

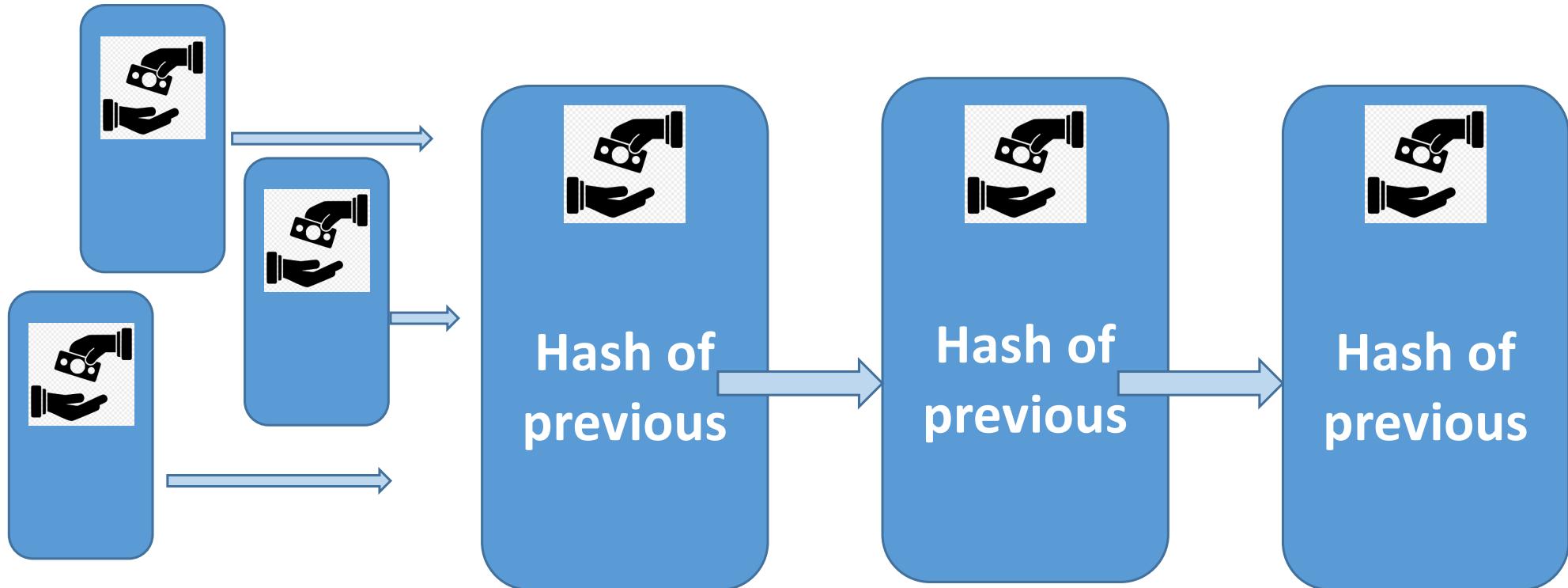
Ethereum Virtual Machine

Highly dependable systems – 2024/25

Lecture 8

Lecturers: Miguel Matos and Paolo Romano

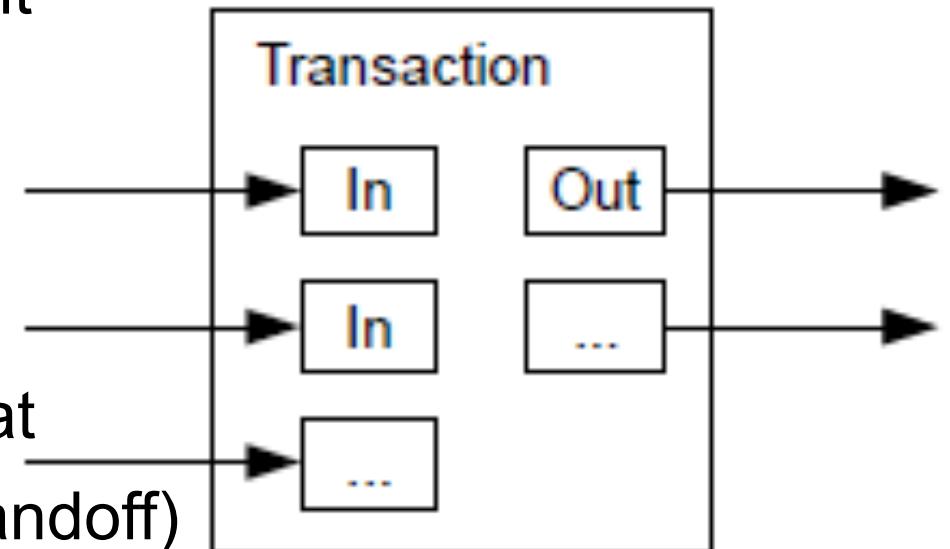
Last lecture: Bitcoin



- PoW → miners compete to append next record
- In some cases may lead to fork → longest chain rule
- Shortcomings: studied several attacks + energy waste

Each block in the blockchain is a set of transactions

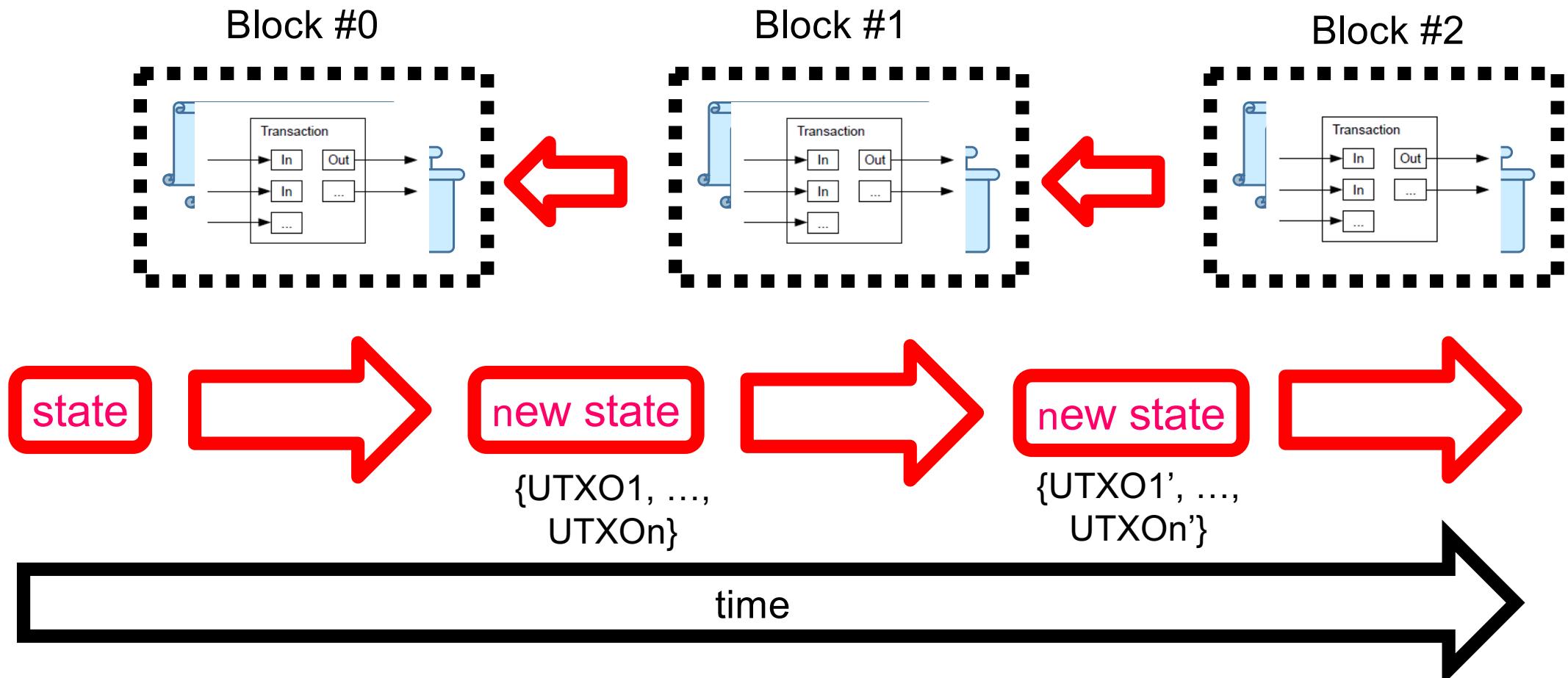
- Transactions can have multiple inputs and ≤ 2 outputs
- Each input describes debit
 - Sender → signs handoff of amount (signed with private key)
- Each output describes credit
 - Recipient → public key of entity (corresponding to private key that later can sign the subsequent handoff)



Bitcoin as a state machine

- State is current set of UTXOs
 - (when a node boots, this is derived from all transactions since inception)
- Transactions cause the following state transition:
 - Remove all UTXOs in “In” from set
 - Add all UTXOs in “Out” to the set

Bitcoin's state machine



Today: Ethereum

- Ethereum is the second largest cryptocurrency after Bitcoin
- Very generic platform, supporting myriad of distributed applications (dapps), with several currencies, NFTs, DAOs, etc.
- Introduced broad support for “smart contracts”
 - Programs that run in the blockchain
 - Ability to read and modify the blockchain state
 - Turing complete
- Adversarial environment without centralized trust
 - Guaranteeing safety is key

Ethereum's layered design

Distributed applications (dapps)

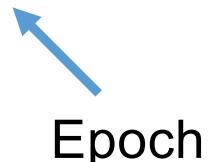
Compute layer (EVM – executes smart contracts)

Consensus layer (PoS based blockchain – “beacon chain”)

Ethereum's PoS blockchain

- Moved from PoW to PoS in December 2020 (“beacon chain”)
- One block every 12 seconds (100-200 txns/block)
- Participants (validators) must stake 32 ETH (~3500 ~1732 EUR today)
 - Large amount, some systems allow for pooling
- Block proposer is chosen at random* among validators
 - Broadcast new block; collects transaction fees
- Remaining validators check correctness and sign correct blocks
 - Block is finalized after a certain threshold of signatures
 - Validators that behave correctly → rewarded (every 32 blocks)
 - Behave incorrectly → part of stake is slashed

* probability is proportional to stake size

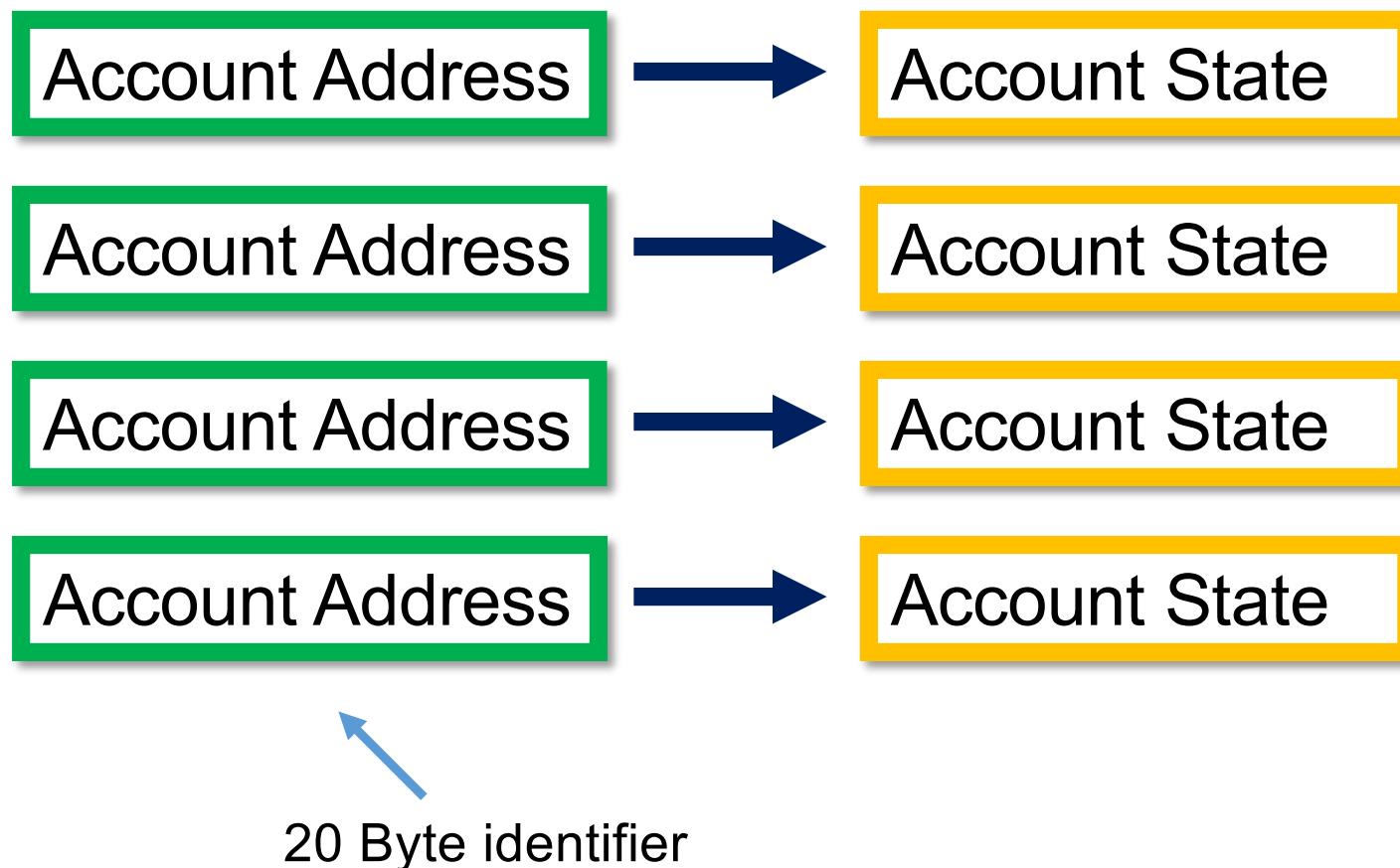


Epoch

Ethereum has an account model

- Instead of UTXOs, the state of Ethereum is a mapping from a set of accounts to their respective state
- At a basic level, the state is the account balance (but it gets more sophisticated)

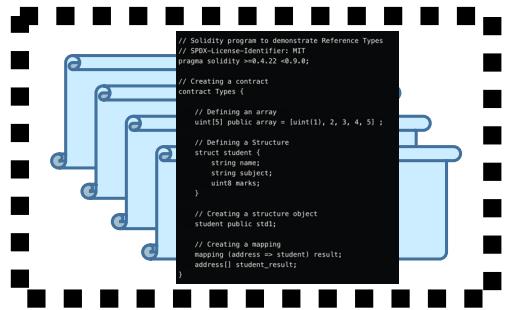
Ethereum State



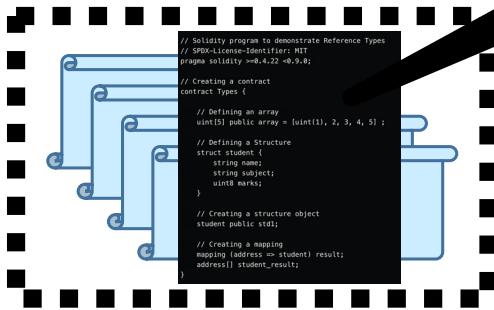
Ethereum's state machine

A state transition executes a whole program

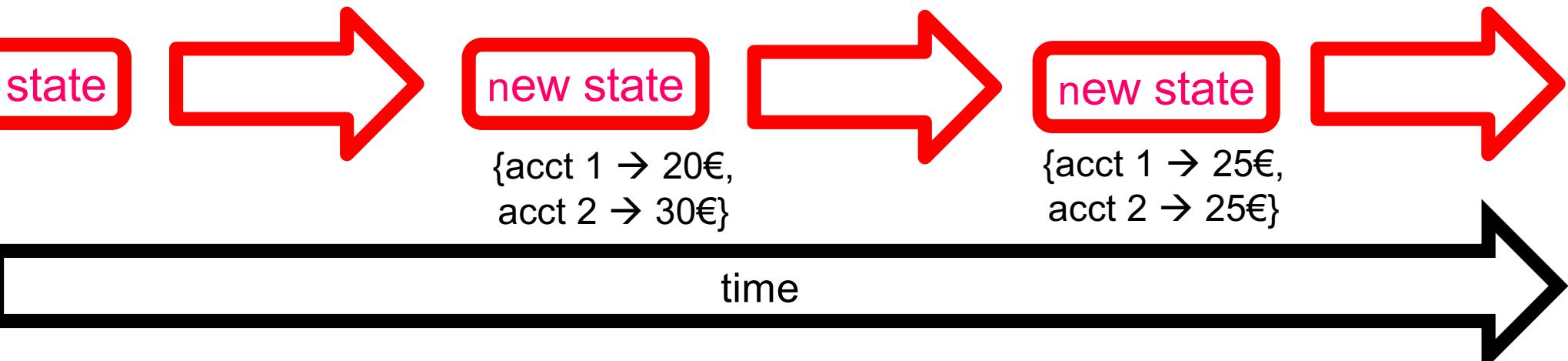
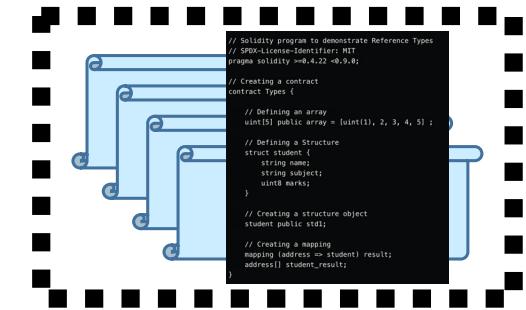
Block #0



Block #1



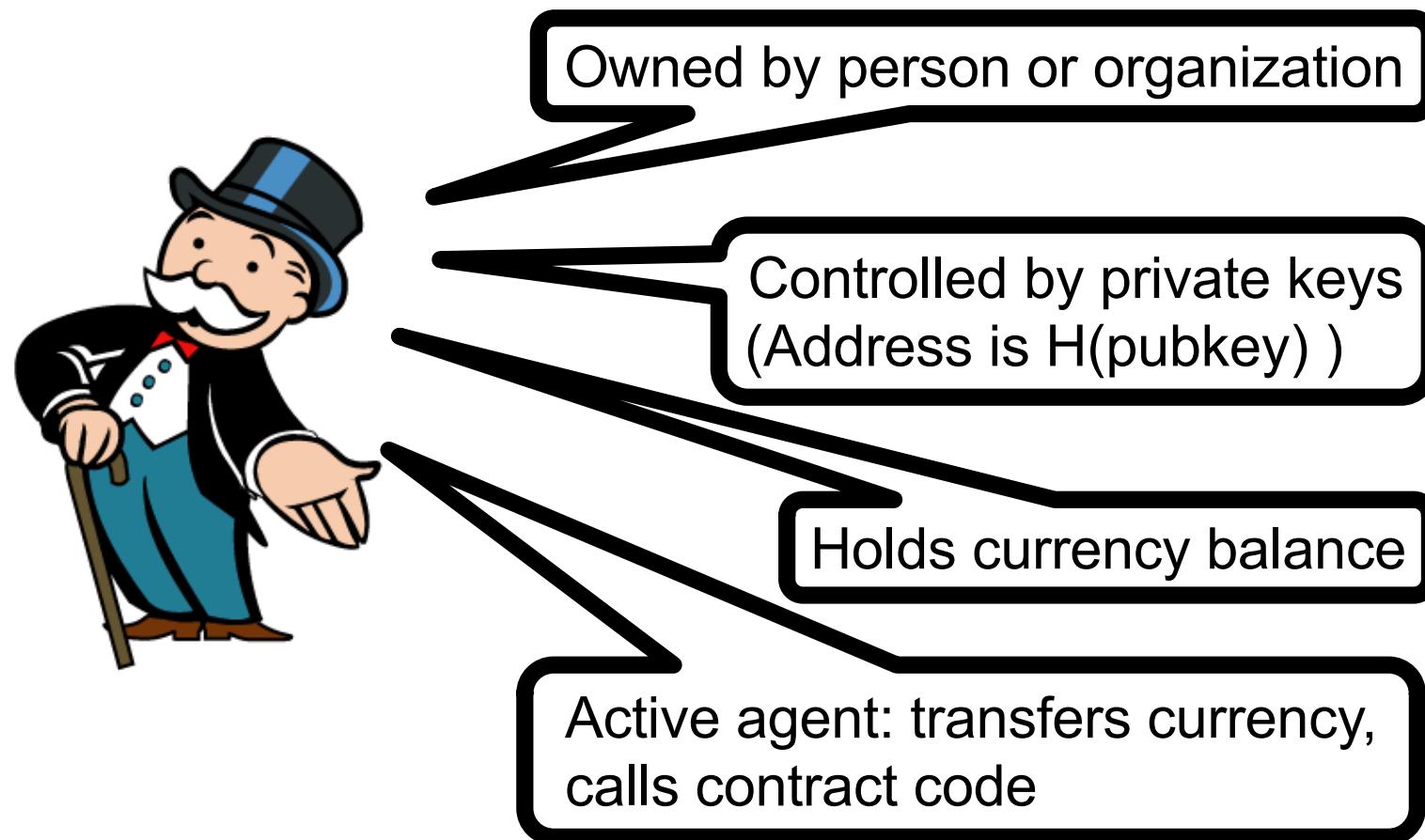
Block #2



Two types of accounts in Ethereum

- External (owned accounts)
- Contract

External Account (“owned”)



External Account (“owned”)

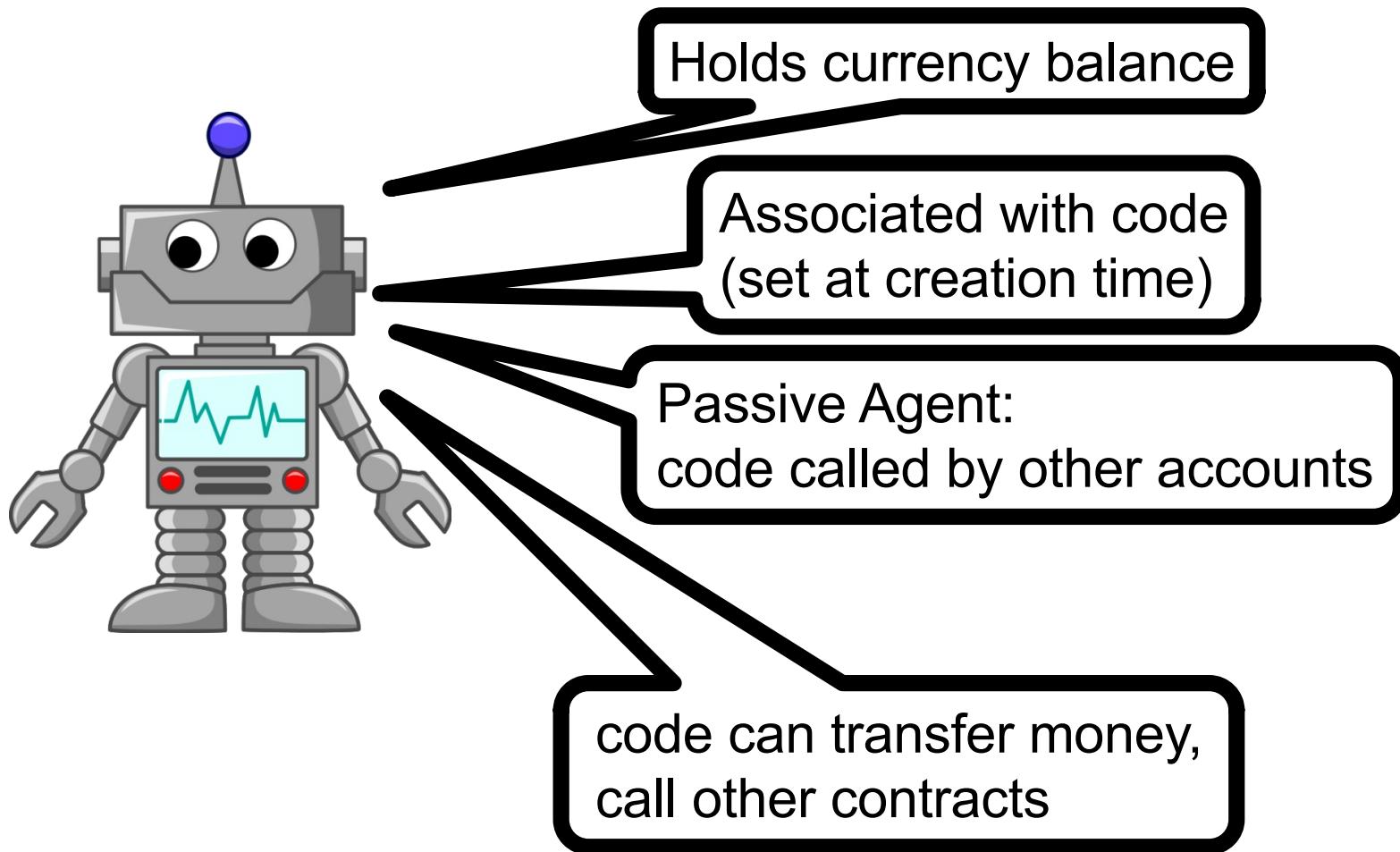


Address

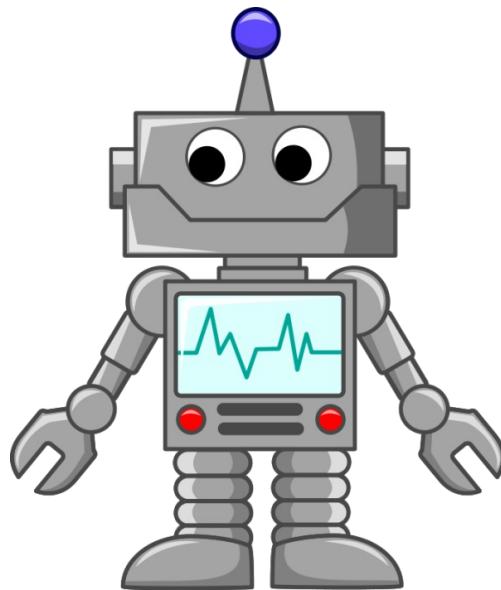


balance

Contract Account



Contract Account



Address



balance
code
storage

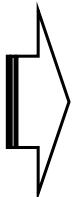
Array of
32-byte cells

Transaction Creation

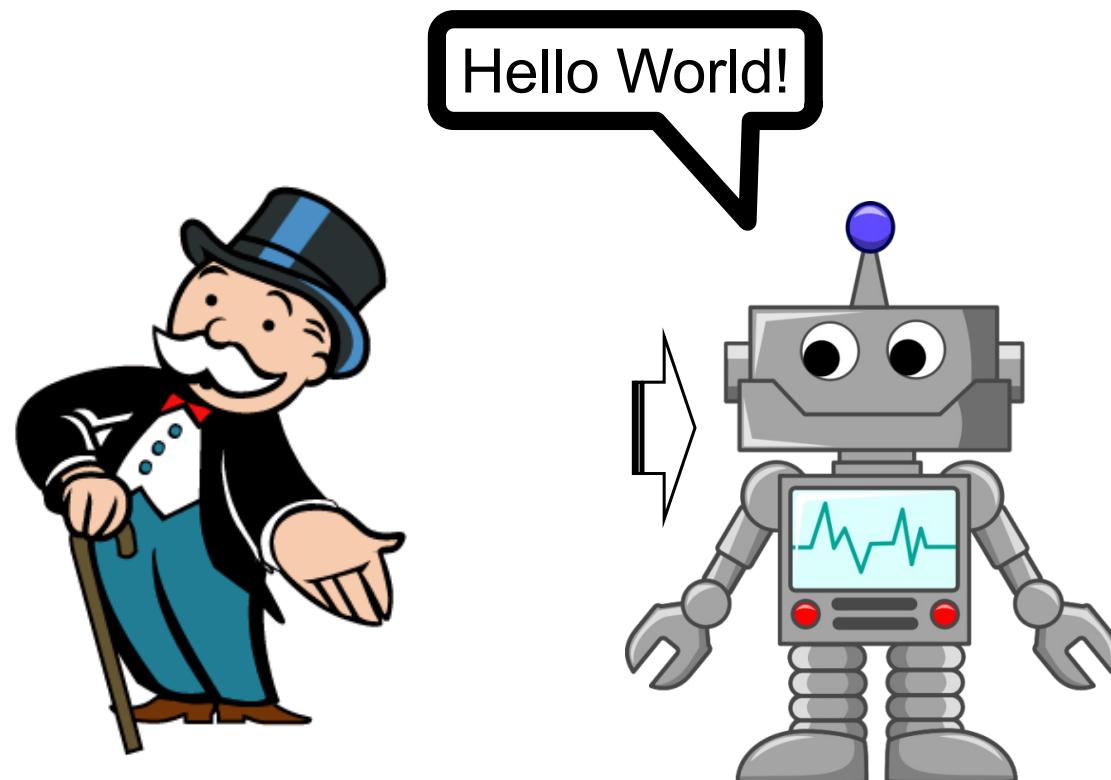


Submitted by external party

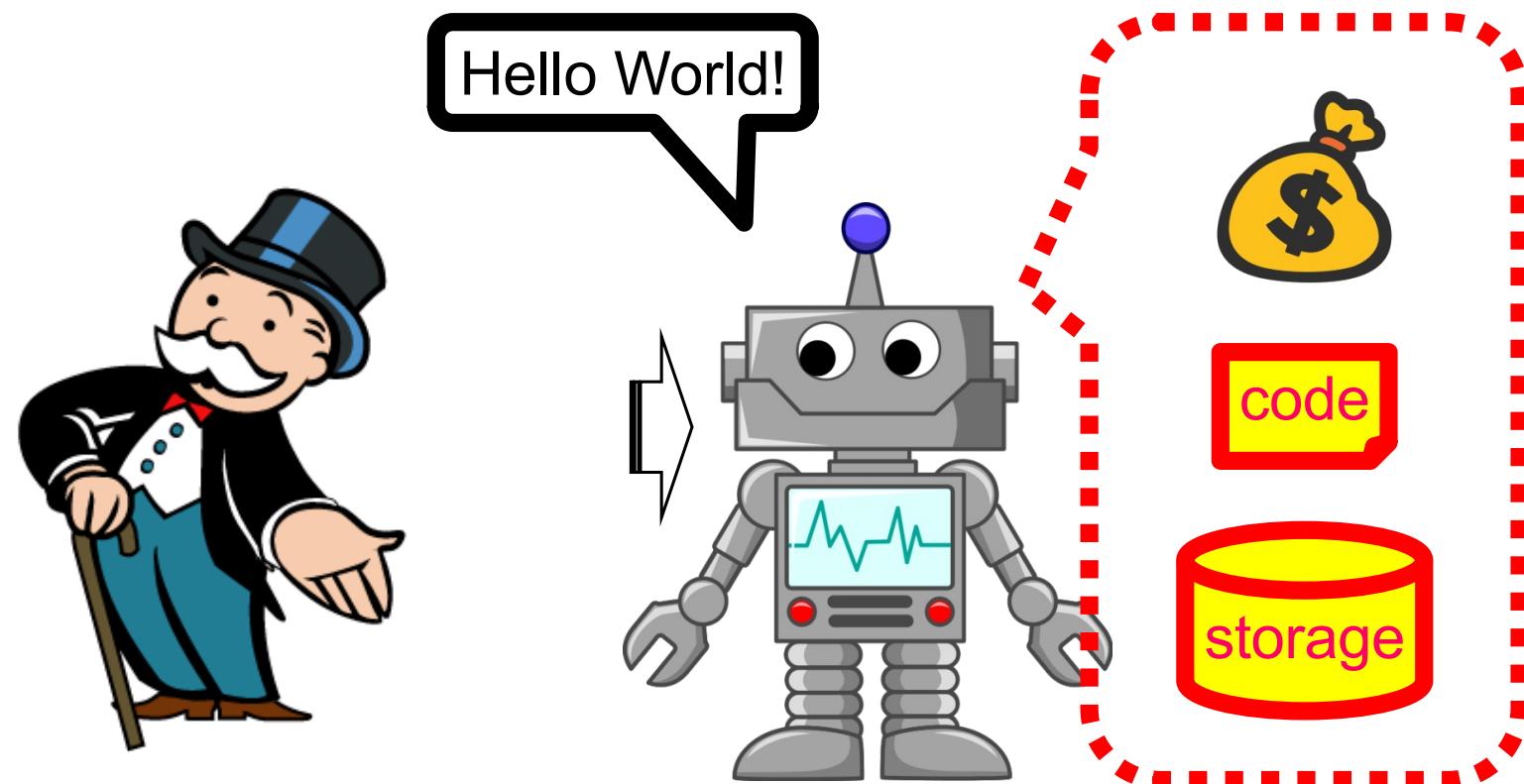
Contract Creation Transaction



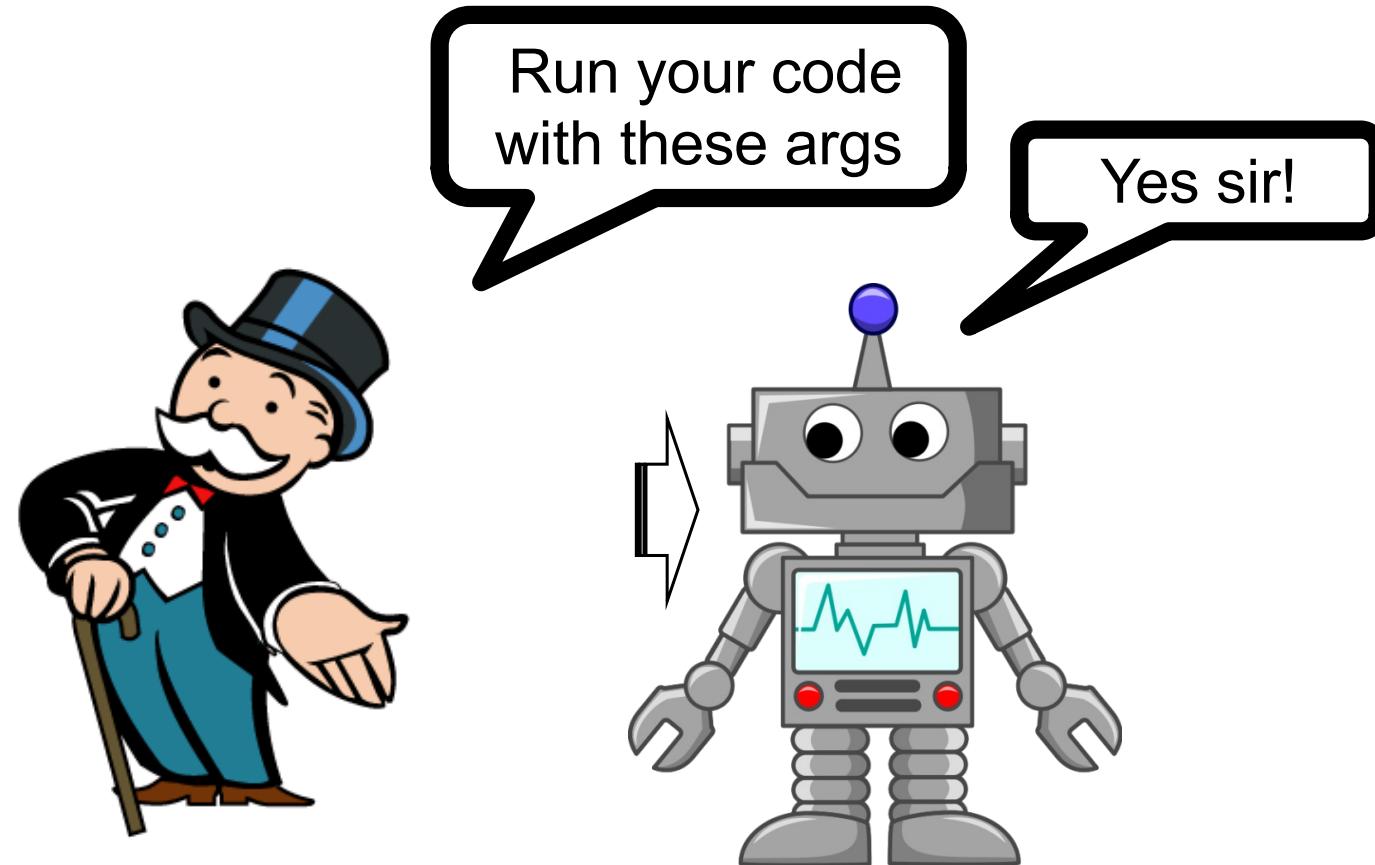
Contract Creation Transaction



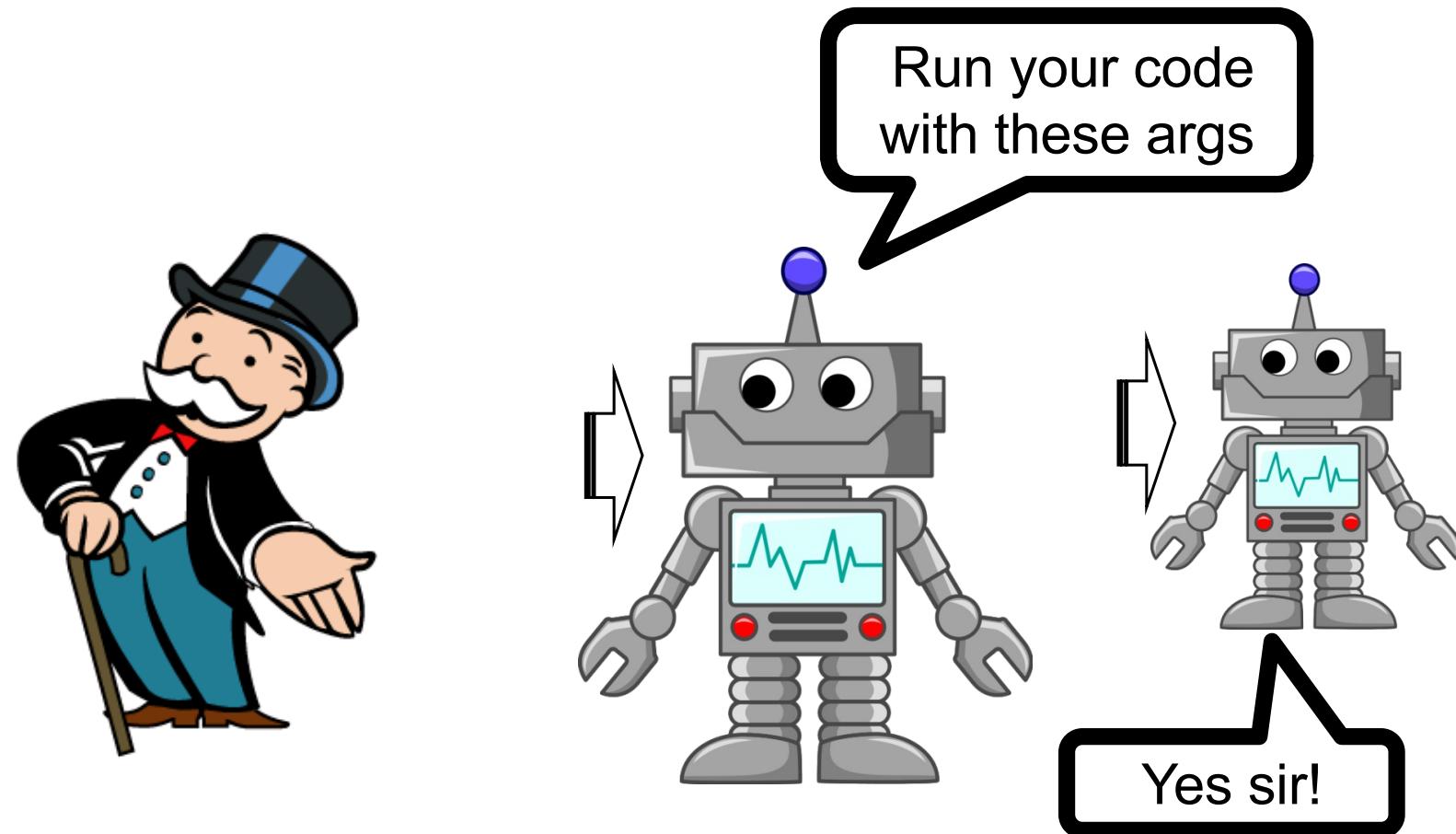
Contract Creation Transaction



Message Call Transaction

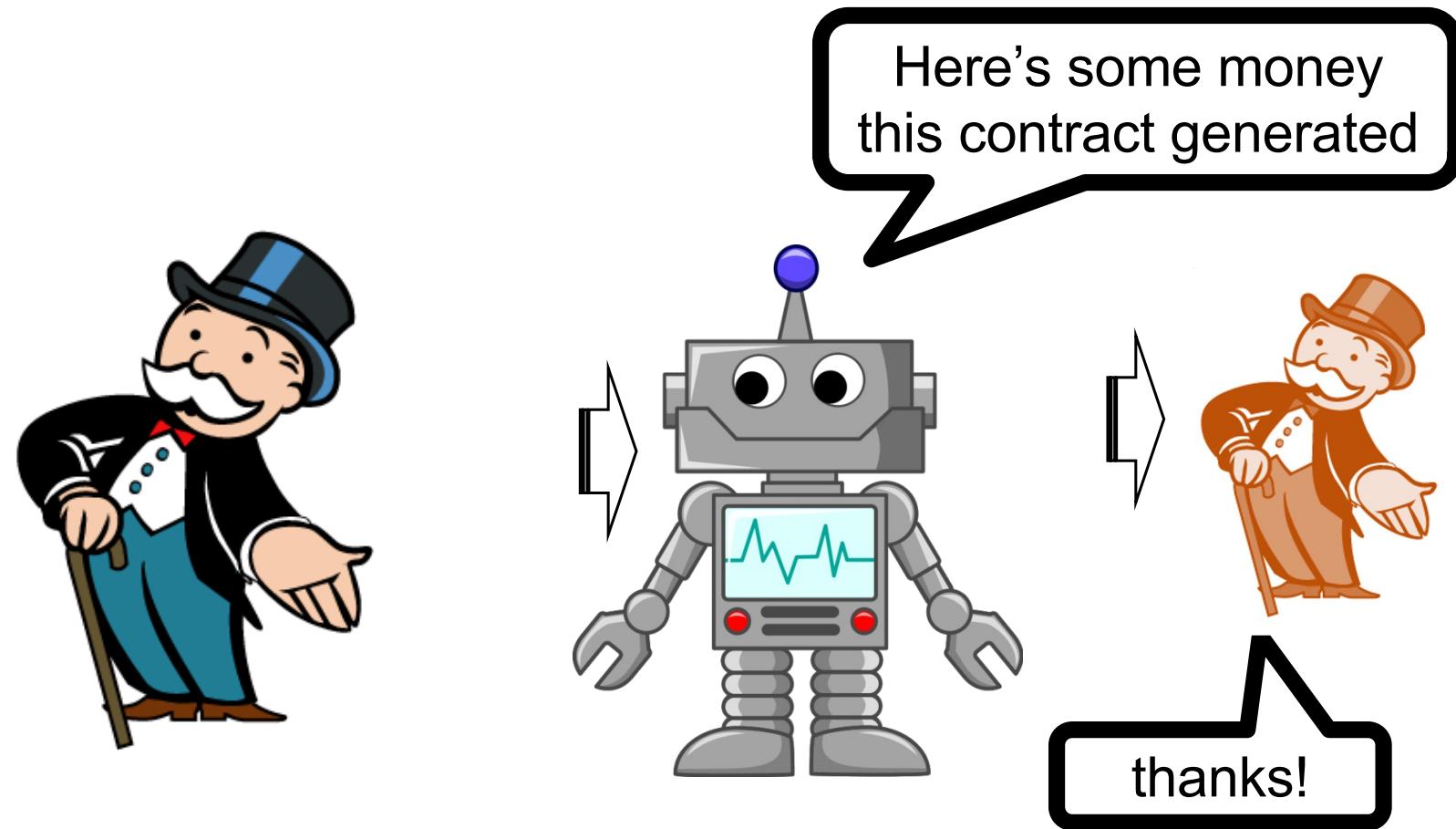


Message Call Transaction



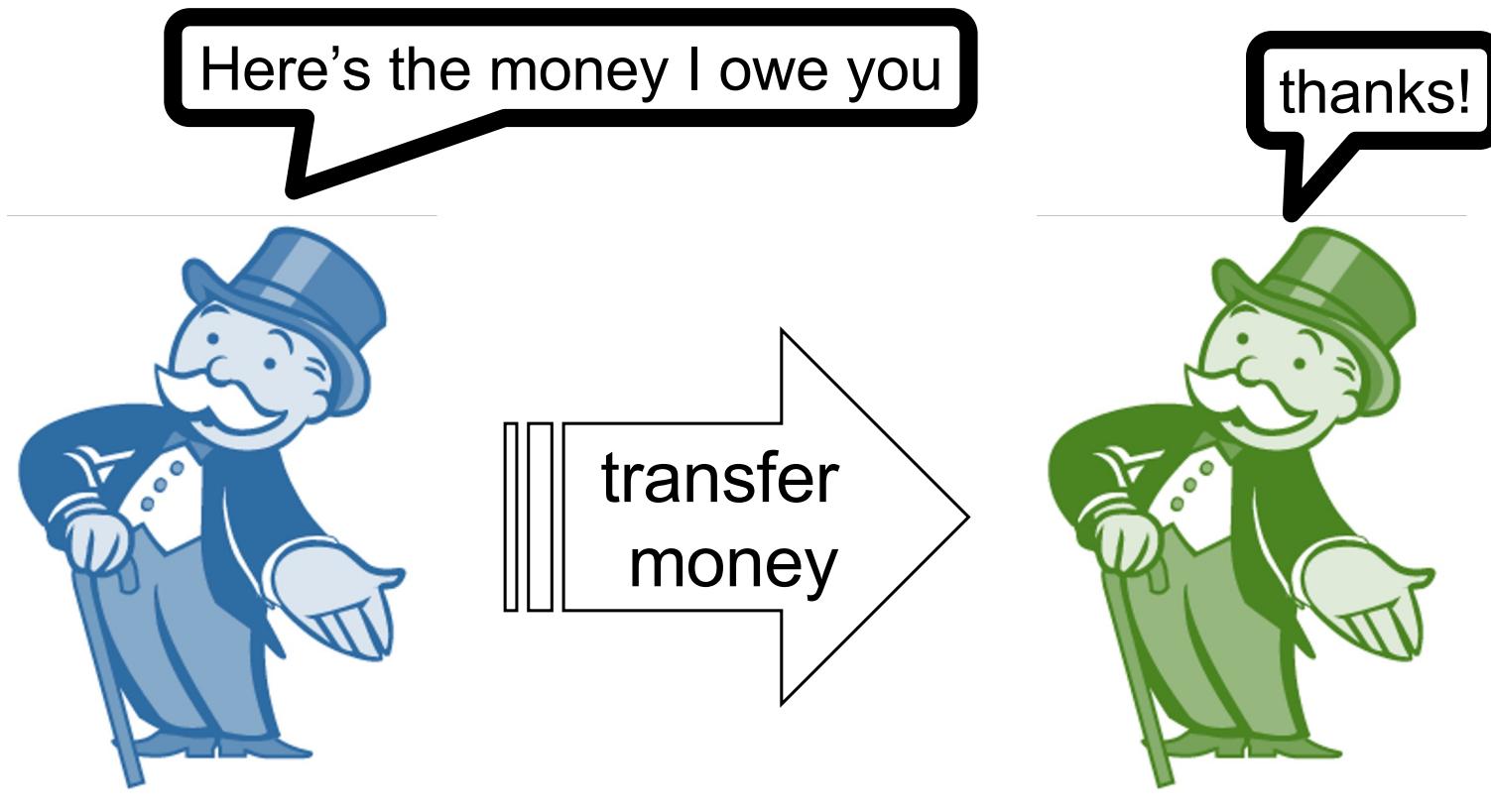
Contracts can call other contracts

Message Call Transaction

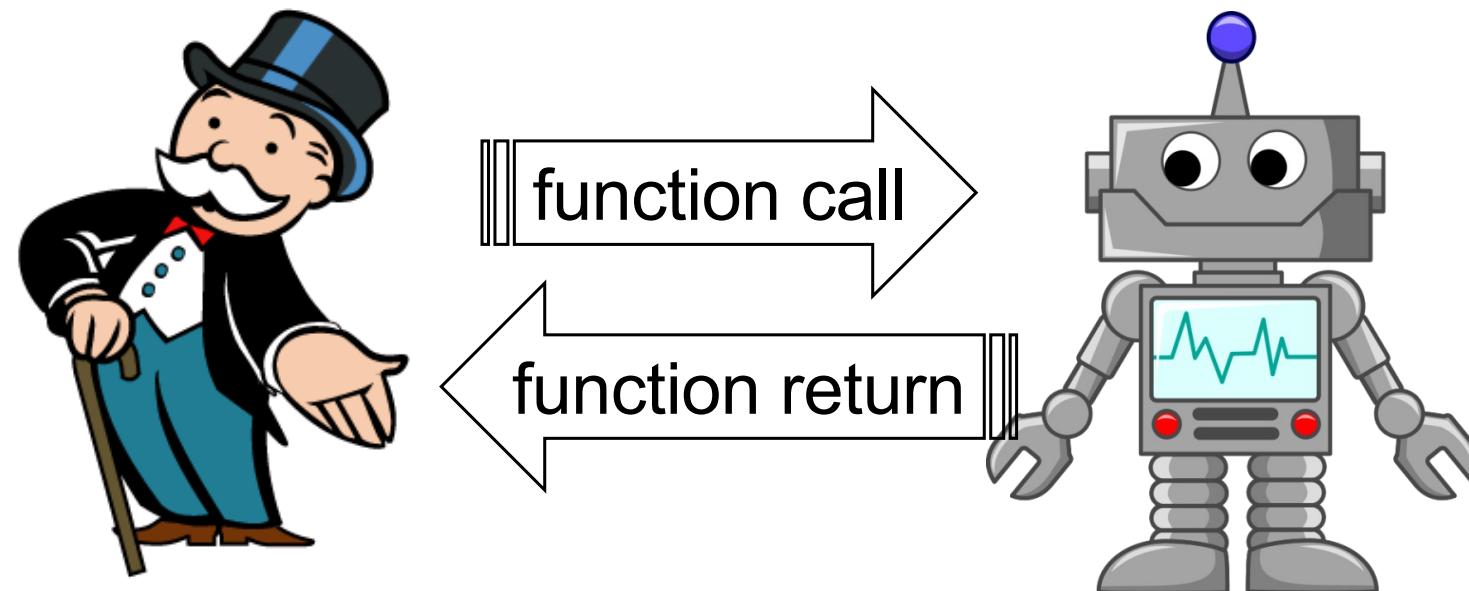


Contracts can also send funds to users

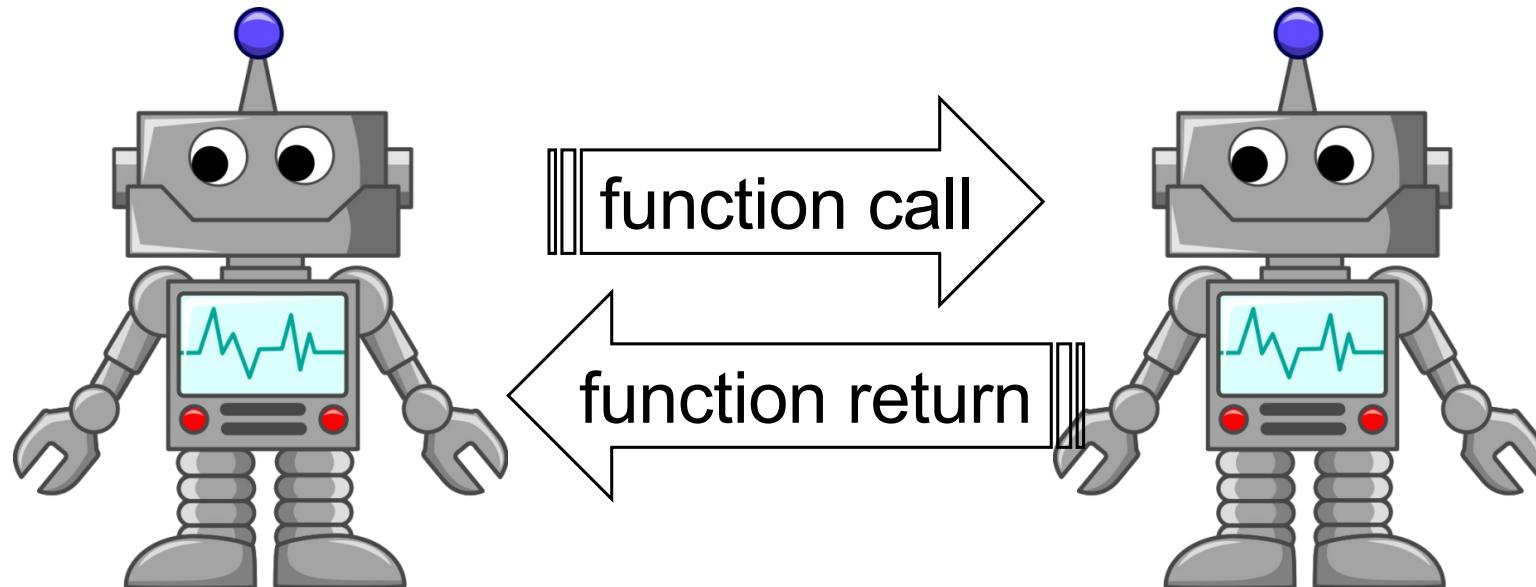
External to External Message



External to Contract & Vice-Versa



Contract to Contract Message

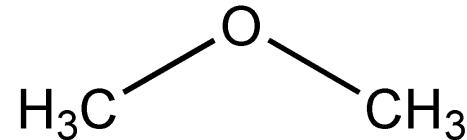


Money, Honey

Native currency called *ether*



not this



but this



Gas

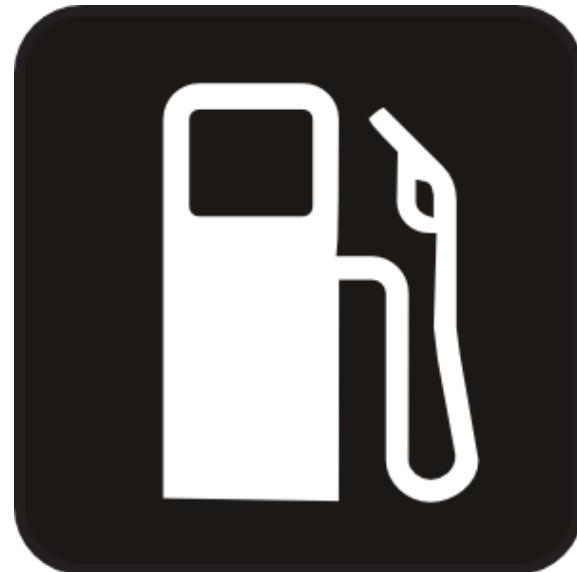
By calling a contract we are essentially running code, written on a Turing complete language, on thousands of nodes worldwide

Problems?

Gas

Caller pays fee for each transaction step

Denial of Service attacks expensive



Gas

Each step has fixed “gas” fee

But gas price in Ether up to caller!

Low price means low priority ...

And vice-versa



Gas

If a call runs out of gas ...

Effects discarded

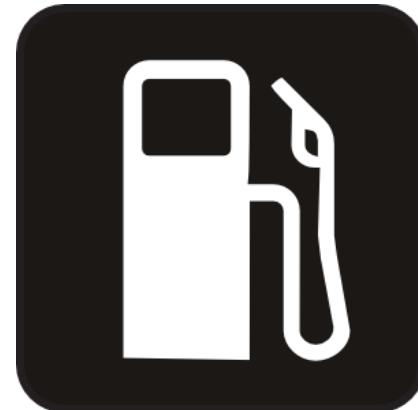
Gas not refunded

If a call has leftover gas ...

Unused gas refunded



Block Gas Limit



Bitcoin has limit on block size

Ethereum has limit on block gas

Block full when transactions' gas costs reach limit

We will see how this can be exploited later

Transaction Fields

Gas price

Value

Gas limit

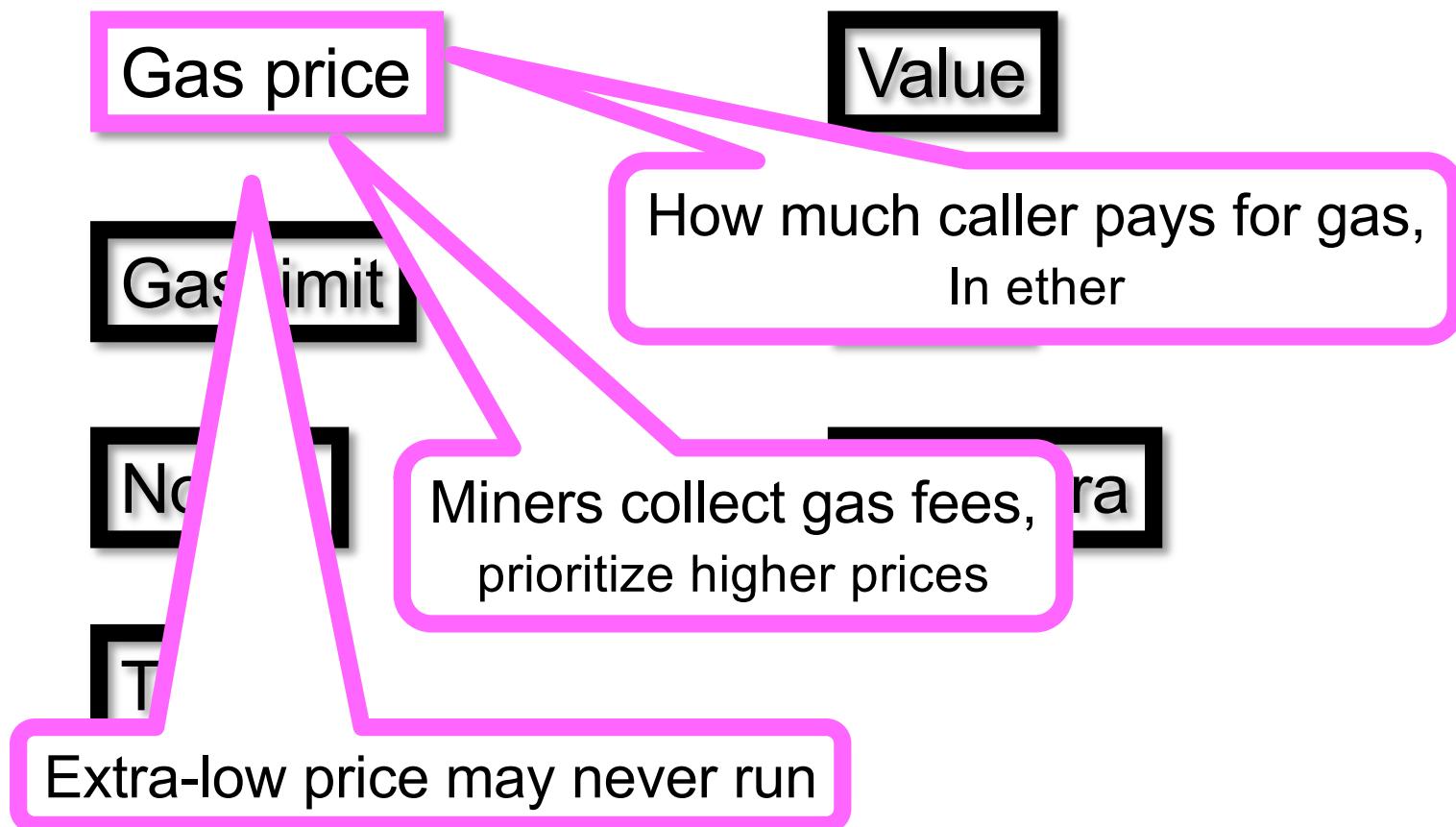
Data

Nonce

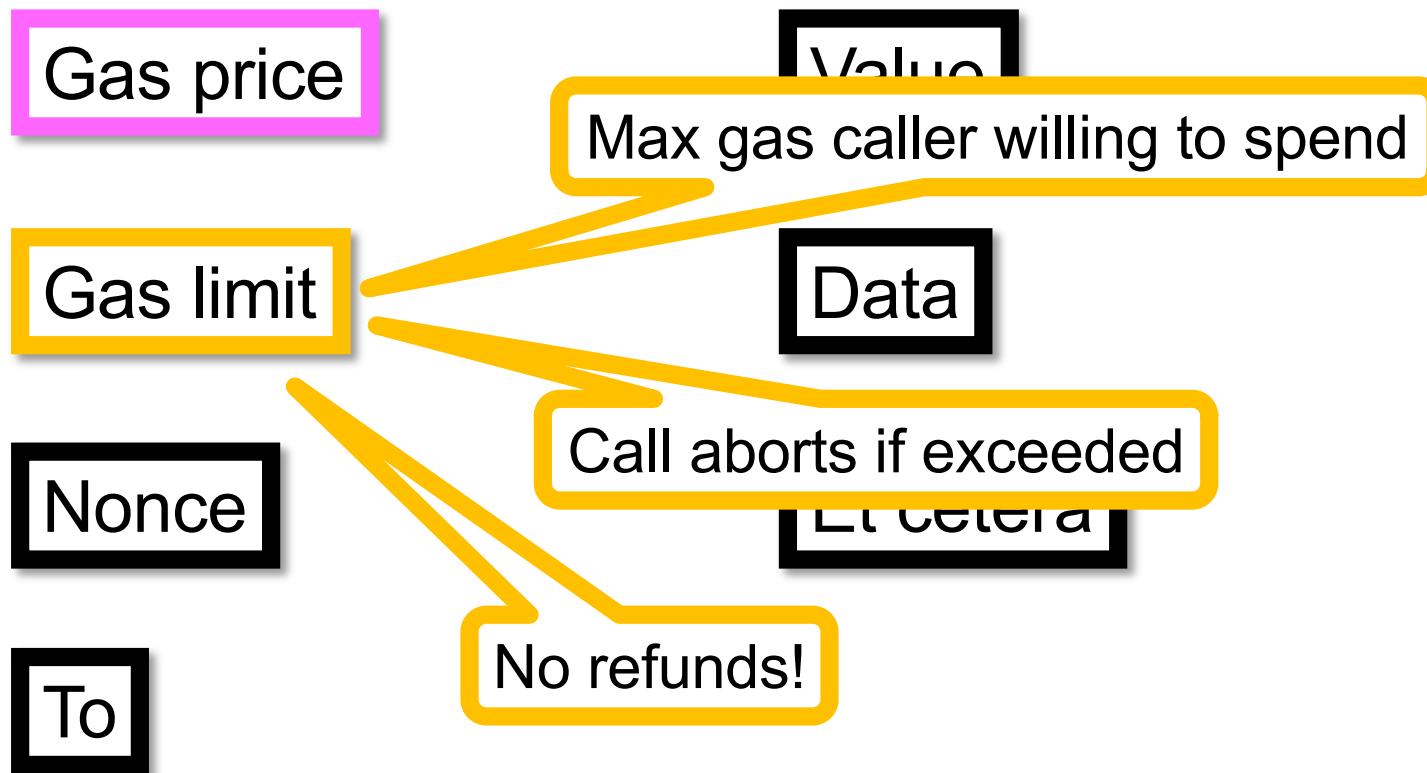
Et cetera

To

Transaction Fields

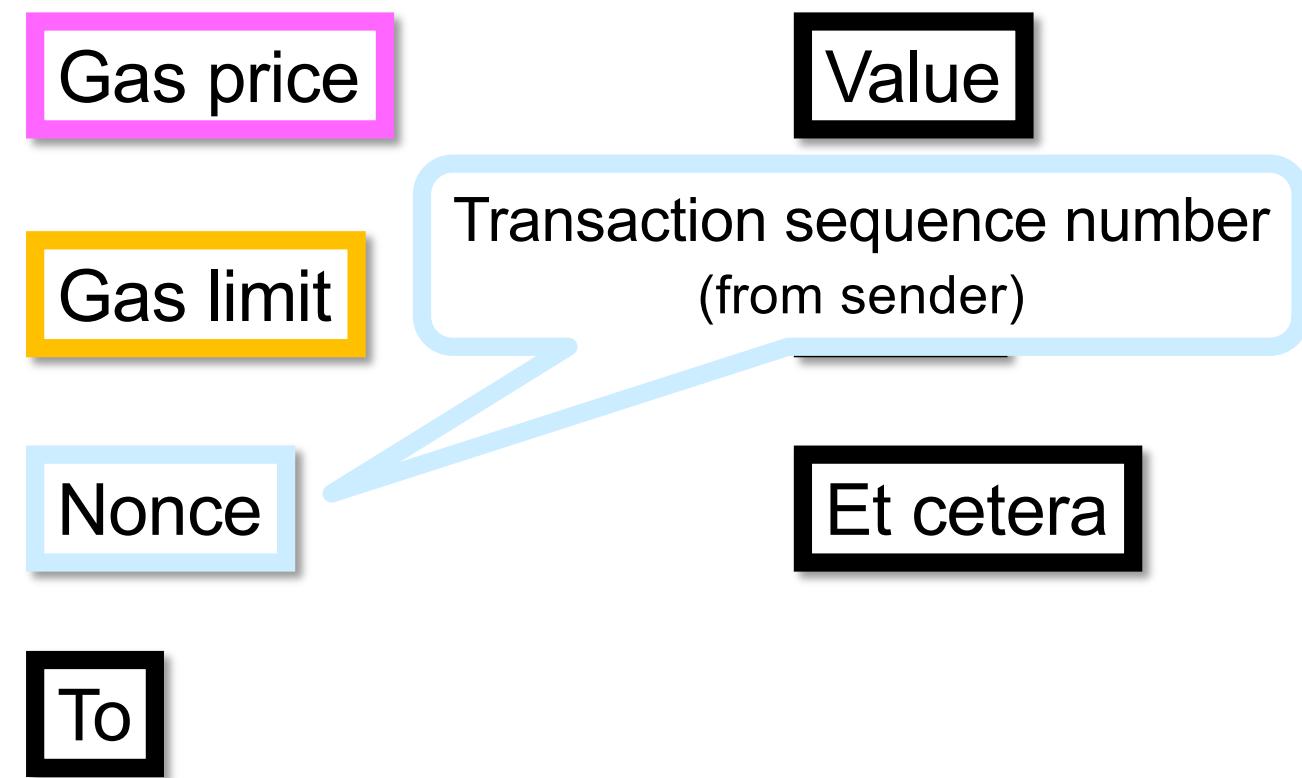


Transaction Fields



Can a caller be sure of the gas that is going to be spent?

Transaction Fields



Transaction Fields

Gas price

Value

Gas limit

Data

Nonce

Et cetera

To

destination address
(external or contract)

Transaction Fields

Gas price

Gas limit

Nonce

To

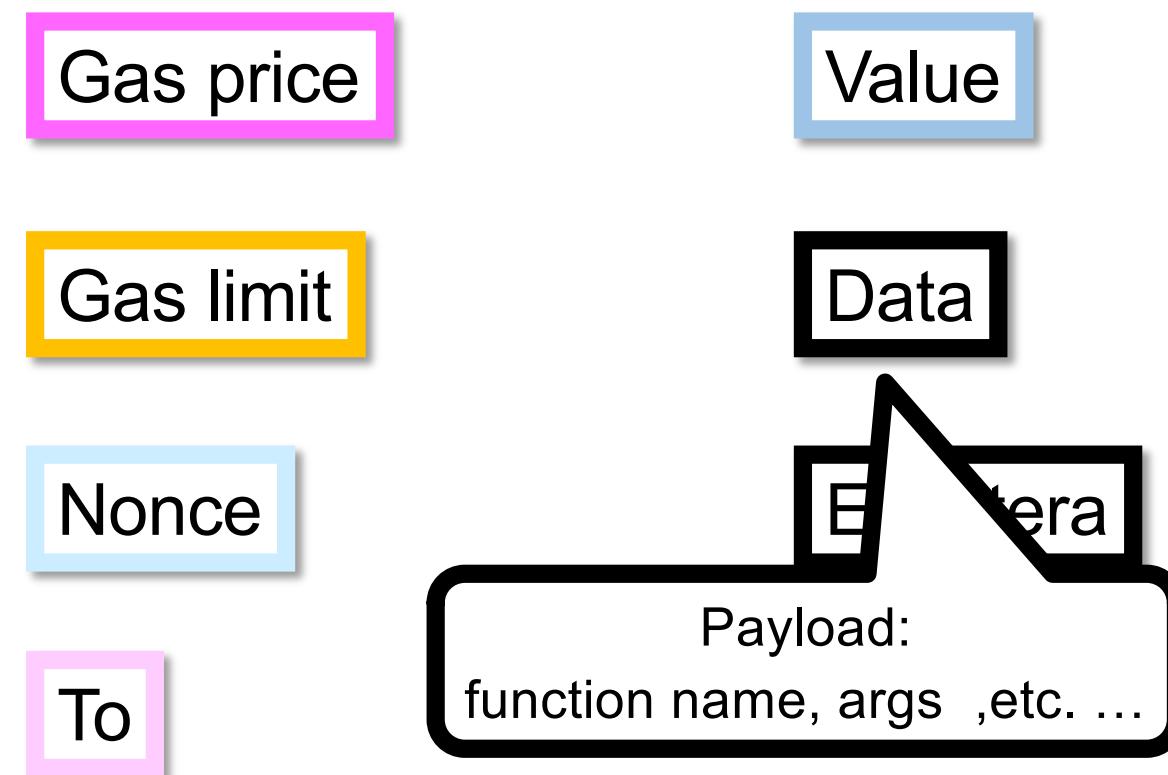
Value

Data

How much ether to transfer

Call criteria

Transaction Fields



Transaction Fields

Gas price

Value

Gas limit

Data

Nonce

Et cetera

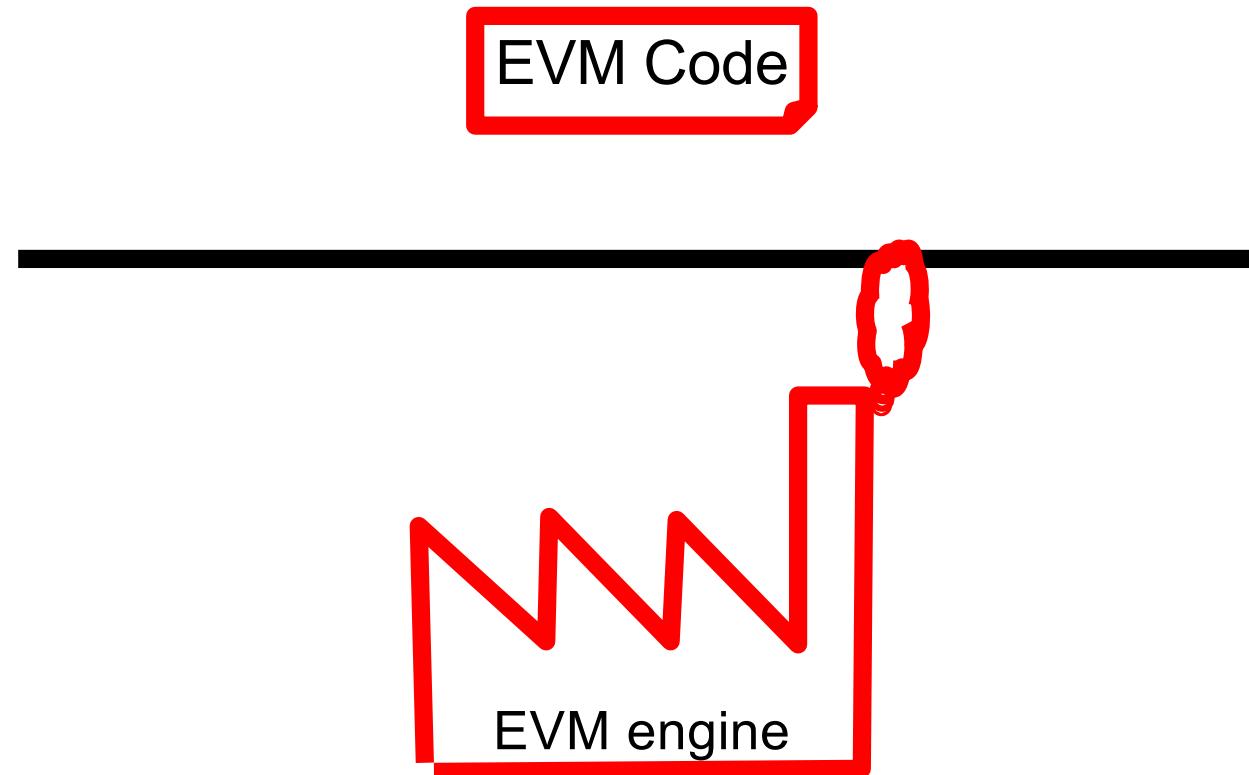
To

ECDSA signature args

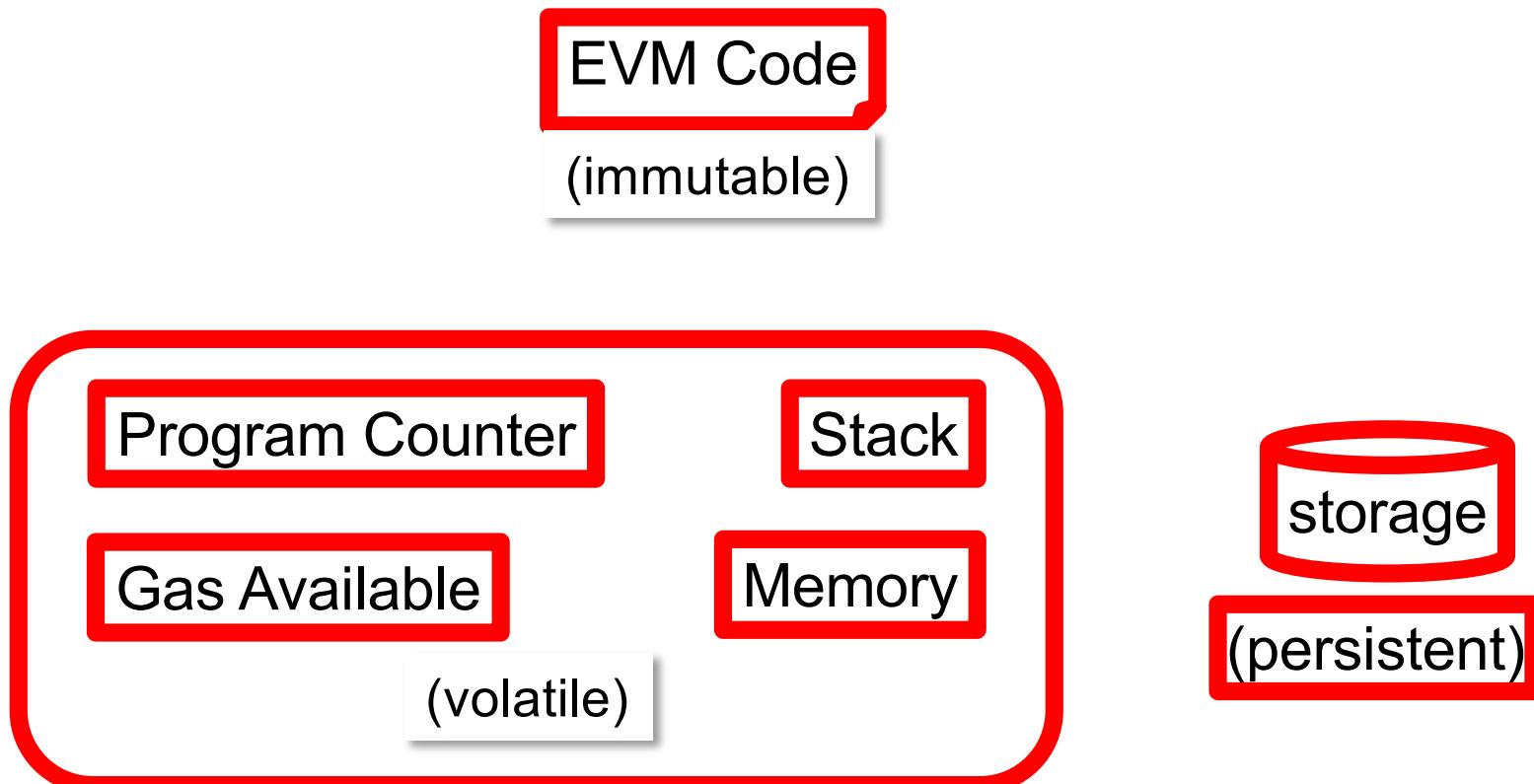
Recent changes in gas pricing

- EIP-1559 (part of “London hard fork” – Aug. 2021) defines new policies for gas pricing
- Idea: replace a first price auction with a base fee mechanism (though users can add a tip for their txns to be prioritized by validators)
- Base fee depends on how full recent blocks are:
 - Blocks do not fill up → base fee decreases
 - Blocks are completely full → base fee increases

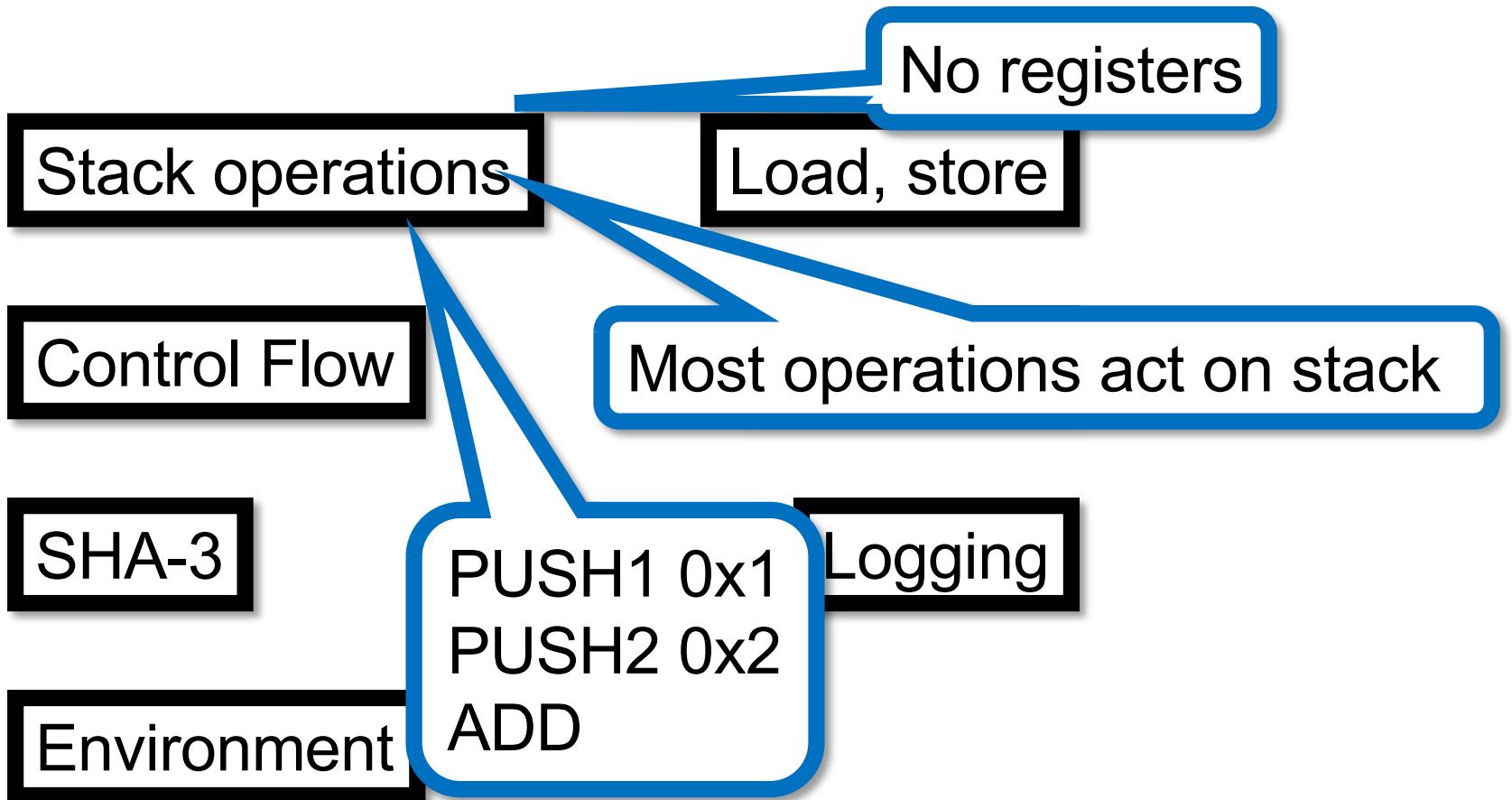
Ethereum Virtual Machine



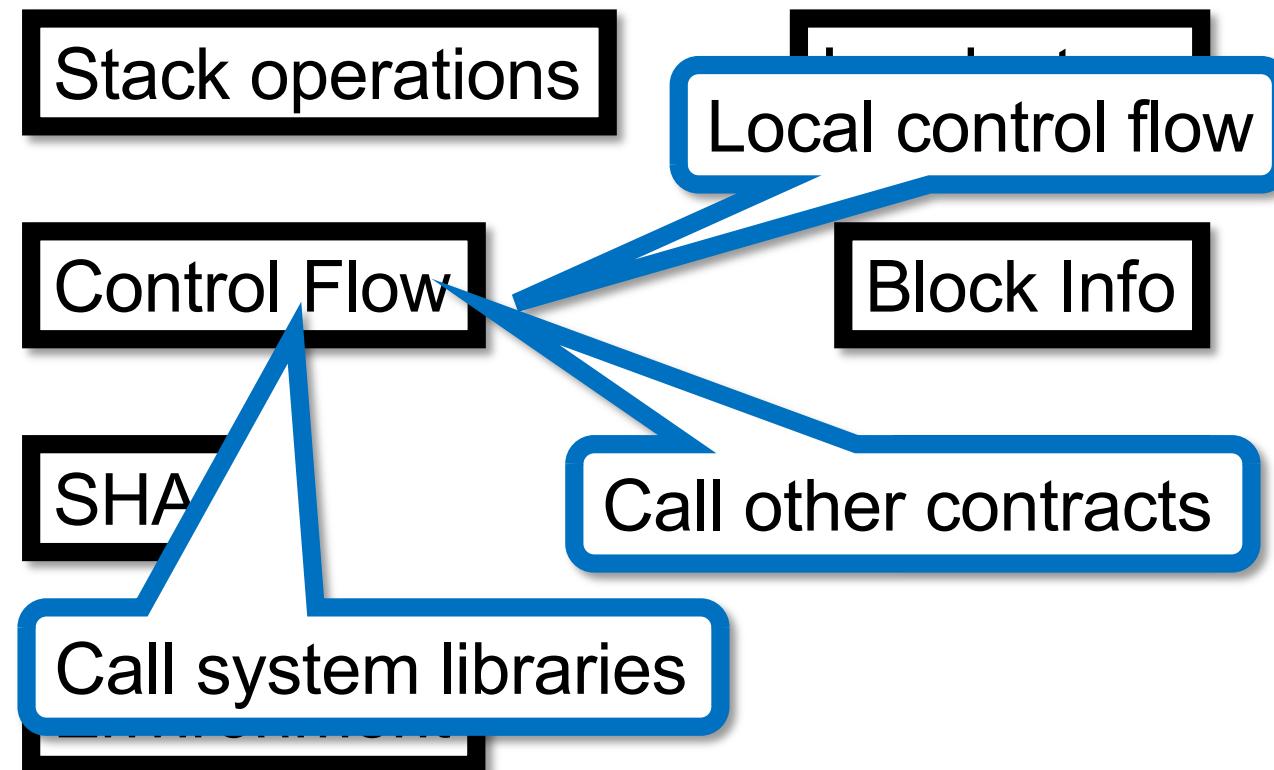
Ethereum Virtual Machine



Types of Instructions



Types of Instructions



Types of Instructions

Stack operations

Load, store

Control Flow

Various crypto hashes provided

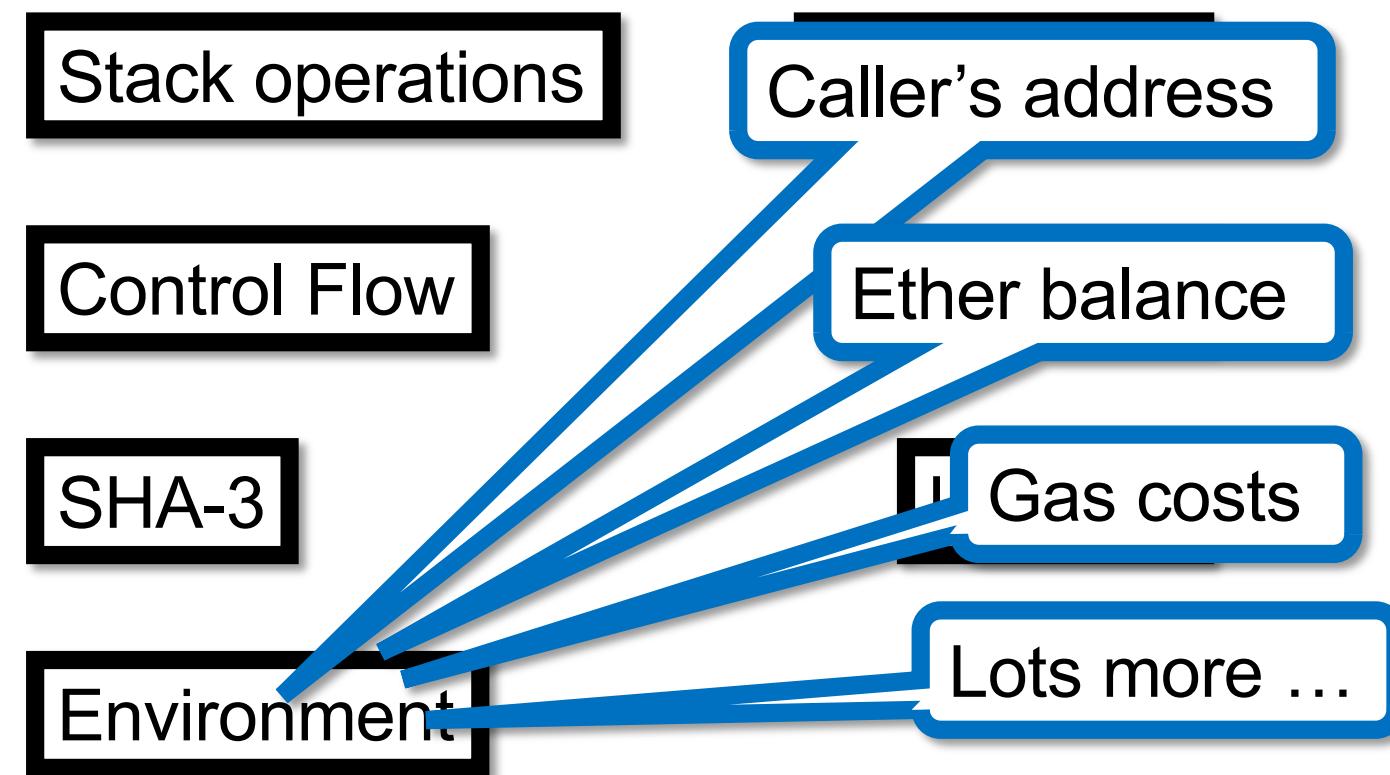
SHA-3

Logging

Environment

Gas costs too expensive to compute directly

Types of Instructions



Types of Instructions

Stack operations

Load, store

Control Flow

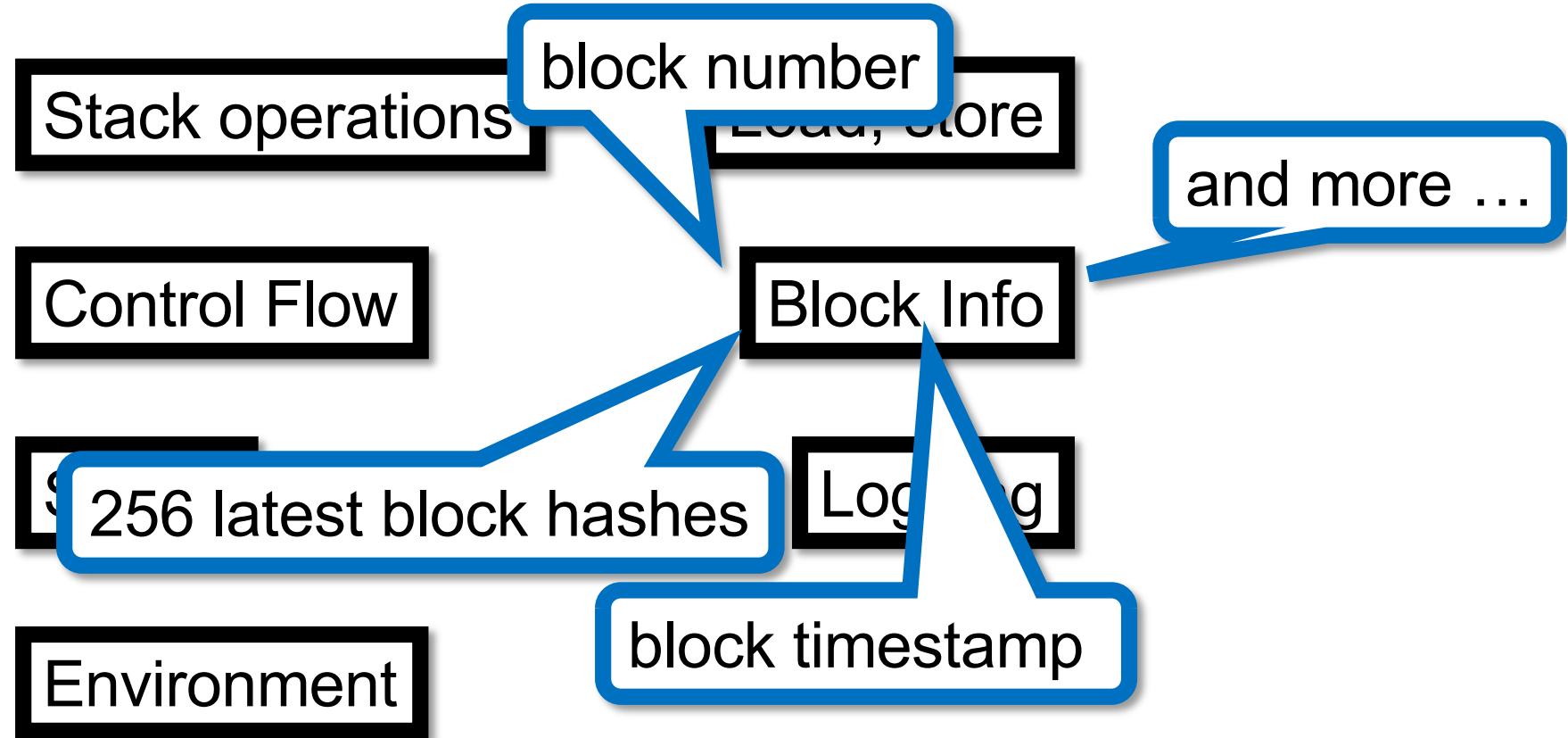
Block

SHA-3

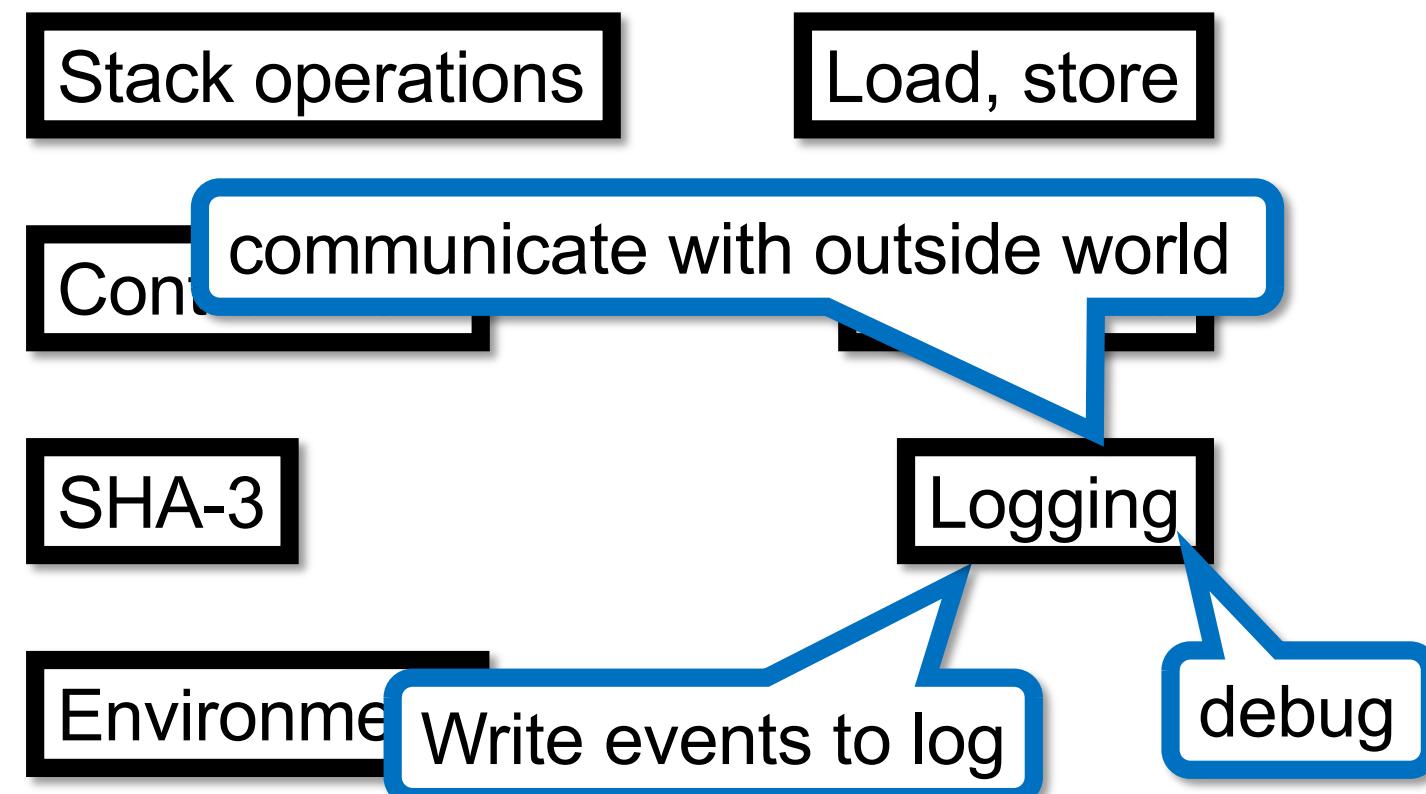
Environment

Load and store
from non-stack memory

Types of Instructions



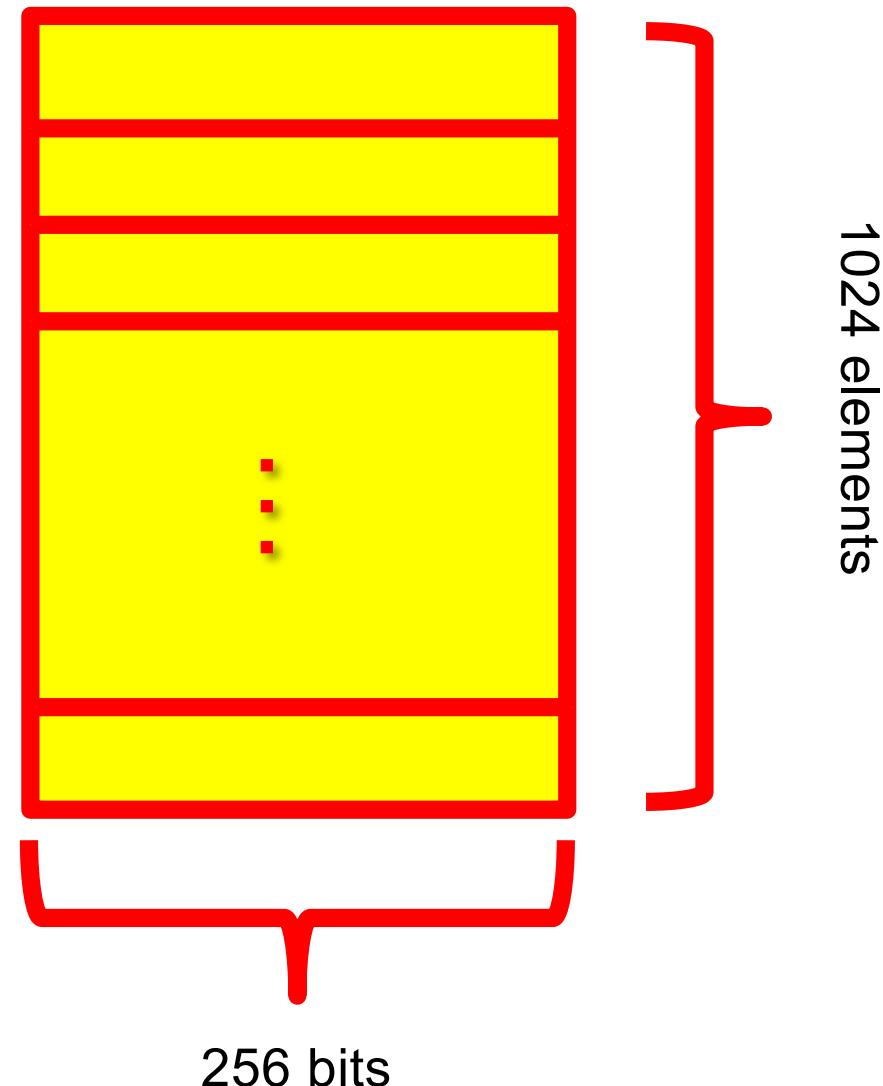
Types of Instructions



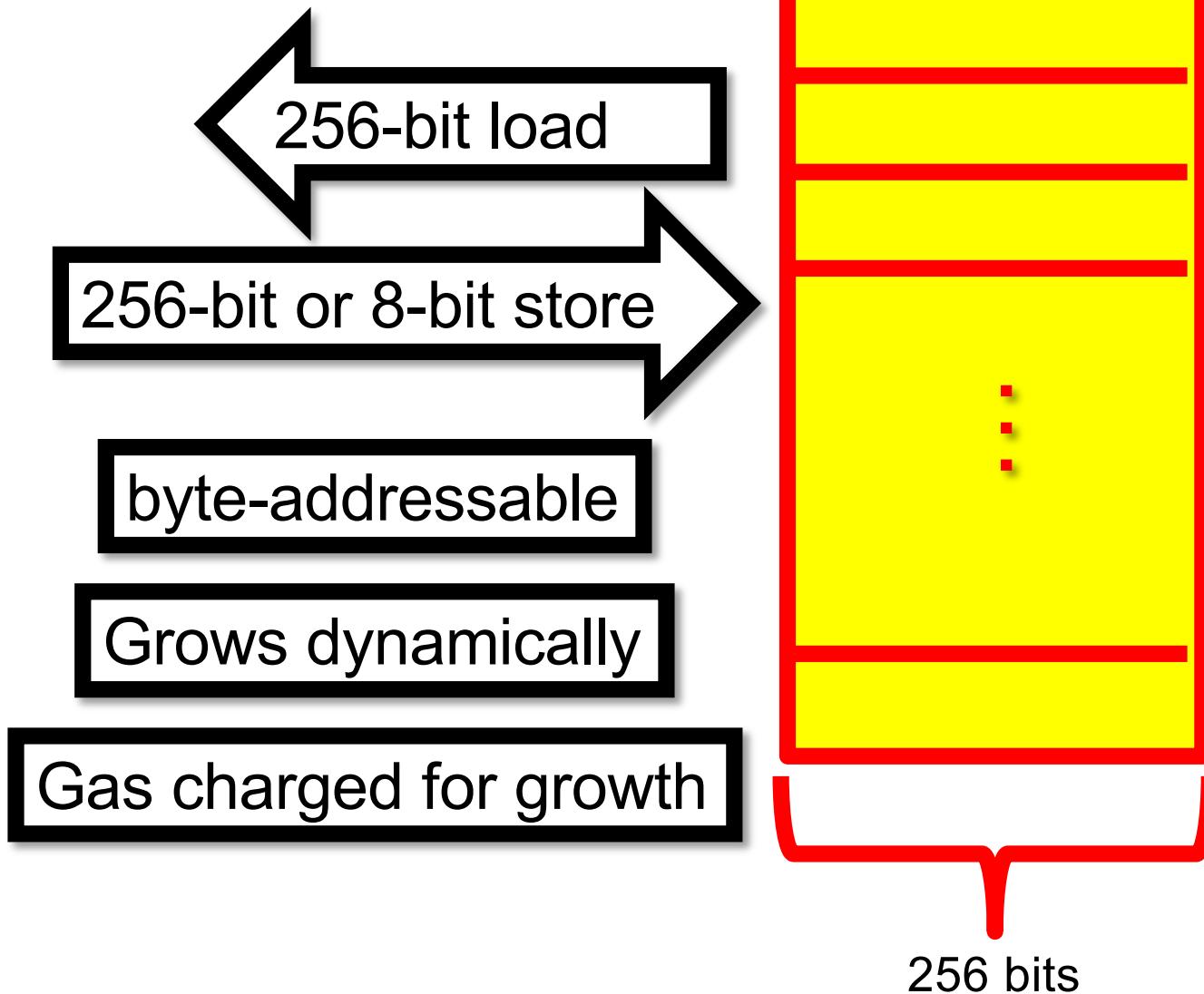
EVM Stack

All operations
act on stack
(e.g., arithmetic
Logic, etc.)

PUSH, POP,
SWAP ...



EVM Memory



EVM Storage

- API/abstraction is identical to memory, but values persist across transactions
- Storage is persisted in the blockchain itself



Storage is Expensive

Users need incentive to free up unneeded storage

Ethereum gives a gas refund for

Freeing storage (overwrite with 0s)

Destroying contracts

Gas Token

(Ethereum token that represents a storage of gas)

GasToken managed by a contract:

Leverage gas refund to “store” gas by creating empty contracts or storage, then tokenize

GasToken contract converts unused gas into GasToken tokens

Tokens can be transferred, sold, traded

Users can generate GasTokens when gas is cheap

GasToken is an Ethereum token that stores gas on the Ethereum network, storing gas when it is expensive. Using GasToken can save users to tokenize gas everything from arbitraging decentralized exchanges to the first contract on the Ethereum network that "hanks" of gas that

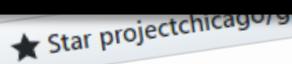
Gas Token Speculation

When gas prices are low

Buy GasTokens

Opposite strategy when prices climb

Speculators aim to profit from price fluctuations



GasToken is a new, cutting-edge Ethereum contract that allows users to tokenize gas on the Ethereum network, storing gas when it is cheap and using / deploying this gas when it is expensive. Using GasToken can subsidize high gas prices on everything from arbitraging decentralized exchanges to buying the first contract on the Ethereum network that "hanks" of gas that

Is That Ethical?

Negative effects:

Uses memory only for speculation

GasToken might drive up prices

Unethical behavior can arise

Is That Ethical?

Positive Effects

Gas-banking service for users

Hedge against increases

Predictability in face of volatility

Further Reading and Acknowledgements

- Ethereum: A Secure Decentralised Generalized Transaction Ledger
- Slides adapted from Maurice Herlihy, Brown University, available under CC BY-NC 4.0 (<https://creativecommons.org/licenses/by-nc/4.0/>)