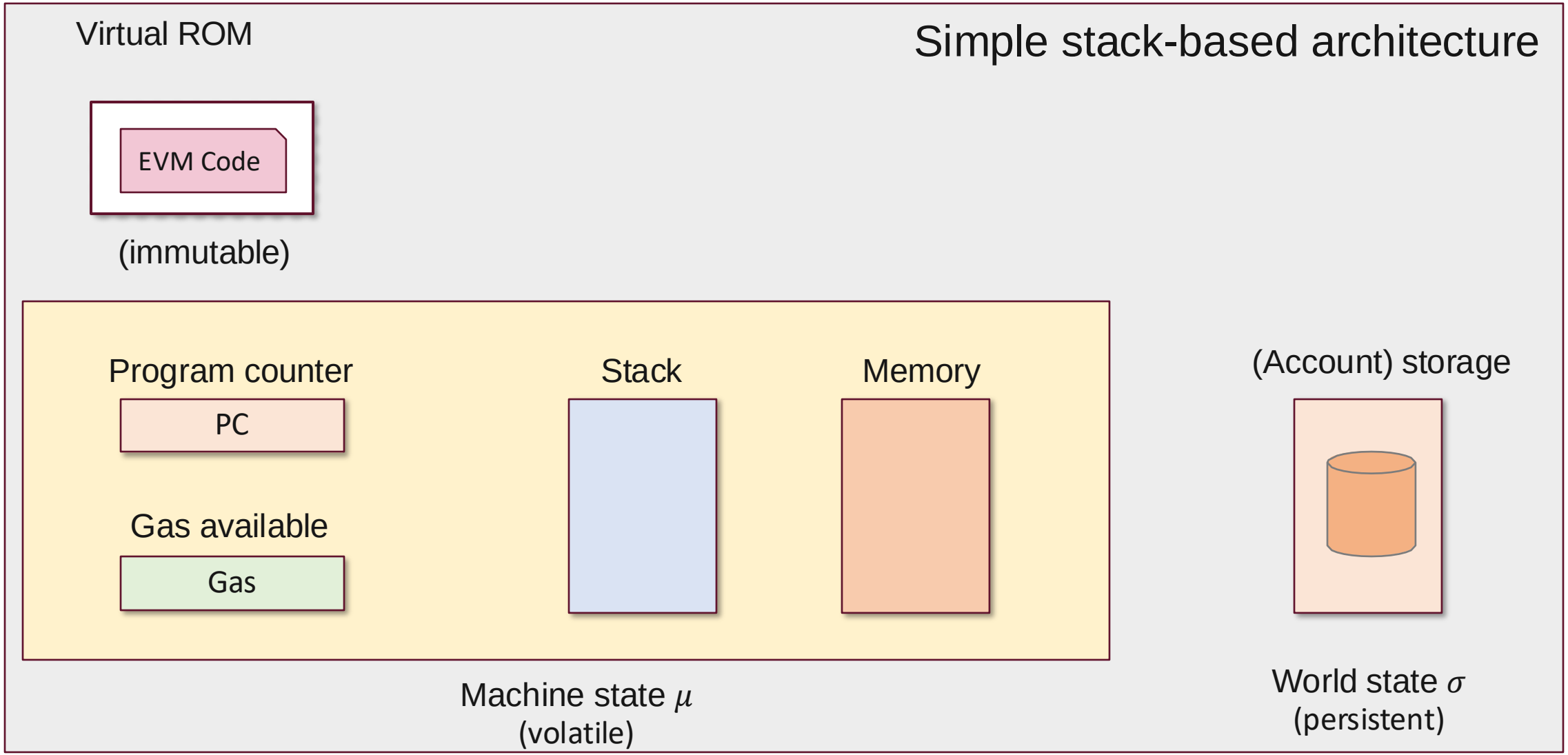


Ethereum Virtual Machine

- EVM code is executed on EVM
- EVM is the runtime environment for smart contracts in Ethereum

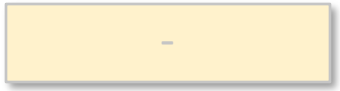


EVM Architecture

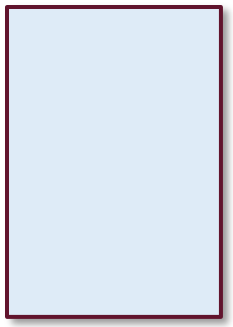


Machine space of EVM

Registers



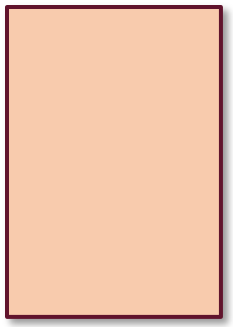
Stack



stack memory

256 bits x 1024 elements

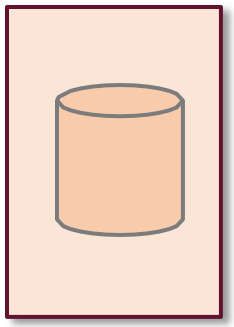
Memory



volatile memory

byte addressing
linear memory

(Account) storage

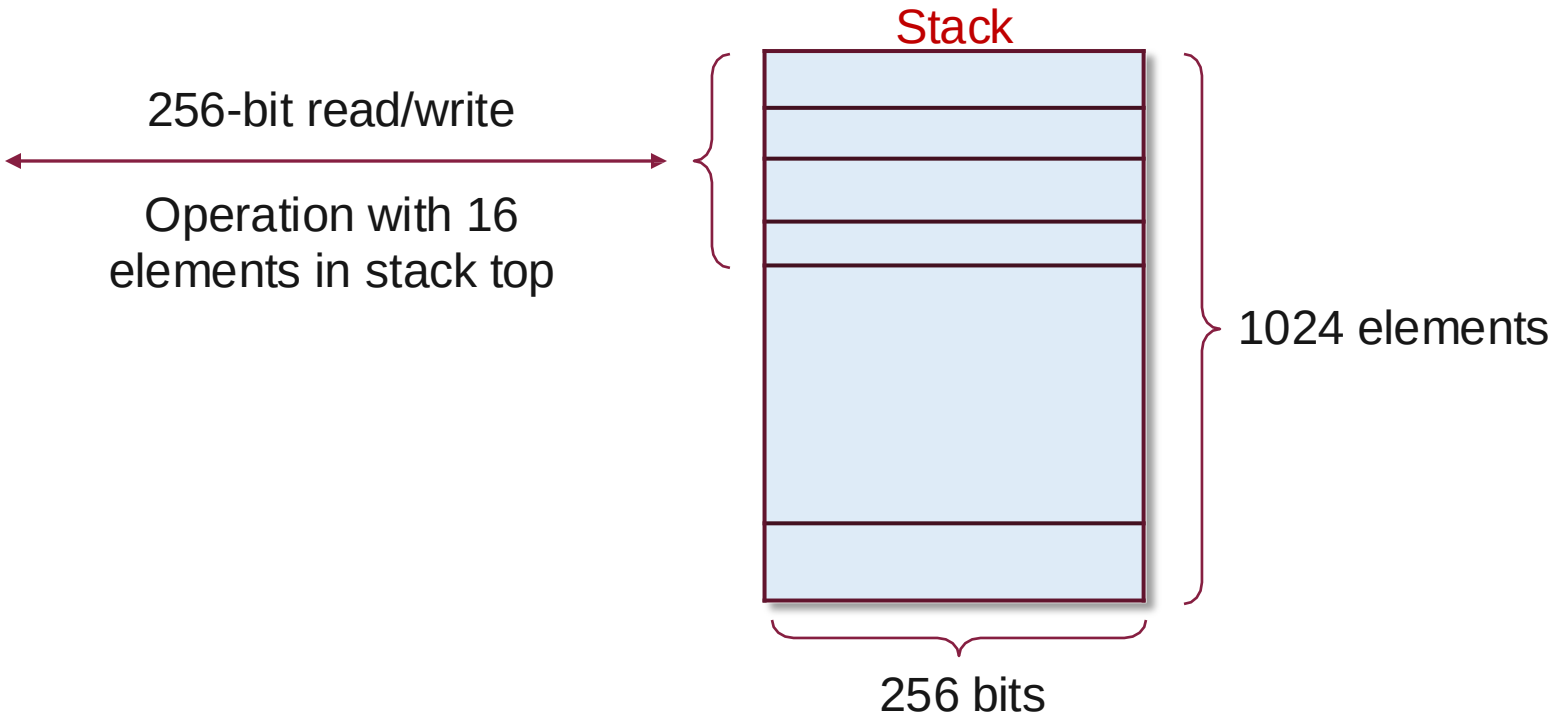


Persistent memory

256 bits – 256 bits
key-value store

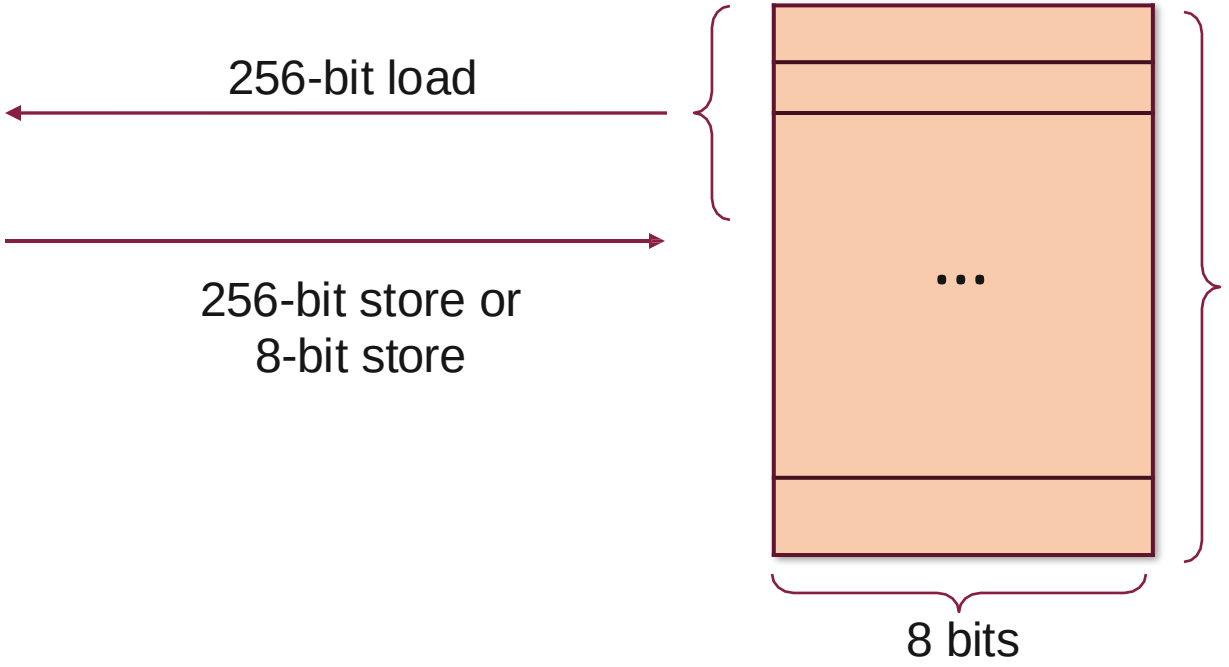
Machine space of EVM

- All operations performed on stack
- Access with stack instructions such as PUSH/POP/COPY/SWAP/JUMP
- Max stack depth = 1024
- Program aborts if stack size exceeded; miner keeps gas



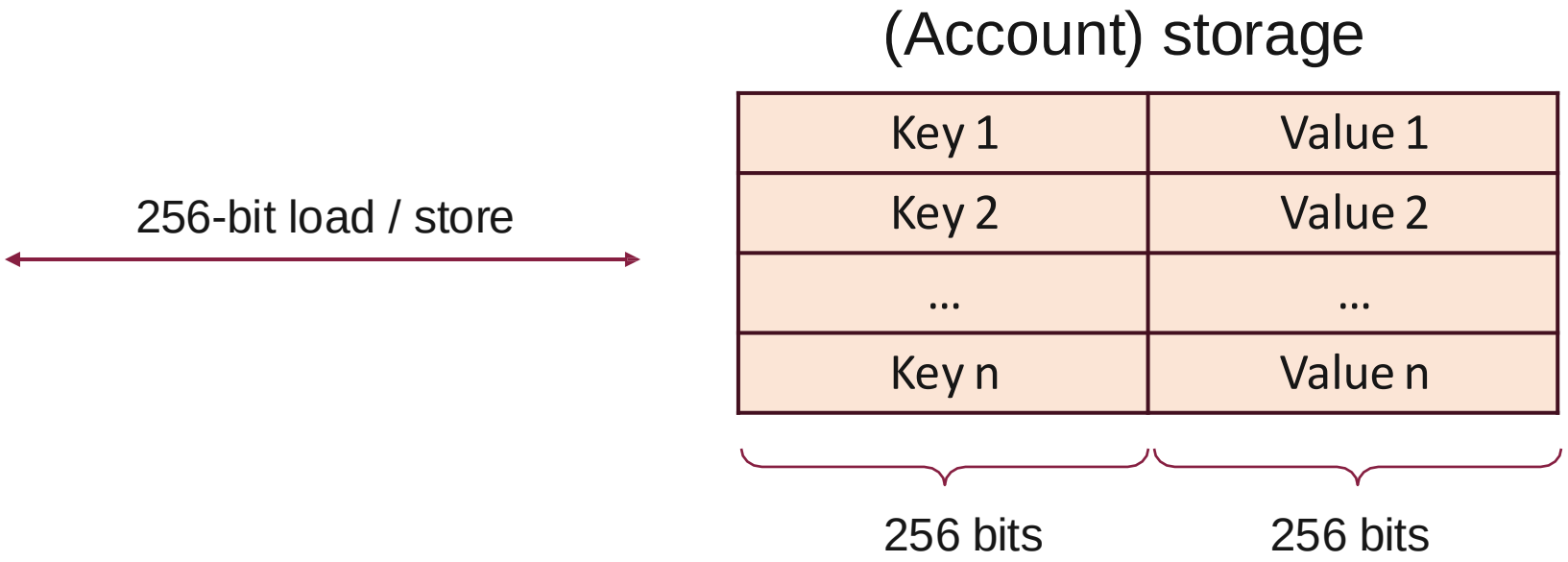
EVM Memory

- Linear memory Byte-level access
- Access with MSTORE/MSTORE8/MLOAD instructions
- All locations in memory are well-defined initially as zero



EVM Account Storage

- Storage is key-value store mapping 256-bit words to 256-bit words
- Access with SSTORE/SLOAD instructions
- All locations in storage are well-defined initially as zero



EVM Code

Assembly view

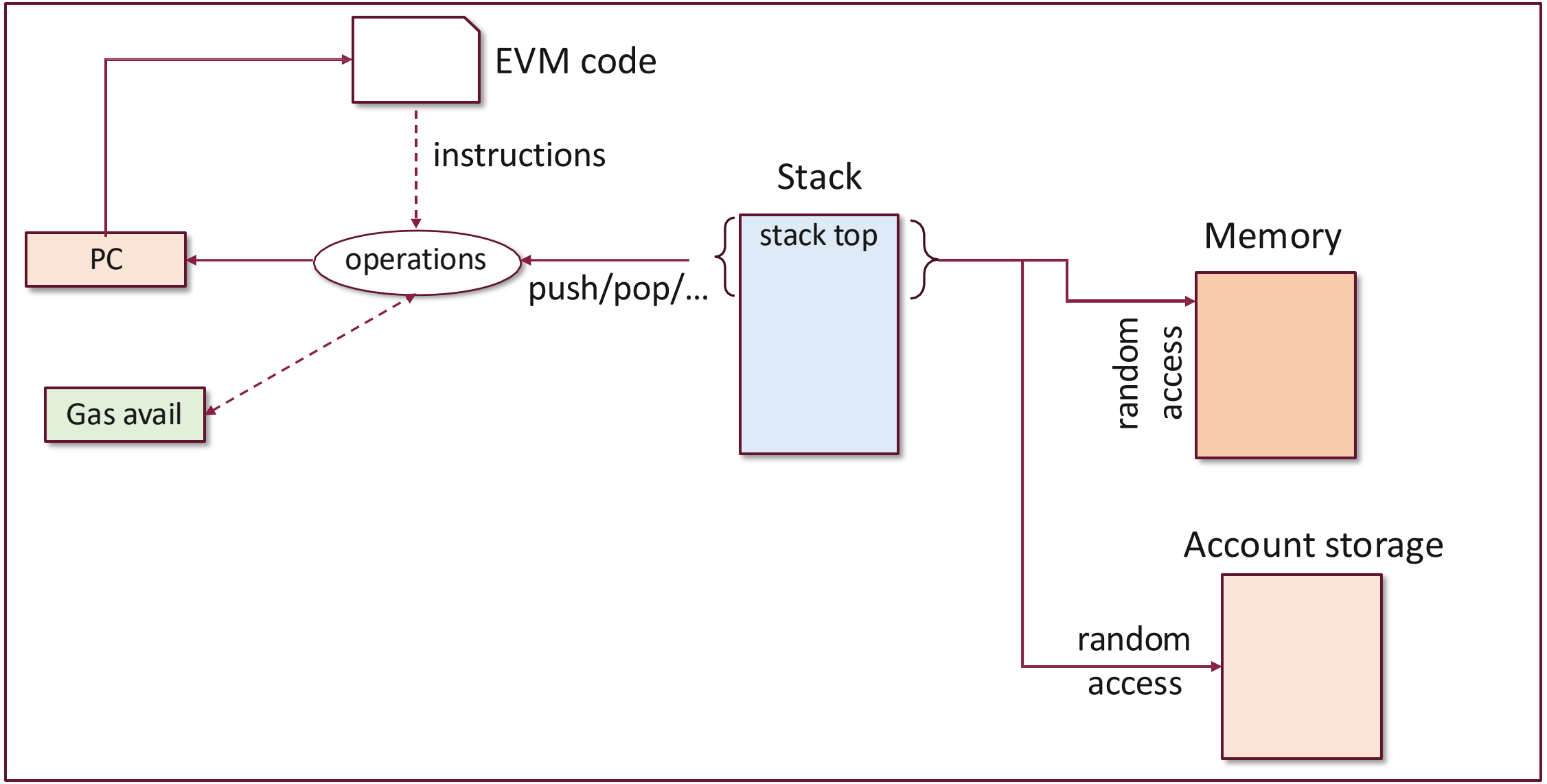
```
PUSH1 e0
PUSH1 02
  EXP
PUSH1 00
CALLDATALOAD
  …
```

Bytecode view

```
0x60e060020a600035
```

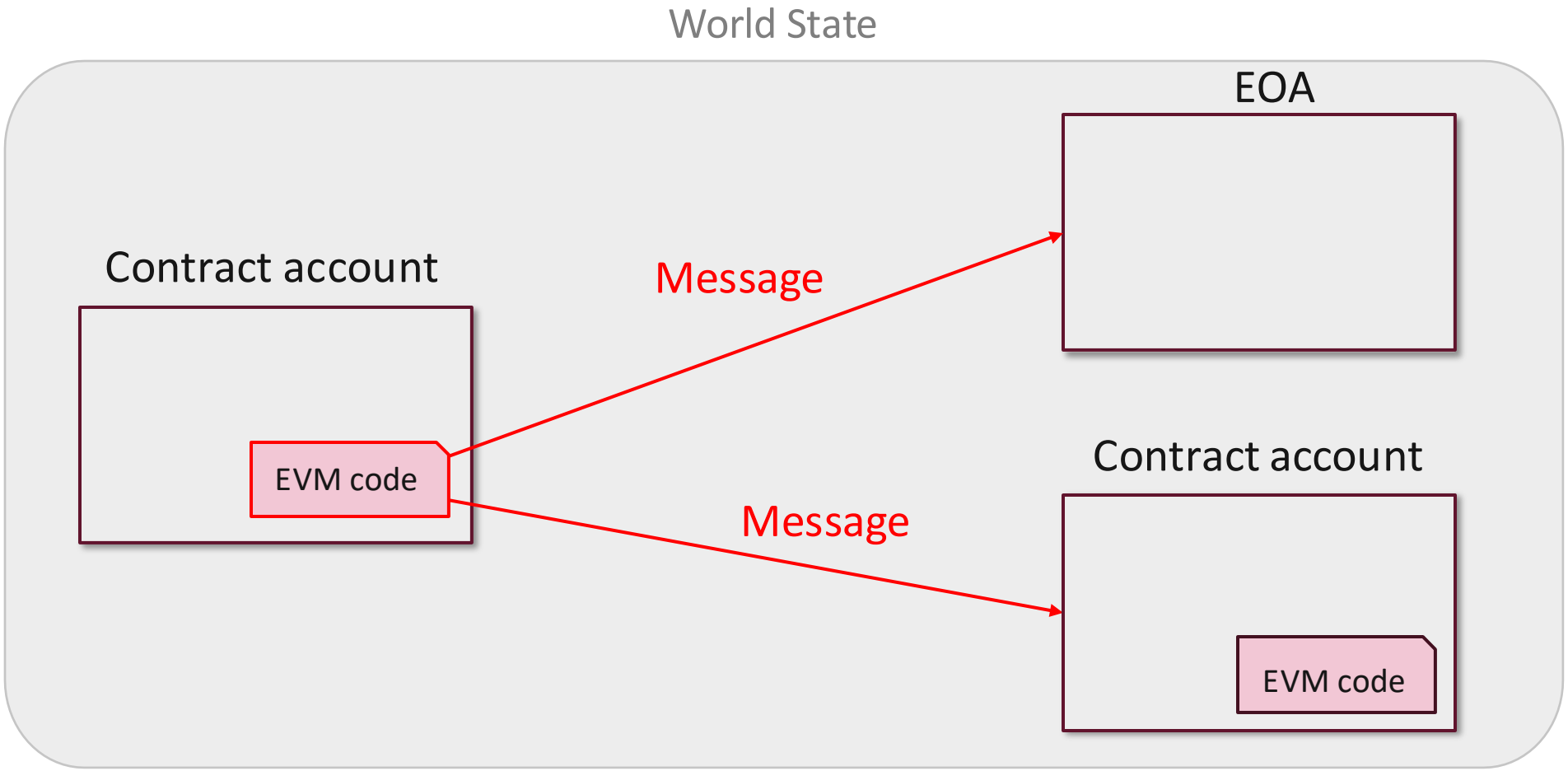
EVM Code is the bytecode that the EVM can natively execute

EVM Execution Model



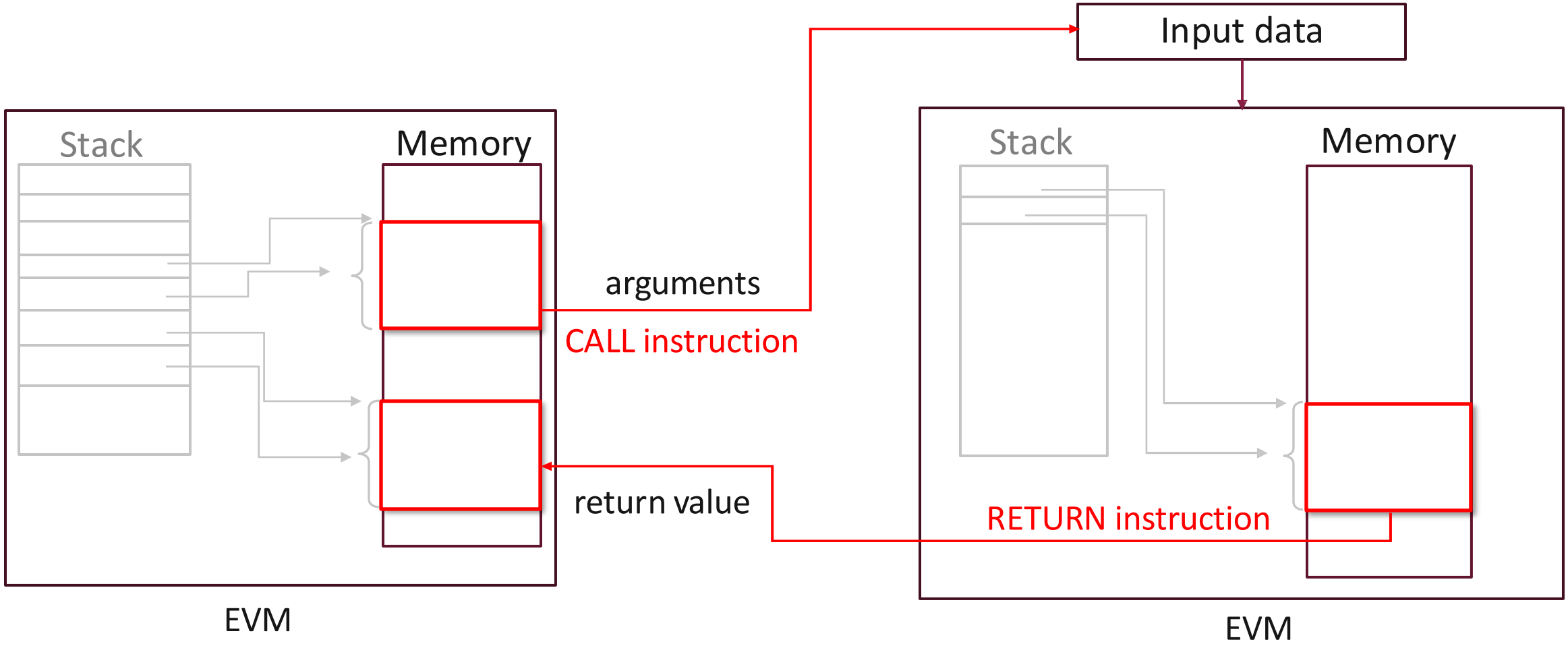
EVM Message Call

- EVM can send a message to other account
- The depth of message call is limited to less than 1024 levels



EVM Message Call Instructions

- Message call triggered by CALL instruction
- Arguments and return values are passed using memory



Outline

- Permissionless Blockchain
- Ethereum
  - Account-based Model
  - External Account and Contract Account
  - Gas
  - Transactions
  - Ethereum Virtual Machine
  - Smart Contract (Solidity)
- Coins and Tokens

Smart contracts

A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitrations and enforcement costs, and other transaction costs.

-Nick Szabo “The Idea of Smart Contracts”
1994

Digitalcurrency is just one application on top of a blockchain

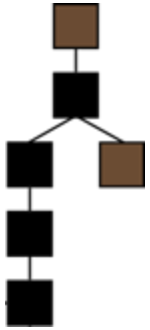
Users

Money

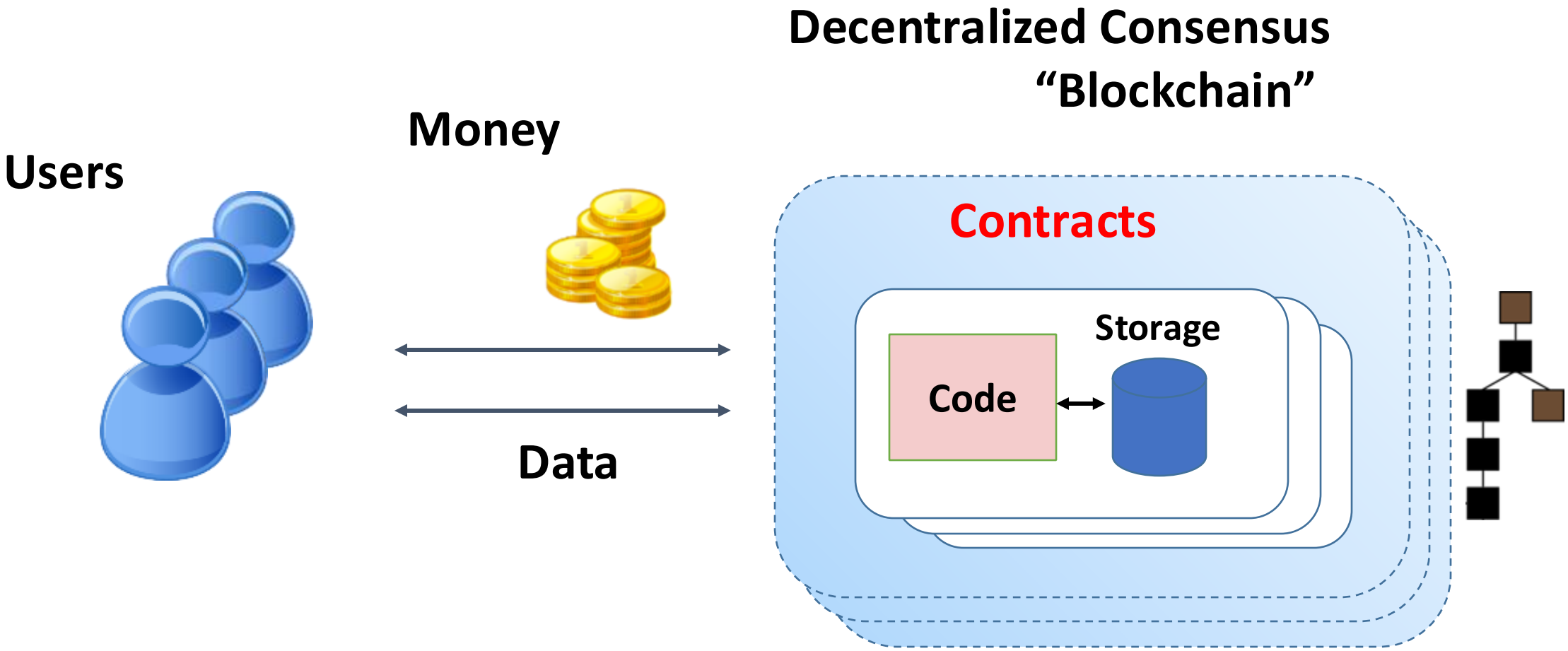
Decentralized Consensus
"Blockchain"

Account Balances

Alice: ฿10
Bob: ฿15
Carol: ฿120



Smart Contracts: user-defined programs running on top of a blockchain



A “dumb contract” example

Alice will reveal to Bob a value x such that SHA-256(x) = 0x2a...

In exchange, Bob will give Alice $10 in cash.

If Alice does not give Bob by July 1, 2018, then she will pay a penalty of US$1 per day that she is late, up to US$100.

Signed: Alice Bob

Traditional contracts vs. smart contracts

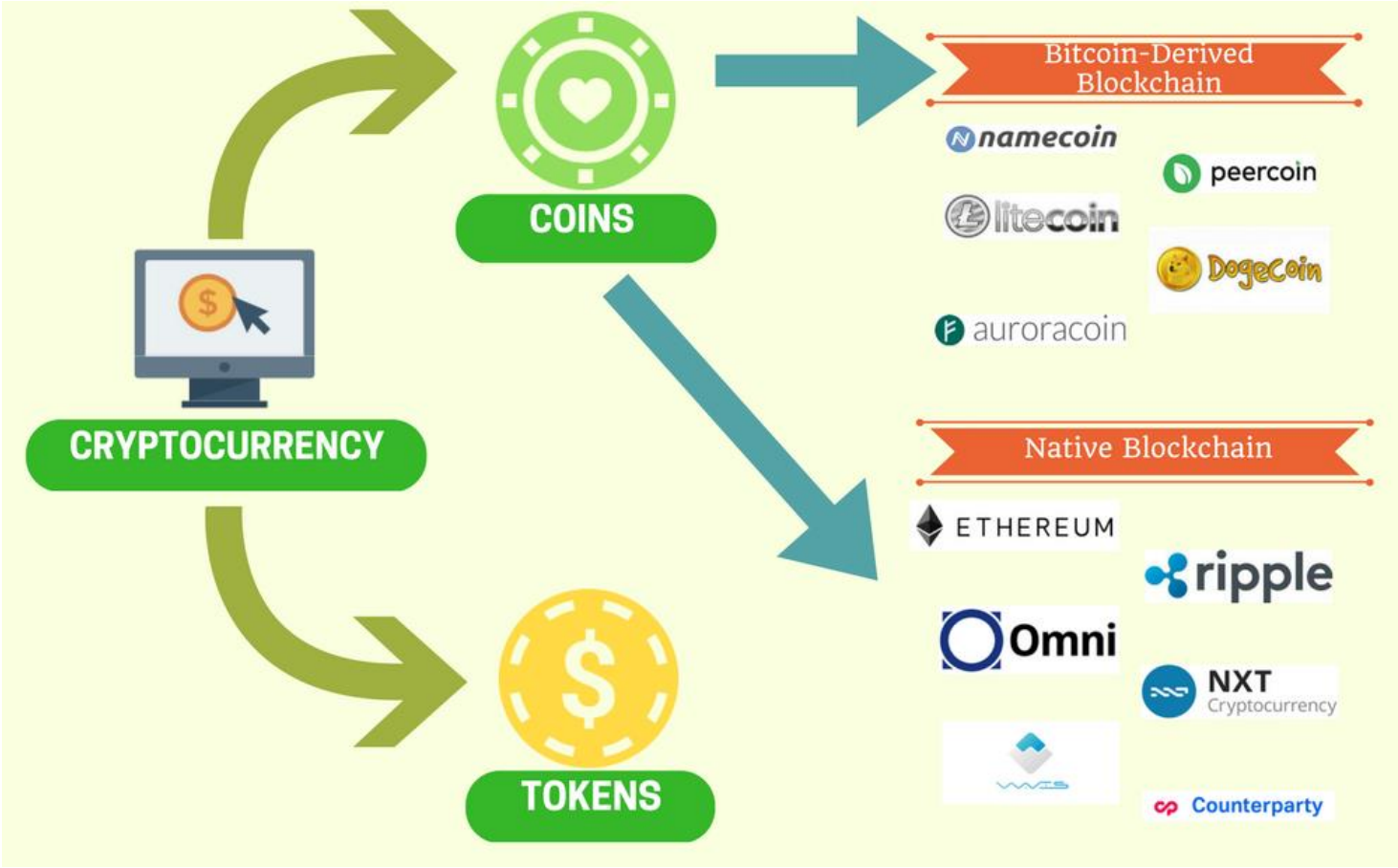
	Traditional contract	Smart contract
Specification	Natural language + “legalese”	Code
Identity & consent	Signatures	Digital signatures
Dispute resolution	Judges, arbitrators	Decentralized platform
Payment	Carried out by parties separately	built-in
Escrow	Trusted third party, settled in $	built-in

Outline

- Permissionless Blockchain
- Ethereum
  - Account-based Model
  - External Account and Contract Account
  - Gas
  - Transactions
  - Ethereum Virtual Machine
  - Smart Contract (Solidity)
- Coins and Tokens

Coins and Tokens

- Coins and Tokens are two different concepts
- Coins (or altcoins) are digital money
  - Created by cryptography
  - Based to public blockchain
  - Can be sent, received, mined

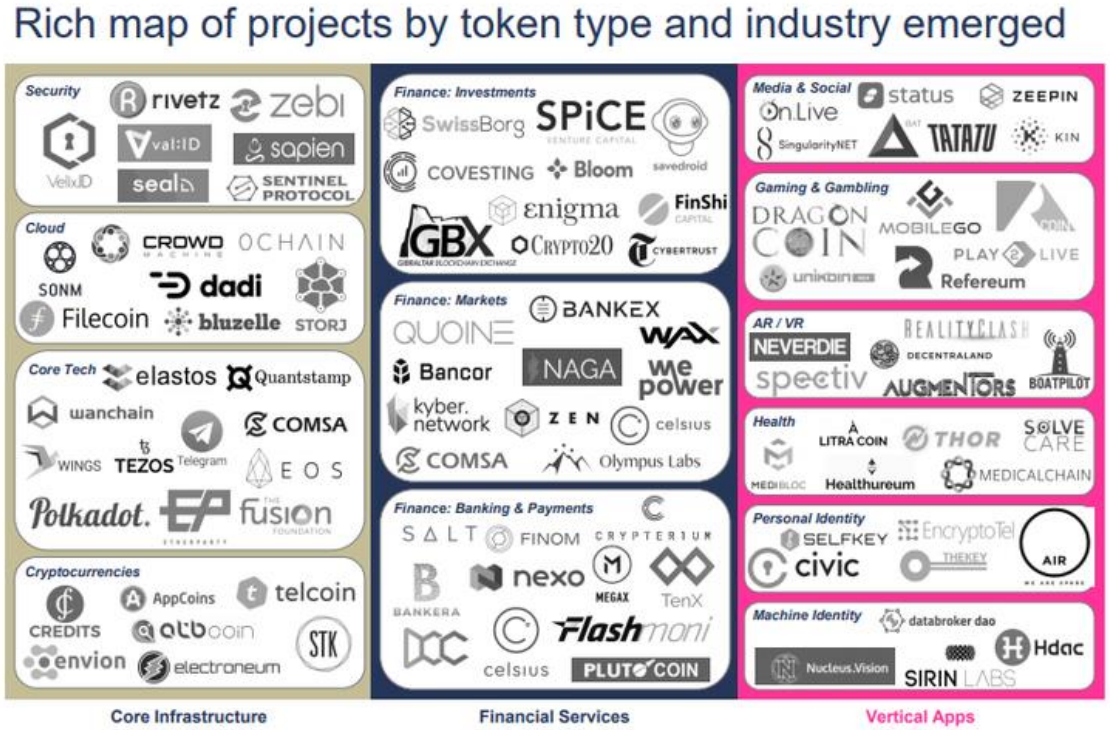


Altcoins

- Altcoins include all alternatives to Bitcoin
  - Since the creation of Bitcoin in 2008, 2000+ alternative cryptocurrencies have been deployed.
- Most of them were created as modified copies of Bitcoin
  - E.g., choose different hash functions, block generation times, block sizes, or mining difficulties
- Different consensus algorithms
  - E.g., Proof of Stake (the most common alternative to PoW), Delegated Proof of Stake, Proof of Burn, Proof of Authority, ...
- Despite sharing some similarities, each altcoin has its own functionalities

Tokens

- Tokens can be used for payment
  - Main difference: gives the holder a right to participate in the network



https://next.autonomous.com/thoughts/tag/taxonomy

- Tokens can act as digital asset
  - E.g., a company's share
- Can have certain use case but only inside certain project
- Creating a token is easier than creating a coin
  - E.g., you can use a standard template from platforms such as Ethereum

Tokens

- Tokens can be programmed to provide different functions
  - Currency
    - A token can serve as a form of currency, with a value determined through private trade
  - Resource
    - A token can represent a resource earned or produced in a sharing economy or resource-sharing environment
      - E.g., a storage or CPU token representing resource that can be shared over a network
  - Asset
    - A token can represent ownership of an intrinsic or extrinsic, tangible or intangible asset.
      - E.g., gold, real estate, a car, oil, energy, …

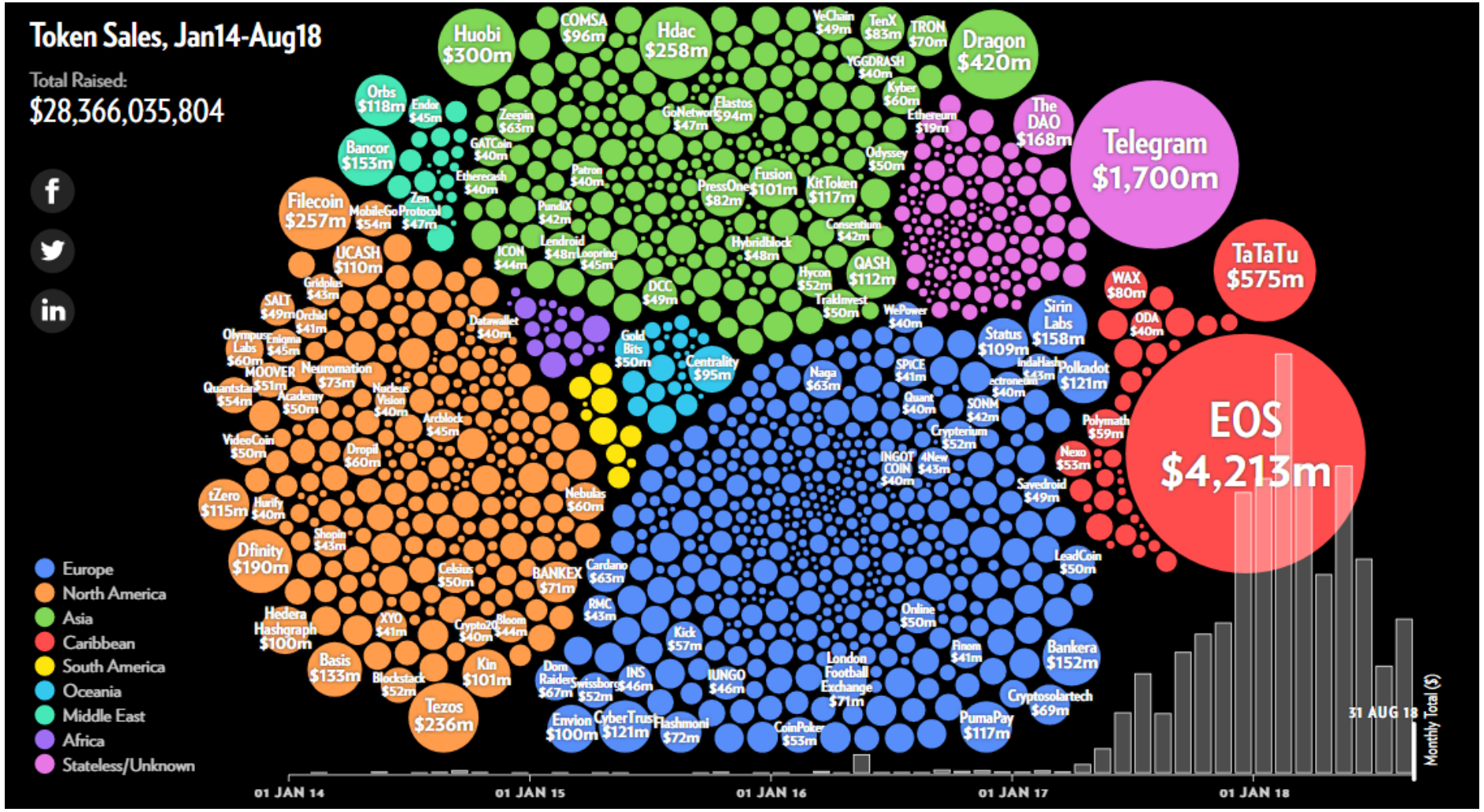
Tokens

- Access
  - A token can represent access rights, and grant access to a digital or physical property
    - E.g., a discussion forum, an exclusive website, a hotel room, or a rental car
- Voting
  - A token can represent voting rights in a digital or legal system
- Identity
  - A token can represent a digital identity (e.g. avatar) or legal identity (e.g. national ID)
- Certification
  - A token can represent a certification by some authority or by a decentralized reputation system
    - E.g., marriage record, birth certificate, college degree

Initial Coin Offering (ICO)

- Initial Coin Offering (ICO) acts as a way to raise funds for a company to create a new coin, app, or service
  - An ICO is the cryptocurrency industry's equivalent to an Initial Public Offering (IPO)
- Interested investors can buy into the offering and receive a new cryptocurrency token issued by the company
- The token may represent a right to use the product this company is offering, or a stake in the company or project

ICO History



https://elementus.io/token-sales-history

Summary

- Permissionless Blockchain
- Ethereum
  - Account-based Model
  - External Account and Contract Account
  - Gas
  - Transactions
  - Smart Contract (Solidity)
- Coins and Tokens