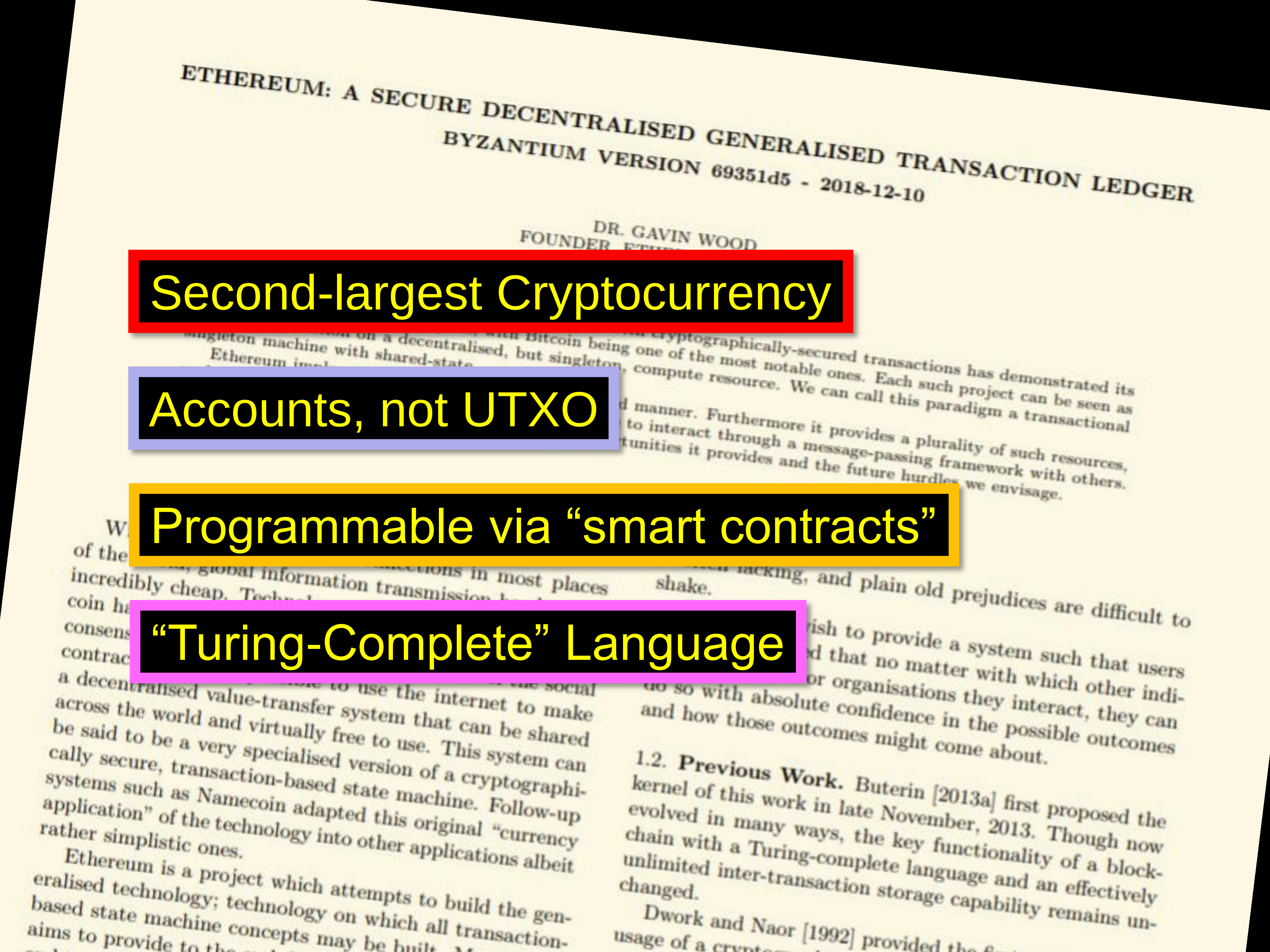


ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER
BYZANTIUM VERSION 69351d5 - 2018-12-10

DR. GAVIN WOOD
FOUNDER, ETH

cryptographically-secured transactions has demonstrated its
with Bitcoin being one of the most notable ones. Each such project can be seen as
on a decentralised, but singleton, compute resource. We can call this paradigm a transactional
singleton machine with shared-state
Ethereum impl
d manner. Furthermore it provides a plurality of such resources,
to interact through a message-passing framework with others.
tunities it provides and the future hurdles we envisage.

W
nnections in most places
of the
ld, global information transmission ha
incredibly cheap. Techn
coin ha
consens
contrac
the social
ble to use the internet to make
a decentralised value-transfer system that can be shared across the world and virtually free to use. This system can be said to be a very specialised version of a cryptographically secure, transaction-based state machine. Follow-up systems such as Namecoin adapted this original "currency application" of the technology into other applications albeit rather simplistic ones.

Ethereum is a project which attempts to build the generalised technology; technology on which all transaction-based state machine concepts may be built. M
aims to provide to th

en lacking, and plain old prejudices are difficult to shake.

ish to provide a system such that users
d that no matter with which other individuals
or organisations they interact, they can do so with absolute confidence in the possible outcomes and how those outcomes might come about.

1.2. **Previous Work.** Buterin [2013a] first proposed the kernel of this work in late November, 2013. Though now evolved in many ways, the key functionality of a blockchain with a Turing-complete language and an effectively unlimited inter-transaction storage capability remains unchanged.

Dwork and Naor [1992] provided the fi
usage of a crypto

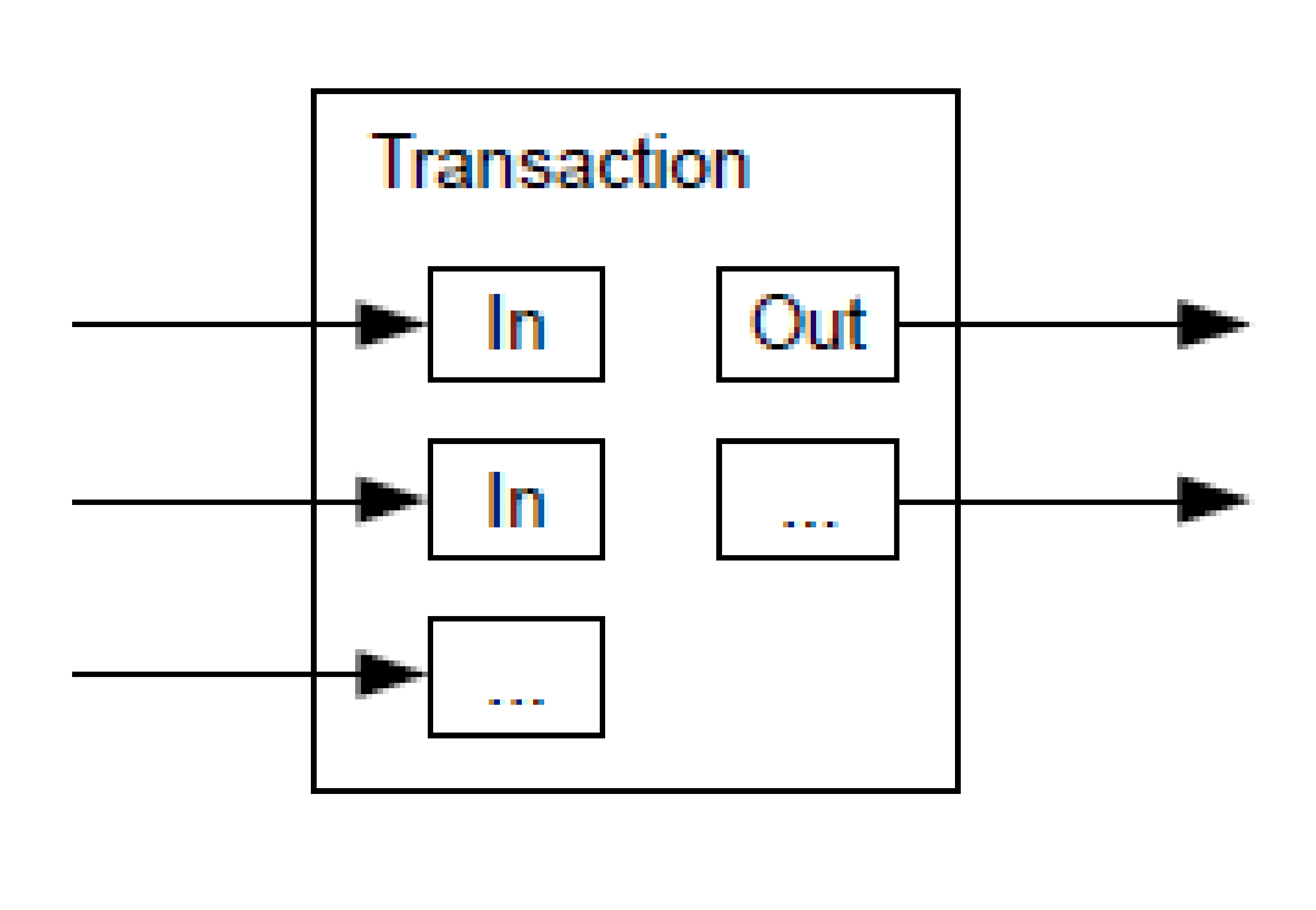
Second-largest Cryptocurrency

Accounts, not UTXO

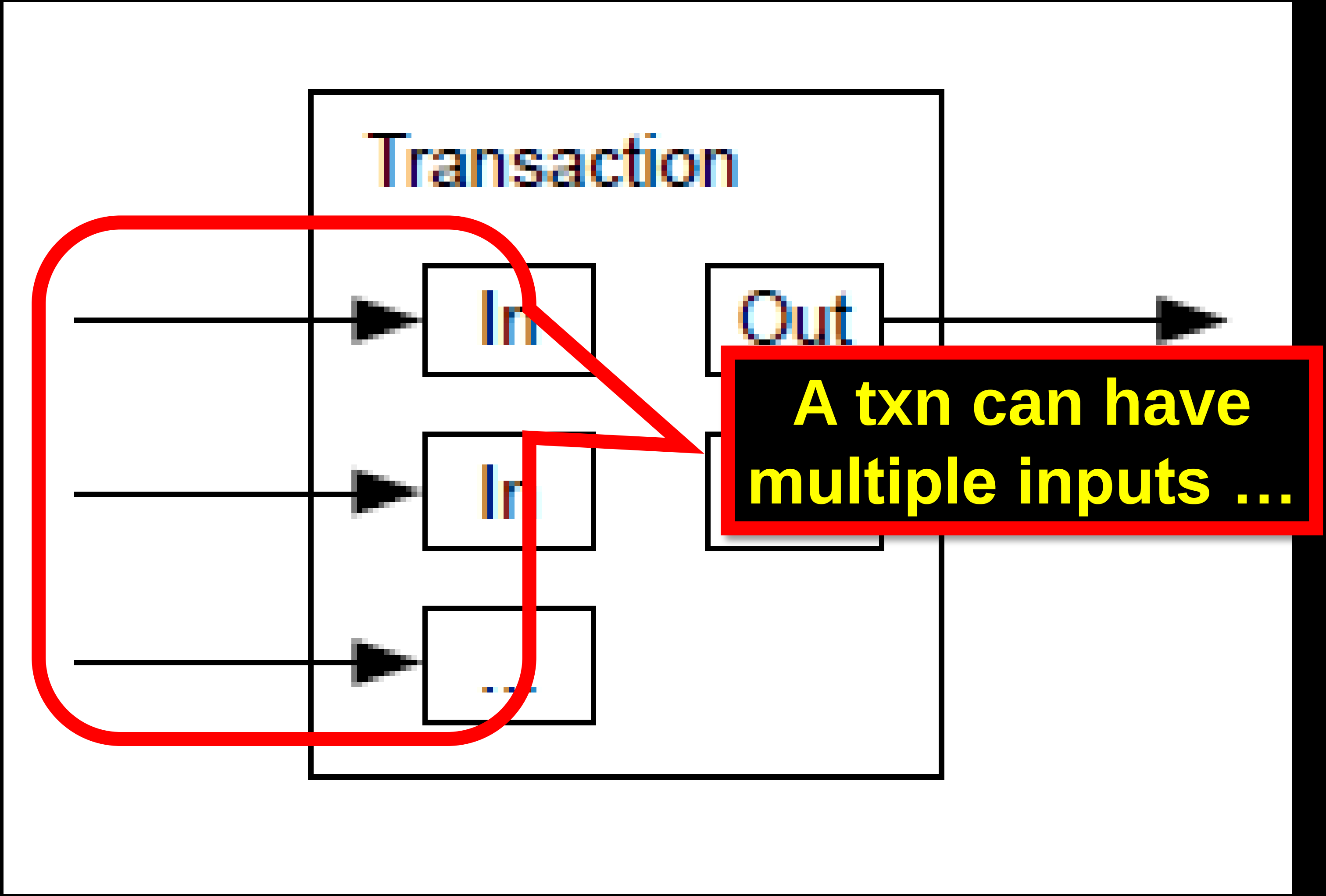
Programmable via "smart contracts"

"Turing-Complete" Language

Bitcoin UTXO Model

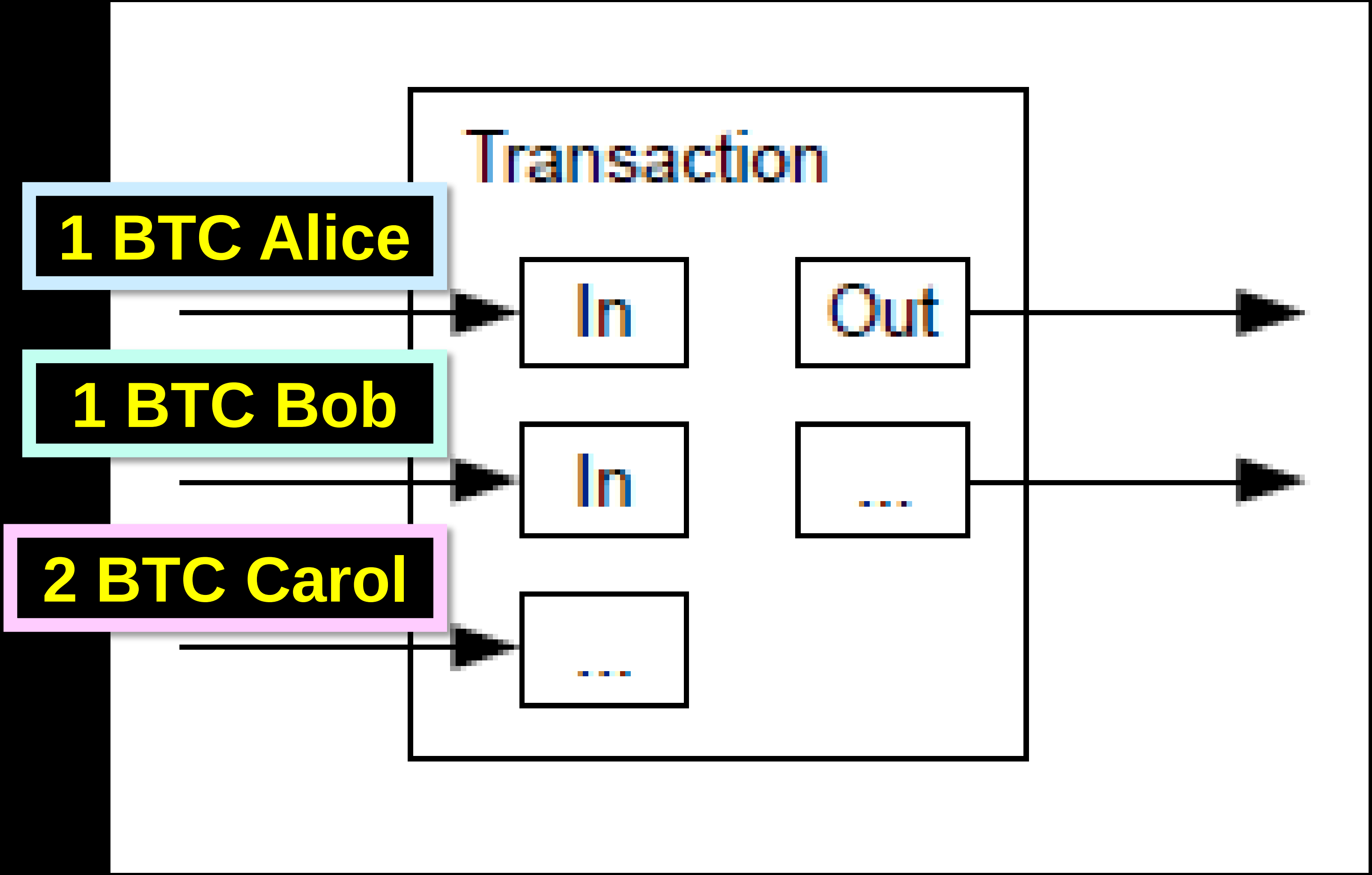


Bitcoin UTXO Model

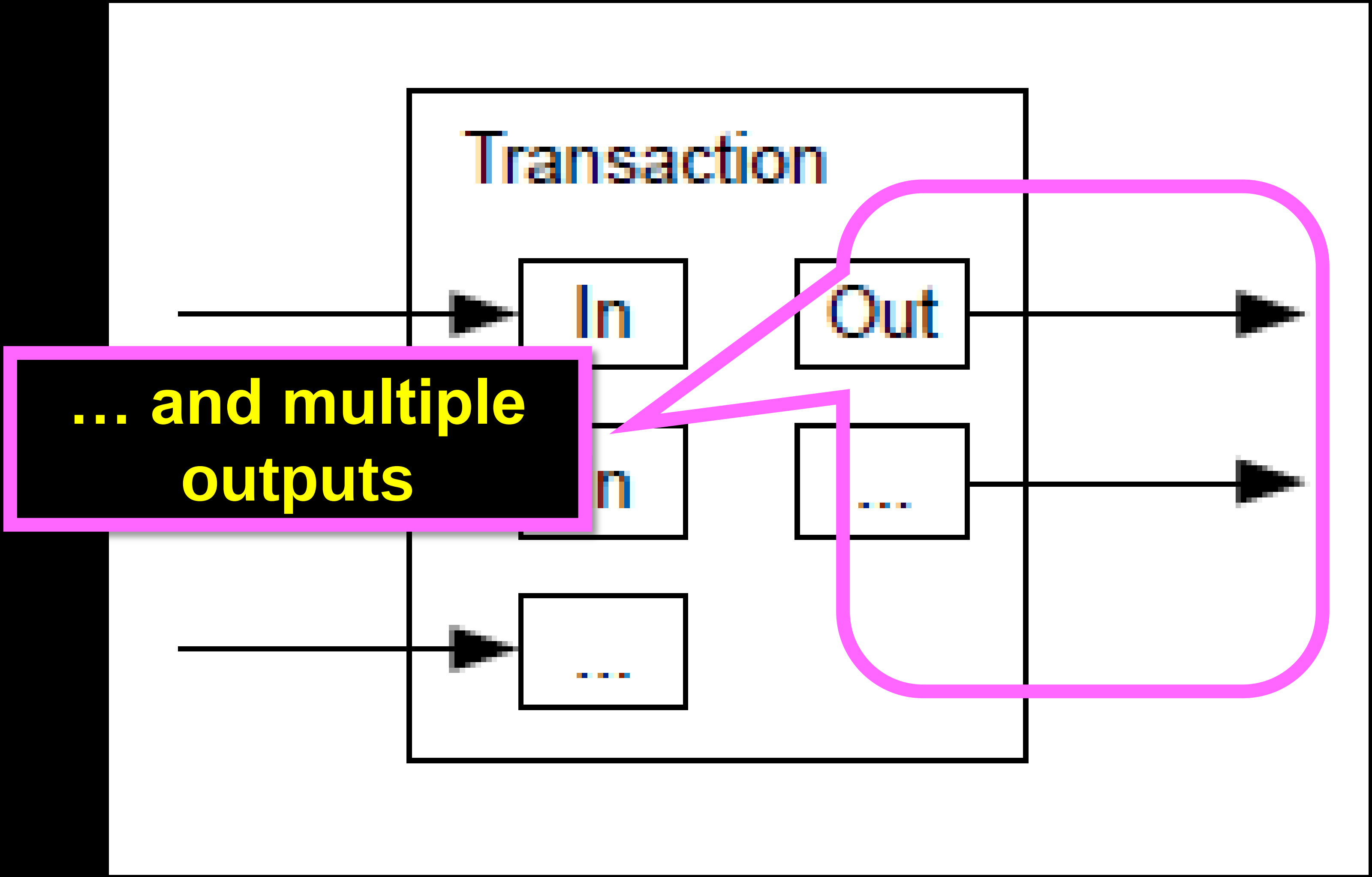


A txn can have multiple inputs …

Bitcoin UTXO Model

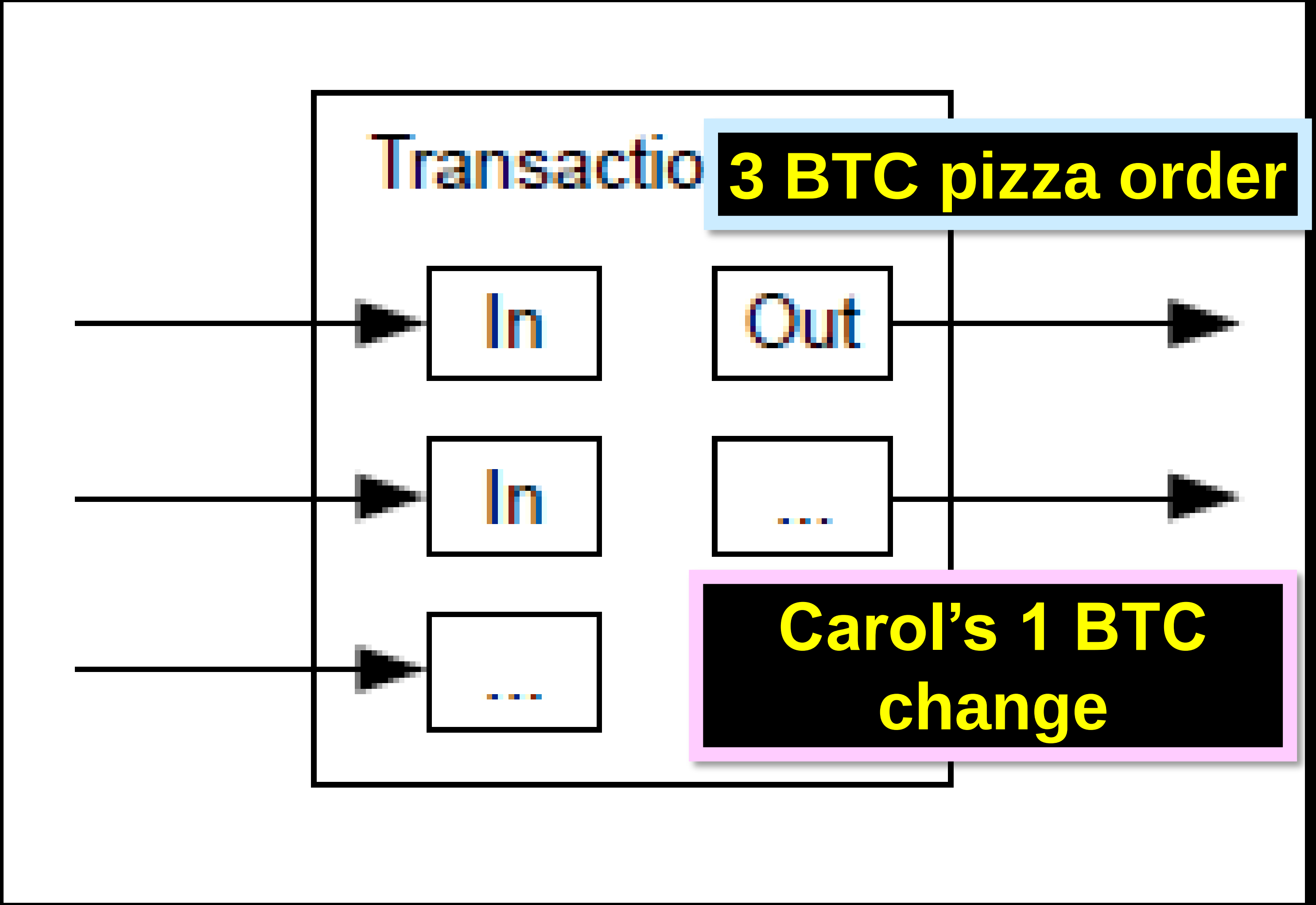


Bitcoin UTXO Model



… and multiple outputs

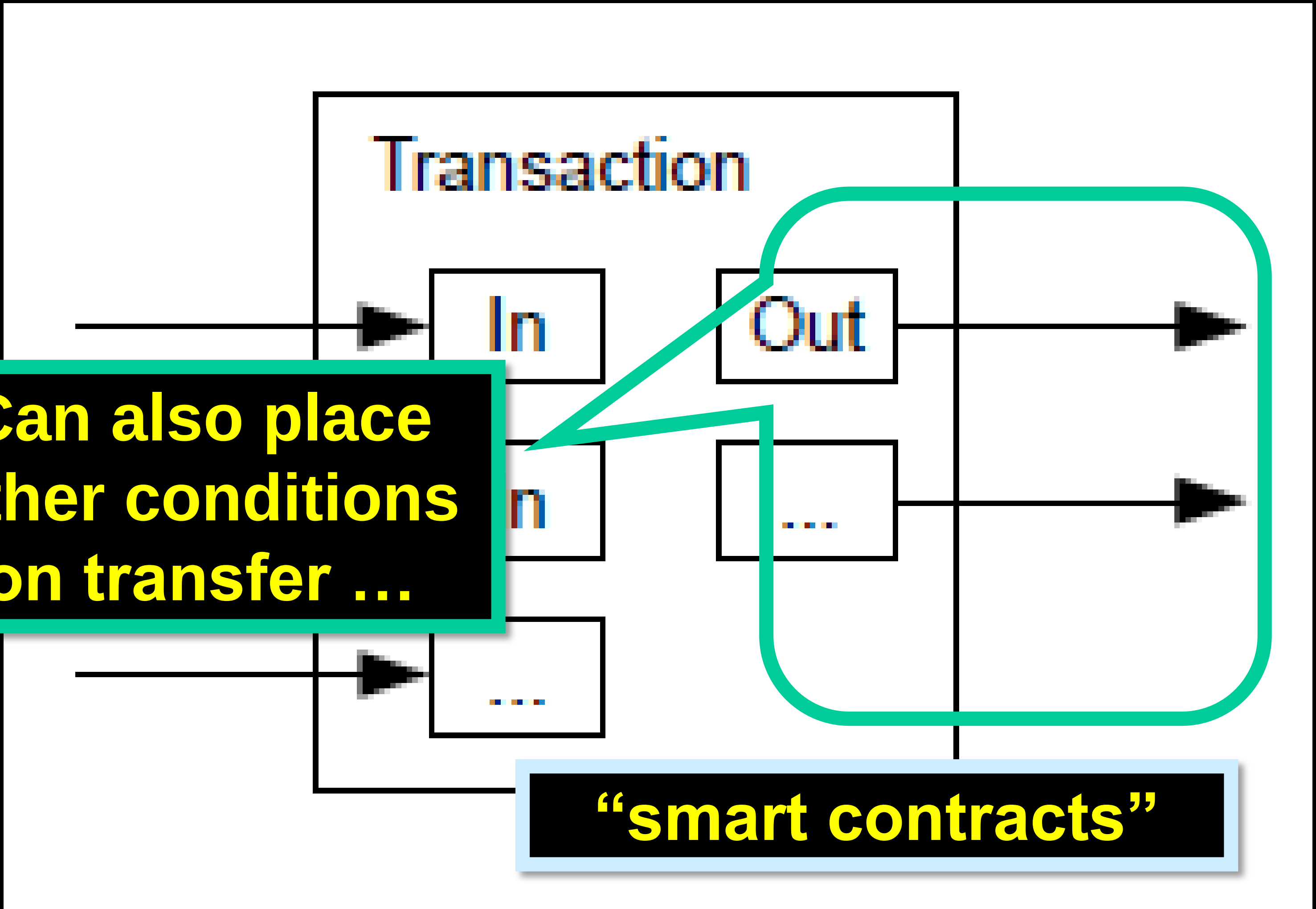
Bitcoin UTXO Model



3 BTC pizza order

Carol’s 1 BTC change

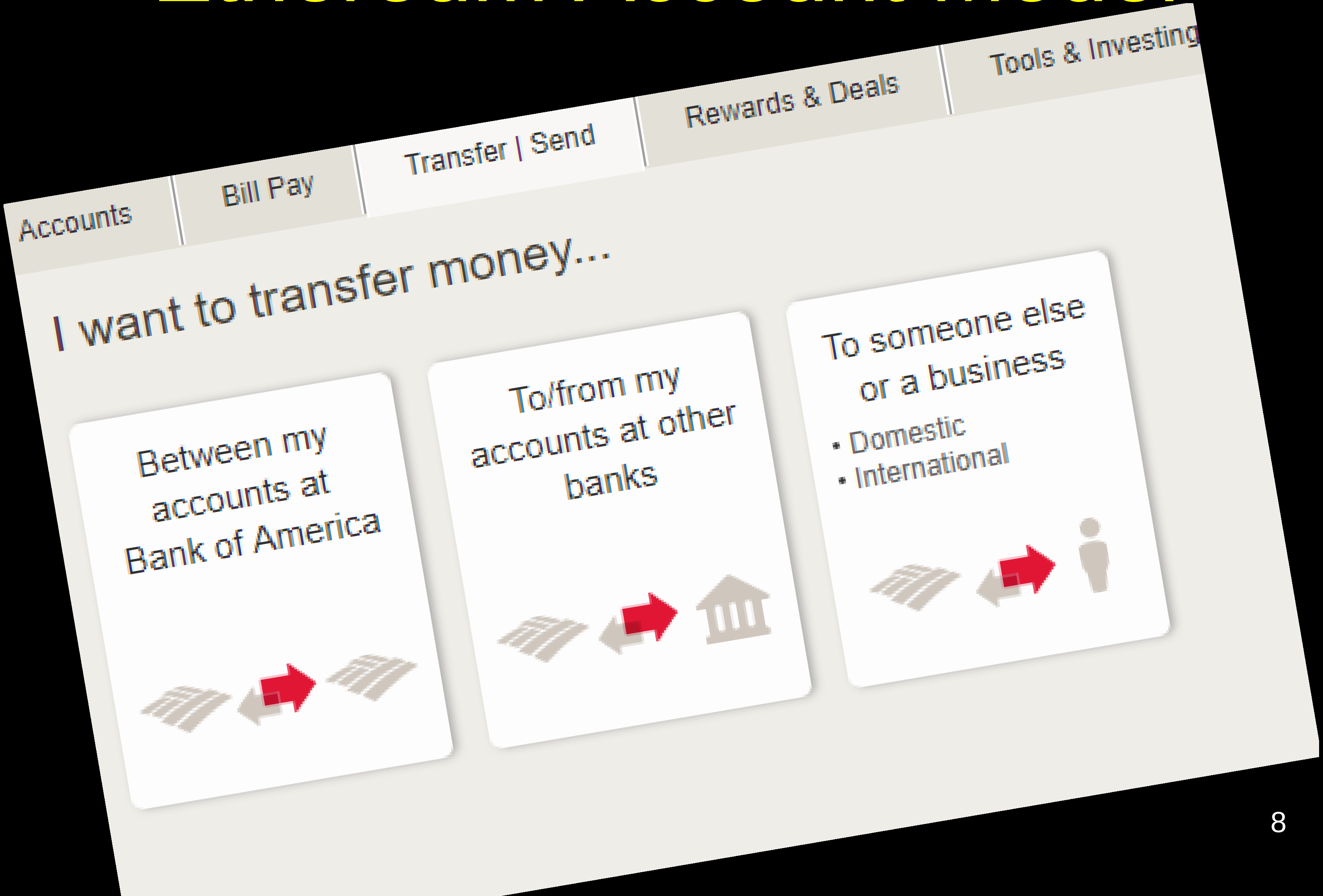
UTXO Model



Can also place other conditions on transfer …

“smart contracts”

Ethereum Account Model



Accounts | Bill Pay | Transfer | Send | Rewards & Deals | Tools & Investing

I want to transfer money...

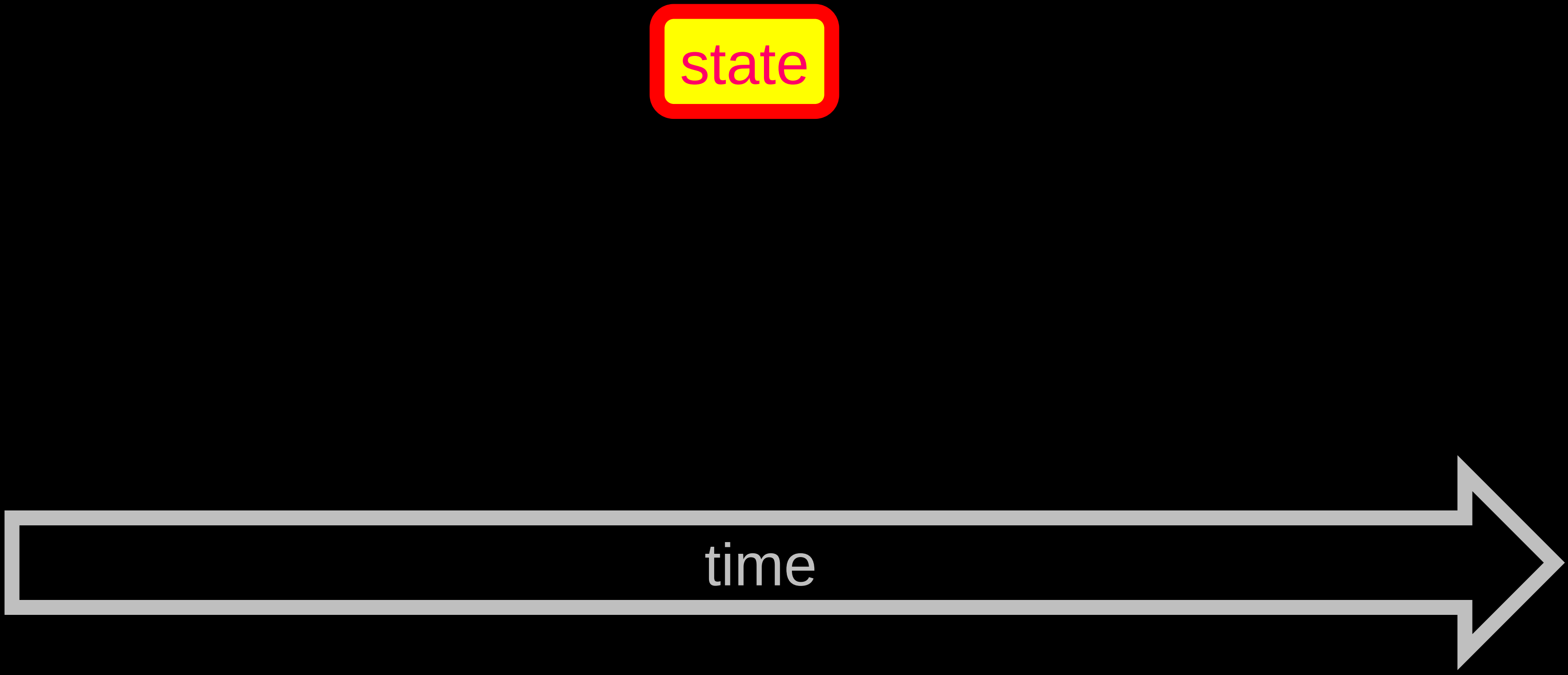
Between my accounts at Bank of America

To/from my accounts at other banks

To someone else or a business

- Domestic
- International

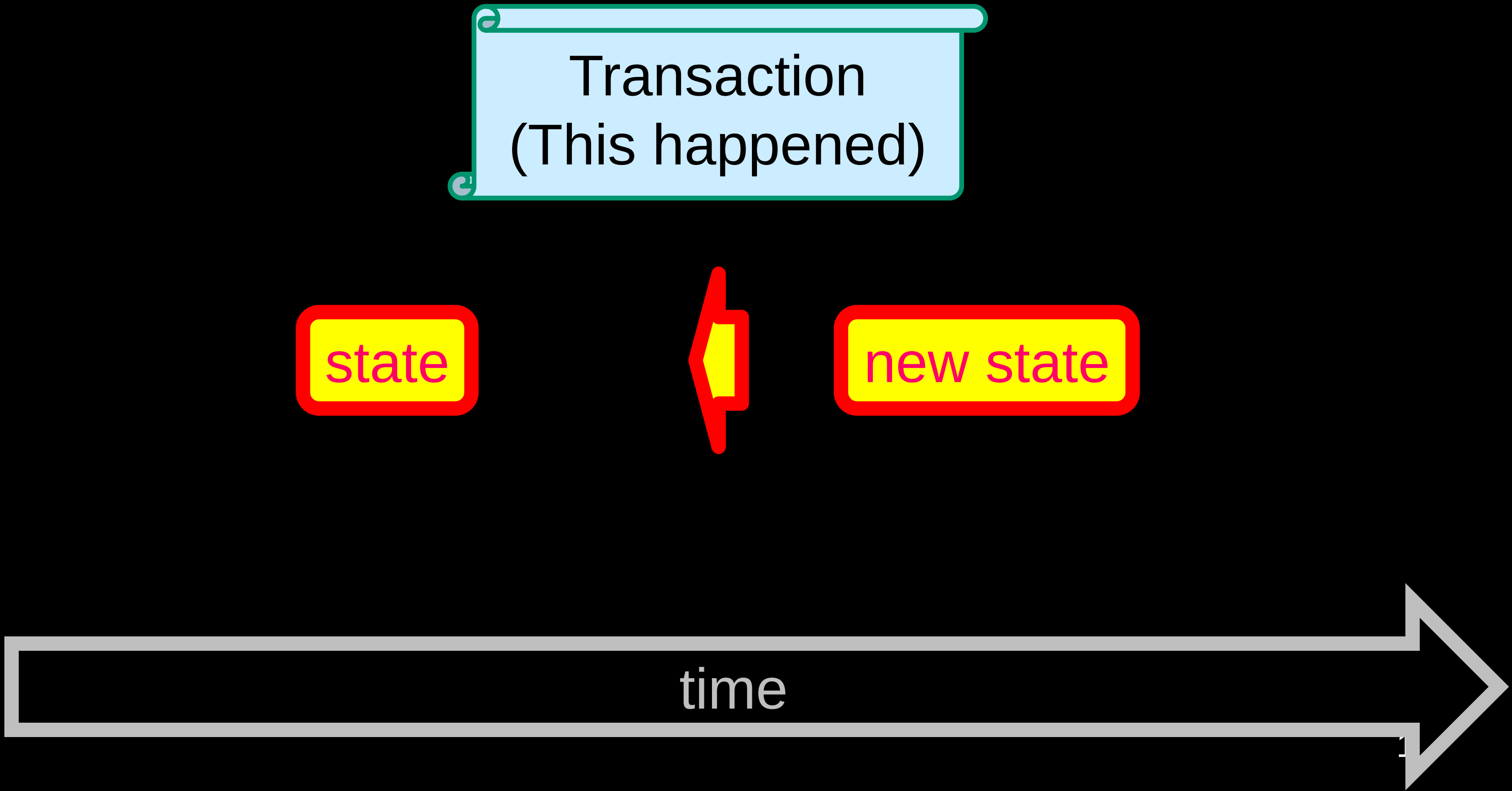
In the beginning …



state

time

On the first day …



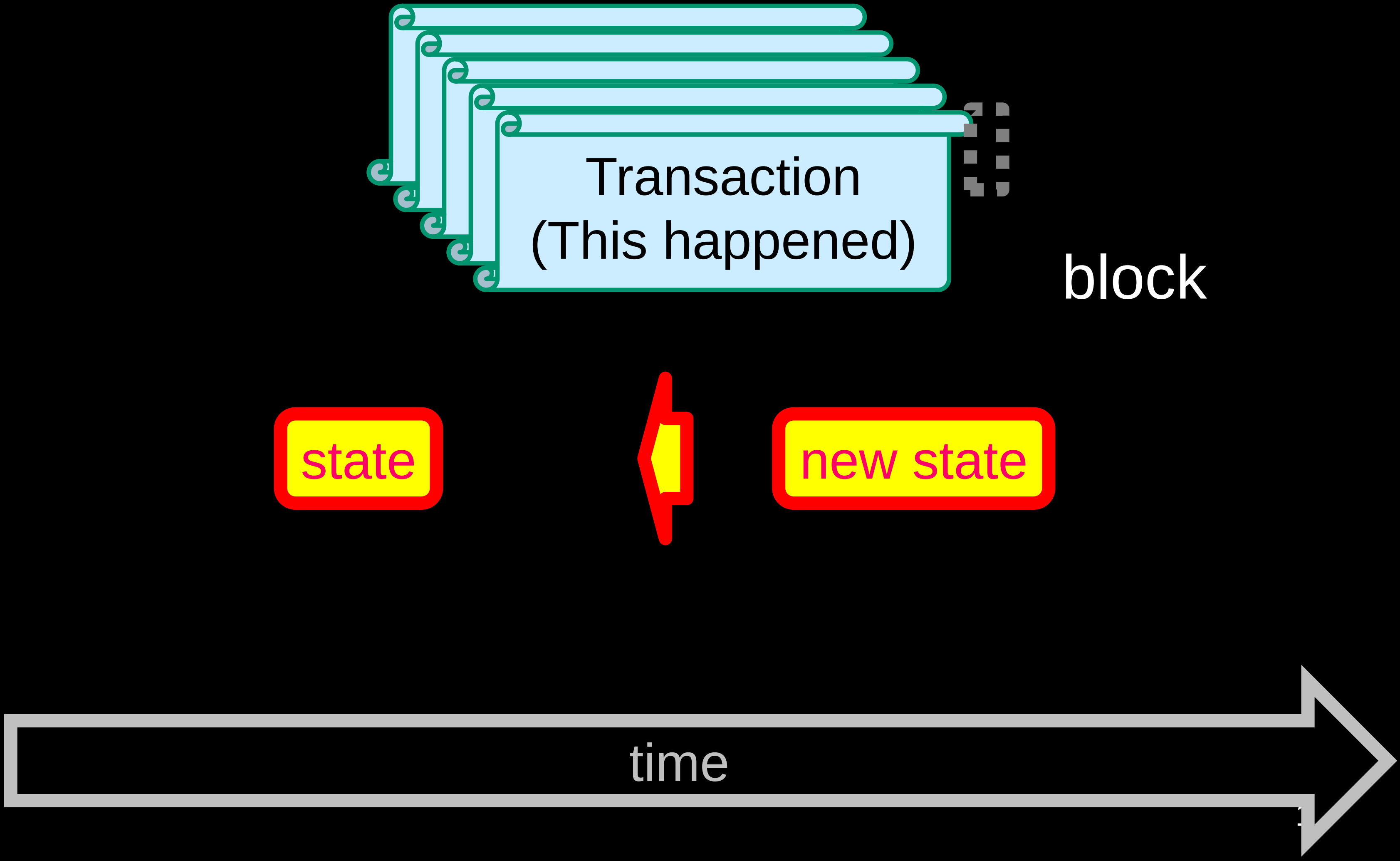
Transaction
(This happened)

state

new state

time

Not long after …



Transaction
(This happened)

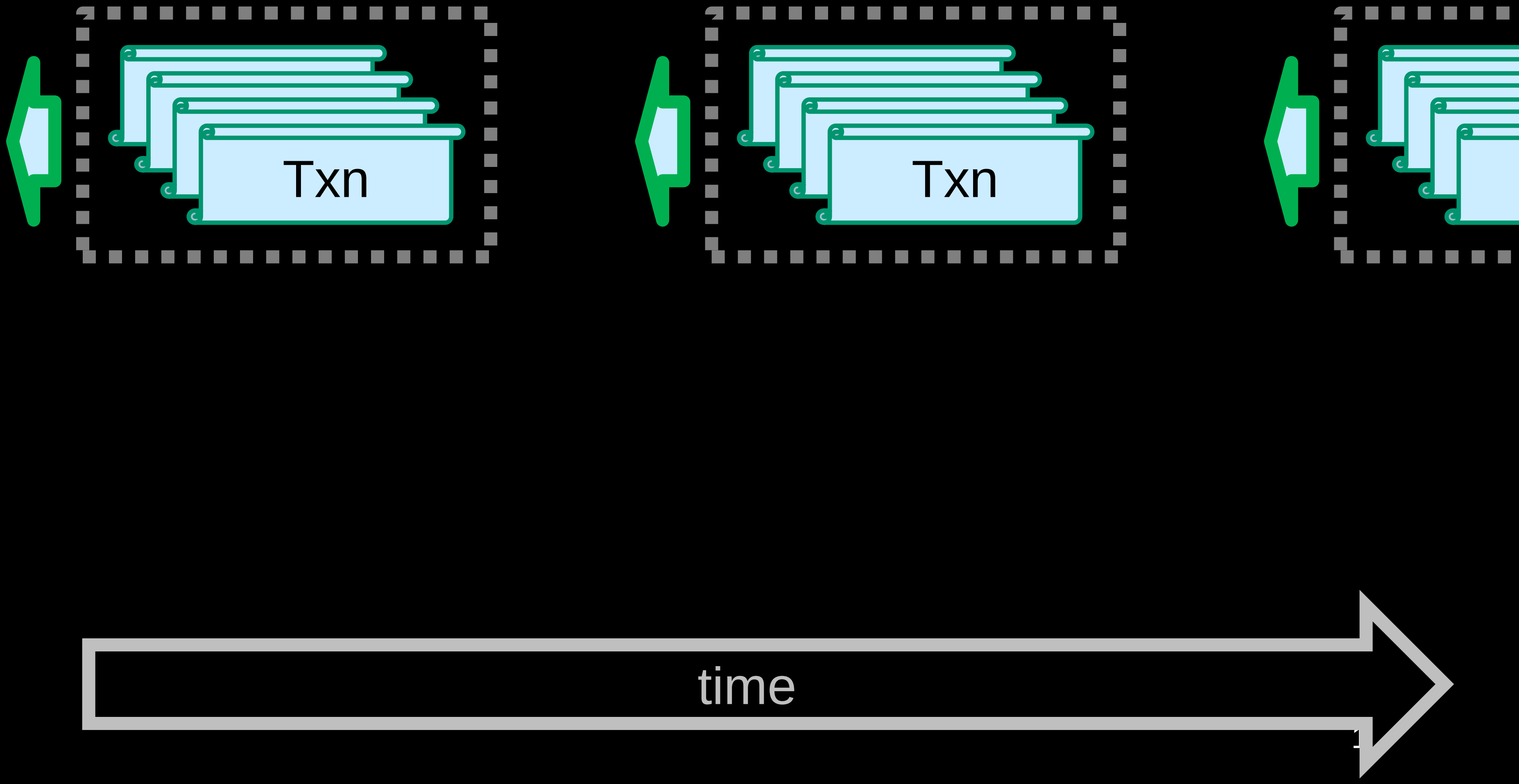
block

state

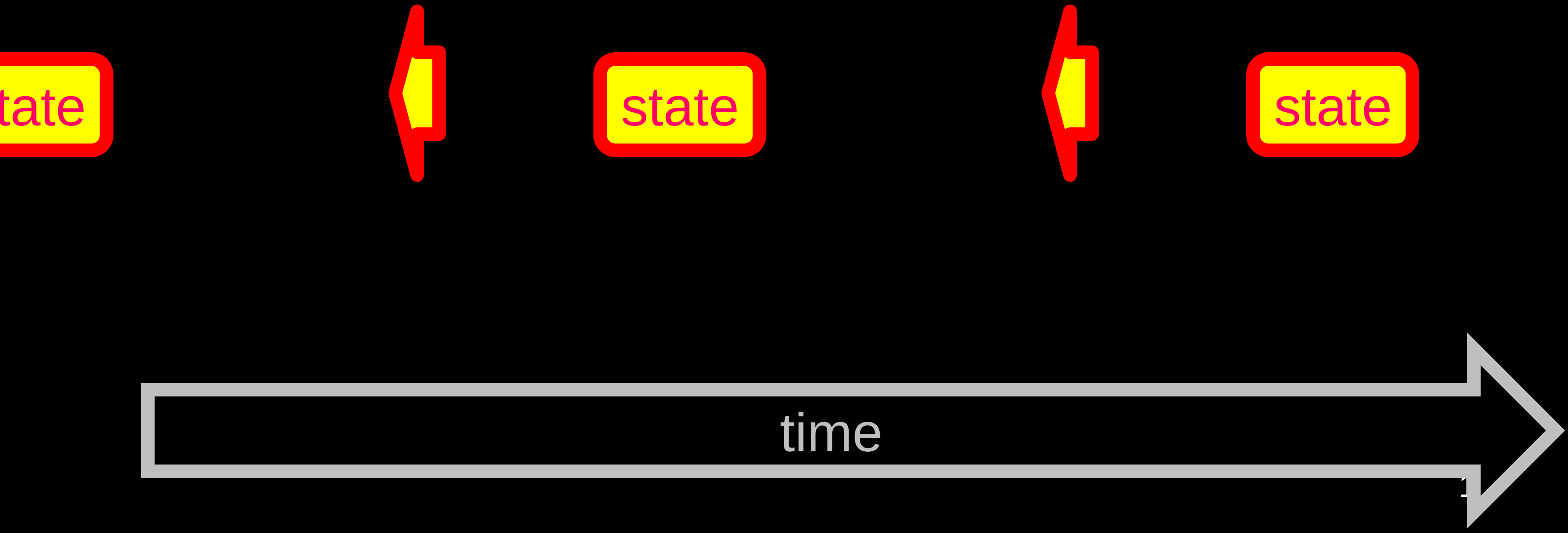
new state

time

Chain of Blocks



Chain of States



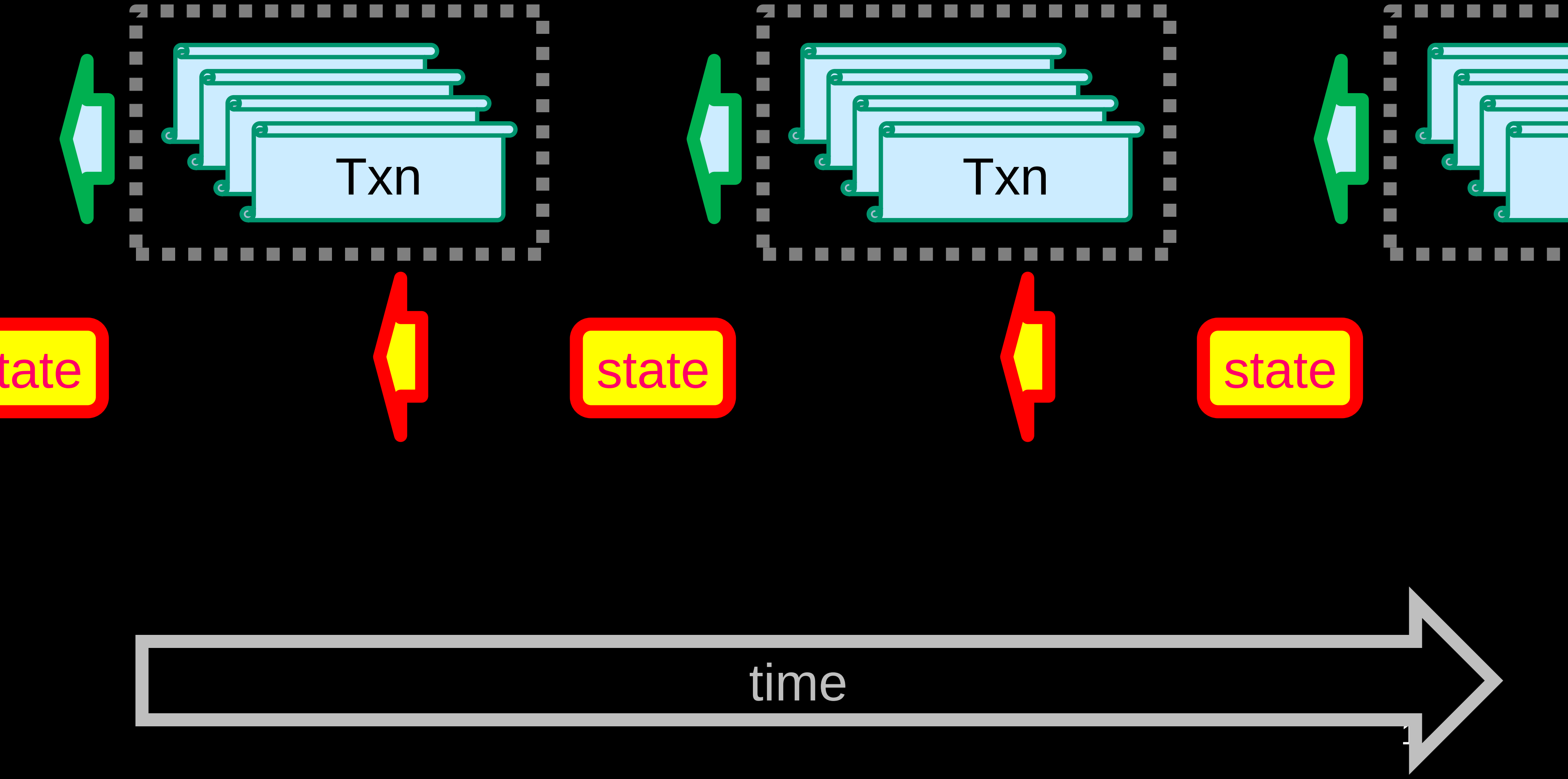
tate

state

state

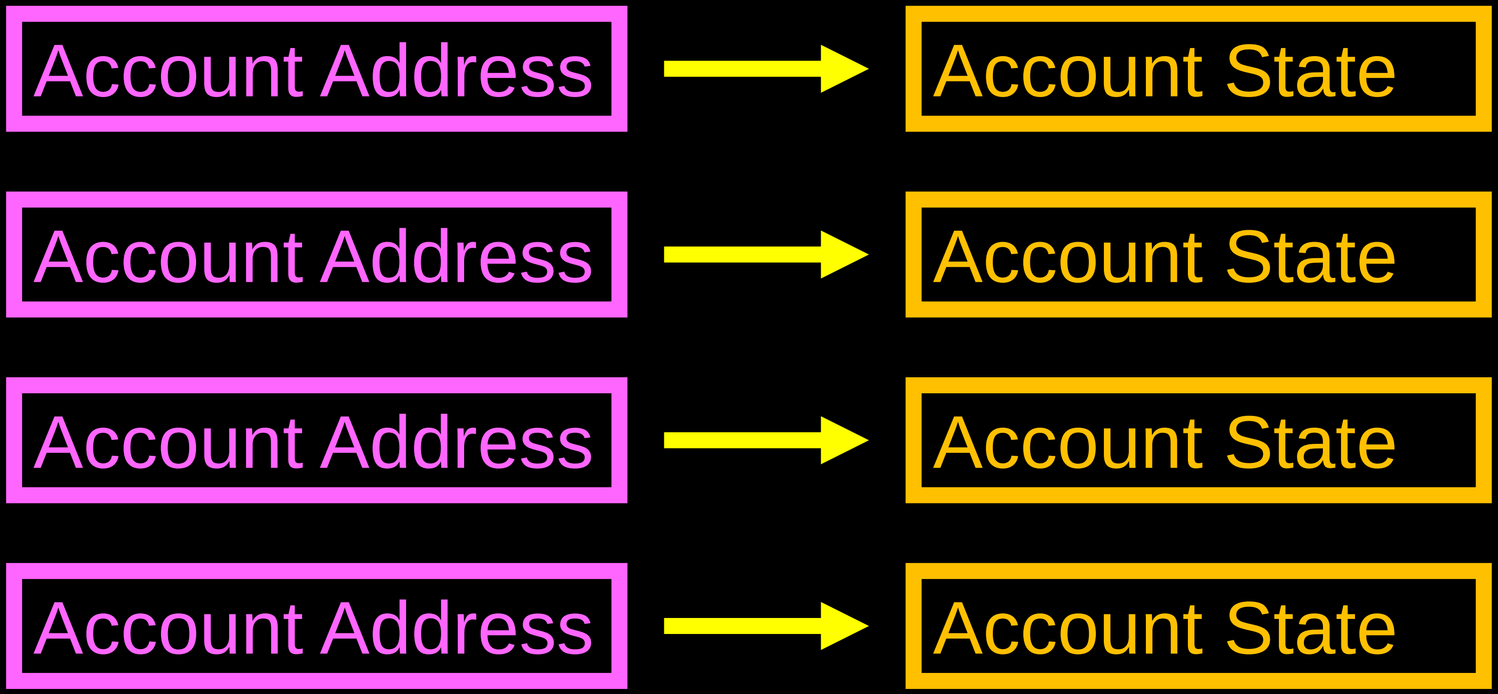
time

Block-State Duality



Ethereum State

Ethereum State



Account Address → Account State

Account Address → Account State

Account Address → Account State

Account Address → Account State

External Account

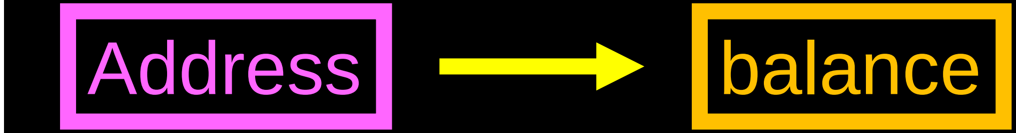
Owned by person or organization

Controlled by private keys

Holds currency balance

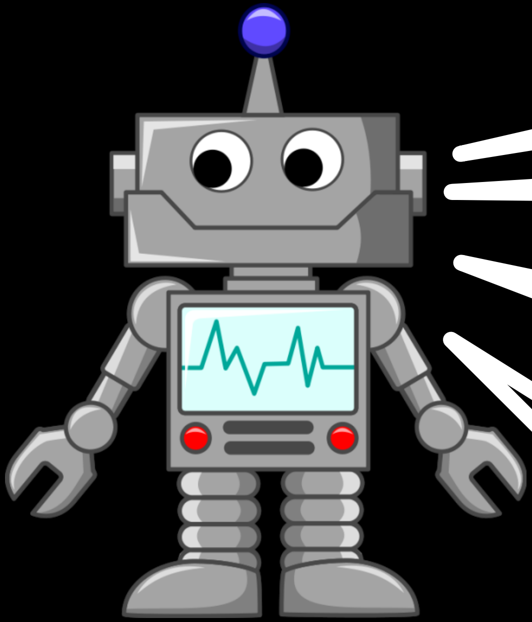
Active agent: transfers currency, calls contract code

External Account



Address → balance

Contract Account



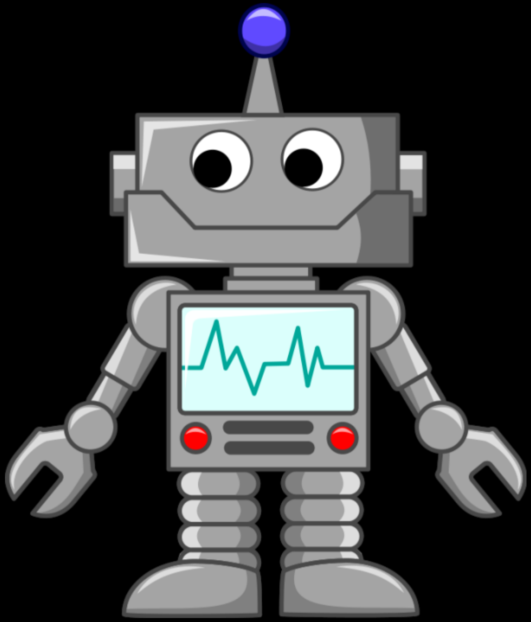
Holds currency balance

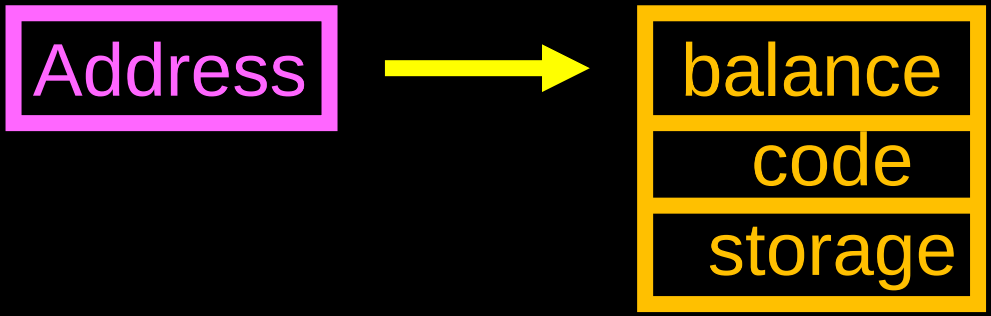
Associated with code

Passive Agent:
code called by other accounts

code can transfer money,
call other contracts

Contract Account





Address

balance

code

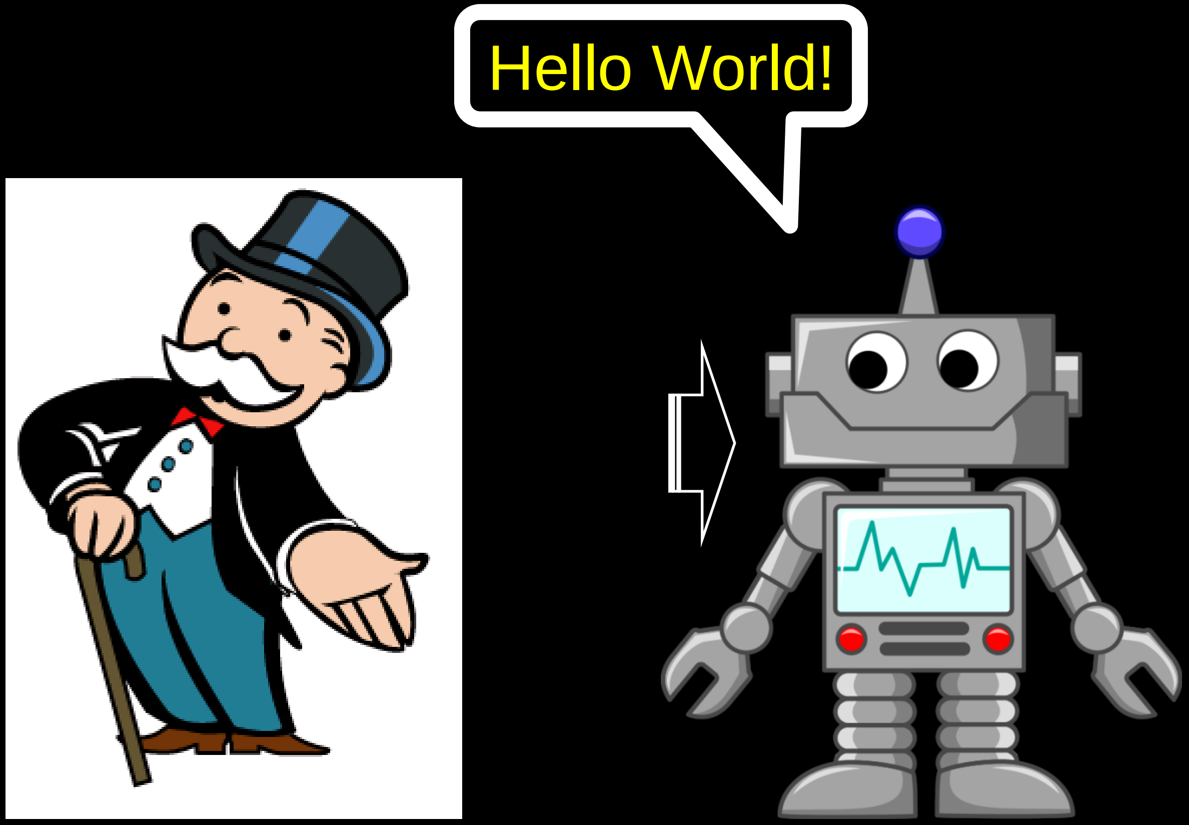
storage

Transaction Creation

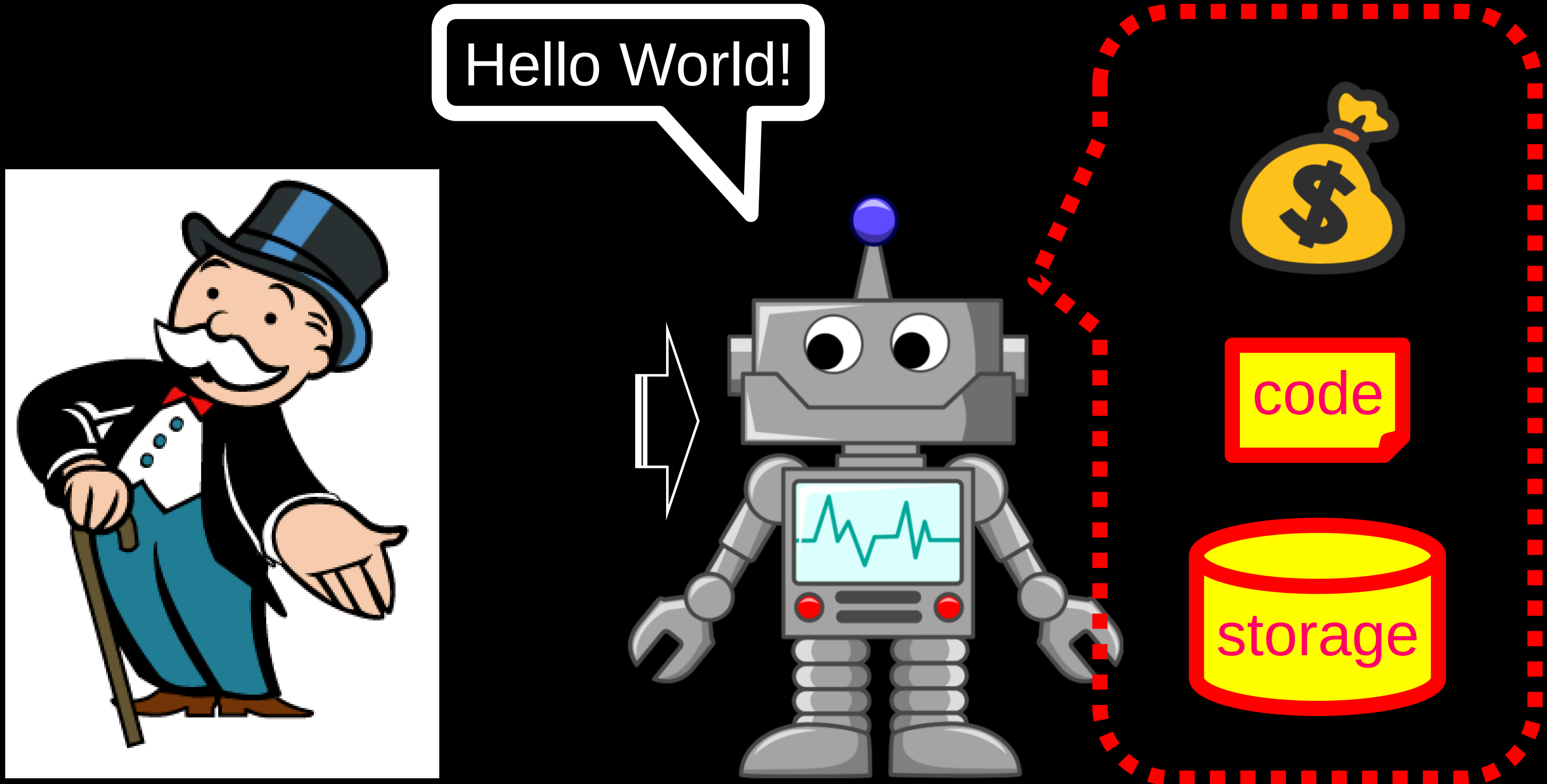
Submitted by external party

Contract Creation Transaction

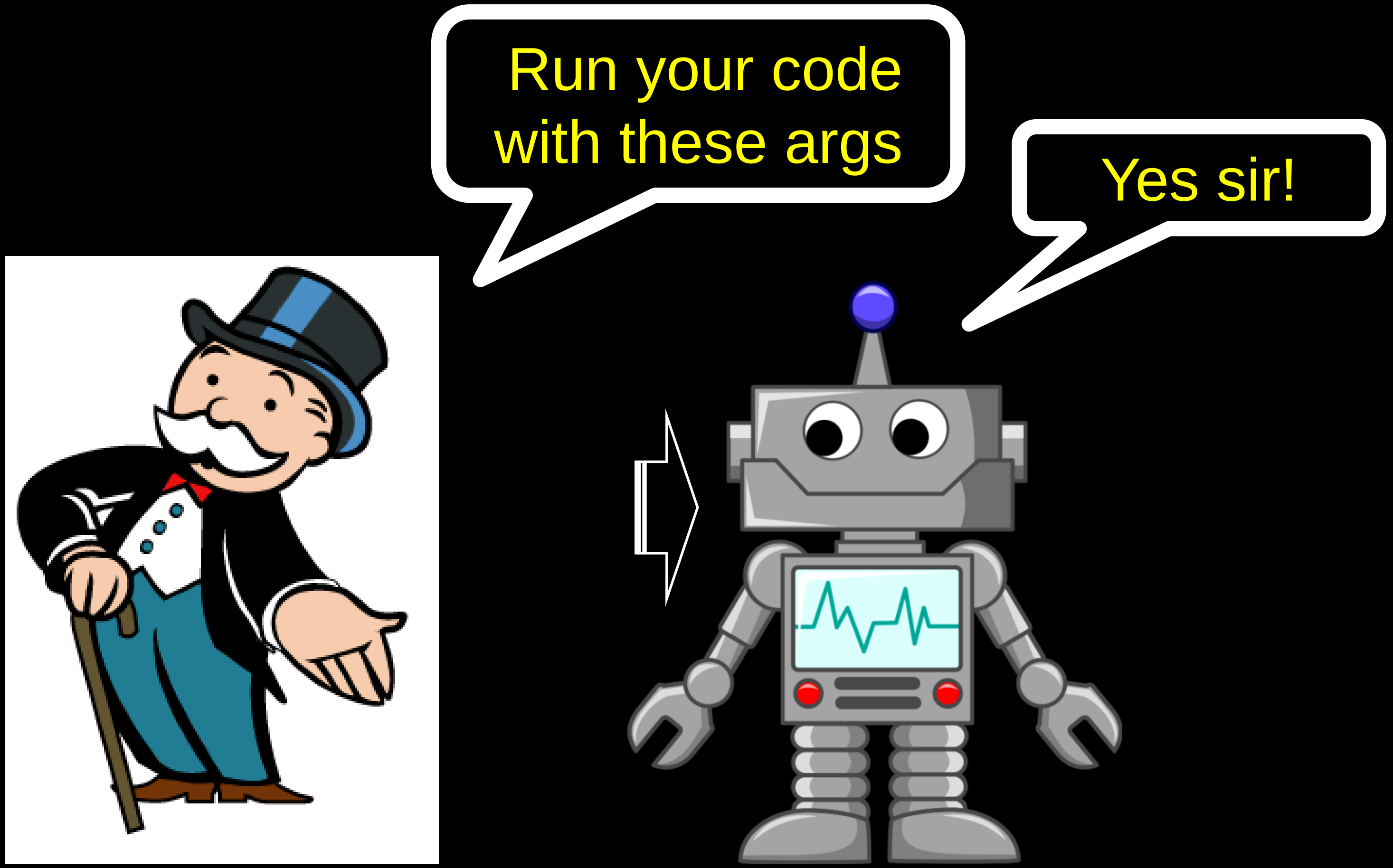
Contract Creation Transaction



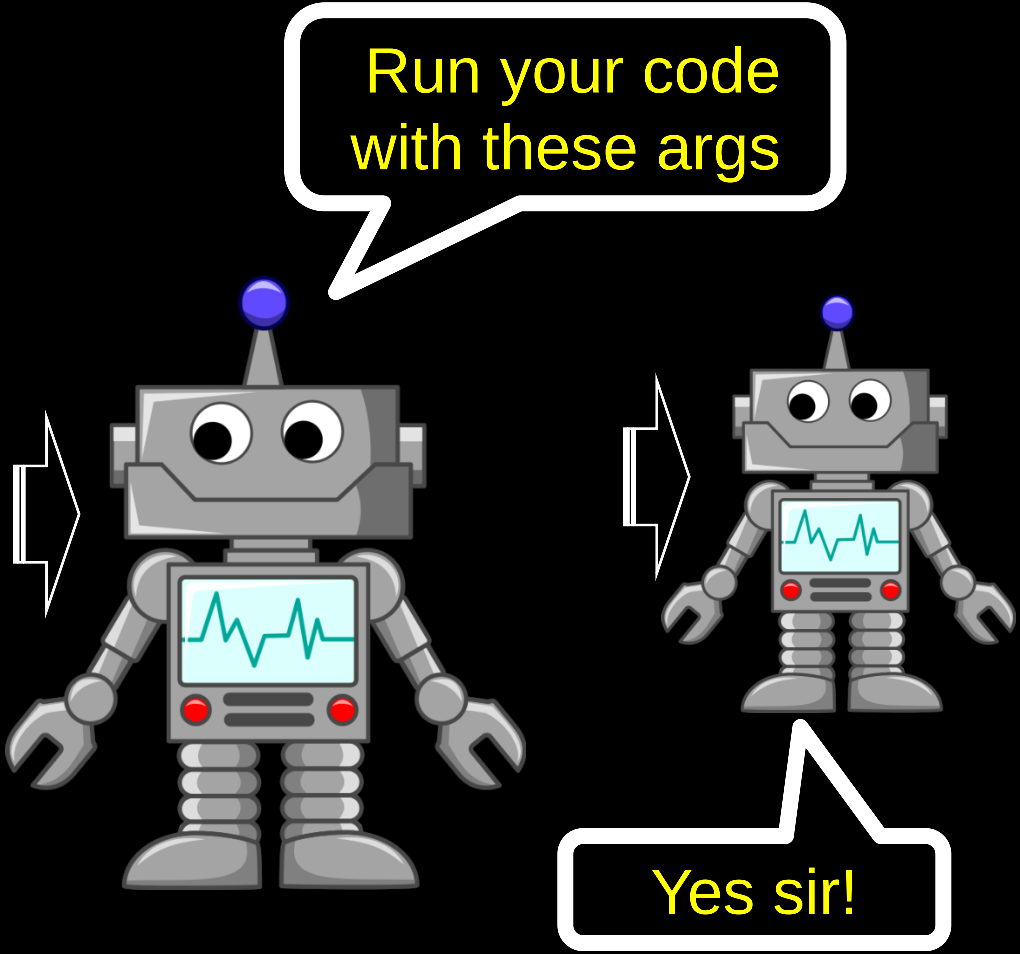
Contract Creation Transaction



Message Call Transaction



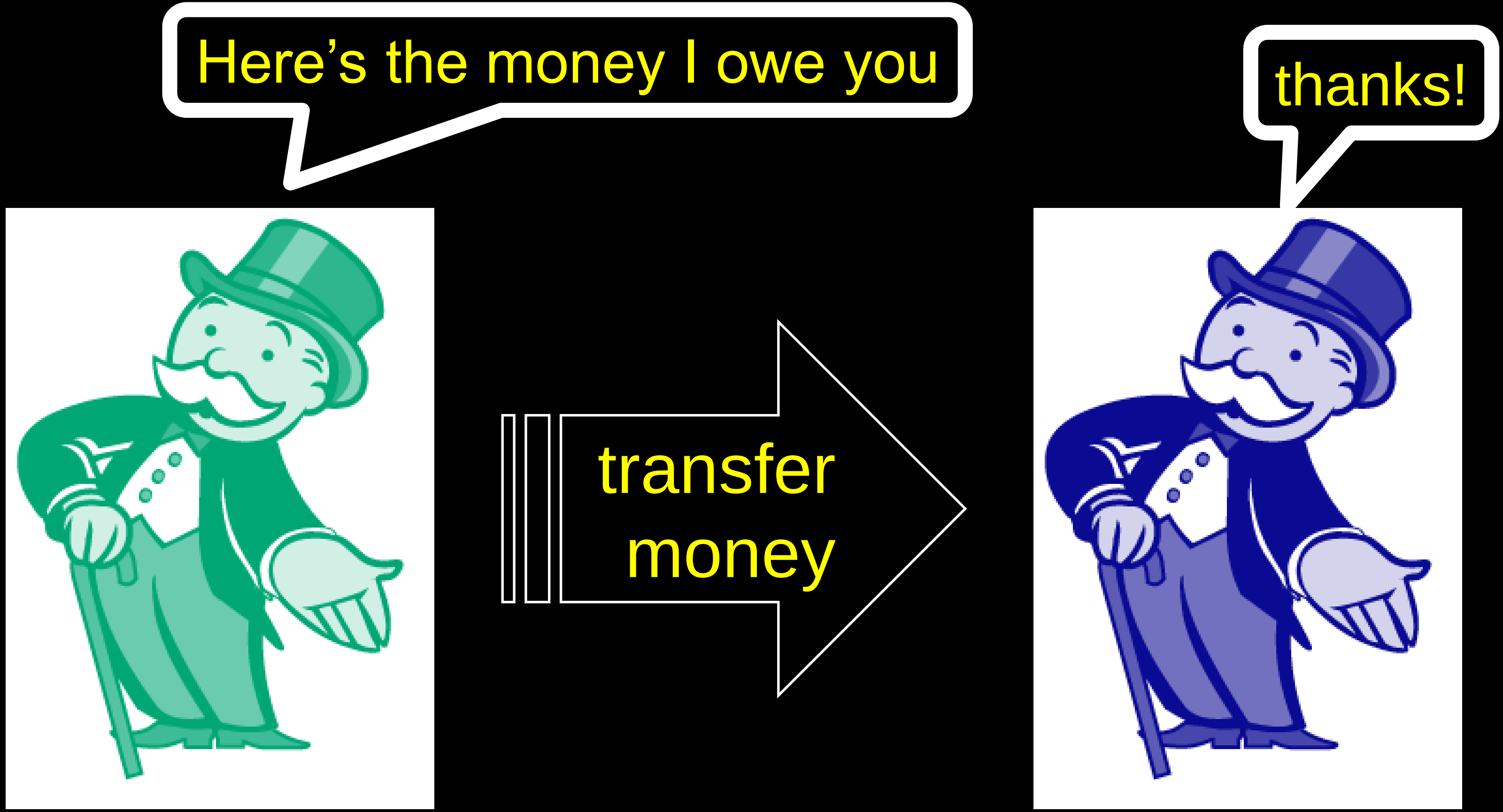
Message Call Transaction



Run your code with these args

Yes sir!

External to External Message

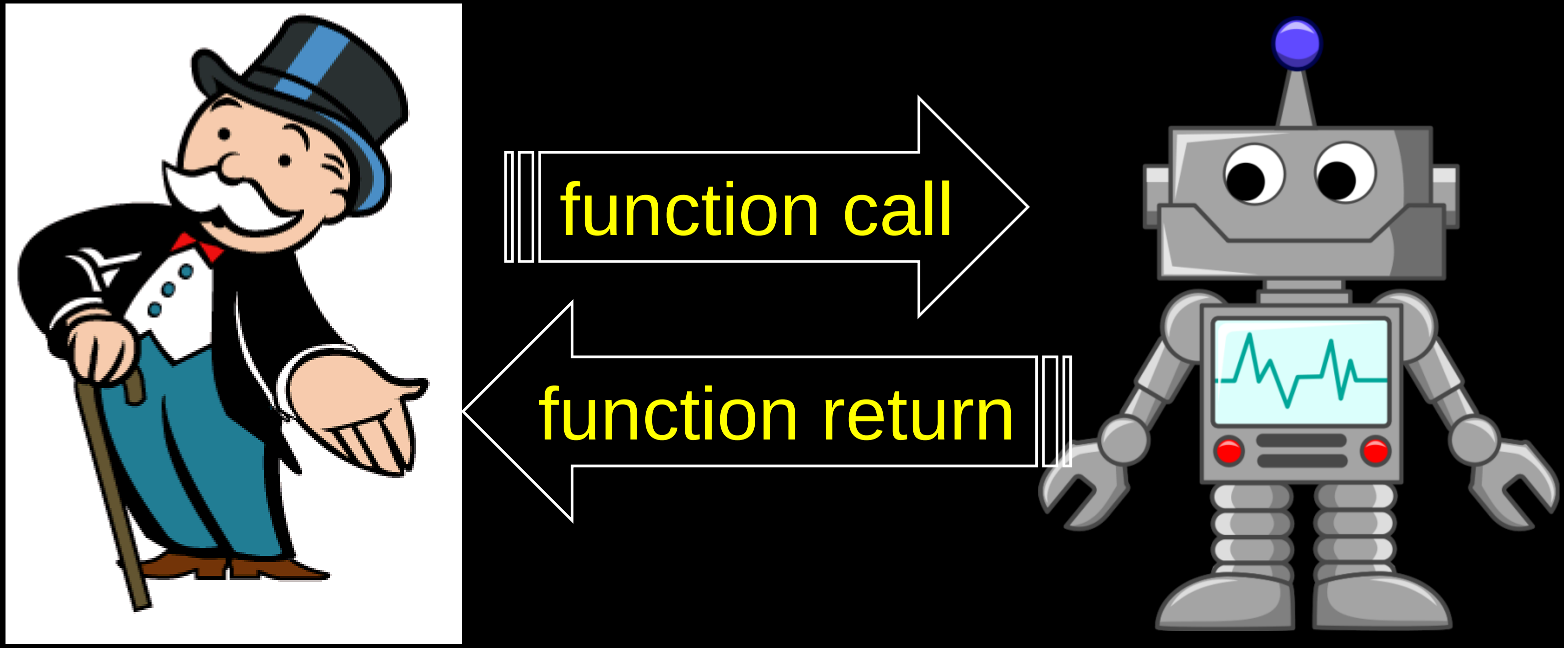


Here's the money I owe you

thanks!

transfer money

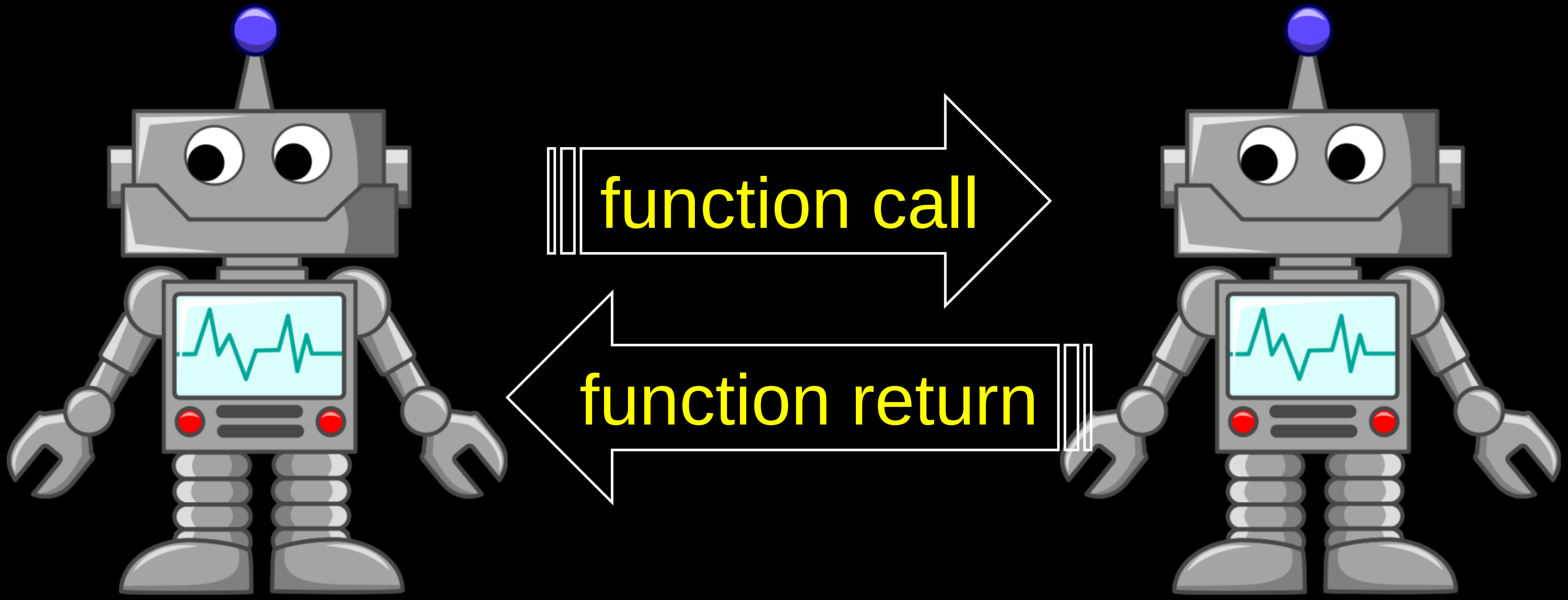
External to Contract & Vice-Versa



function call

function return

Contract to Contract Message



function call

function return

Questions?

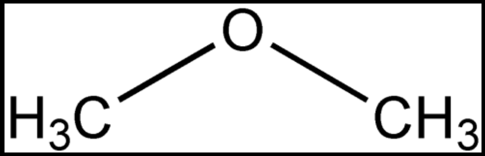
Take 15 seconds to reflect …

Unmute and ask!

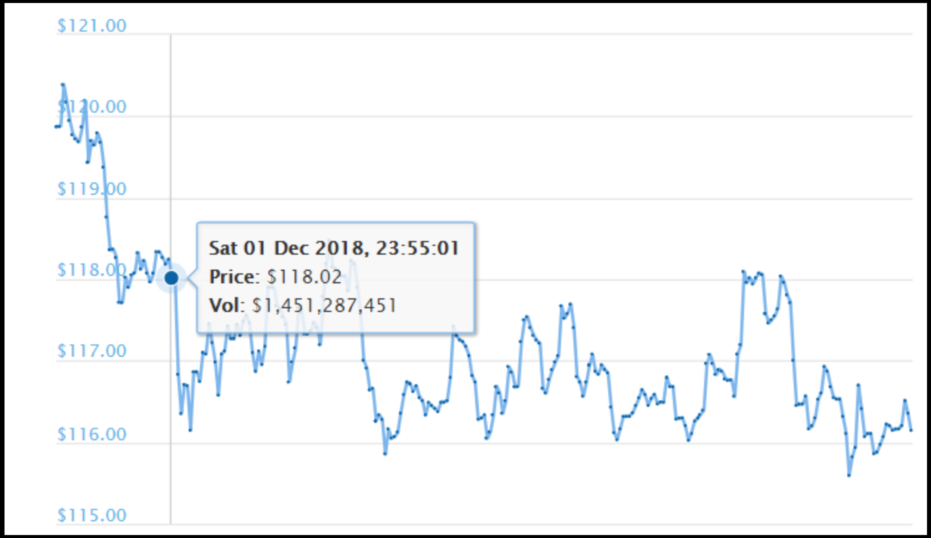
Money, Honey

Native currency called *ether*

not this



but this



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

Questions?

Take 15 seconds to reflect …

Unmute and ask!

Transaction Fields

Gas price

Gas limit

Nonce

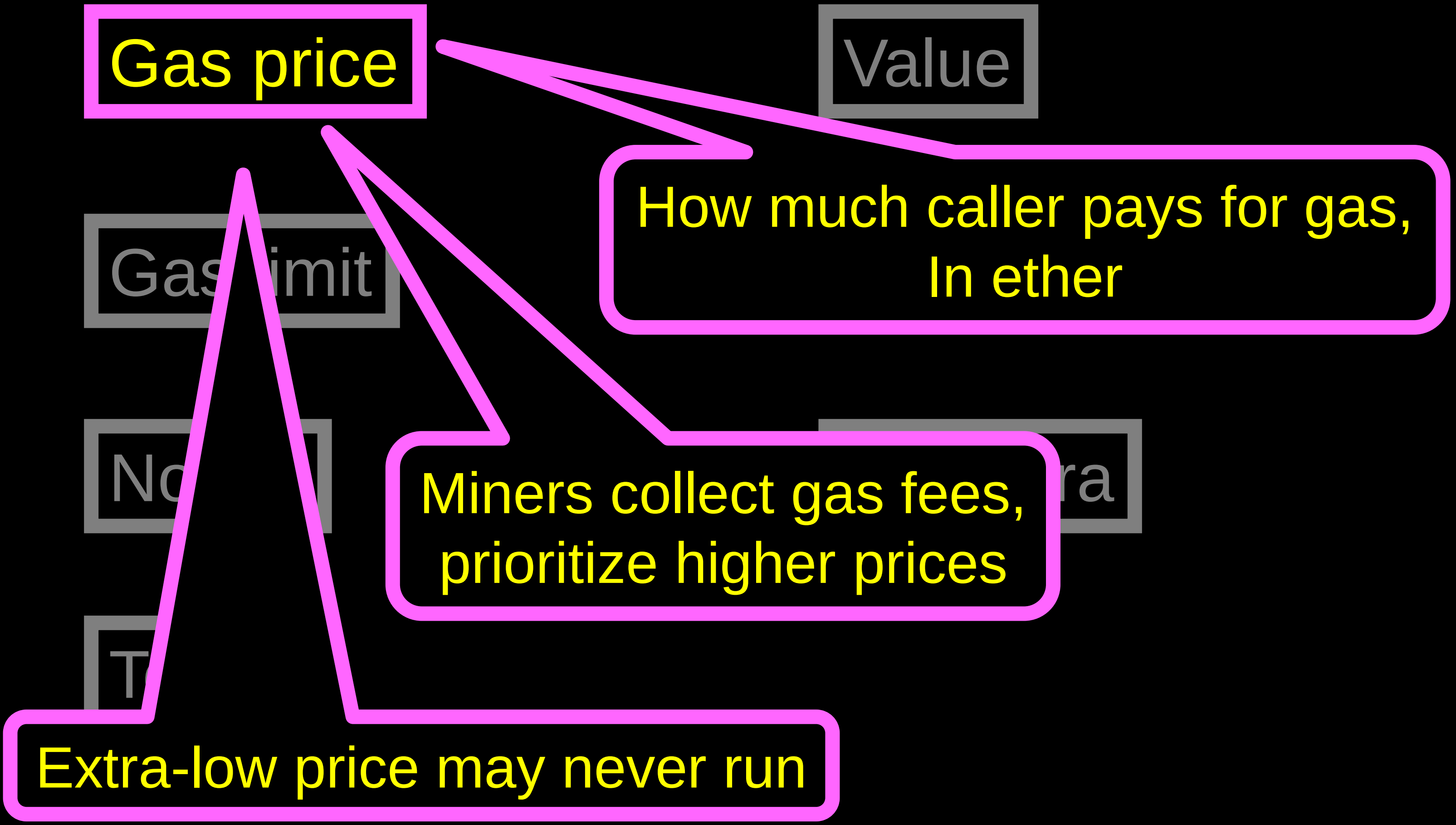
To

Value

Data

Et cetera

Transaction Fields



Gas price

Value

Gas limit

No

ra

T

How much caller pays for gas,
In ether

Miners collect gas fees,
prioritize higher prices

Extra-low price may never run

Transaction Fields

Gas price

Gas limit

Max gas caller willing to spend

Call aborts if exceeded

No refunds!

Nonce

To

Value

Data

Et cetera

Transaction Fields

Gas price

Value

Gas limit

Transaction sequence number
(from sender)

Nonce

Et cetera

To

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

Dat

Et cetera

How much ether to transfer

Transaction Fields

Gas price

Value

Gas limit

Data

Nonce

E ra

To

Payload:
function name, args ,etc. …

Transaction Fields

Gas price

Gas limit

Nonce

To

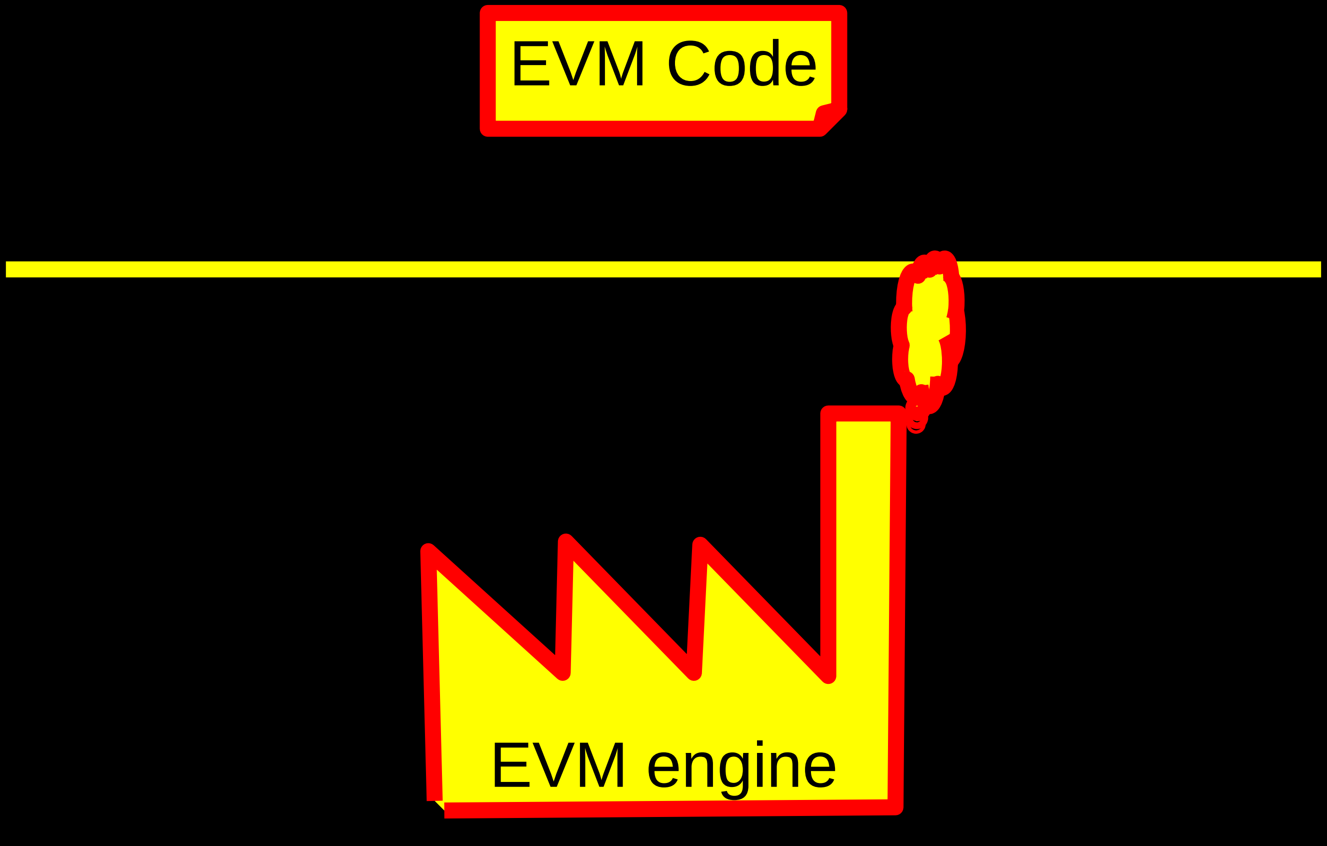
Value

Data

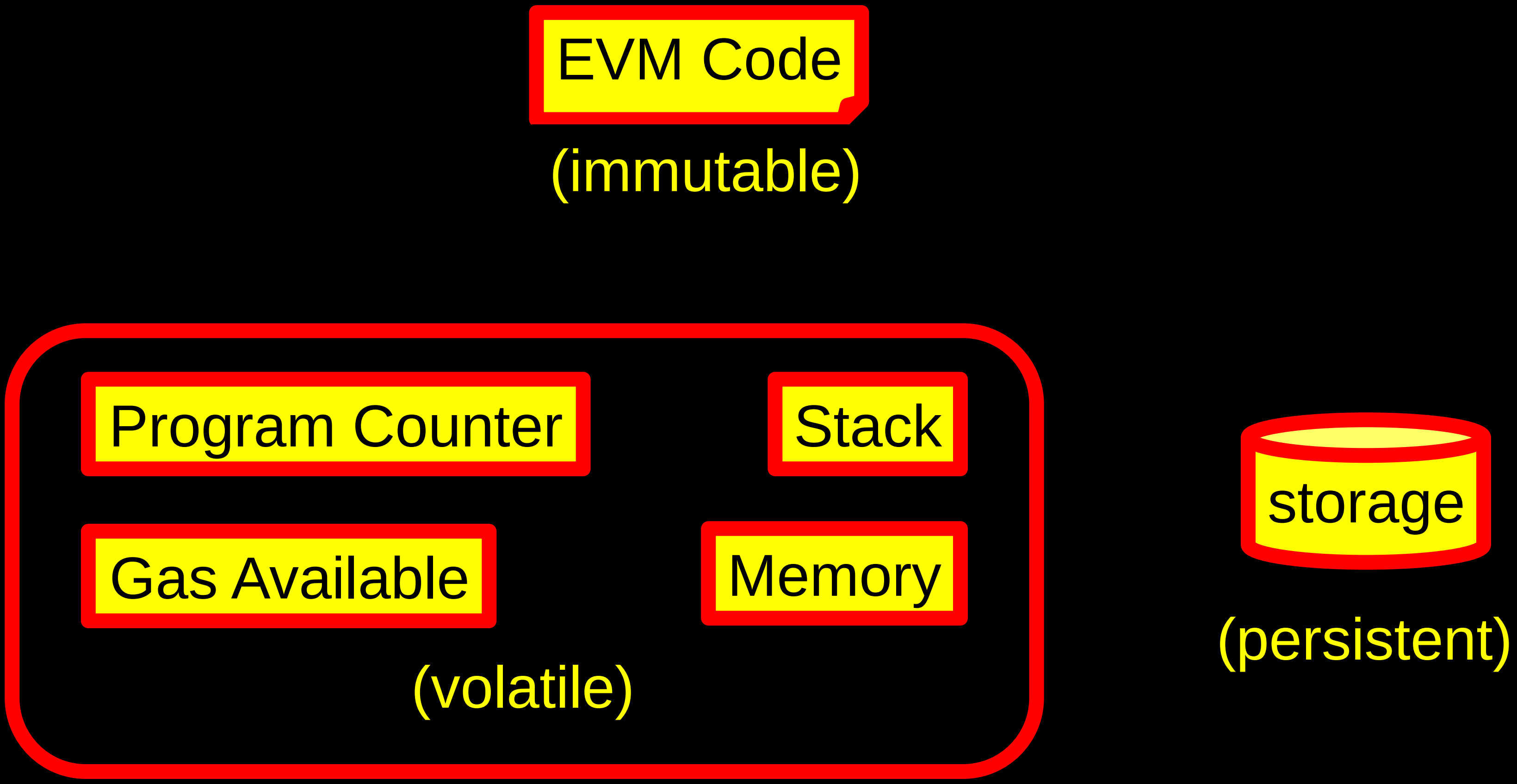
Et cetera

ECDSA signature args ….

Ethereum Virtual Machine



Ethereum Virtual Machine



EVM Code

(immutable)

Program Counter

Stack

Gas Available

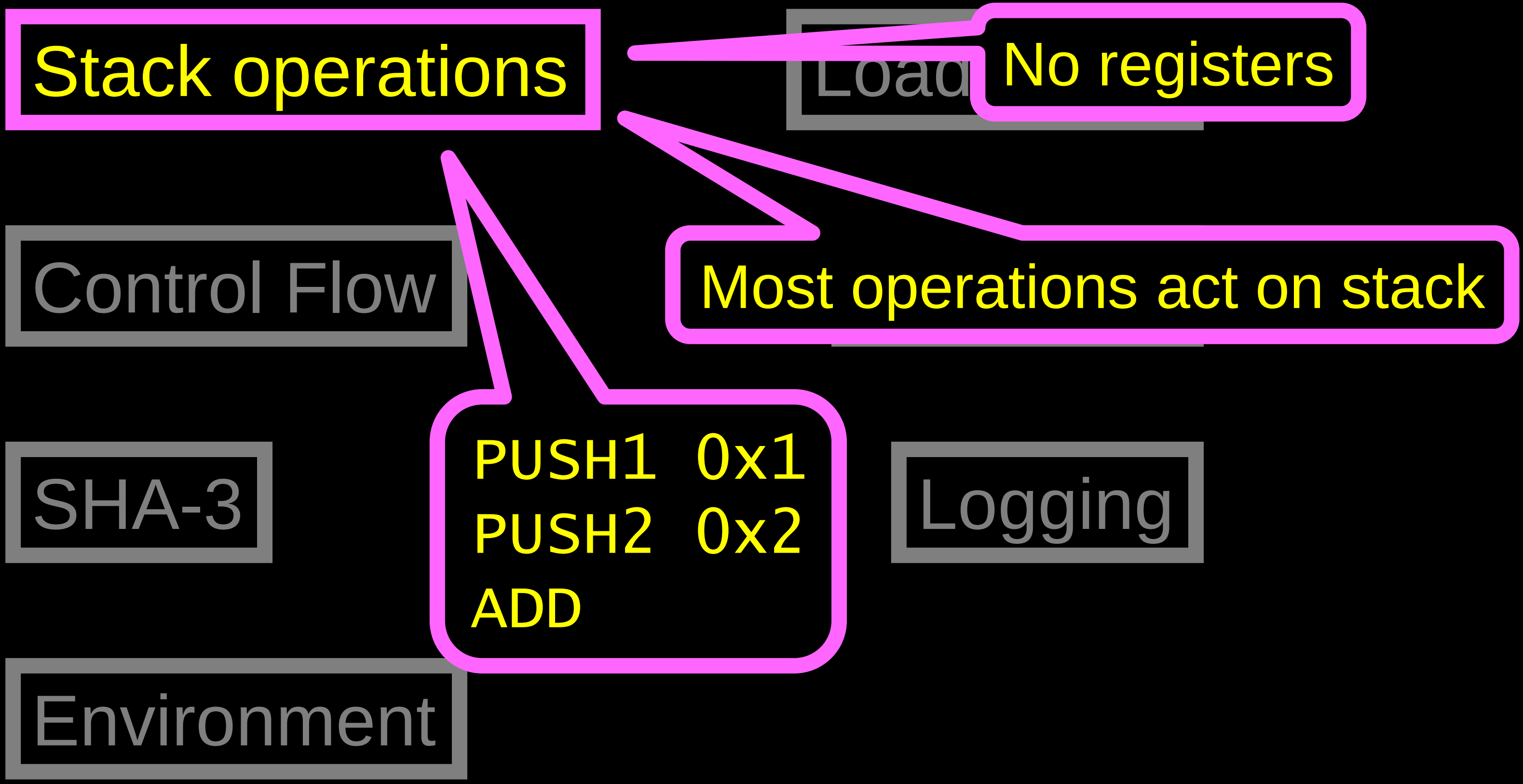
Memory

(volatile)

storage

(persistent)

Types of Instructions



Stack operations

Load

No registers

Control Flow

Most operations act on stack

SHA-3

```
PUSH1 0x1
PUSH2 0x2
ADD
```

Logging

Environment

Types of Instructions

Stack operations

Load, store

Local control flow

Control Flow

Block Info

SHA

Call other contracts

g

Call system libraries

Environment

Types of Instructions

Stack operations

Load, store

Control Flow

Various crypto hashes provided

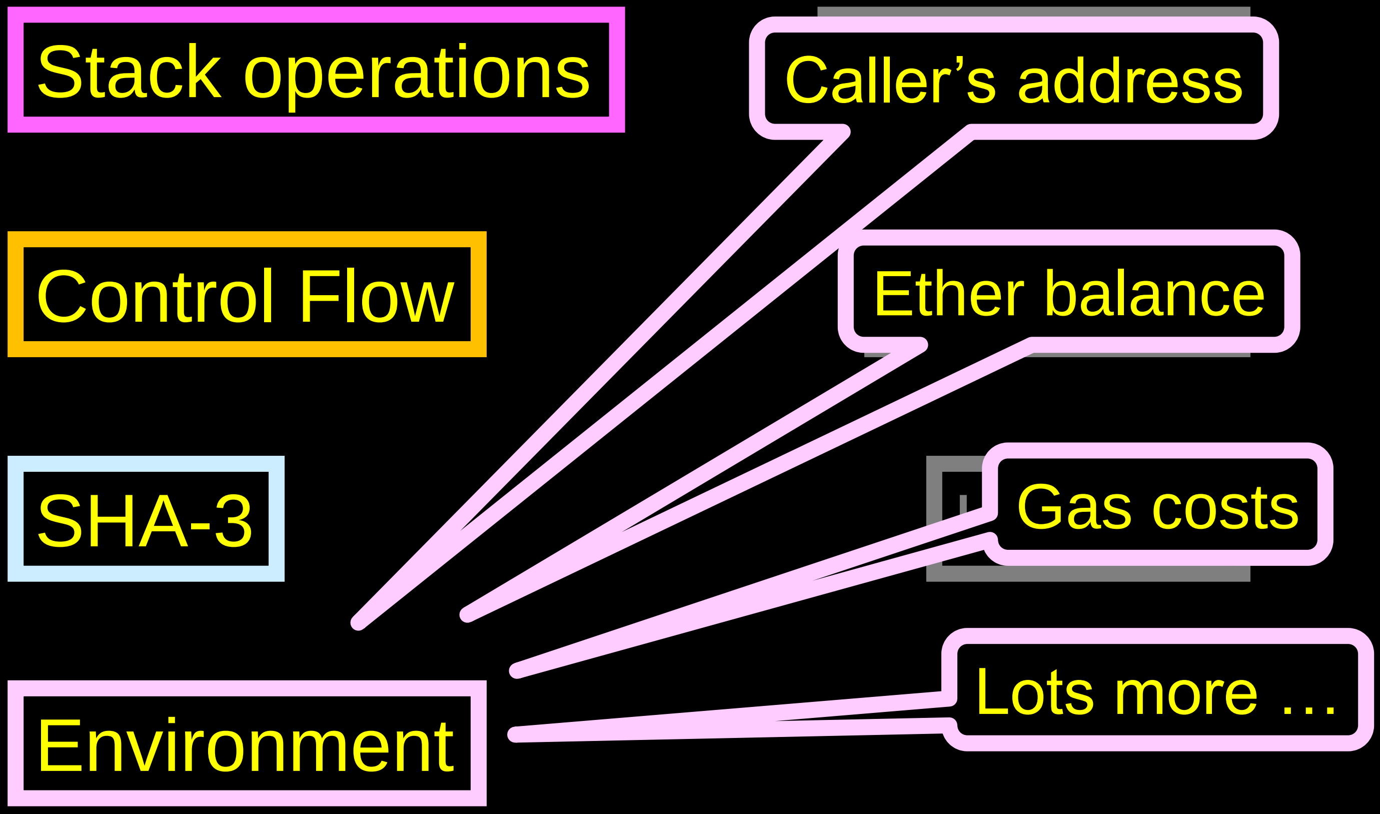
SHA-3

Logging

Environme

Gas costs too expensive to compute directly

Types of Instructions



Stack operations

Control Flow

SHA-3

Environment

Caller’s address

Ether balance

Gas costs

Lots more …

Types of Instructions

Stack operations

Load, store

Control Flow

Bloc nfo

SHA-3

Load and store
from non-stack memory

Environment

Types of Instructions

Stack operations

Load, store

Control Flow

Block Info

SHA-3

Logging

Environment

block number

and more …

256 latest block hashes

block timestamp

Types of Instructions

Stack operations

Load, store

Control Flow

ck info

SHA-3

Logging

Environme

communicate with outside world

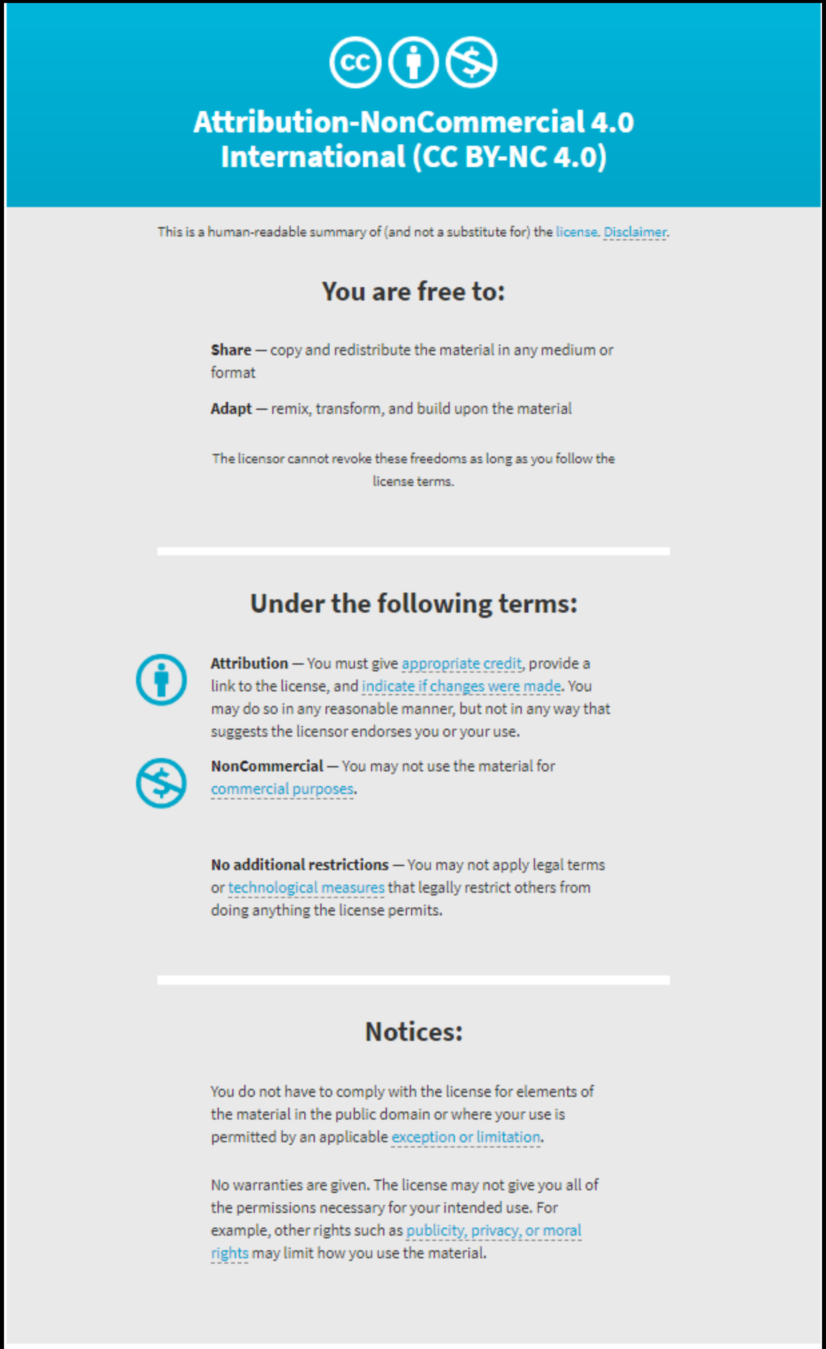
Write events to log

debug

Source : Ethereum Virtual Machine, CS1951 L by

Maurice Herlihy

Brown University



Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)

This is a human-readable summary of (and not a substitute for) the license. Disclaimer.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.