

CSE446: Blockchain & Cryptocurrencies

Lecture - 13: Ethereum - 2

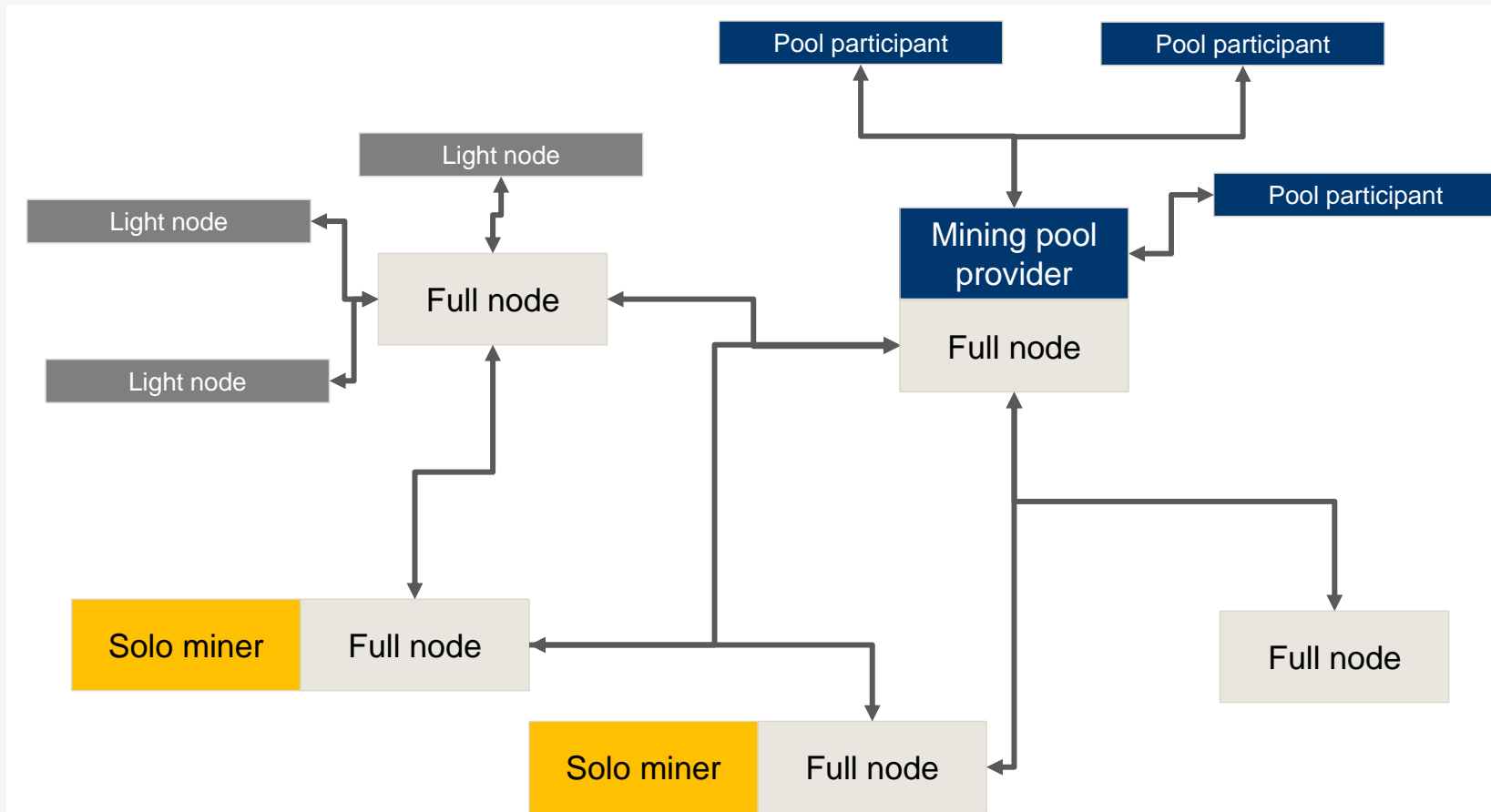


Inspiring Excellence

Agenda

- Ethereum network
- Ethereum EVM
- Ethereum Transactions
- Gas and Gas price

Ethereum network



Ethereum network node types

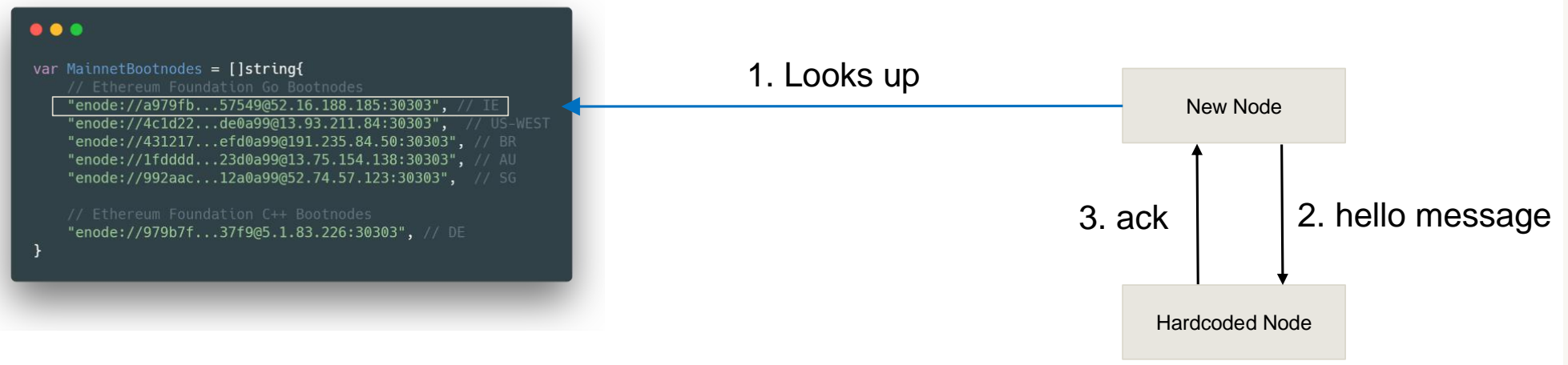
- Full nodes are the foundation of the Ethereum network
- Each full node holds a copy of the entire blockchain and syncs it with other nodes
- Transactions must be sent to a full node which distributes it among the network participants
- A light node is a client that is connected to a full node for the sake of not having to sync and download the entire blockchain
- For most private people, light nodes are the most comfortable way of interacting with the Ethereum blockchain
- One of the most common light nodes is <https://myetherwallet.com> (always triple check the domain)

Ethereum implementation

- To connect to the Ethereum network, you will need to download an Ethereum implementation
- Not all Ethereum nodes are using the same code base
- Since the specification for an Ethereum node is open source, basically anyone could create a different implementation
- The two major Ethereum implementations are: Geth and OpenEthereum

Node discovery

- Both Geth and OpenEthereum have a maintained list of default peers hardcoded into their source code
- Otherwise, it would be possible that no nodes are found, and the sync will always fail
- The Geth client comes with 6 hardcoded peers
- Once a node is selected, a hello message is sent to make an initial connection with the node



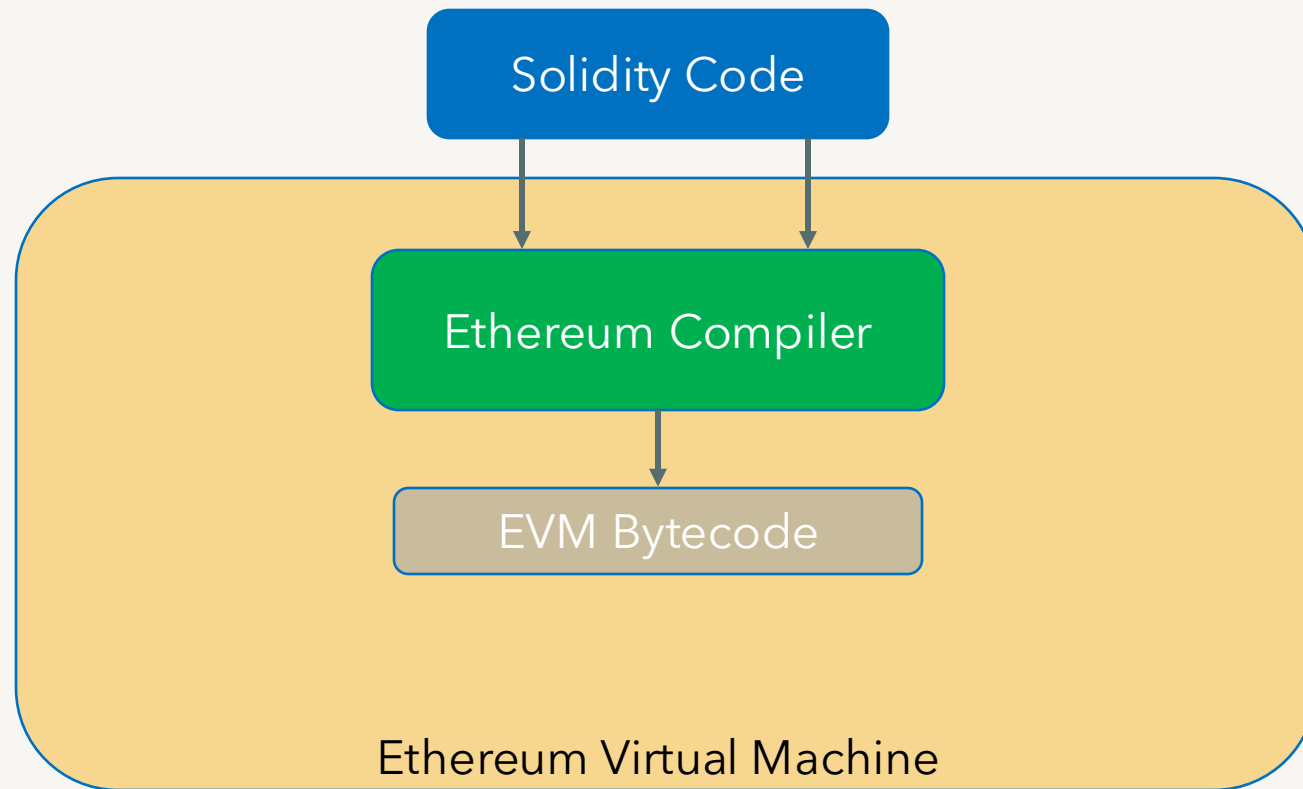
Ethereum Virtual Machine (EVM)

- The Ethereum Virtual Machine (EVM) is a simple but powerful, Turing complete 256bit Virtual Machine that allows anyone to execute arbitrary EVM Byte Code
- Turing completeness means you can write programs (contracts) that can (for the most part) solve any reasonable computational problem or write sophisticated logic
- It allows anyone to execute arbitrary code in a trust-less environment in which the outcome of an execution can be guaranteed and is fully deterministic

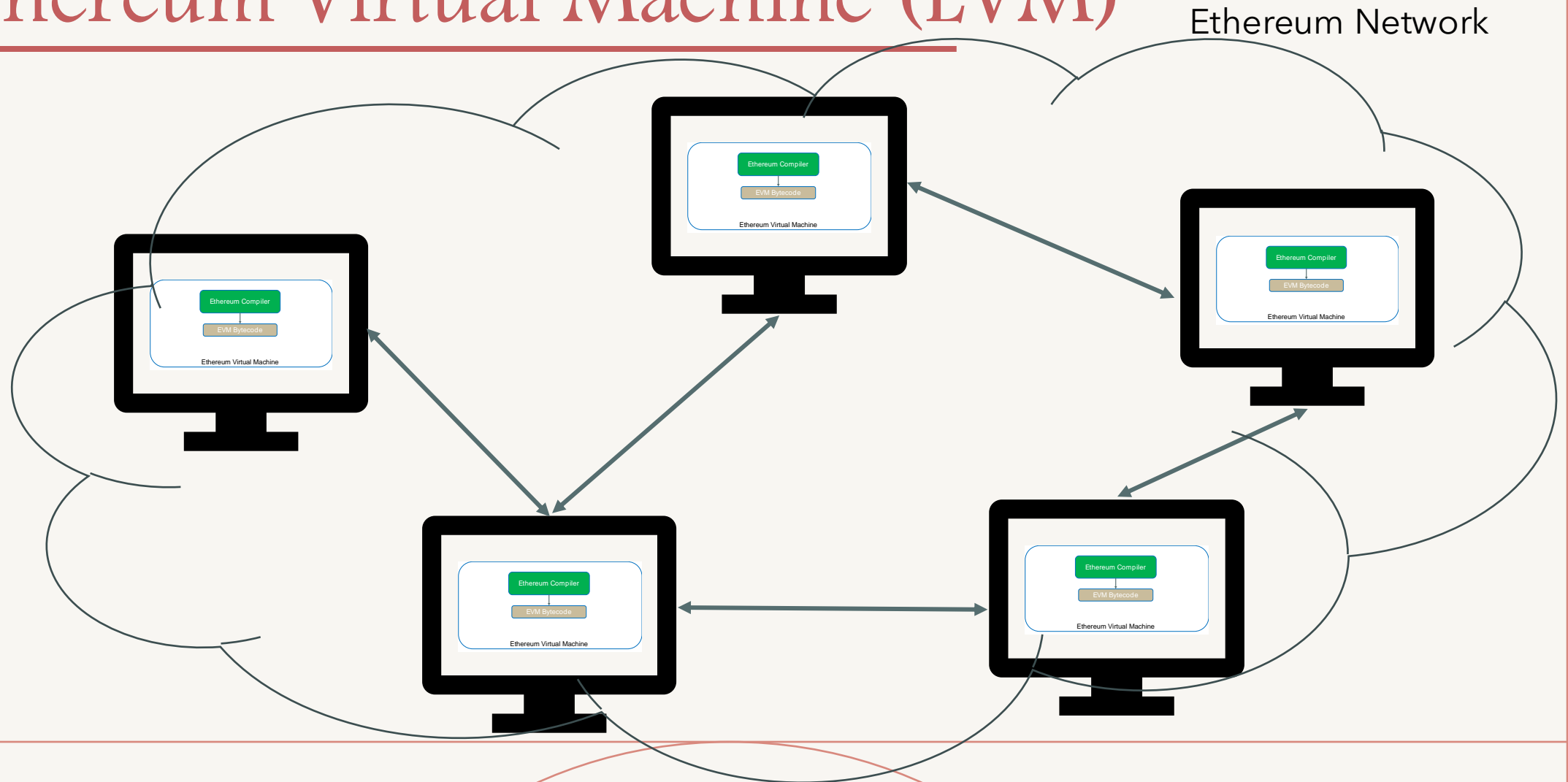
Ethereum Virtual Machine (EVM)

- Smart contract code is usually written in a high level programming language such as Solidity
- This code gets compiled to something called the EVM bytecode which gets deployed to the Ethereum blockchain
- This is very similar to a programming language like Java where the code gets converted to JVM Byte code
- The Ethereum runtime environment only understands and can execute the bytecode

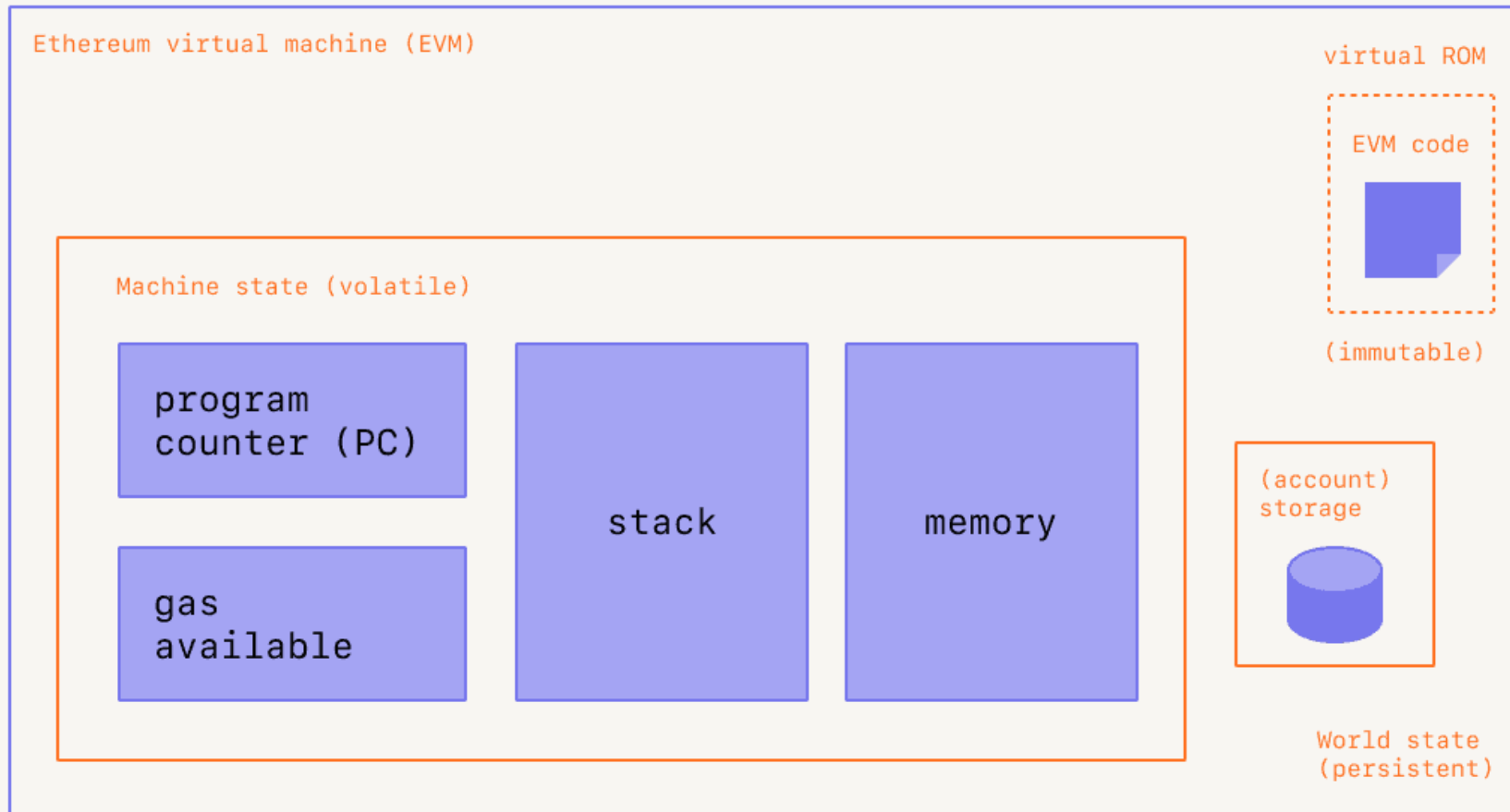
Ethereum Virtual Machine (EVM)



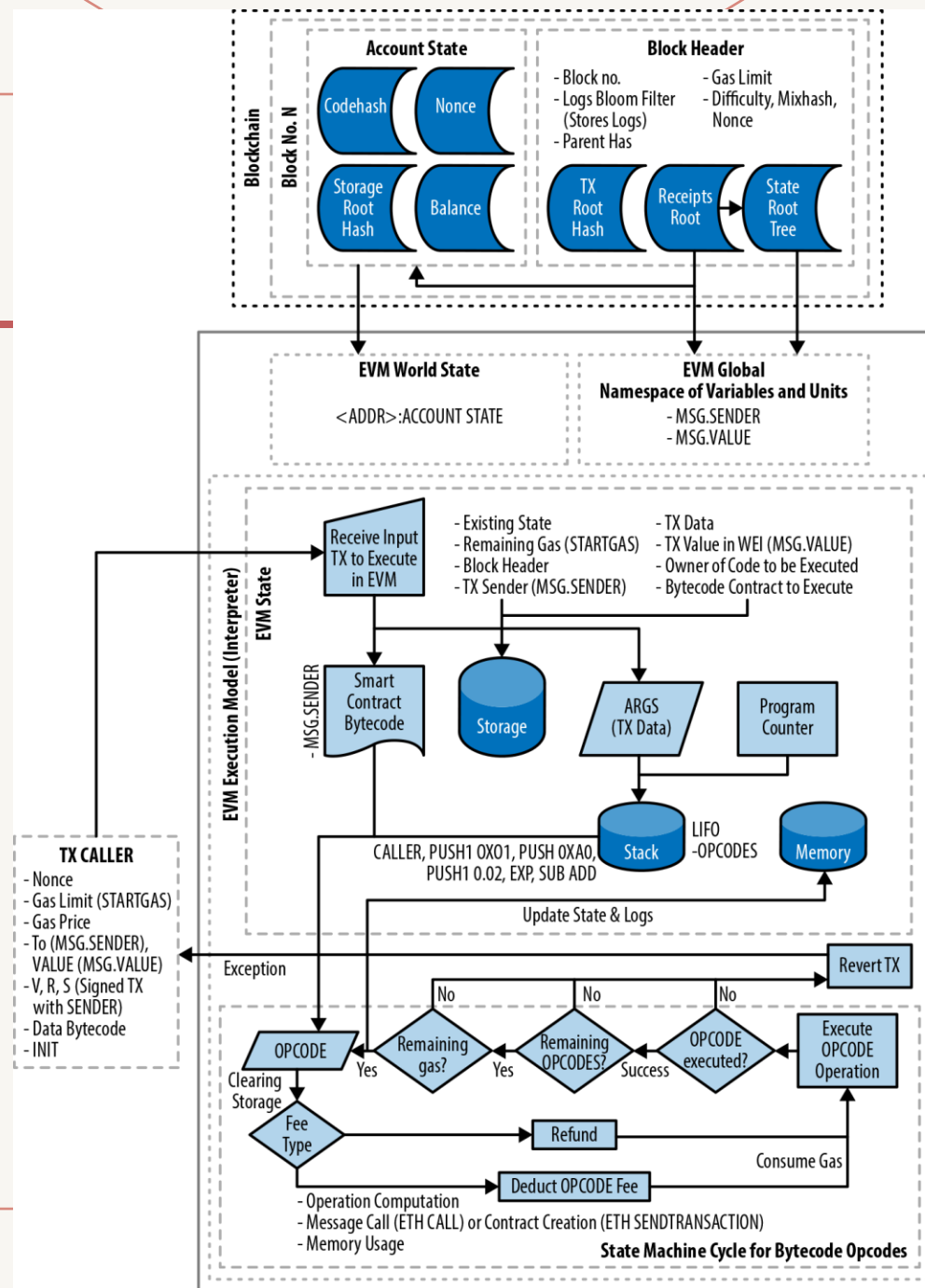
Ethereum Virtual Machine (EVM)



Ethereum Virtual Machine (EVM)



EVM



EVM Bytecode

- EVM bytecode is the instruction sets for the EVM
- The EVM instruction set offers most of the operations you might expect, including
 - Arithmetic and bitwise logic operations
 - Stack, memory, and storage access
 - Control flow operations
 - Logging, calling, and other operators
 - Access to account information (e.g., address and balance) and block information (e.g., block number and current gas price) (Blockchain state information)

Ethereum blockchain

- Like Bitcoin, there are three major sub-components in the Ethereum blockchain
 - Transactions
 - Blocks
 - Blockchain

Ethereum transactions

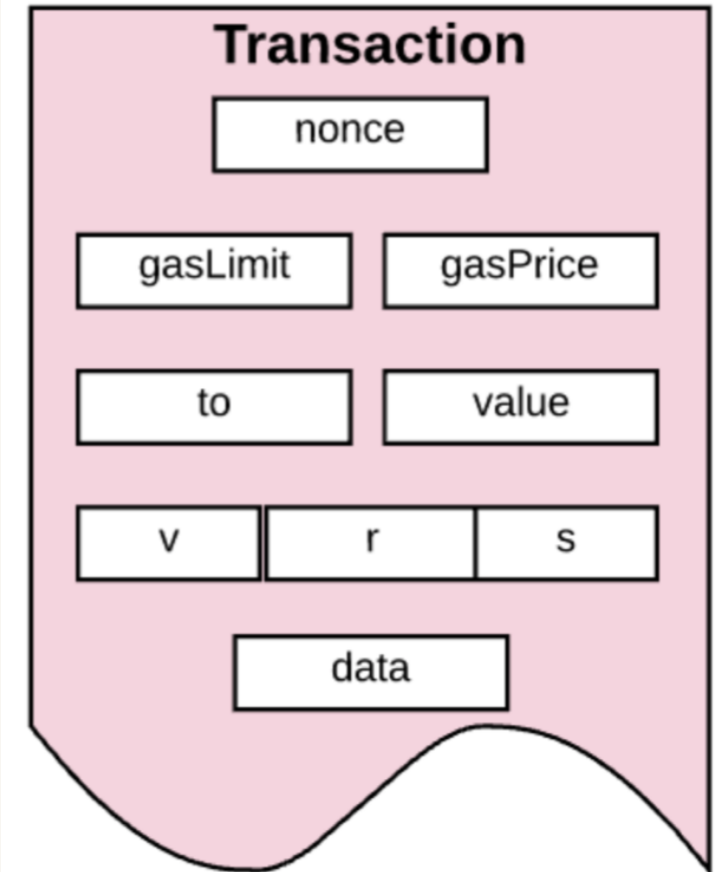
- On Ethereum there are mainly two transactions
 - External transactions
 - Internal transactions
- An external transaction
 - is generated by a user, serialised and sent to the Ethereum network, put on the blockchain (just like Bitcoin)
 - Includes payment (regular) transactions and contract creation & contract call (execution) transactions

Ethereum external transactions

- Regular transactions
 - a transaction from one account to another (either to an EOA or to a contract account)
- Contract deployment transactions:
 - to deploy a smart-contract
 - a transaction without a 'to' address, where the data field is used for the contract code
- Execution of a contract:
 - a transaction that interacts with a deployed smart contract
 - In this case, 'to' address is the smart contract address
 - Data field contains the relevant value for the contract (function)

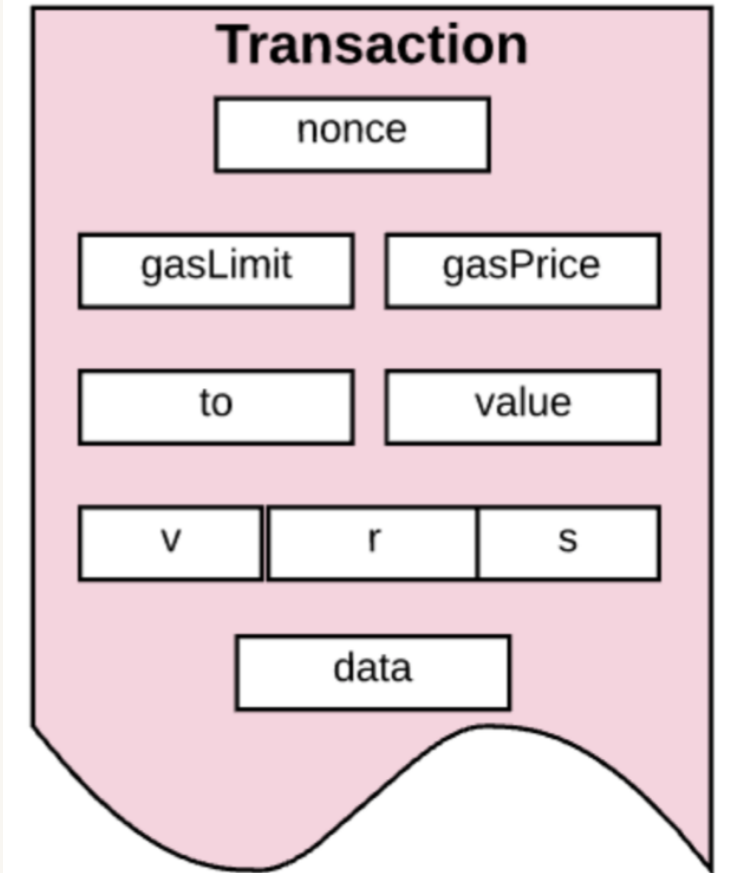
Ethereum external transactions: structure

- nonce: A count of the number of transactions sent by the sender (EOA) number, used to prevent message replay
- gasPrice: The price of gas (in wei) the originator (sender) is willing to pay
- gasLimit: The maximum amount of gas the originator is willing to buy for this transaction



Ethereum transactions

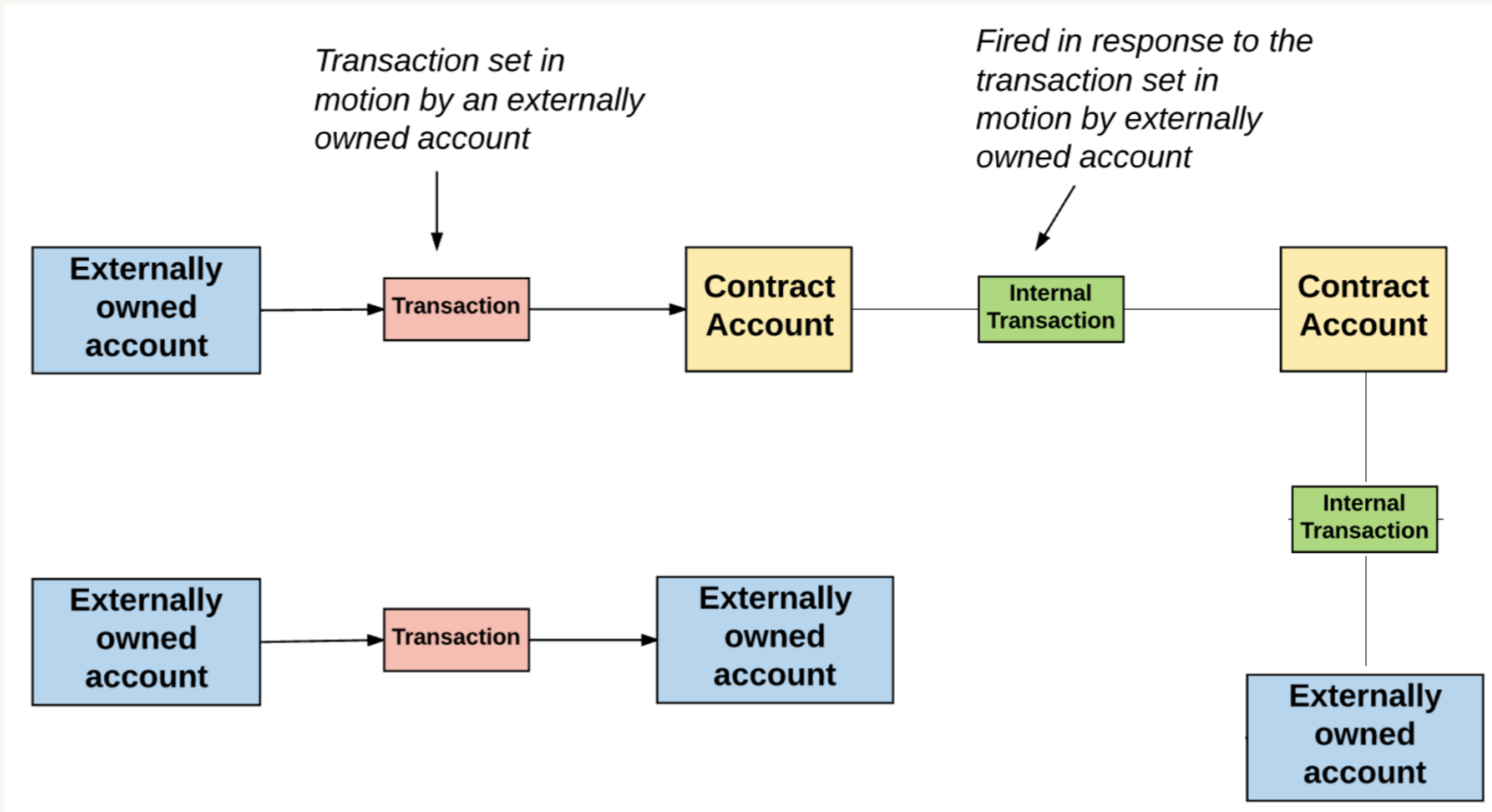
- to: Recipient's/contract address
- value: Amount of Wei transferred from the sender to the receiver
- v,r,s: The three components of an ECDSA digital signature of the originating EOA
- data: optional field to include code to create a contract or data for calling a function of an existing contract



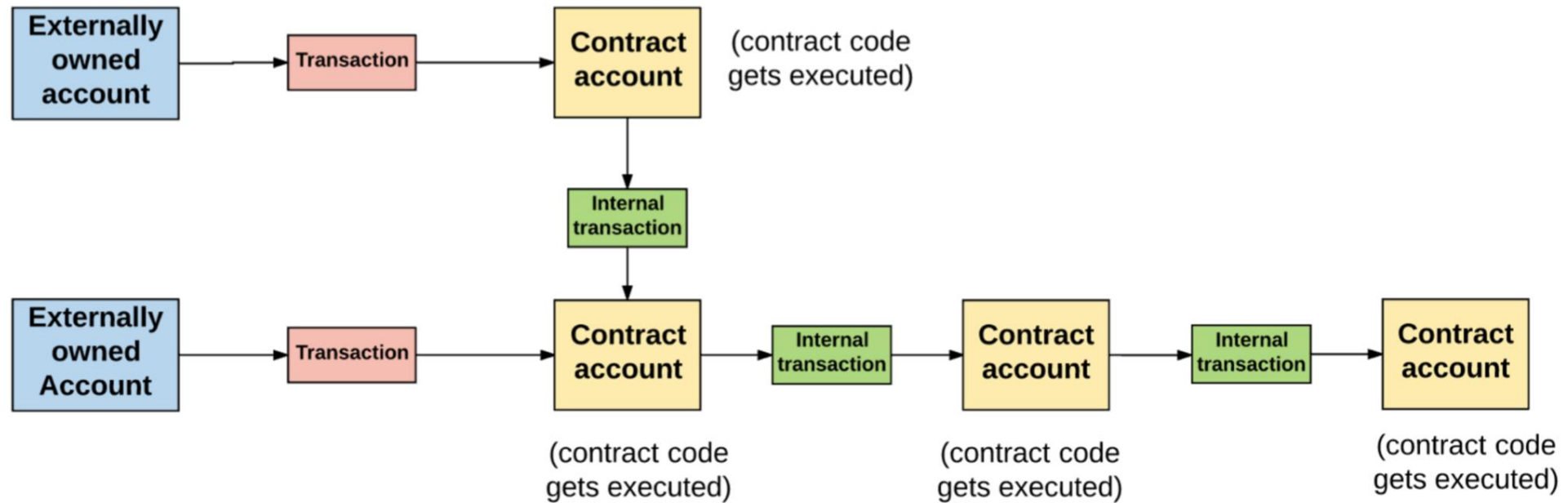
Ethereum transactions types

- Internal transactions
 - Contracts A can call a function in Contract B
 - These are not serialised and not put on the blockchain!
 - When these transactions transfer value (Eth) to another account, the account update is stored in the blockchain

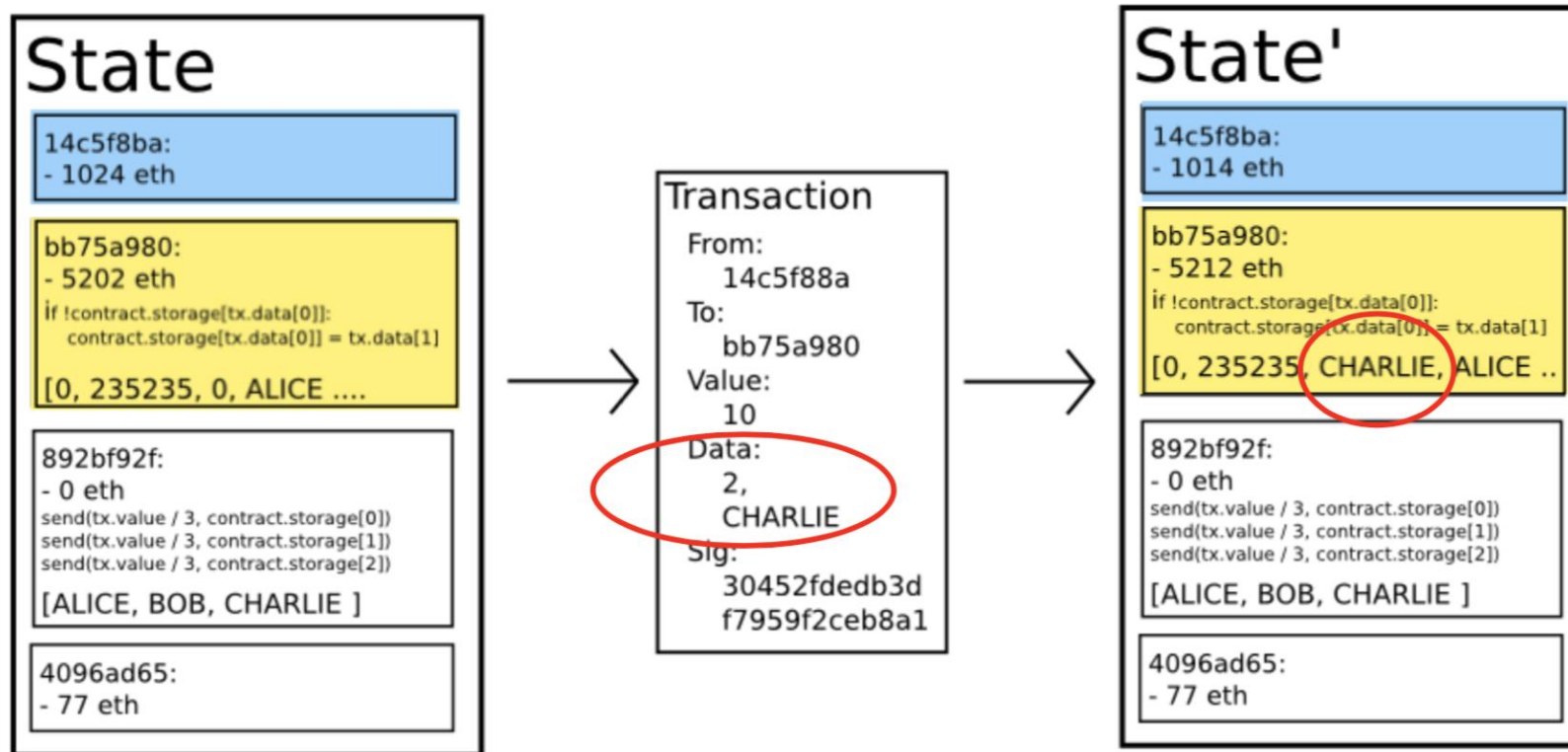
Ethereum transactions types



Ethereum transactions types



Ethereum transaction execution



Question?

