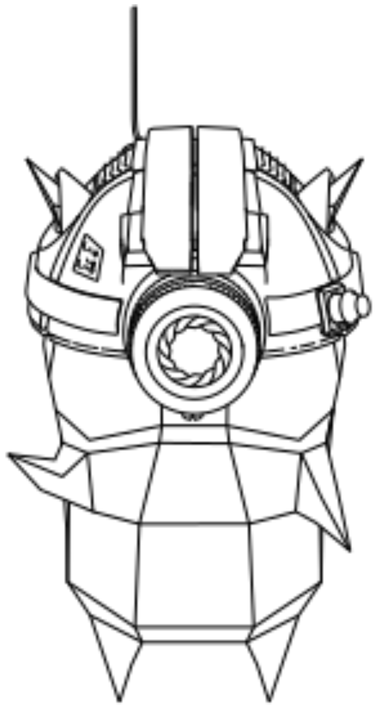
Talking to geth

Diving into go-ethereum API



Sebastian Ławniczak, 19.01.2023

About me

- Senior Software Engineer at Monerium
  - We provide IBAN for your Crypto Wallet
  - Check us out at: https://monerium.app/

What’s blockchain actually?

- Chain of blocks connected by cryptographic hashes
- Every new block contains reference to its parent block
- Block is a batch of changes made to the previous block
- All participants (accounts) on the network agree on the number and history of blocks

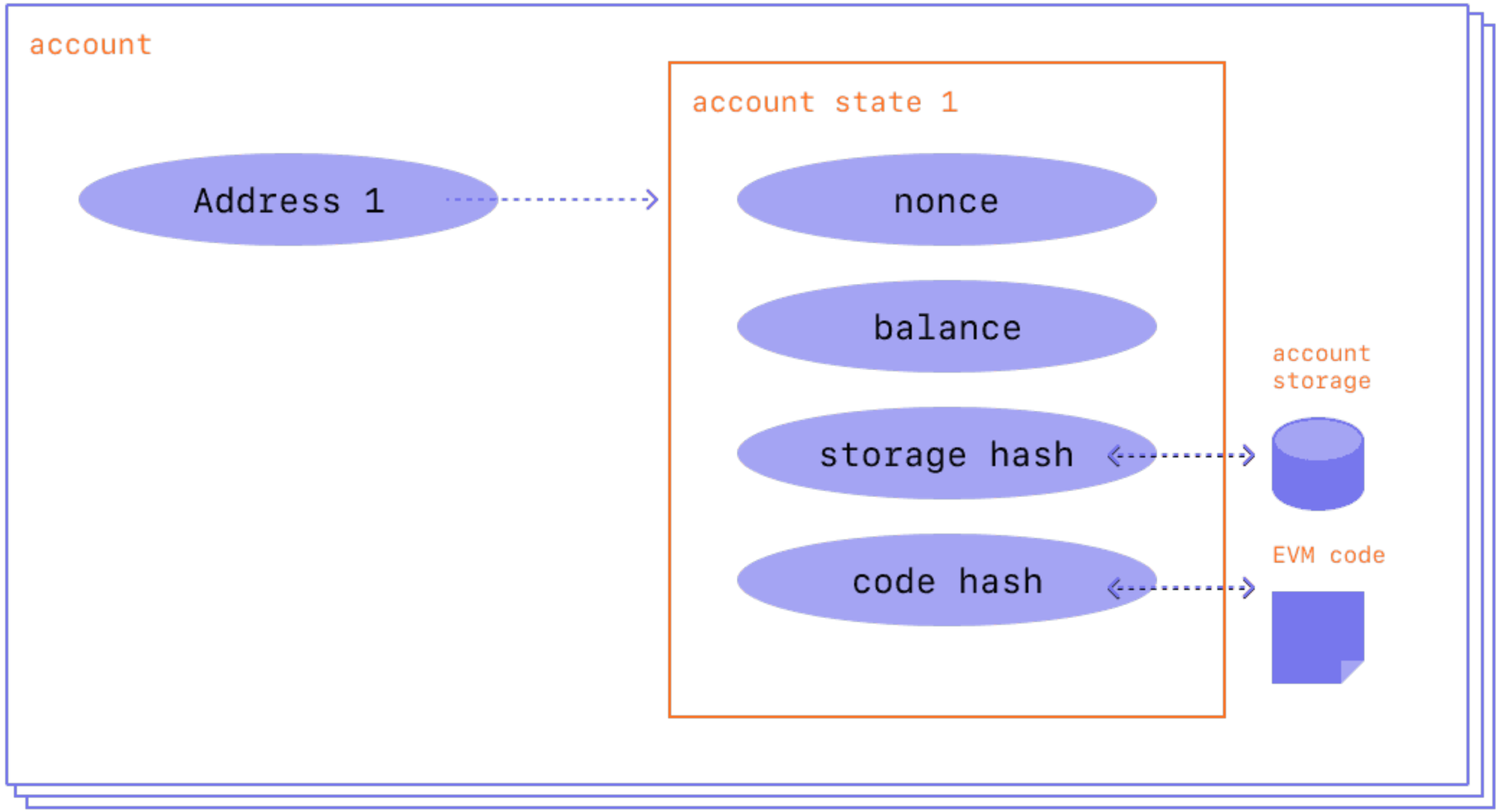
What’s blockchain actually?

- Chain of blocks connected by cryptographic hashes
- Every new block contains reference to its parent block
- Block is a batch of changes made to the previous block
- All participants (accounts) on the network agree on the number and history of blocks.
- Sounds like **Git**, right?

Accounts

- Externally Owned Accounts
  - controlled by private keys (ECDSA)
  - “not your keys, not your coins”
- Contracts accounts
  - smart contracts deployed to the network
  - controller by code

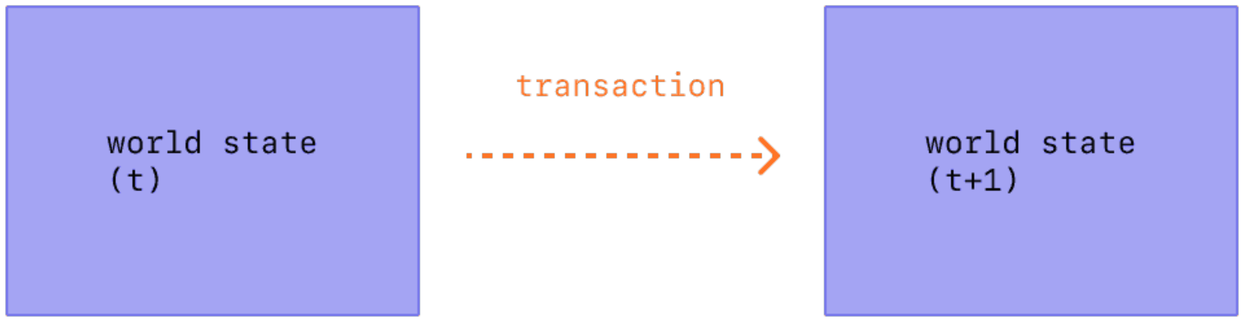
Accounts



https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

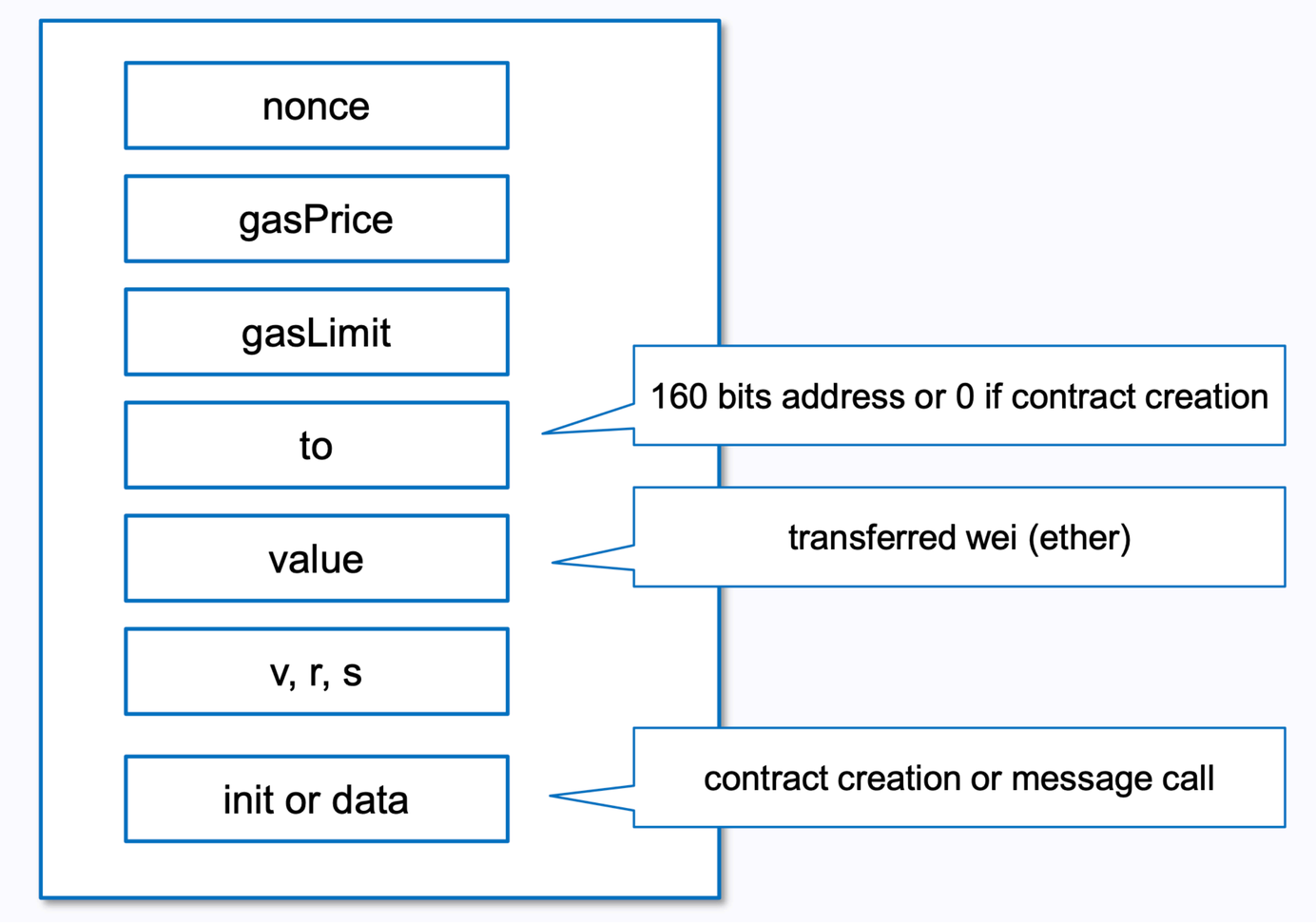
Transactions

- Cryptographically signed
- Transactions modify state of the Ethereum
- Examples:
  - transferring Ether between accounts
  - creating / interacting with smart contracts



https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

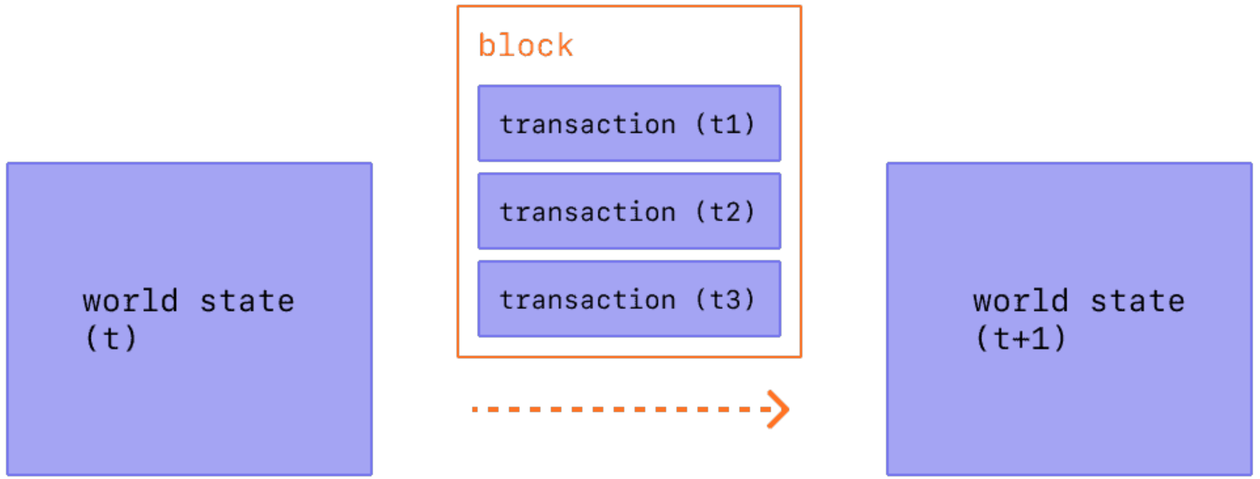
Transactions



https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

Blocks

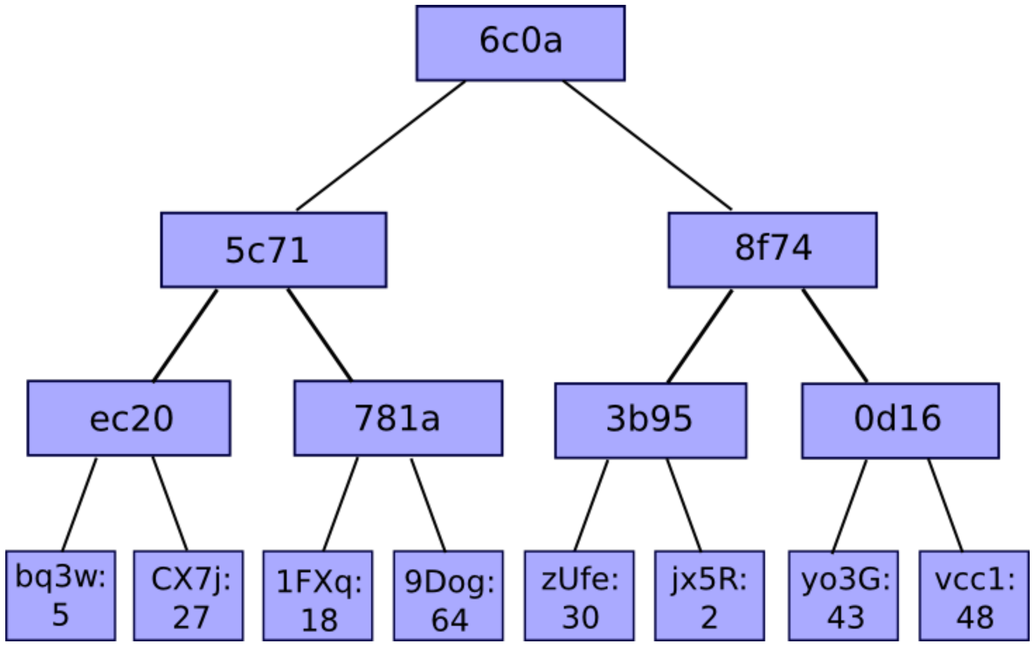
- Block is a batch of transactions with hash of the prev. block
- Produced every 12 seconds



https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

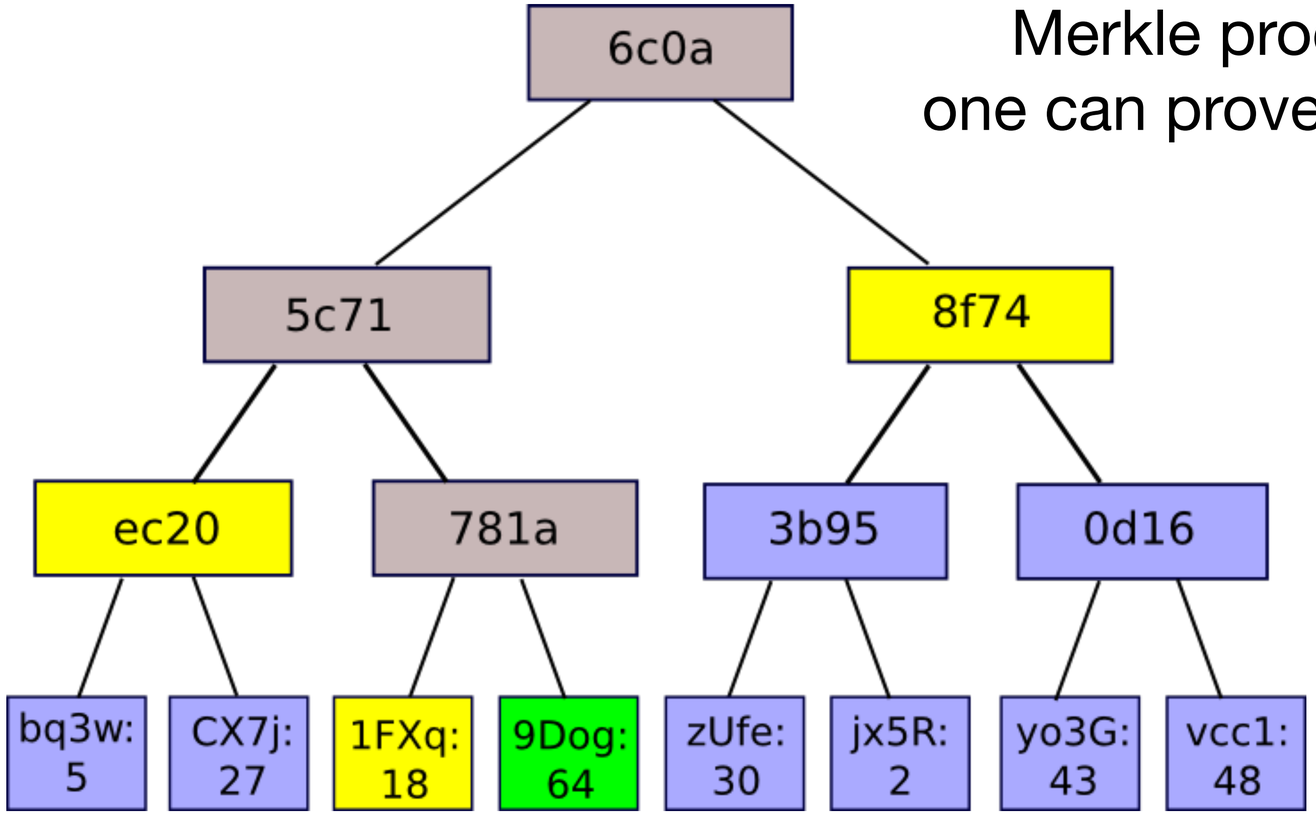
Merkle trees

- Storing all data in block is expensive
- Ethereum's block stores roots of three merkle trees
  - Transactions
  - Receipts ('effects' of transactions)
  - State



https://blog.ethereum.org/2015/11/15/merkling-in-ethereum

Merkle trees



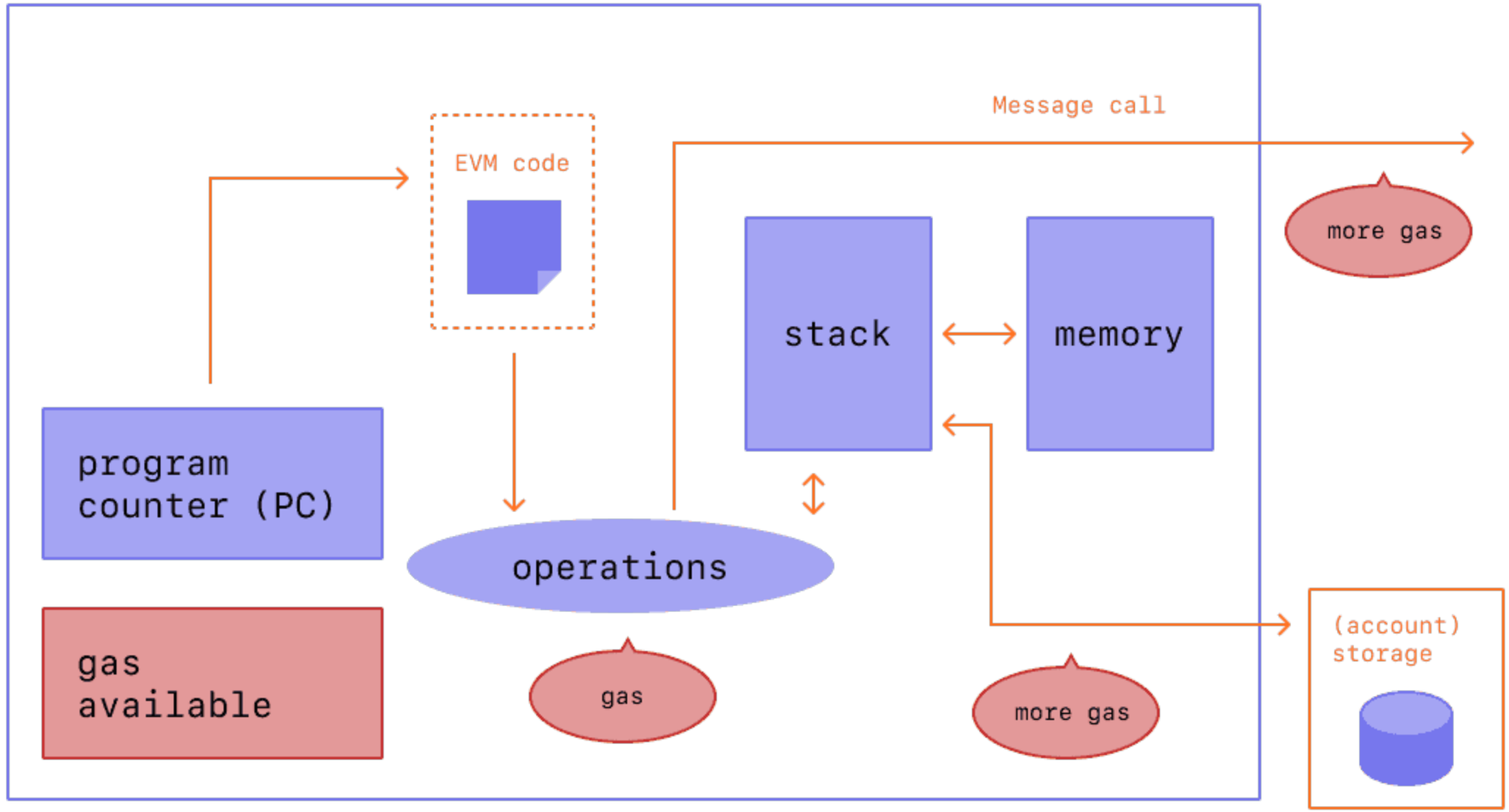
Merkle proof - by providing yellow hashes
one can prove the green transaction is included.

https://blog.ethereum.org/2015/11/15/
merkling-in-ethereum

Ethereum Virtual Machine (EVM)

- Distributed State Machine
  - Specification: Ethereum's Yellowpaper
- EVM its own:
  - instruction set (think assembler)
  - volatile memory (like RAM)
  - persistent storage (like disk, expensive)

Ethereum Virtual Machine (EVM)



https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf

Gas

- Fuel that allows Ethereum to operate
- ETH - Ether, native cryptocurrency of Ethereum
  - used for paying for computation and storage
  - also award for validator (fee)
  - 1 ether = 10^9 gwei = 10^18 wei
- Example: storing 32-byte word costs 20 000 gas units
  - 1 gas unit is ~33 gwei
  - 20 000 * 33 gwei = 66 000 gwei (~ $0.1)

Gas

- Example:
  - Alice sends 1 ETH to Bob
  - Minimum gas that guarantees execution = 21 000 units
  - Base fee, set by the network = 14 gwei
  - Priority tip, set by the EOA = 2 gwei
  - 21 000 * (10 + 2) = 252 000 gwei (~ $0,38)
  - Result: 1.000252 ETH is deducted from Alice’s account.
- cheap ETH -> cheap transactions :)

https://coinmarketcap.com/currencies/ethereum/

Smart contracts - solidity

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Storage {
    uint public value;

    event ValueChanged(address indexed sender, uint indexed value);

    function setValue(uint _value) public {
        value = _value;
        emit ValueChanged(msg.sender, _value);
    }
}
```

Smart contracts - ABI


```
[
  [...]
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "_value",
        "type": "uint256"
      }
    ],
    "name": "setValue",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  [...]
]
```


- ABI indicates how to build a stream of bytes that must be sent in ‘data’ field of transaction when we want to interact with smart contract.

Smart contracts - opcodes

```
/* "storage.sol":174:290  function setValue(uint _value) public */
tag_4:
  tag_9
  0x04
  dup1
  calldatasize
  sub
  dup2
  add
  swap1
  tag_10
  swap2
  swap1
  tag_11
  jump
```

OPCODE	NAME	MINIMUM GAS	DESCRIPTION	Expand
00	STOP	0	Halts execution	
01	ADD	3	Addition operation	
02	MUL	5	Multiplication operation	
03	SUB	3	Subtraction operation	
04	DIV	5	Integer division operation	
05	SDIV	5	Signed integer division operation (truncated)	
06	MOD	5	Modulo remainder operation	

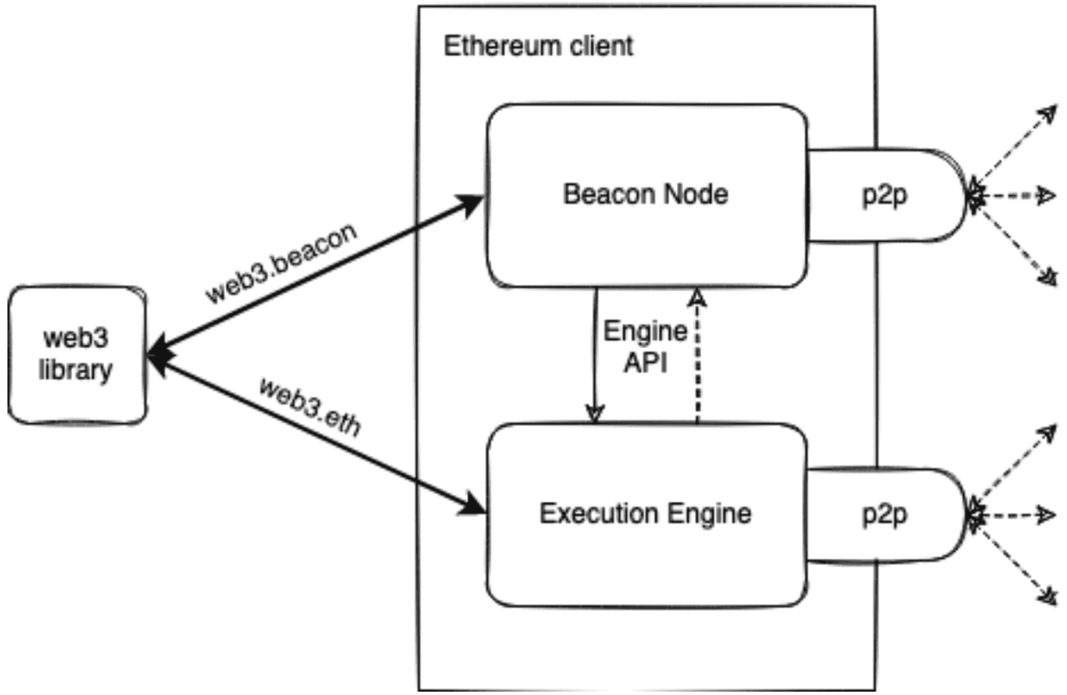
https://www.evm.codes

Demo: Compiling smart contracts

```
$ brew tap ethereum/ethereum
$ brew install solidity             # solc
$ brew install ethereum         # abigen
```

```
$ solc --abi storage.sol -o .
$ solc --evm storage.sol -o .
$ solc --bin storage.sol -o .
$ abigen --abi=storage.abi --bin storage.bin --pkg storage --out pkg/storage/storage.go
```

Nodes and Networks



- Networks:
  - mainnet
  - public testnets: goerli, sepolia
- Node components
  - execution engine - processing and broadcasting transactions
  - beacon node - handling consensus algorithm

Nodes and Networks

- Node types:
  - Full node - full blockchain data
  - Light node - just a minimum to connect to the network
  - Archive node
- How do I connect?
  - By hosting a Light node
  - Via web3 IaaS: Infura, Alchemy (centralized!)

Demo: Deploying smart contract to testnode

- Anvil, part of Foundry toolset, is used to create a local testnode
  - https://getfoundry.sh/

Demo: Setting the Value

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Storage {
    uint public value;

    event ValueChanged(address indexed sender, uint indexed value);

    function setValue(uint _value) public {
        value = _value;
        emit ValueChanged(msg.sender, _value);
    }
}
```

Demo: Getting the Value

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Storage {
    uint public value;

    event ValueChanged(address indexed sender, uint indexed value);

    function setValue(uint _value) public {
        value = _value;
        emit ValueChanged(msg.sender, _value);
    }
}
```

Demo: Observing Value changes

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.17;

contract Storage {
    uint public value;

    event ValueChanged(address indexed sender, uint indexed value);

    function setValue(uint _value) public {
        value = _value;
        emit ValueChanged(msg.sender, _value);
    }
}
```

Further reading

- Docs: Ethereum
- Docs: go-ethereum
- Docs: Solidity
- Book: Ethereum Development with Go
- Blog post: What happens when you send 1 DAI?
- Blog post: Understanding Ethereum by studying the source code
- Blog post: My first impressions of web3

Going beyond

- Tokens - more than just ETH
- NFTs
- DeFi

Tokens

- ERC-20 - Token Standard
  - Each token can be exchanged
  - Good to represent tickets, fiat currencies, so on
- Basically: **map[address]int**
- Monerium use case: EURe <=> EUR

Tokens

- Ethereum's EIP-20
- OpenZeppelin's IERC20

Demo: Generating bindings for ERC20

NFTs

- ERC721 - Non-fungible Token Standard
  - each token is unique
  - useful for storing ownership of an asset
  - as long as you can calculate hash from it… ;)
- Think of it as: **map[hash]address**

NFTs

- Ethereum's EIP-721
- OpenZeppelin's IERC721

DeFi - Decentralized Finance

- Concept built on top of ERC20 & ERC721 and more.
- Lending & Borrowing
  - https://aave.com/
  - http://compound.finance
- AMMs - Automated Market Makers
  - http://uniswap.org
  - https://curve.fi/
- How do I bridge money to DeFi? http://monerium.app ;)

Thank you!

Questions?