Introduction to

Blockchain Science & Engineering

An informatics Master's level course

Aggelos Kiayias

Dionysis Zindros, Christos Nasikas

Ethereum

Overview

- What is Ethereum ?
- Ethereum accounts
- Ethereum transactions
- Ethereum blockchain
- Solidity (programming language)

Extending Bitcoin functionality: adding new opcodes

- Building a protocol on top of Bitcoin:
  - Pros:
    - Take advantage of the underlying network and mining power.
    - Very low development cost
  - Cons:
    - No flexibility.
    - No SPV clients.
- Build an independent network:
  - Pros:
    - Easy to add and extend new opcodes.
    - Flexibility.
  - Cons:
    - Need to attract miners to sustain the network.
    - Difficult to implement.

Alternative blockchain applications

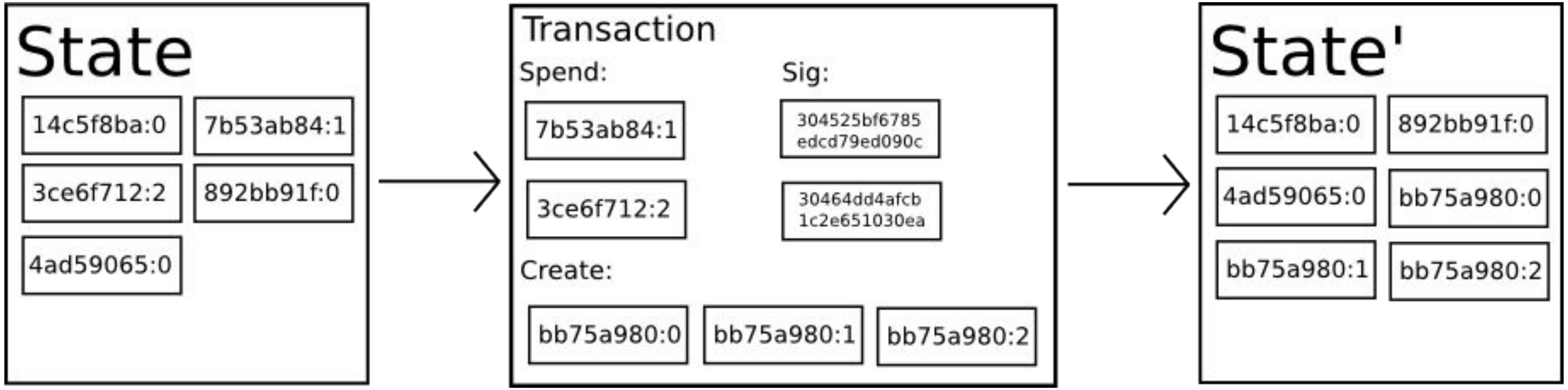
- Namecoin:
  - Bitcoin fork: Currency NMC
  - Decentralized name registration database: DNS, identities etc
- Colored coins:
  - On top of Bitcoin
  - Allows people to create their own digital currencies
- OmniLayer (formerly Mastercoin)
  - On top of Bitcoin
  - Distributed exchange, smart property, distributed e-commerce, etc
- OpenBazaar
  - On top of Bitcoin
  - Decentralized marketplace

Bitcoin’s scripting language limitations

- Lack of Turing-completeness: No loops
- Lack of state: Cannot keep internal state.
- Value-blindness: Cannot denominate the amount being sent
- Blockchain-blindness: Cannot access block header values such as nonce, timestamp and previous hash block.

What about user defined functionality ?

Bitcoin as a state transition system



State = UTXO
Transaction is applied to state to give a new state

Ethereum: A universal RSM

- Transaction-based deterministic state machine
  - Global singleton state
  - A virtual machine that applies changes to global state
- A global decentralized computing infrastructure
- Anyone can create their own state transition functions

Ethereum: A universal RSM

- Stack-based bytecode language
- Turing-completeness
- Smart contracts
- Decentralized applications

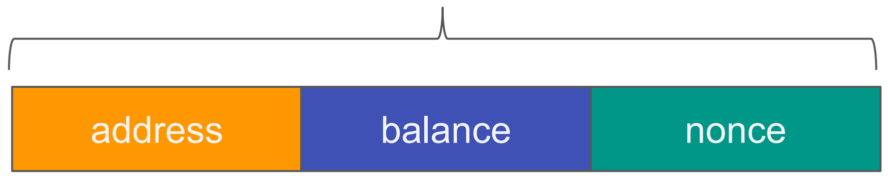
Same principles as Bitcoin

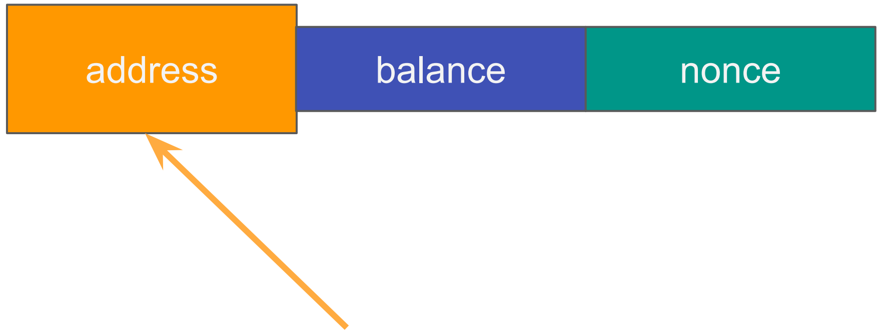
- **A peer-to-peer network:** connects the participants
- **A consensus algorithm**: Proof of Work (will move to PoS)
- **A digital currency:** ether
- **A global ledger**: the blockchain
    - Addresses: key pair
    - Wallets
    - Transactions: digital signatures
    - Blocks

Ethereum accounts

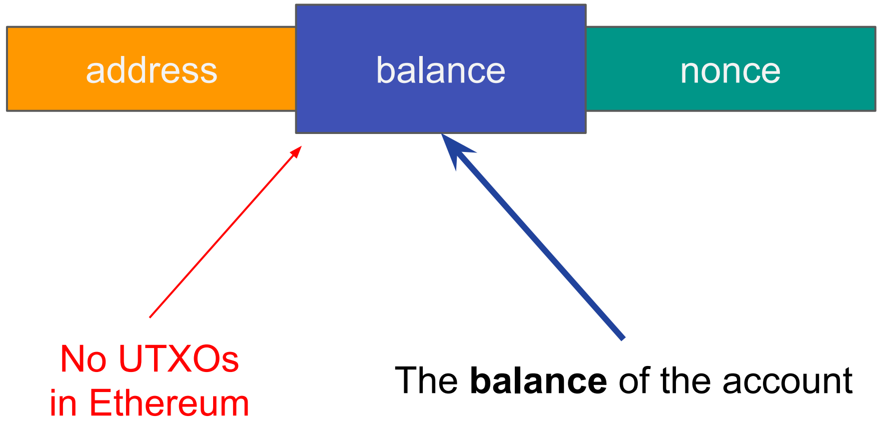
- Global state of Ethereum: **accounts**
- They **interact** to each other **through transactions** (messages)
- A **state associated** with it and a 20-byte **address** (160-bit identifier)

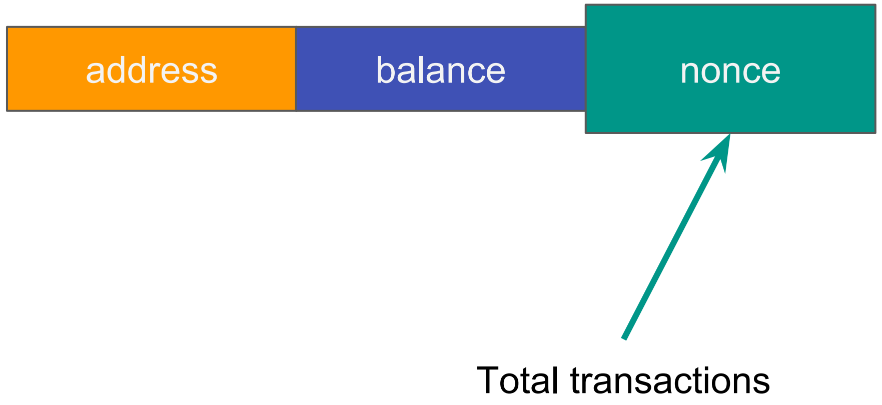
Ethereum account





The **address** of the account

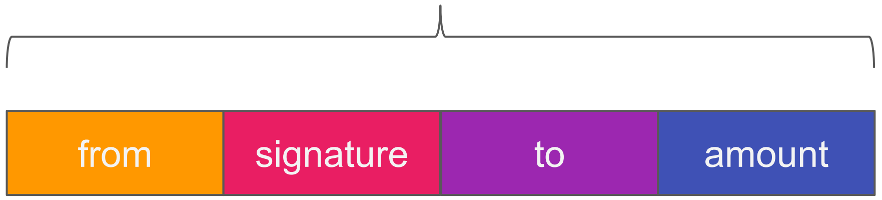

address
balance
nonce
No UTXOs
in Ethereum
The **balance** of the account

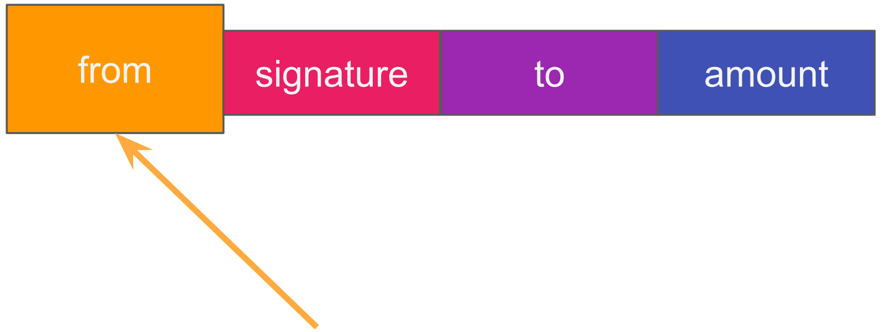

address
balance
nonce
Total transactions

UTXO vs Accounts

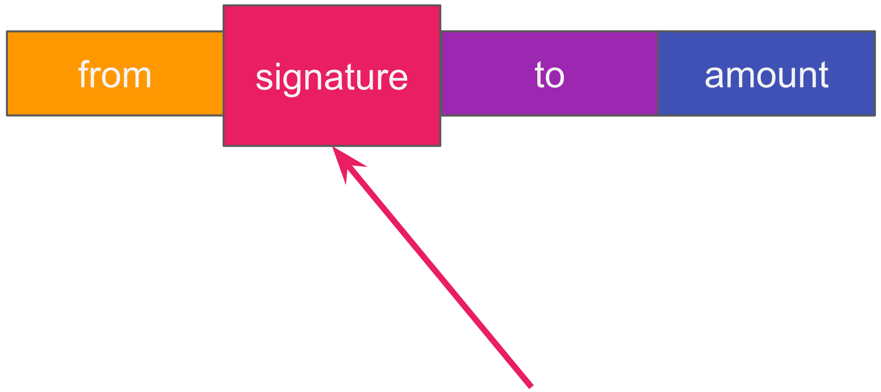
- UTXOs pros:
  - Higher degree of privacy
  - Scalability (parallelism, sharding)
- Accounts pros:
  - Space saving
  - Better fungibility
  - Simplicity
  - Efficiency

Ethereum transaction

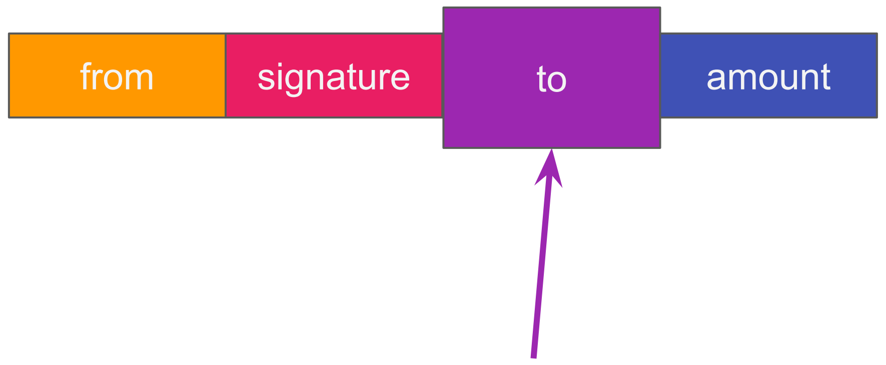




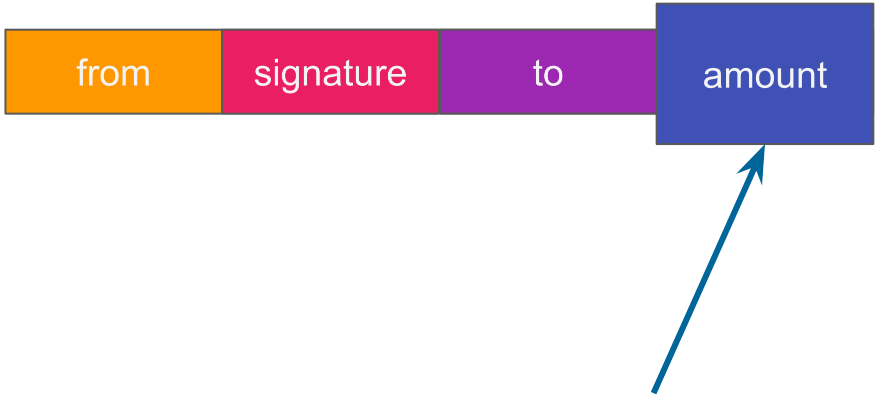
The **sender** of the transaction



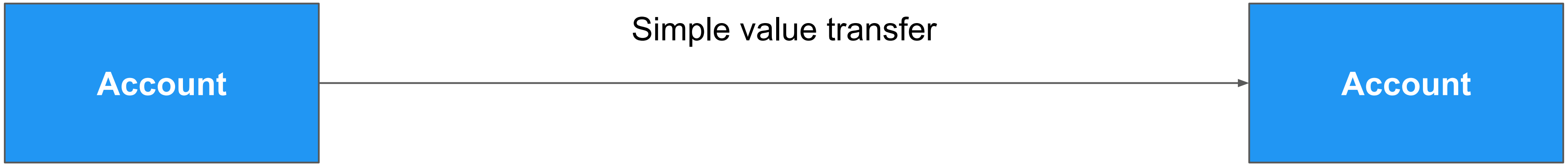
Digital signature on the **new transaction** created by **the sender’s private key**



Receiver of the transaction



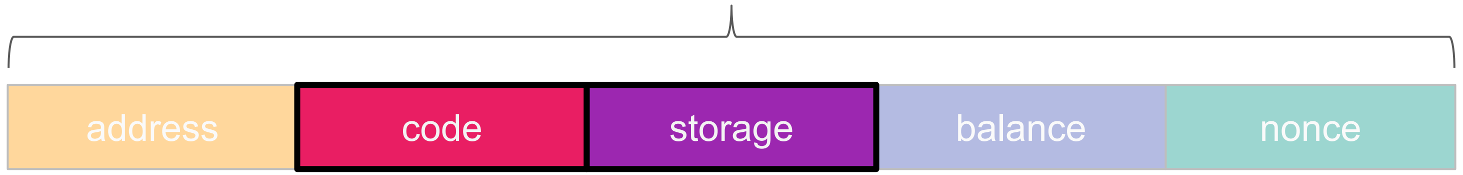
Amount transferred by transaction
Given in Wei


Account
Simple value transfer
Account

Two types of accounts

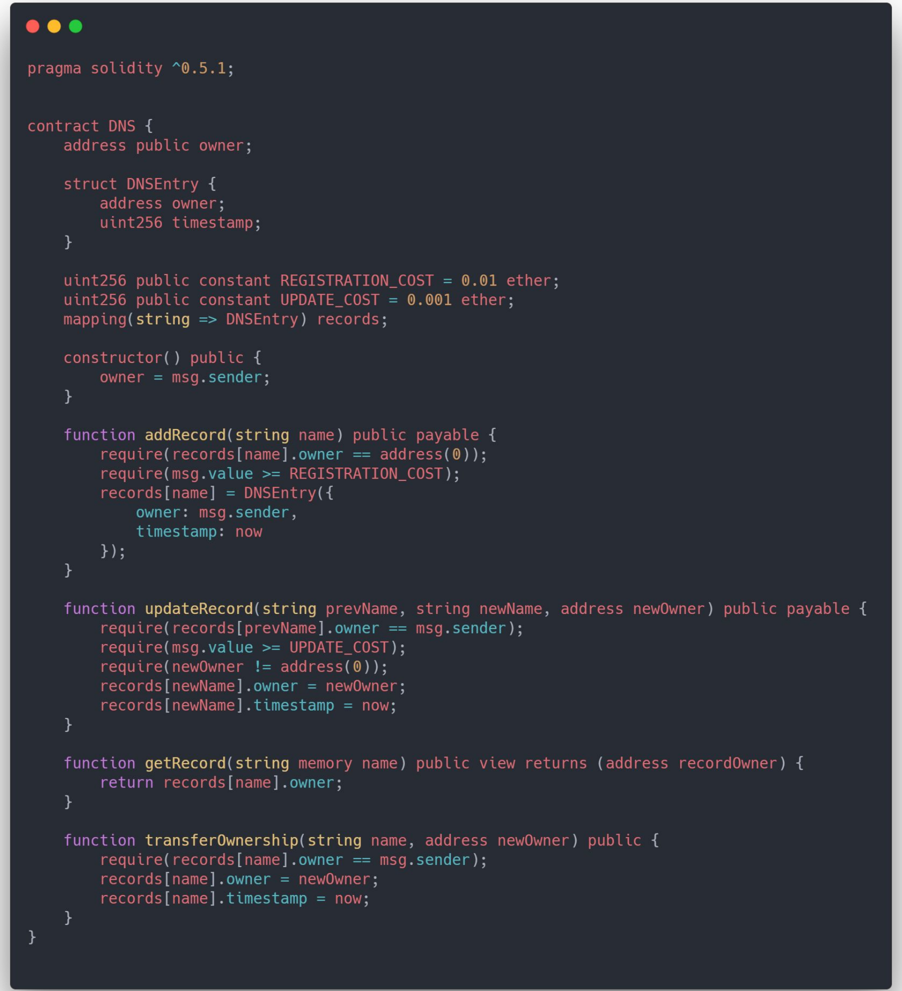
- Personal accounts (what we’ve seen)
- **Contract accounts**

Ethereum contract account



What is a smart contract?

- Computer programs
- Contract code is executed by all full nodes
- The outcome of a smart contract is the same for everyone
- Context: Internal storage, transaction context, most recent blocks
- The code of a smart contract cannot change



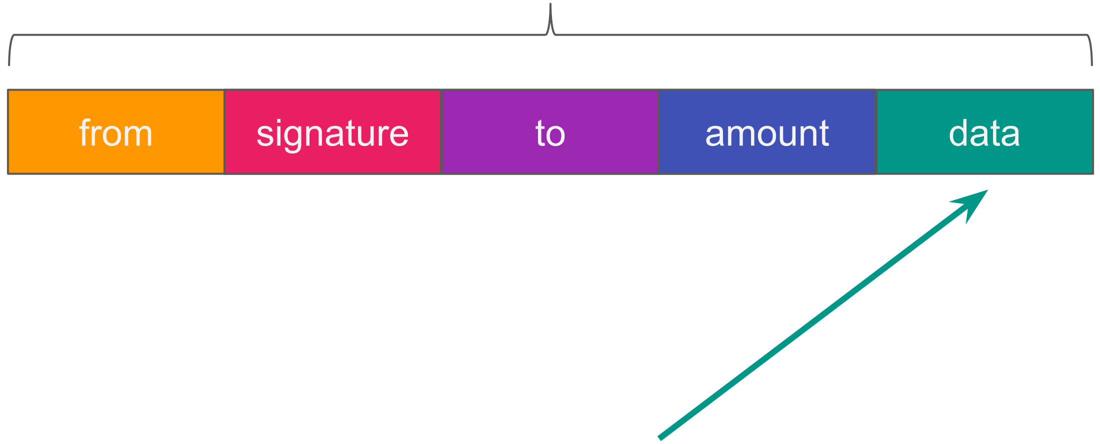
```
pragma solidity ^0.5.1;


contract DNS {
    address public owner;

    struct DNSEntry {
        address owner;
        uint256 timestamp;
    }

    uint256 public constant REGISTRATION_COST = 0.01 ether;
    uint256 public constant UPDATE_COST = 0.001 ether;
    mapping(string => DNSEntry) records;

    constructor() public {
        owner = msg.sender;
    }

    function addRecord(string name) public payable {
        require(records[name].owner == address(0));
        require(msg.value >= REGISTRATION_COST);
        records[name] = DNSEntry({
            owner: msg.sender,
            timestamp: now
        });
    }

    function updateRecord(string prevName, string newName, address newOwner) public payable {
        require(records[prevName].owner == msg.sender);
        require(msg.value >= UPDATE_COST);
        require(newOwner != address(0));
        records[newName].owner = newOwner;
        records[newName].timestamp = now;
    }

    function getRecord(string memory name) public view returns (address recordOwner) {
        return records[name].owner;
    }

    function transferOwnership(string name, address newOwner) public {
        require(records[name].owner == msg.sender);
        records[name].owner = newOwner;
        records[name].timestamp = now;
    }
}
```

Ethereum accounts

	Personal account	Contract account
address	H(pub_key)	H(creator, nonce)
code	∅	Code to be executed
storage	∅	Data of the contract
balance	ETH balance (in Wei)	ETH balance (in Wei)
nonce	# transaction sent	# transaction sent

address
code
storage
balance
nonce

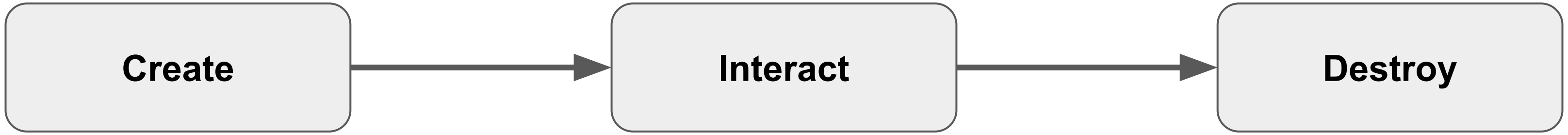
a transaction about a contract

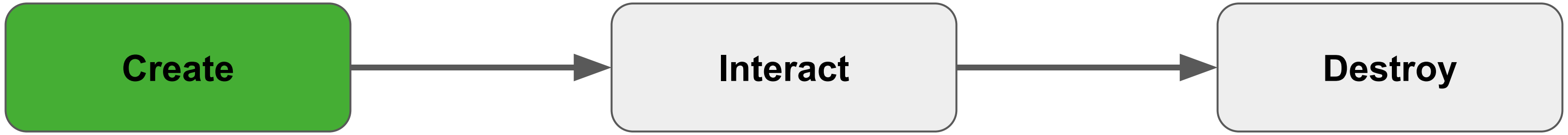


Transaction **about personal accounts**:
Field is unused

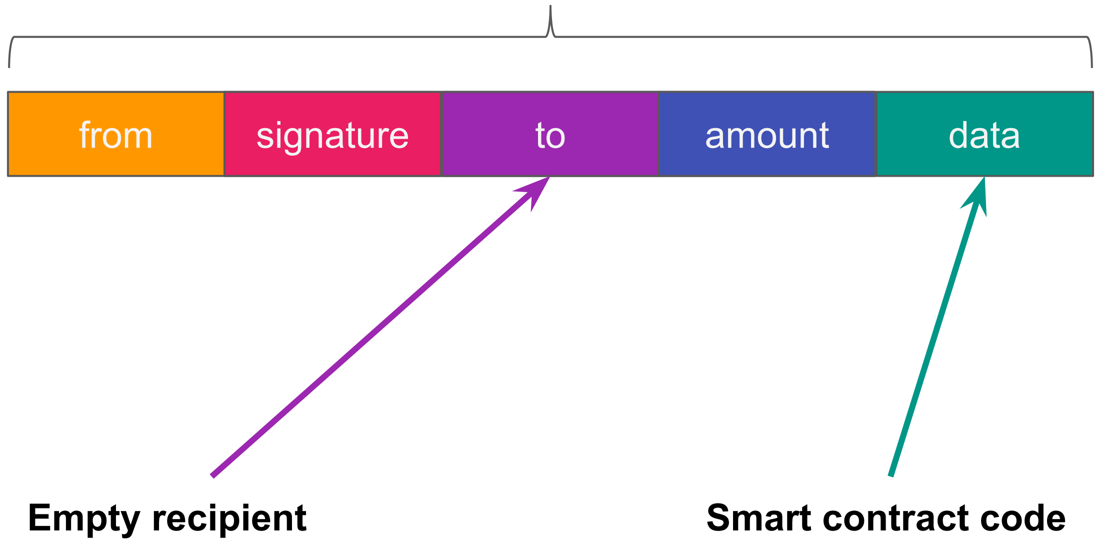
Transaction **about contracts**:
Will contain **data about the contract**

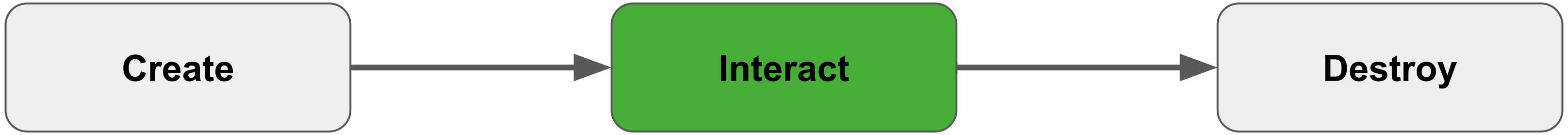
Smart contract lifecycle



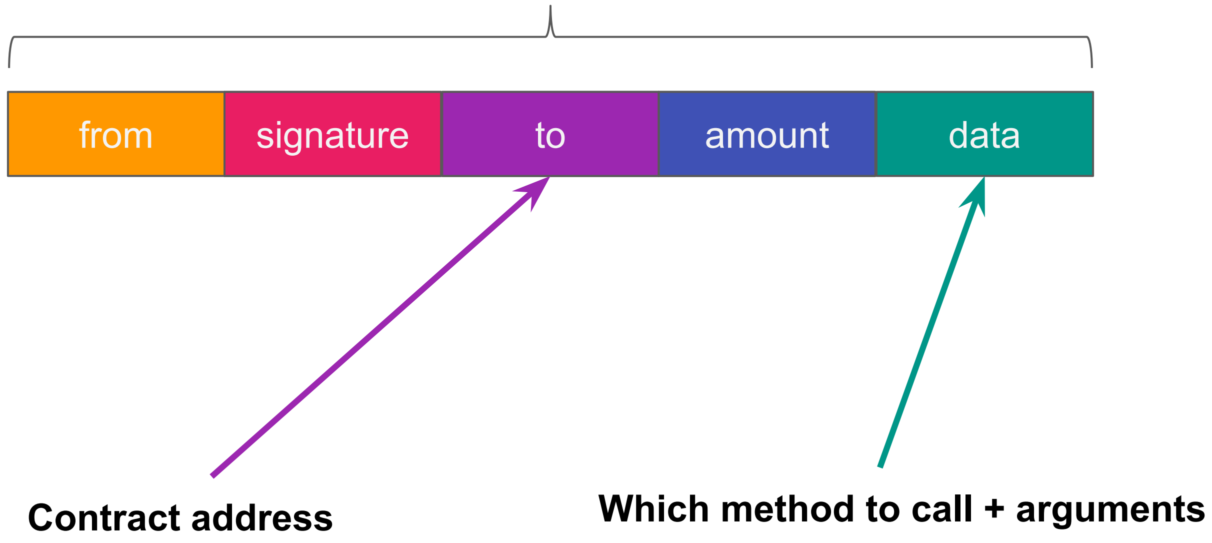

Create
Interact
Destroy

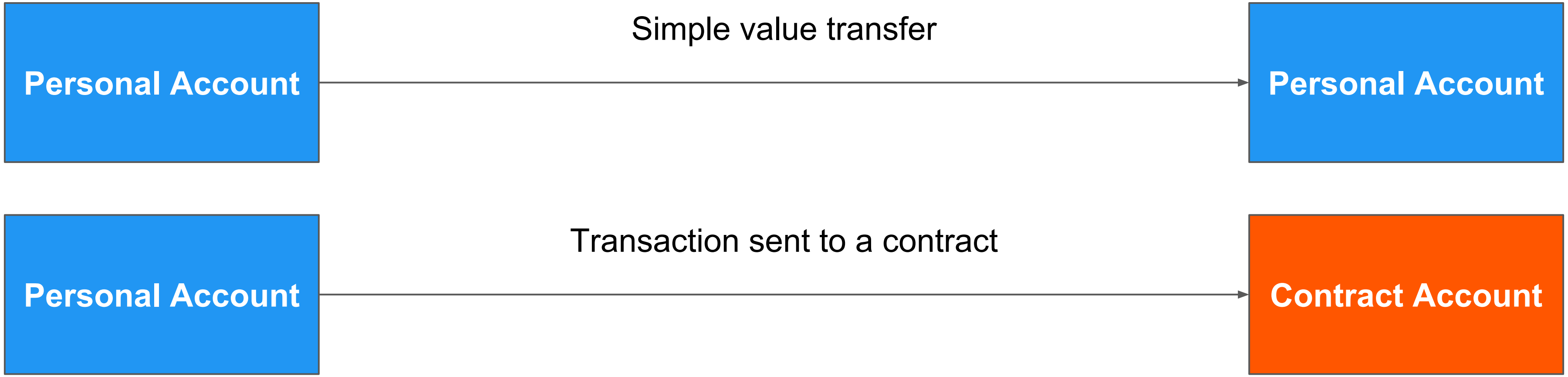
Transaction for contract creation




Create
Interact
Destroy

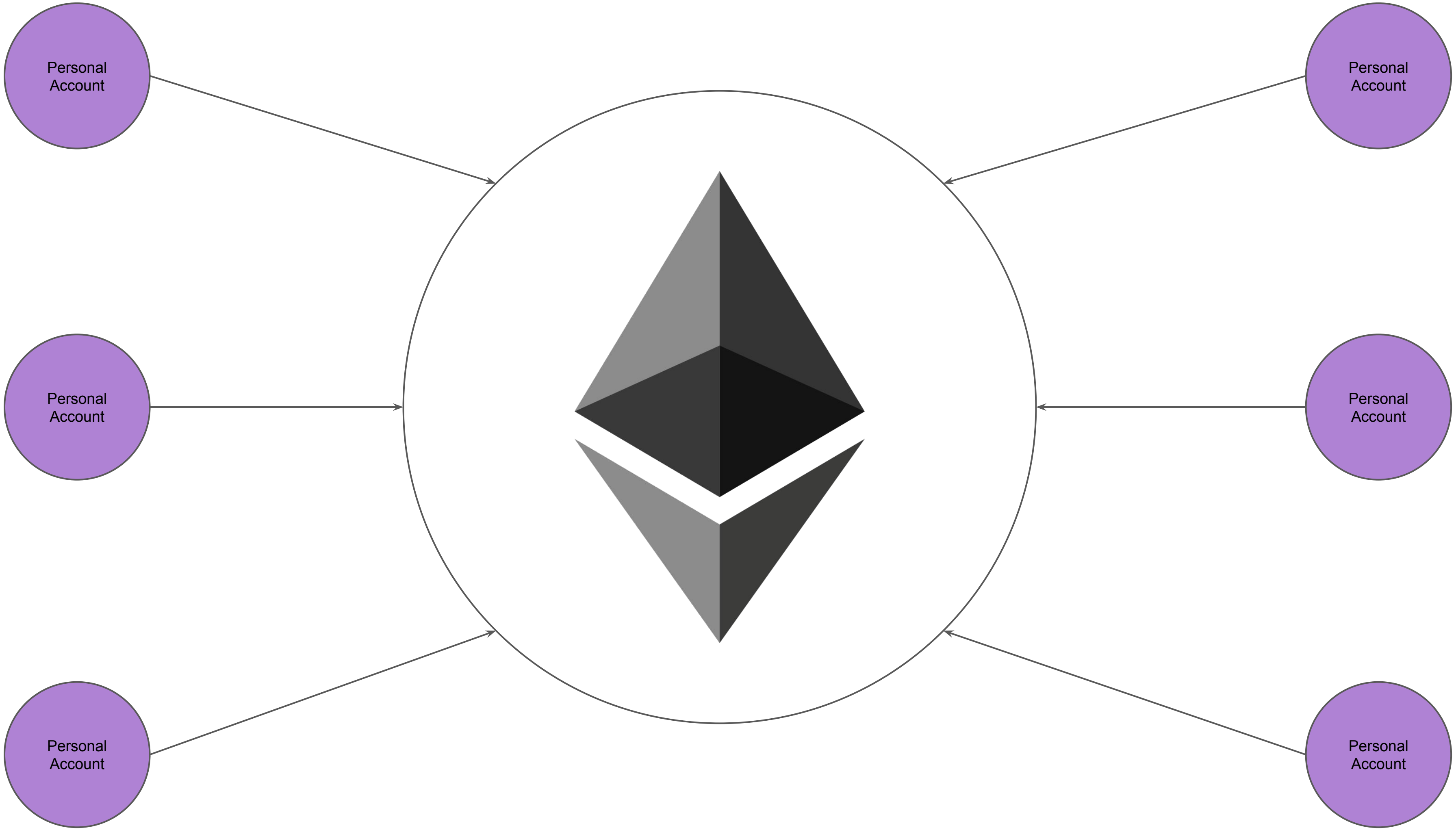
Transaction for contract interaction




Personal Account
Simple value transfer
Personal Account
Personal Account
Transaction sent to a contract
Contract Account

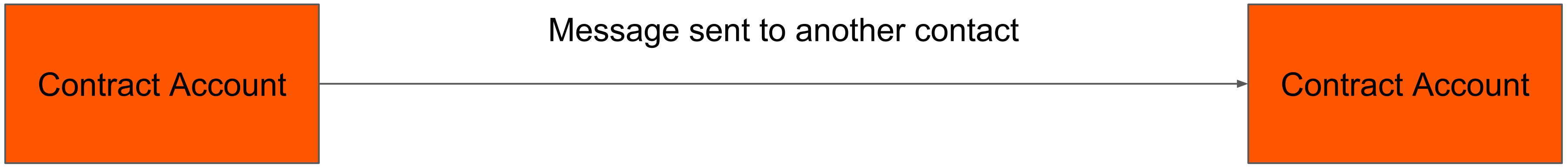
Contract method call

- When contract account is activated:
    a. Contract **code** runs
    b. It can read / write to **internal storage**
    c. It can **send other transactions** or **call other contracts**
- Can’t initiate new transactions on their own
- Can only fire transactions in response to other transactions received

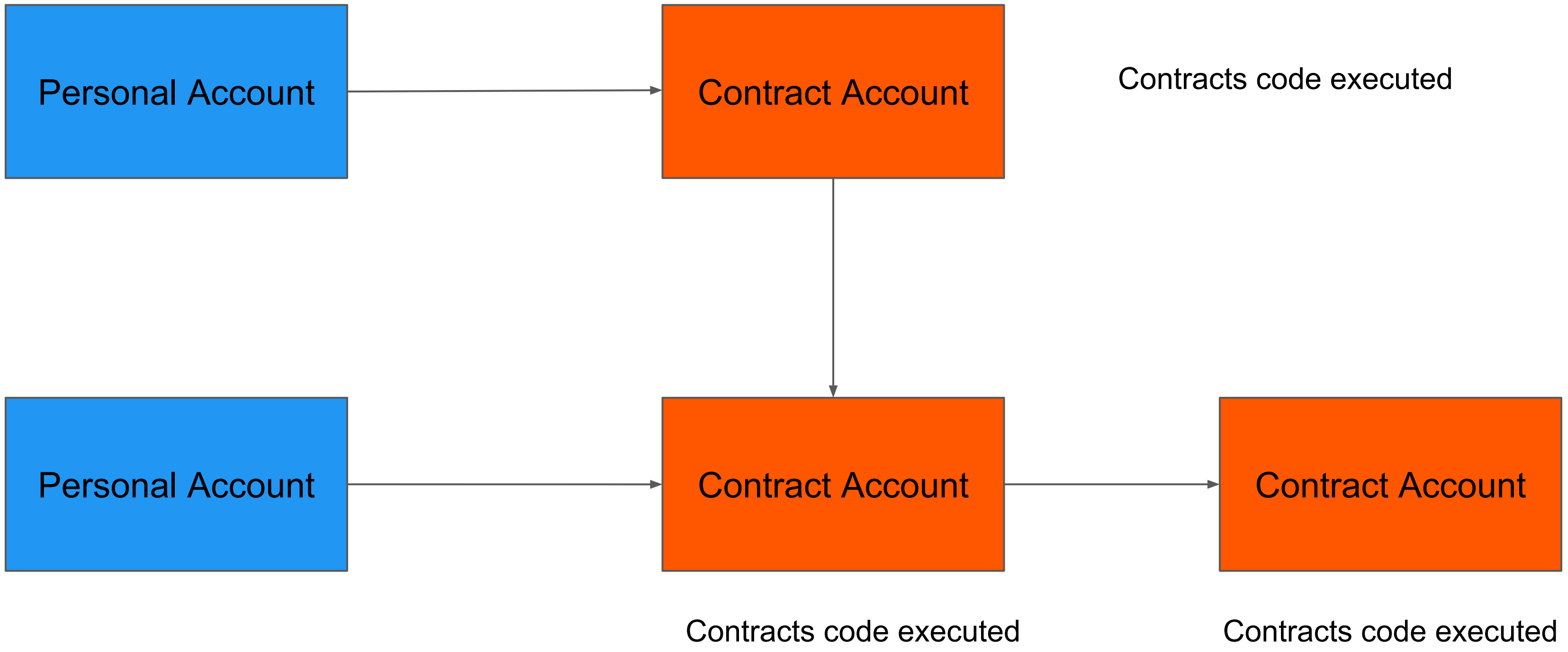

Personal Account
Personal Account
Personal Account
Personal Account
Personal Account
Personal Account

Messages

- Like a **transaction** except it is **produced by a contract**
- Virtual objects
- Exist **only** in the **Ethereum execution environment**
- A message leads to the recipient account running its code
- **Contracts** can have **relationships** with **other contracts**

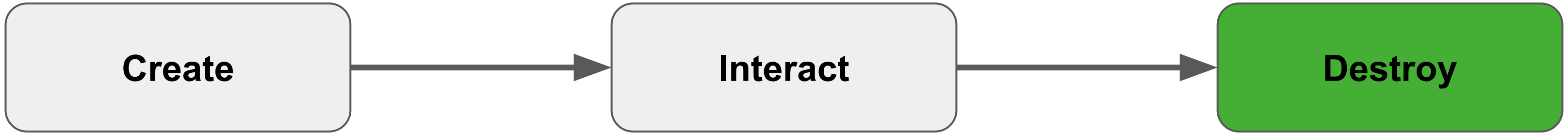

Contract Account
Message sent to another contact
Contract Account

Transactions & messages

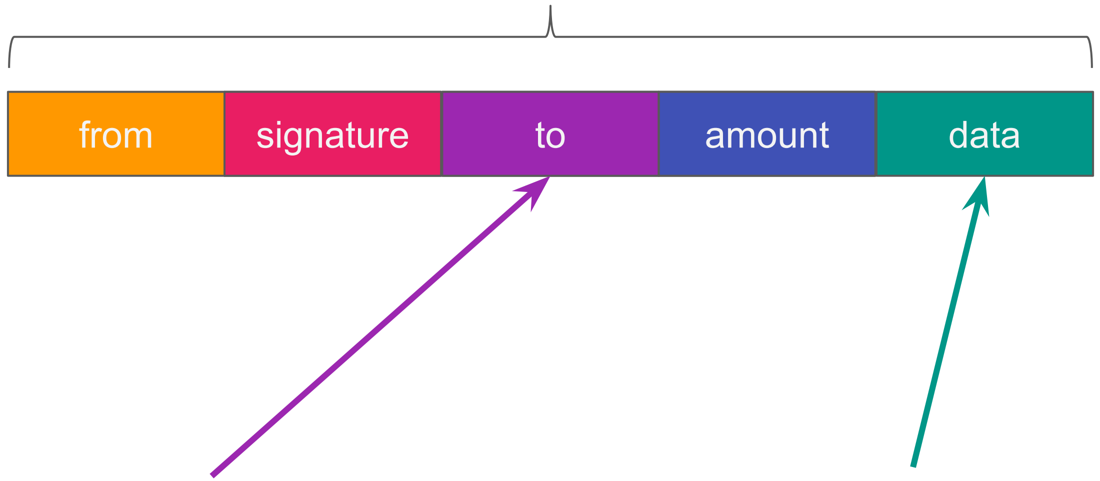


Types of transactions

	create	send	call
from	creator	sender	caller
signature	sig	sig	sig
to	∅	receiver	contract
amount	ETH	ETH	ETH
data	code	∅	f, args


Create
Interact
Destroy

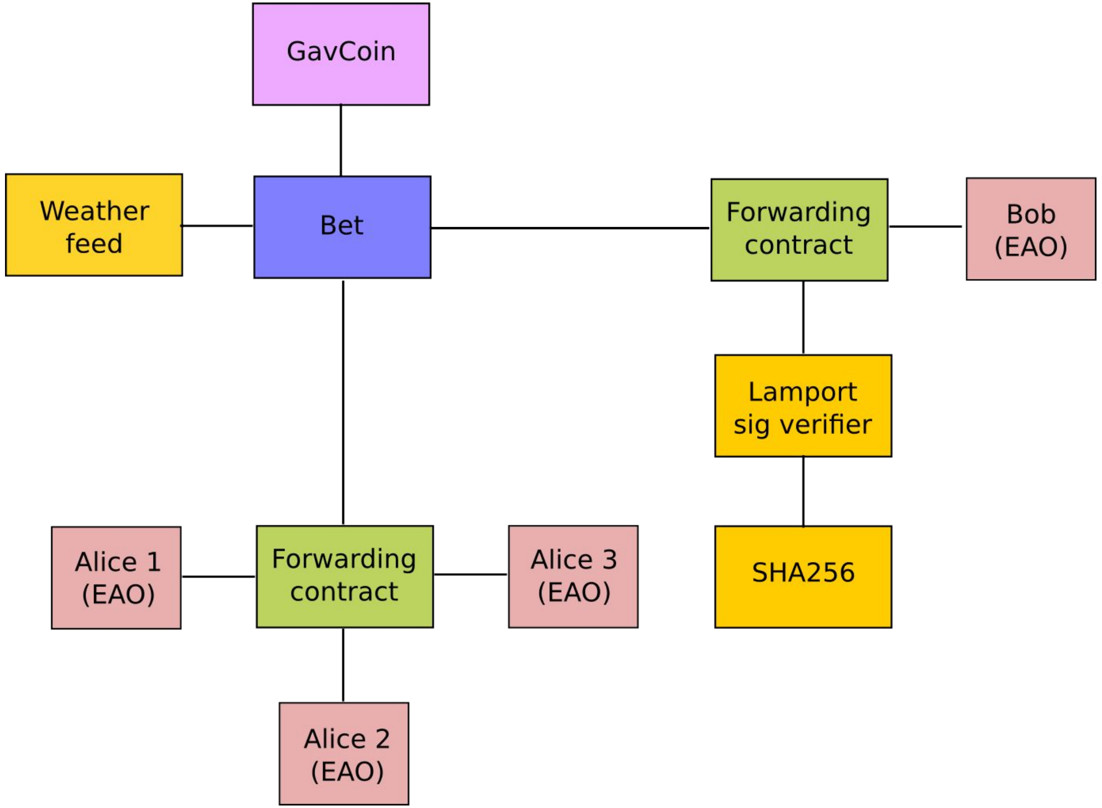
a transaction for contract destruction



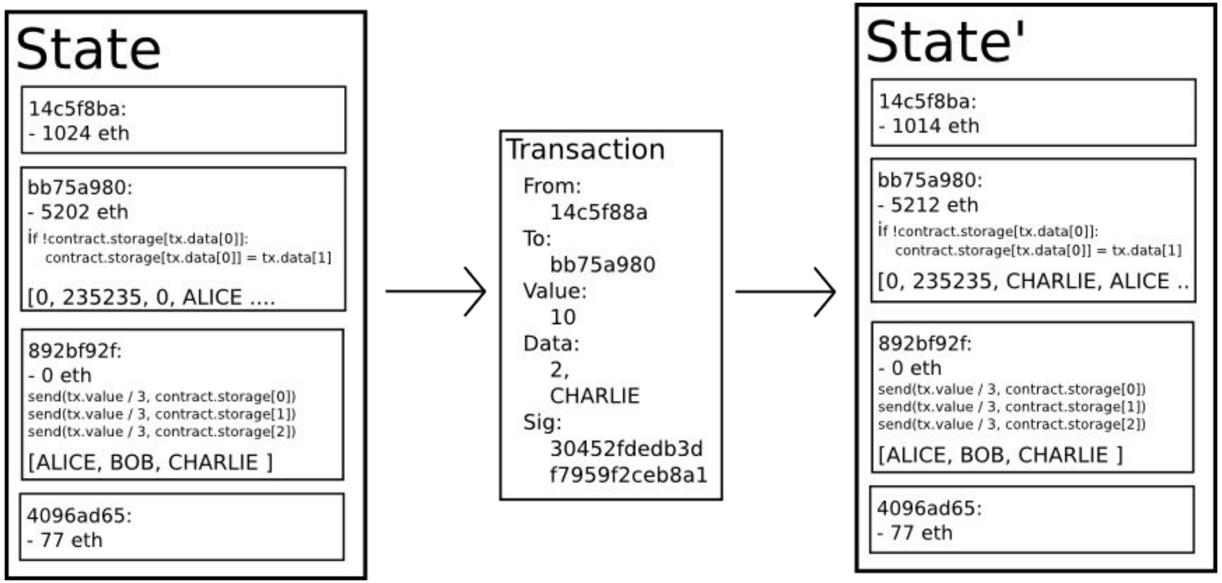
Contract address

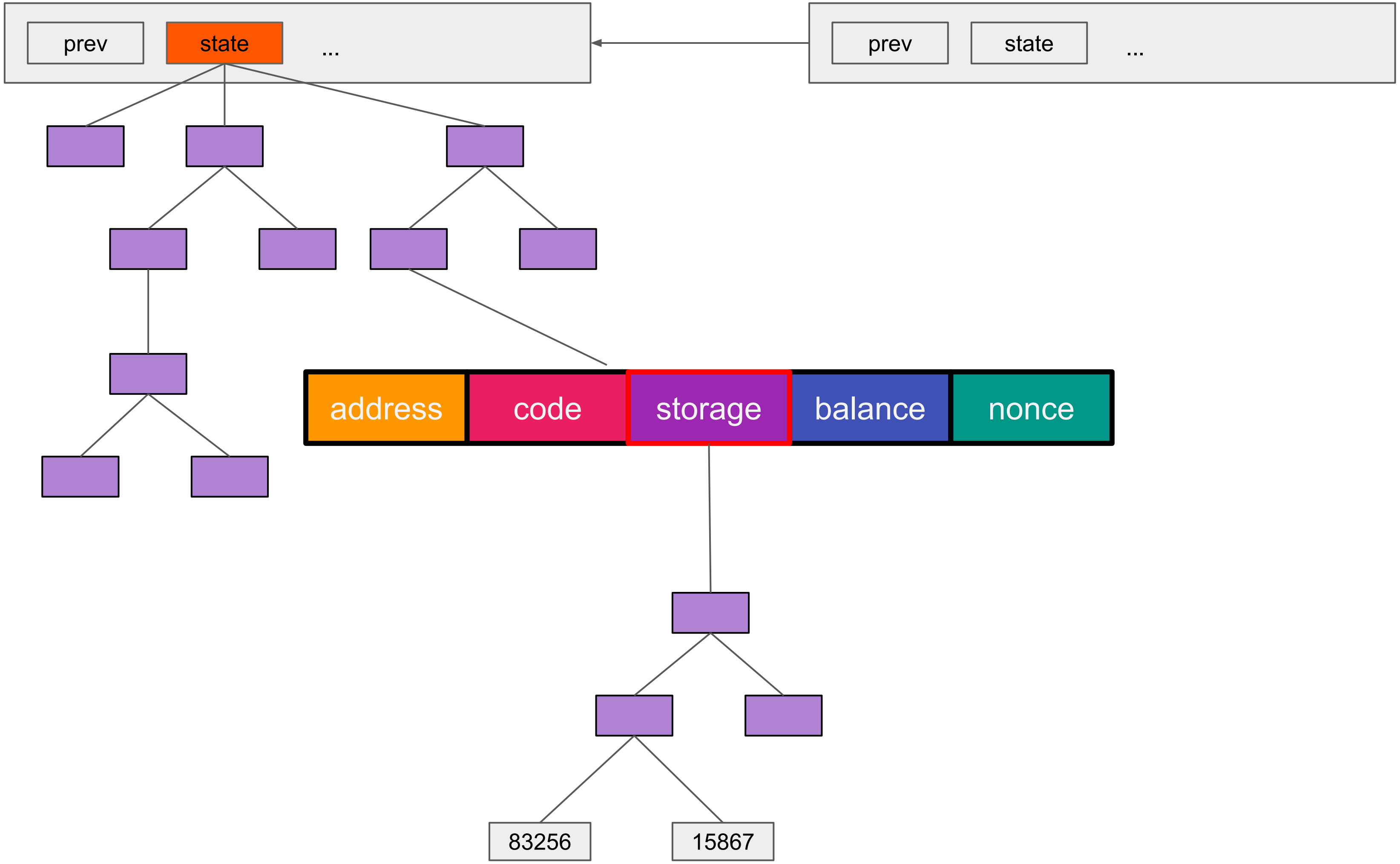
The name of a method that calls the `selfdestruct` **opcode**

Example of contract and account interaction

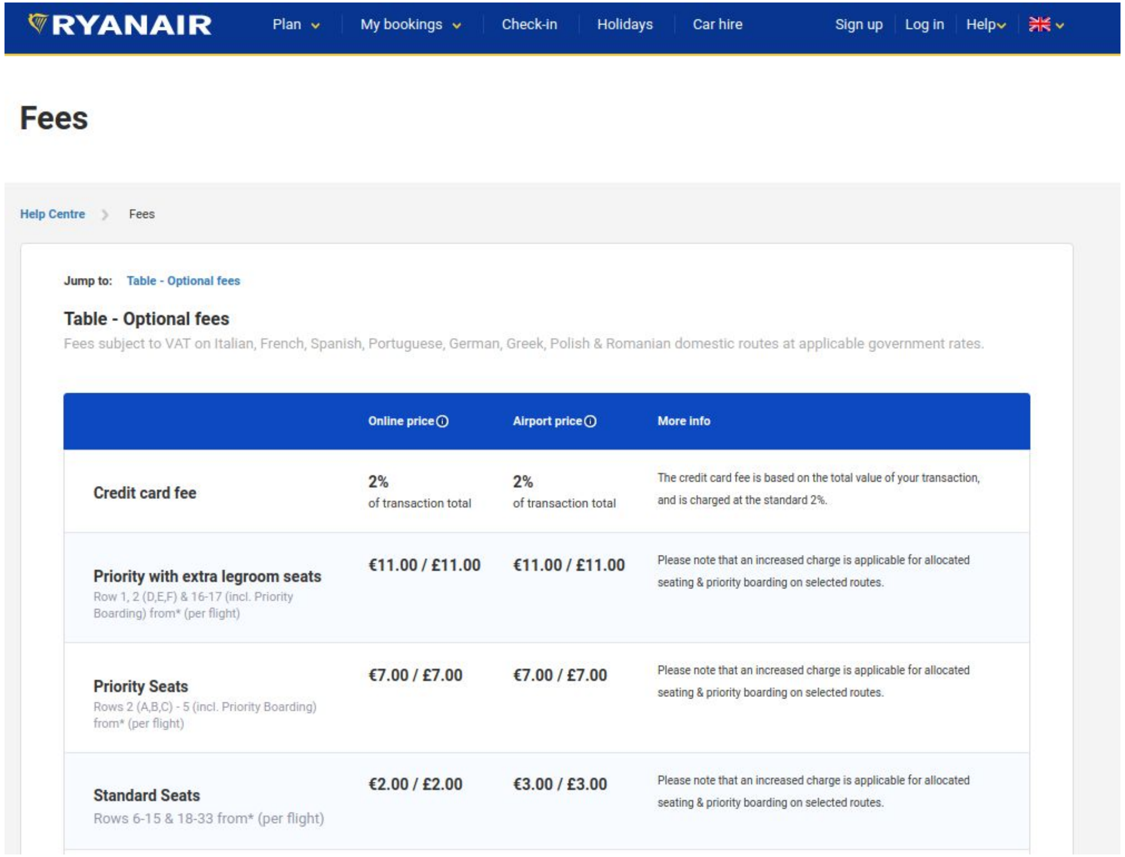


Ethereum state machine


State
14c5f8ba:
- 1024 eth
bb75a980:
- 5202 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, 0, ALICE ....
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65:
- 77 eth
Transaction
From:
14c5f88a
To:
bb75a980
Value:
10
Data:
2,
CHARLIE
Sig:
30452fdedb3d
f7959f2ceb8a1
State'
14c5f8ba:
- 1014 eth
bb75a980:
- 5212 eth
if !contract.storage[tx.data[0]]:
contract.storage[tx.data[0]] = tx.data[1]
[0, 235235, CHARLIE, ALICE ..
892bf92f:
- 0 eth
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
[ALICE, BOB, CHARLIE]
4096ad65:
- 77 eth


prev
state
...
prev
state
...
address
code
storage
balance
nonce
83256
15867

“Ethereum is Ryanair”: pay to board, then keep paying



Fees

Help Centre > Fees

Jump to: Table - Optional fees

Table - Optional fees

Fees subject to VAT on Italian, French, Spanish, Portuguese, German, Greek, Polish & Romanian domestic routes at applicable government rates.

	Online price	Airport price	More info
Credit card fee	2% of transaction total	2% of transaction total	The credit card fee is based on the total value of your transaction, and is charged at the standard 2%.
Priority with extra legroom seats Row 1, 2 (D,E,F) & 16-17 (incl. Priority Boarding) from* (per flight)	€11.00 / £11.00	€11.00 / £11.00	Please note that an increased charge is applicable for allocated seating & priority boarding on selected routes.
Priority Seats Rows 2 (A,B,C) - 5 (incl. Priority Boarding) from* (per flight)	€7.00 / £7.00	€7.00 / £7.00	Please note that an increased charge is applicable for allocated seating & priority boarding on selected routes.
Standard Seats Rows 6-15 & 18-33 from* (per flight)	€2.00 / £2.00	€3.00 / £3.00	Please note that an increased charge is applicable for allocated seating & priority boarding on selected routes.

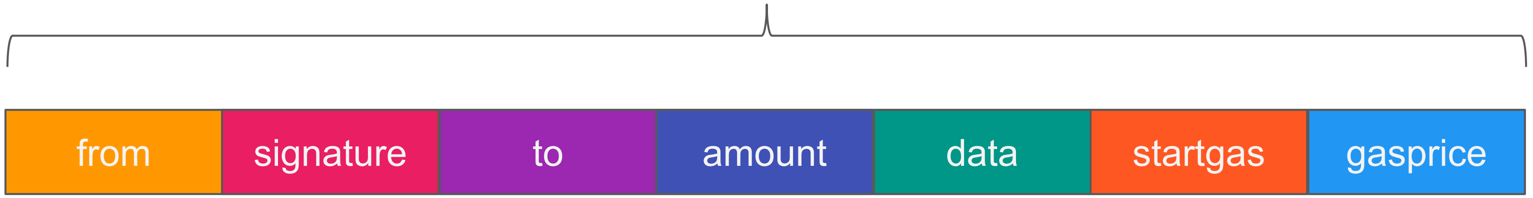
Gas: a necessary evil

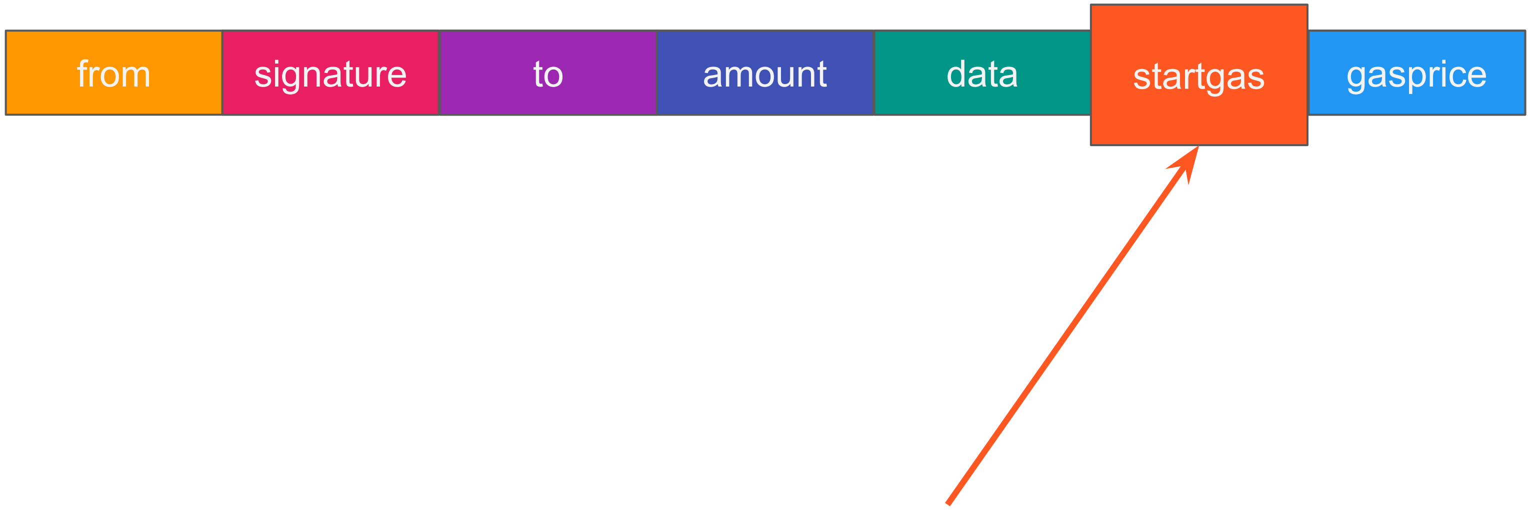
- Every node on the network:
  - evaluate all **transactions**
  - store all **state**
- Halting problem

Gas: a necessary evil

- Every **computation step** has a **fee**
- Is **paid** in **gas**
- **Gas** is the **unit** used to **measure computations**

Ethereum transaction

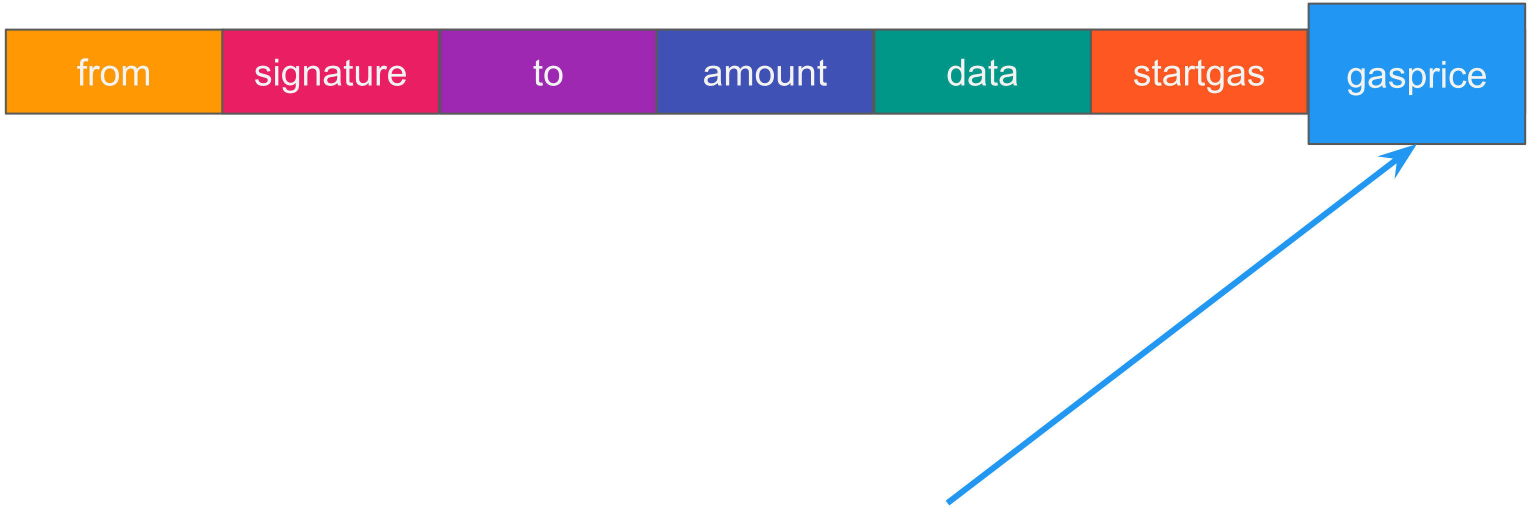




Maximum amount of gas willing to pay

Gas Limit

- All **unused gas** is **refunded** at the end of a transaction
- **Out of gas** transaction are **not refundable**



Price to pay per gas unit

Gas Price

- Measured in **gwei** (1 × 10^9 Wei)
- Determines how **quickly** a transaction will be **mined**

Transaction Fees

Gas Limit	×	Gas Price	=	Max transaction fee
50.000		20 Gwei		0.001 ETH

Gas costs

Operation	Gas	Description
ADD/SUB	3	Arithmetic operation
MUL/DIV	5	Arithmetic operation
ADDMOD/MULMOD	8	Arithmetic operation
AND/OR/XOR	3	Bitwise logic operation
LT/GT/SLT/SGT/EQ	3	Comparison operation
POP	2	Stack operation
PUSH/DUP/SWAP	3	Stack operation
MLOAD/MSTORE	3	Memory operation

Gas costs

Operation	Gas	Description
JUMP	8	Unconditional jump
JUMPI	10	Conditional jump
SLOAD	200	Read from storage
SSTORE	20.000	Write to storage
BALANCE	400	Get balance of an account
CREATE	32.000	Create a new account using CREATE
CALL	25.000	Message-call into an account

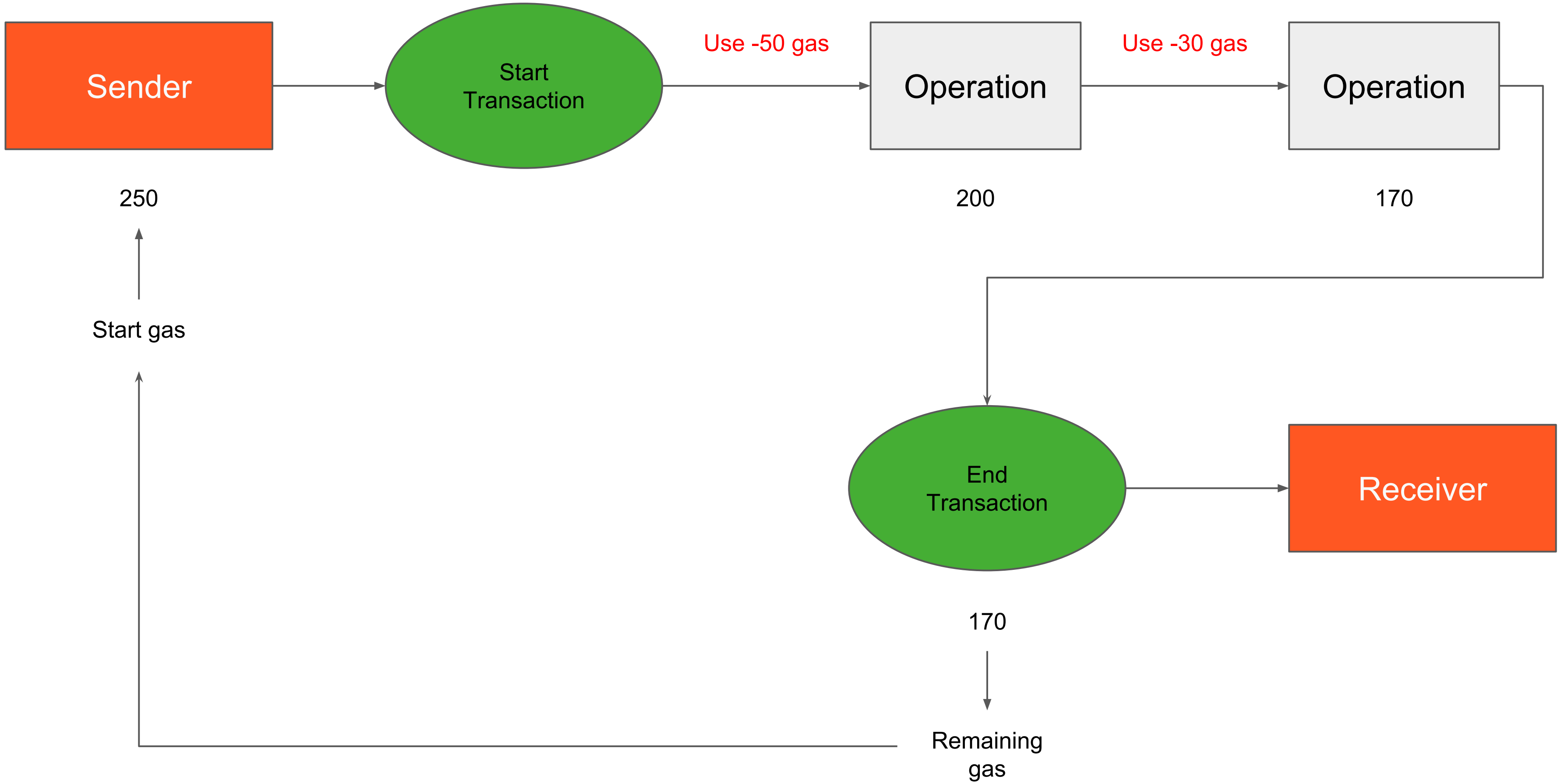
Storage in Ethereum

ETH Price: $166.41 (Apr 17, 2019) - Gas Price: 3 Gwei

Size	Gas	Cost (ETH)	Cost ($)
32 bytes	21.000	0.000063	$0.36088
1KB	724.664	0.002174	$0.01046
1MB	~697.325.562	2.09198	$347.268
10MB	~7.000.000.000	~21	$3,486
100MB	~70.000.000.000	~210	$34,860
1GB	~700.000.000.000	~2100	$348,600

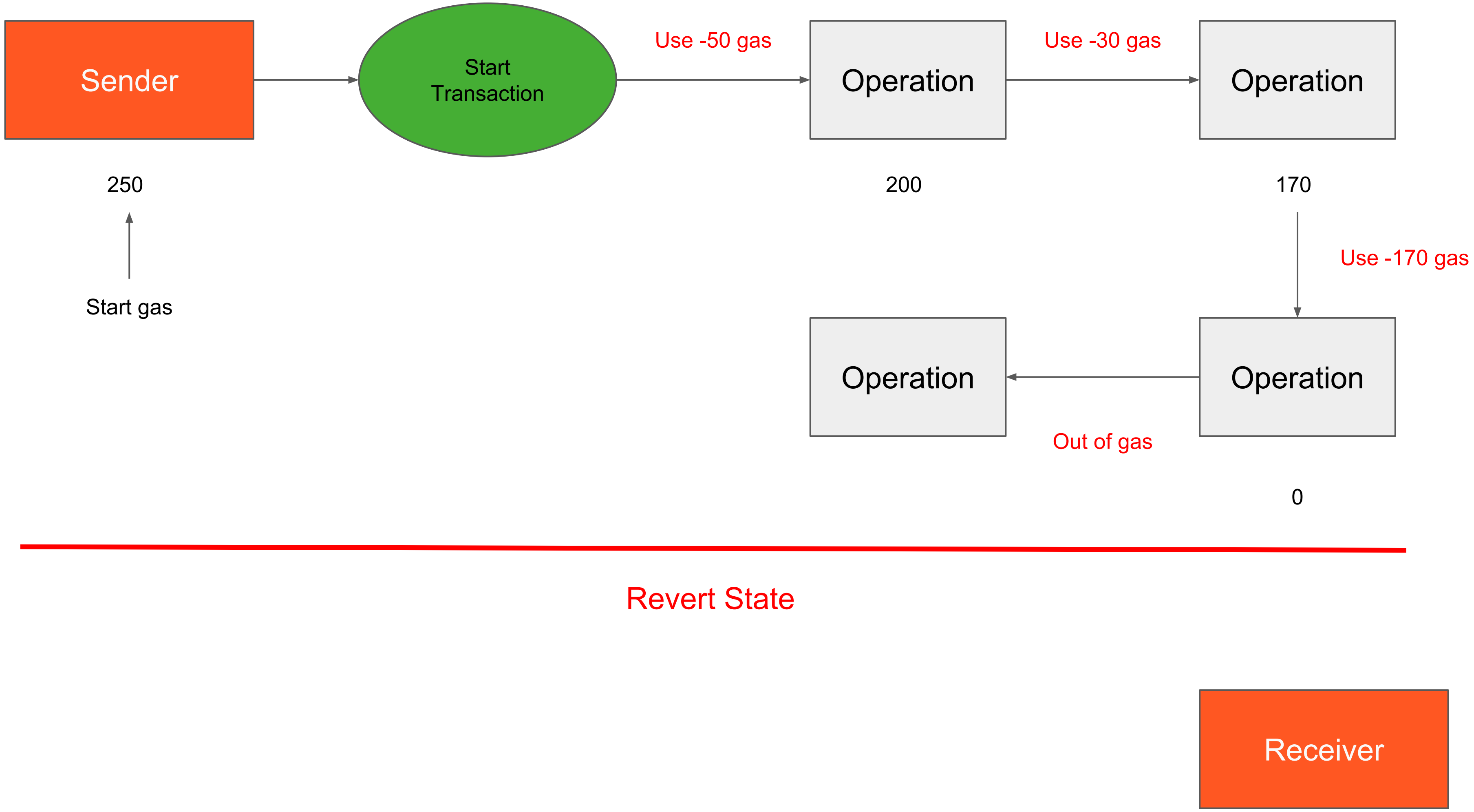
Computation steps

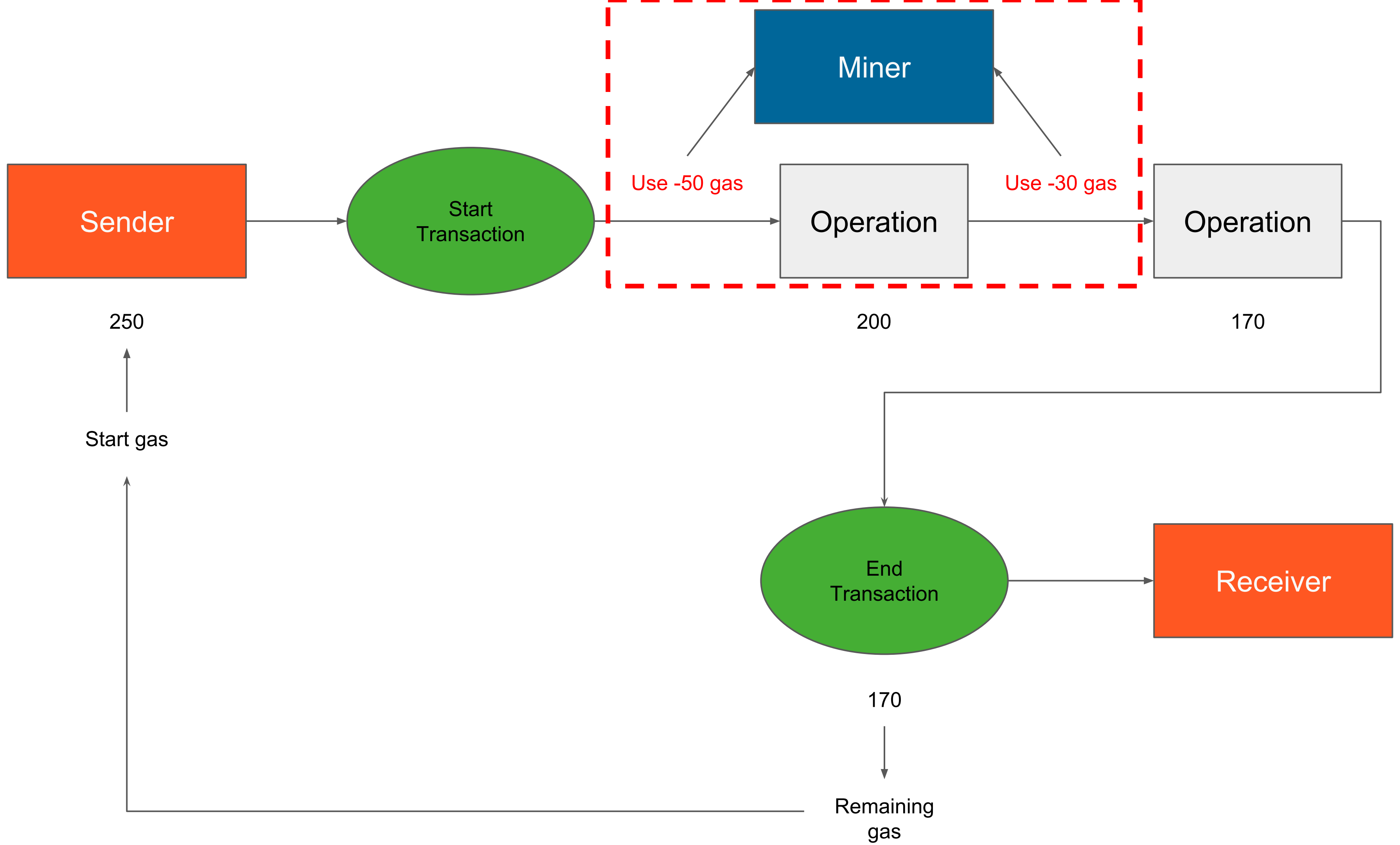
1. If **gas_limit * gas_price** > **balance** then **halt**
2. **Deduct** gas_limit * gas_price from **balance**
3. Set gas = gas_limit
4. **Run code** deducting from gas
5. After termination **return remaining gas** to **balance**


Sender
Start Transaction
Use -50 gas
Operation
Use -30 gas
Operation
250
200
170
Start gas
End Transaction
Receiver
170
Remaining gas

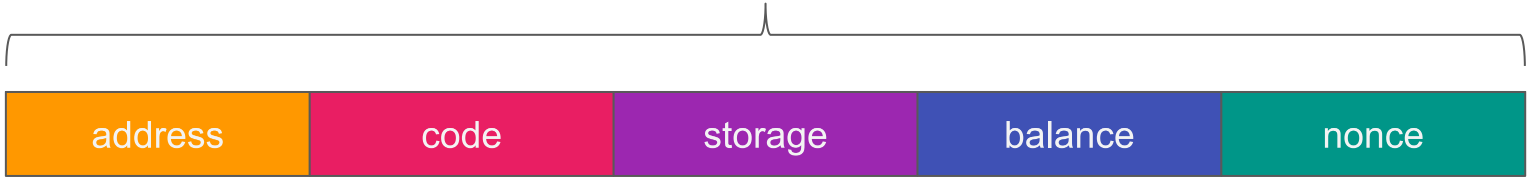
Out of gas exceptions

- State **reverts** to **previous state**
- gas_limit * gas_price is **still deducted** from **balance**

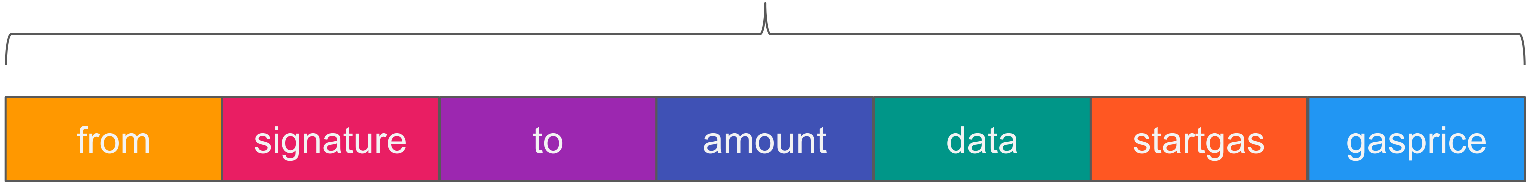

Sender
Start Transaction
Use -50 gas
Operation
Use -30 gas
Operation
250
200
170
Start gas
Use -170 gas
Operation
Operation
Out of gas
0
Revert State
Receiver


Miner
Sender
Start Transaction
Use -50 gas
Operation
Use -30 gas
Operation
250
200
170
Start gas
End Transaction
Receiver
170
Remaining gas

Account



Transaction



Ethereum Virtual Machine

- Series of **bytecode** instructions (EVM code)
- Each **bytecode** represents an **operation** (opcode)
- A quasi **Turing complete** machine
- **Stack-based** architecture (1024-depth)
- **32-byte** words (256-bit words)
- **Crypto** primitives

EVM bytecode

```
PUSH1 0
CALLDATALOAD
SLOAD
NOT
PUSH1 9
JUMPI
STOP
JUMPDEST
PUSH1 32
CALLDATALOAD
PUSH1 0
CALLDATALOAD
SSTORE
```

EVM: contract execution

- Three types of storage:
  - **Stack**
  - **Memory** (expandable byte array)
  - **Storage** (key/value store)
- All memory is **zero-initialized**
- Access: **value**, **sender**, **data**, **gas** limit and **block header** data (depth, timestamp, miner, hash)

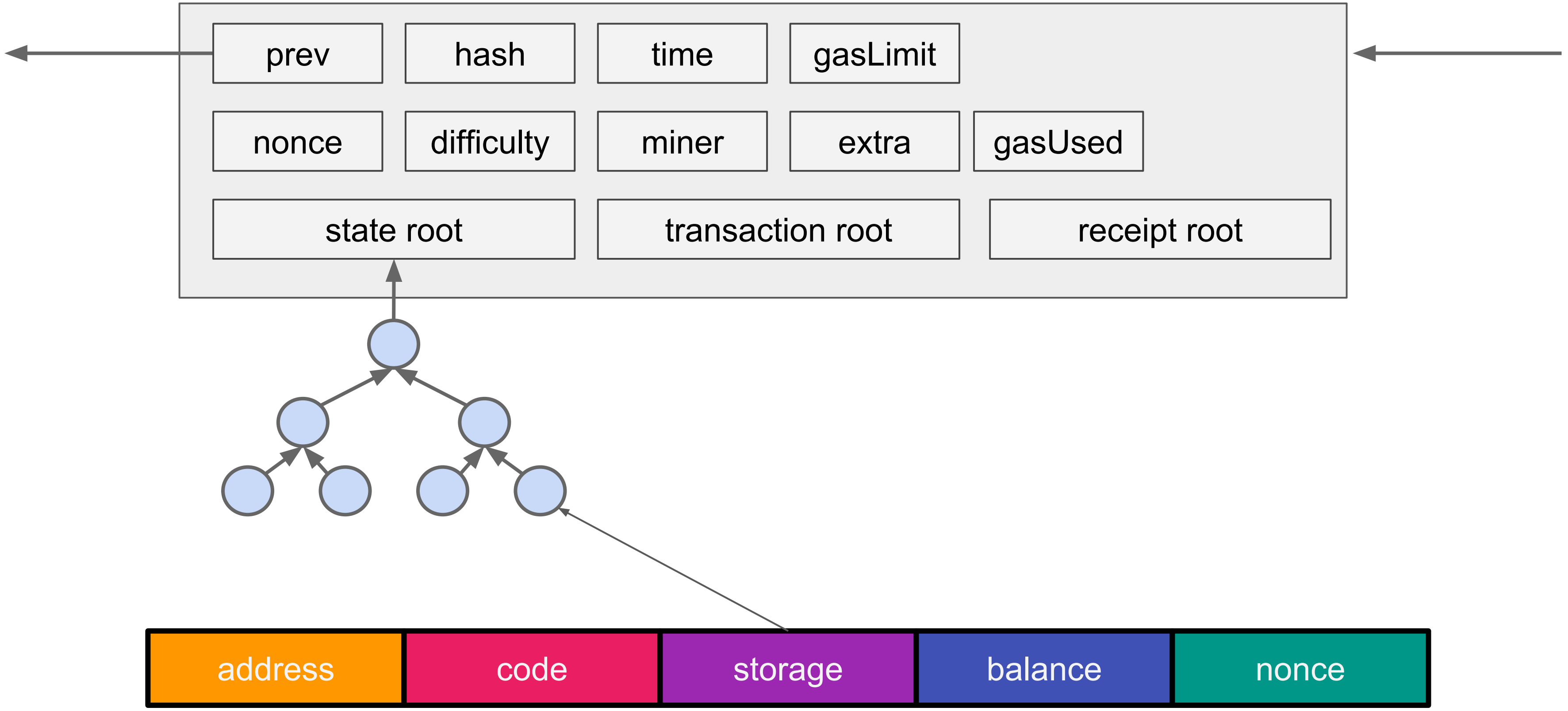
Ethereum Mining

- **Similar** to **Bitcoin**
- **Blocks** contain: **transaction** list and most **recent state**
- Block **time**: ~12 - 15 **seconds**
- **Proof-of-work**: Ethash (designed to be **memory-hard**)
- Casper: Future transition to **proof-of-stake**
- **Winner** of the block: **3 ETH**

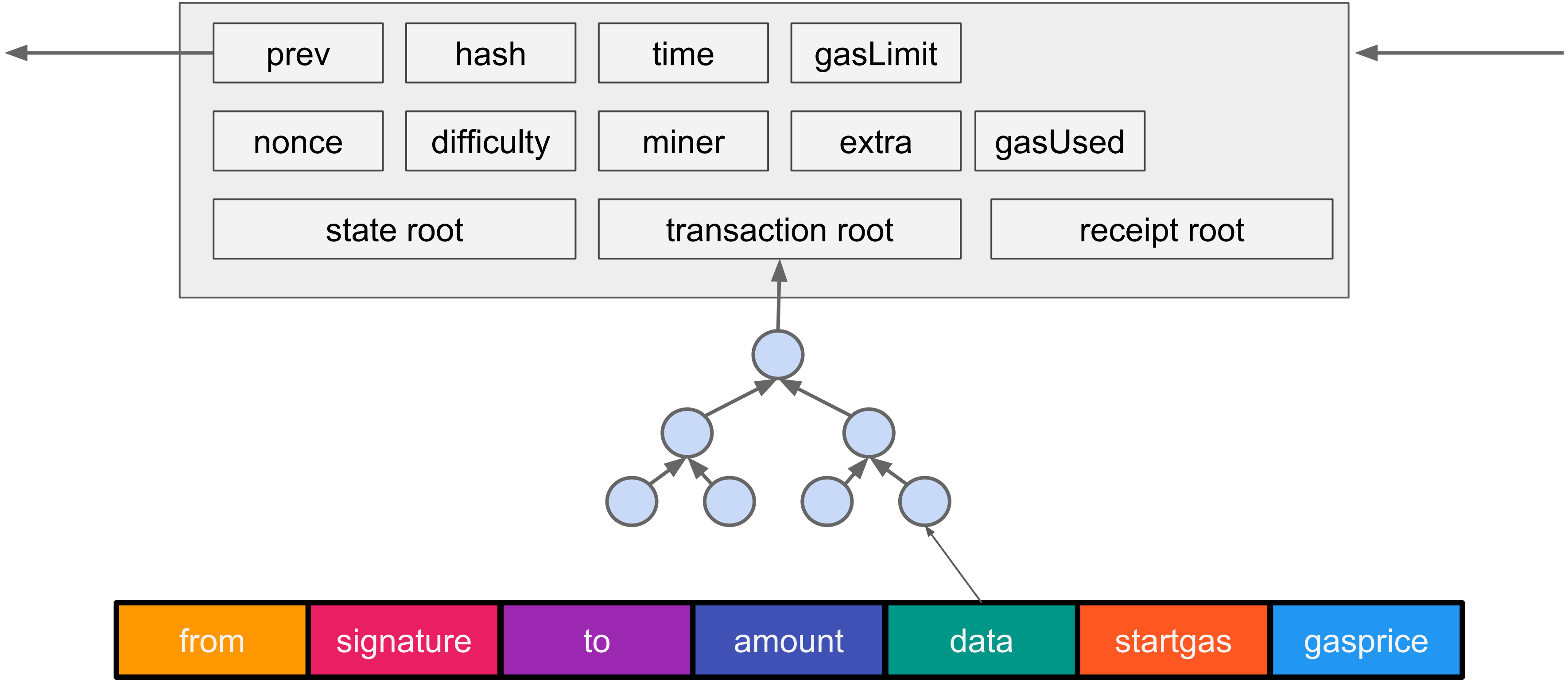
Ethereum Mining

- Uses a variant of **GHOST** (Greedy Heaviest Observed Subtree) protocol to **reward stale blocks**
- The GHOST protocol rule **picks the chain** that has had the **most computation** done upon it
- Planned hard forks:
    - Frontier, Homestead, and Byzantium (Metropolis phase 1)
    - Next one at 2019: Constantinople (Metropolis phase 2)

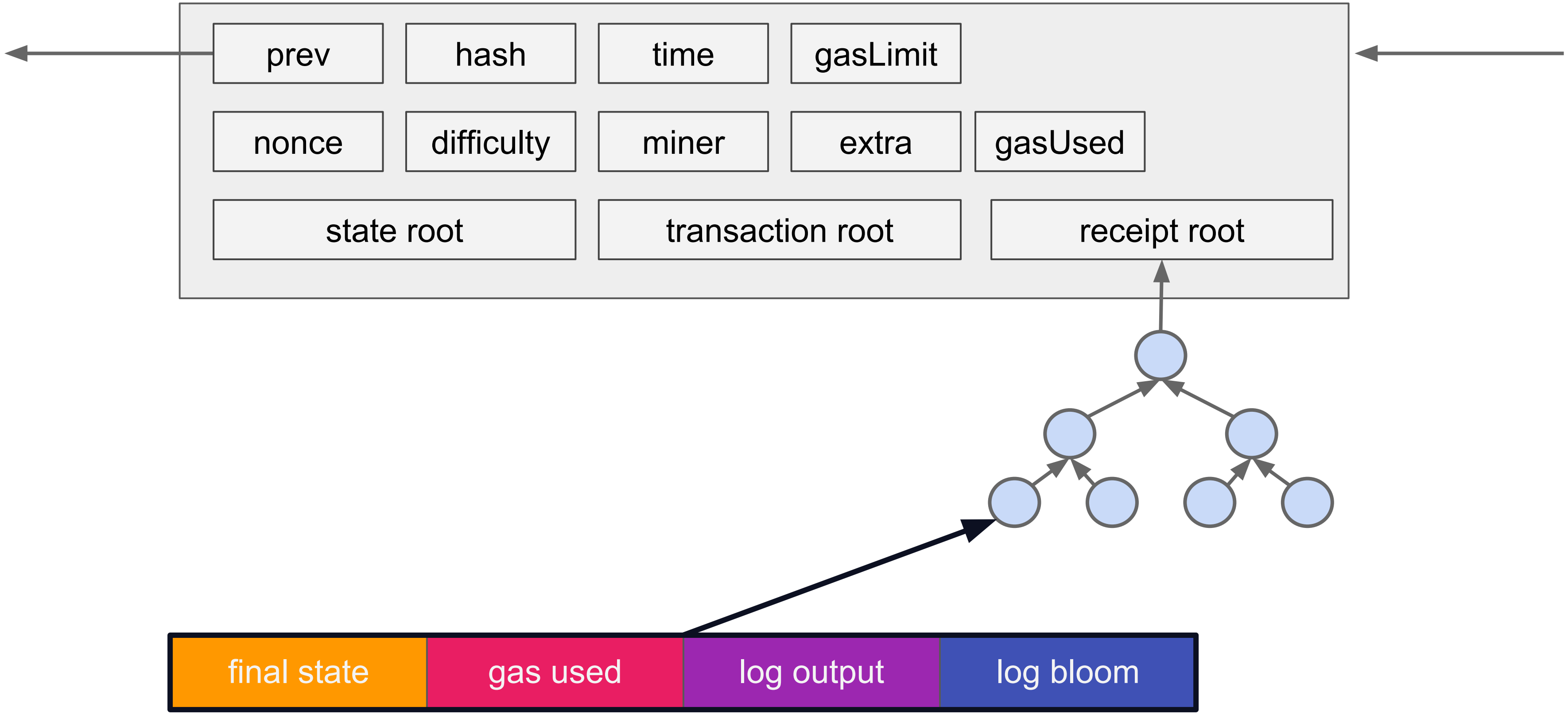
Ethereum block



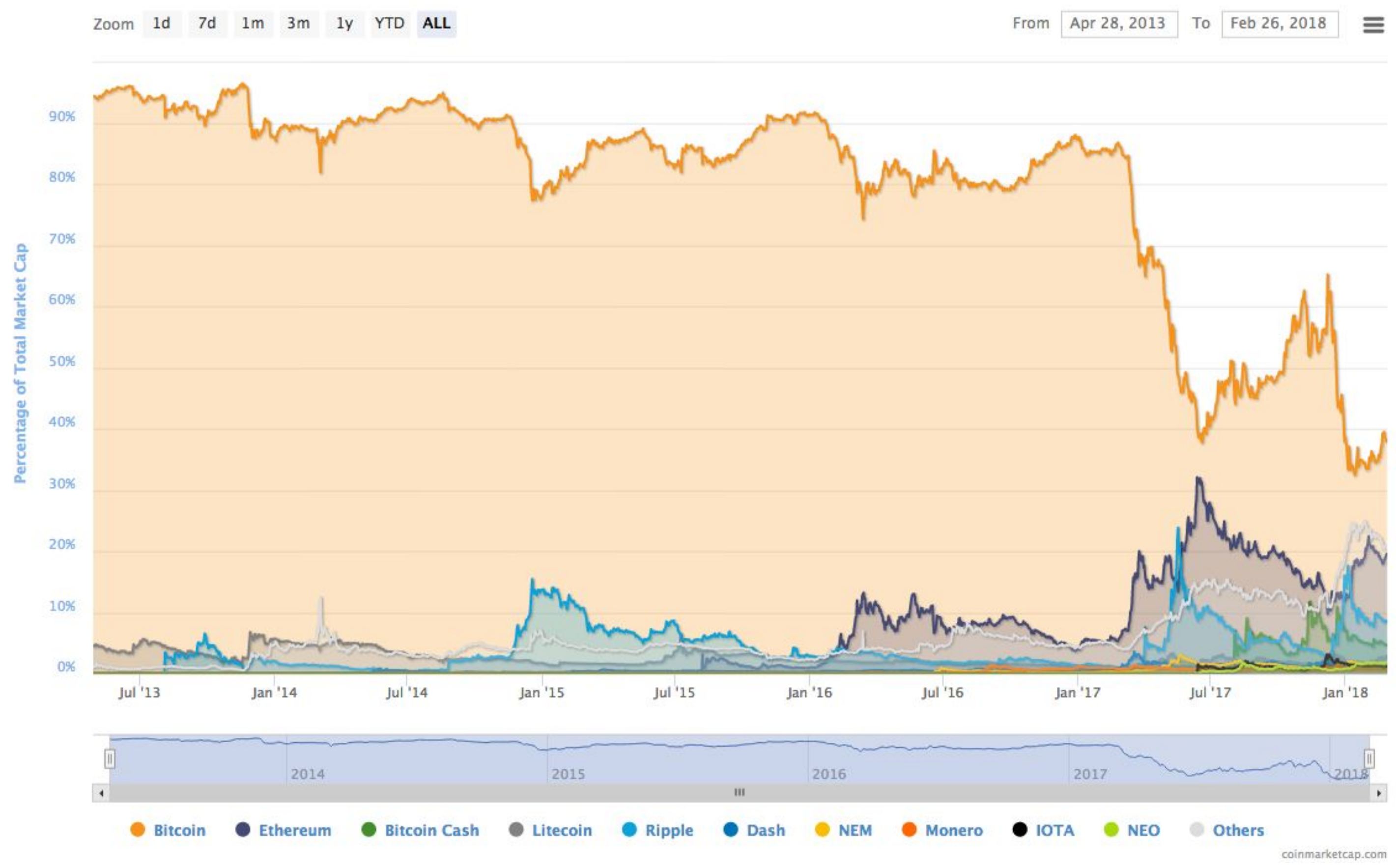
Ethereum block

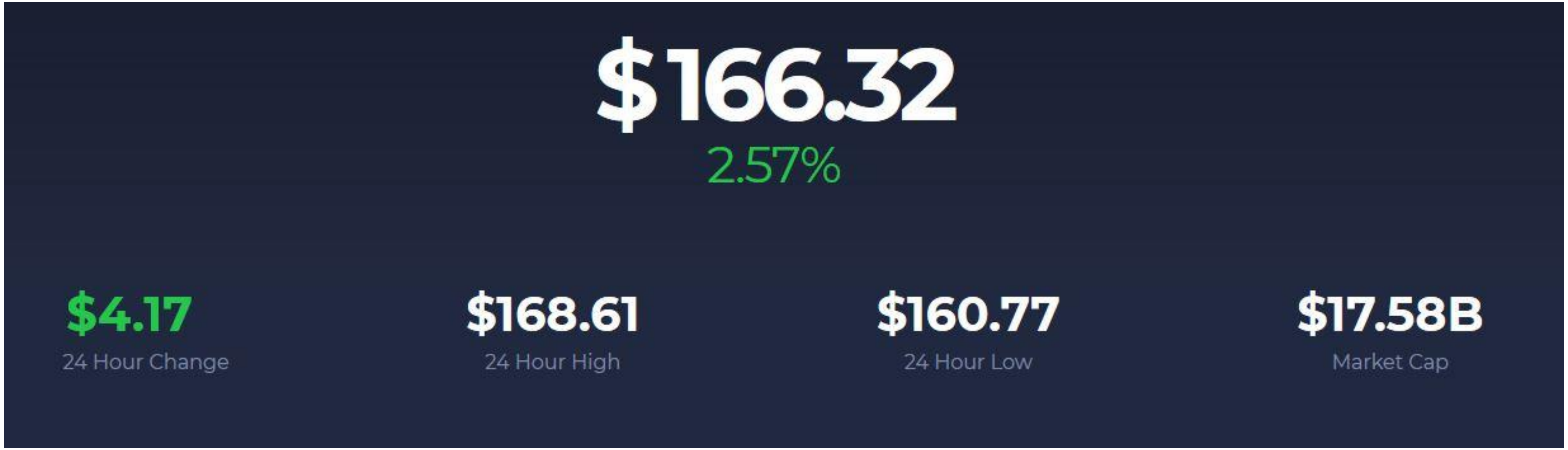


Ethereum block



Percentage of Total Market Capitalization (Dominance)





$166.32

2.57%

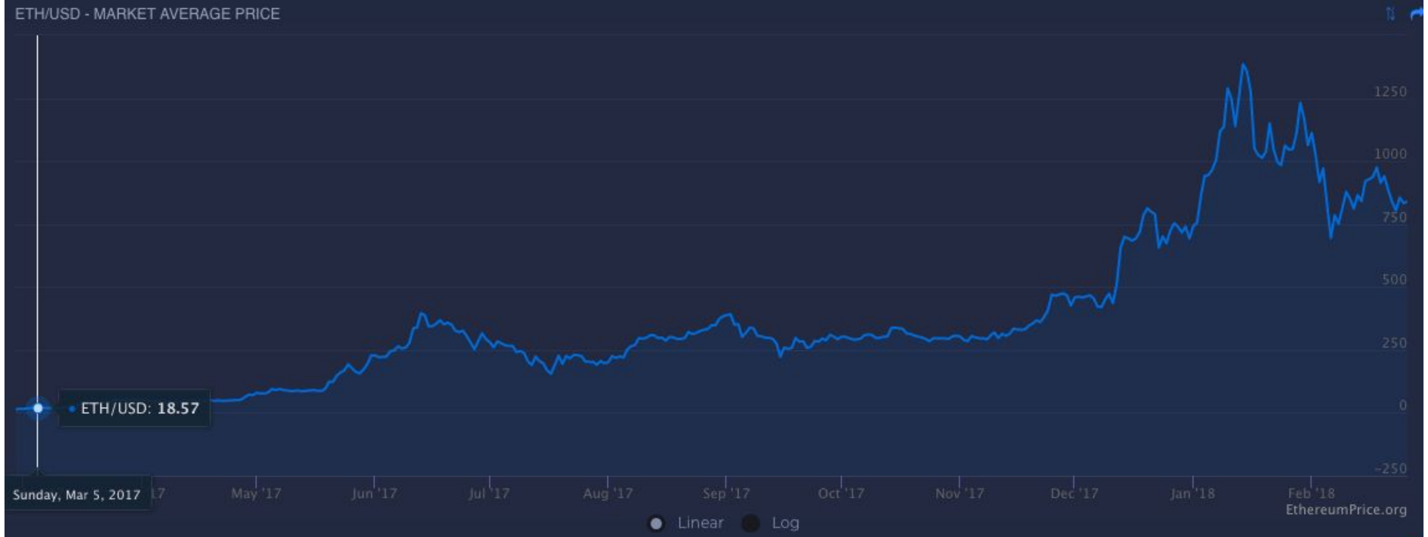
$4.17
24 Hour Change

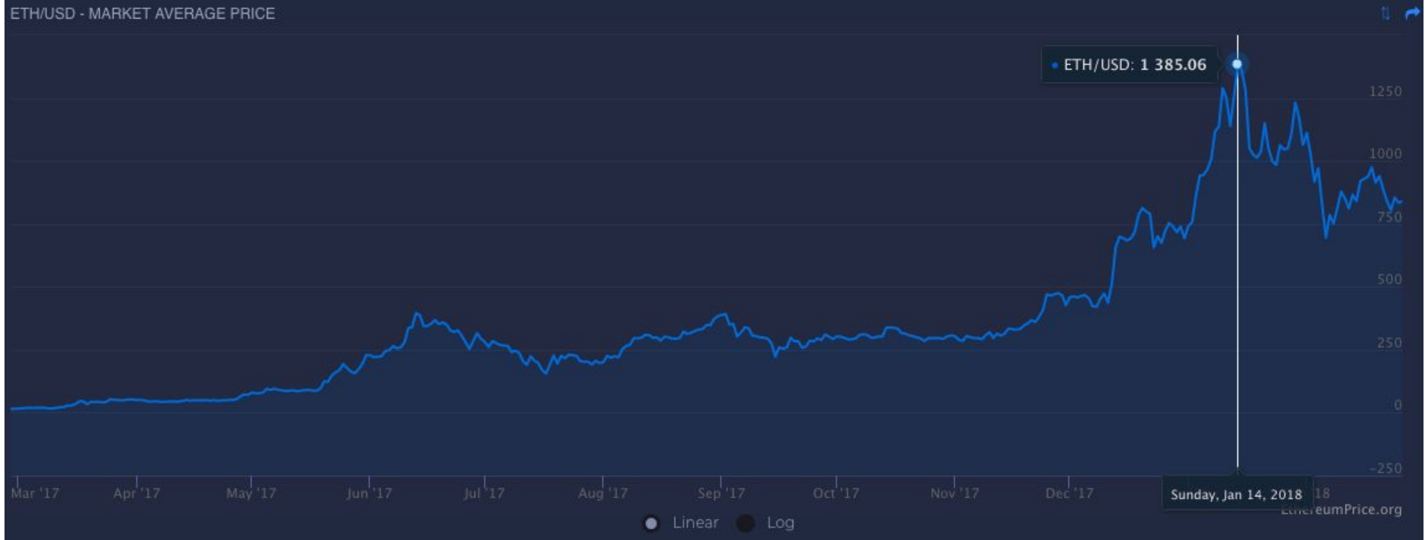
$168.61
24 Hour High

$160.77
24 Hour Low

$17.58B
Market Cap

17 April 2019


ETH/USD - MARKET AVERAGE PRICE
ETH/USD: 18.57
Sunday, Mar 5, 2017
1250
1000
750
500
250
0
-250
May '17
Jun '17
Jul '17
Aug '17
Sep '17
Oct '17
Nov '17
Dec '17
Jan '18
Feb '18
EthereumPrice.org
Linear
Log


ETH/USD - MARKET AVERAGE PRICE
ETH/USD: 1 385.06
Sunday, Jan 14, 2018
1250
1000
750
500
250
0
-250
Mar '17
Apr '17
May '17
Jun '17
Jul '17
Aug '17
Sep '17
Oct '17
Nov '17
Dec '17
EthereumPrice.org
Linear
Log

Thank you!