5 min break

Discussion

1. Discuss the benefits and drawbacks of smart contracts (technical limitations?)
2. Design a lending smart contract (what functions would you have, how would you store data?)
3. How does Bitcoin implement the distributed ledger?

Blockchain at Michigan

W4: Ethereum

2022

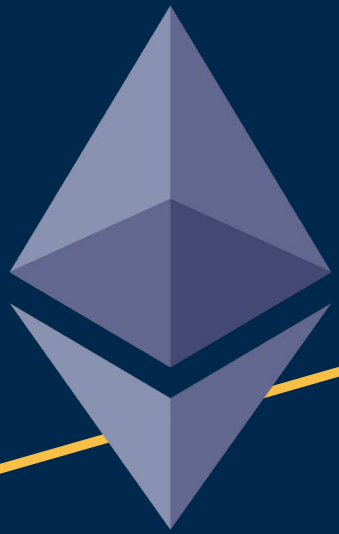
Today's Goals

To be able to understand:

1. What Ethereum is and how its different.
2. Exactly how Ethereum works from a high level.
3. The applications of the Ethereum platform.

To be able to recognize and understand what you read online about Ethereum.

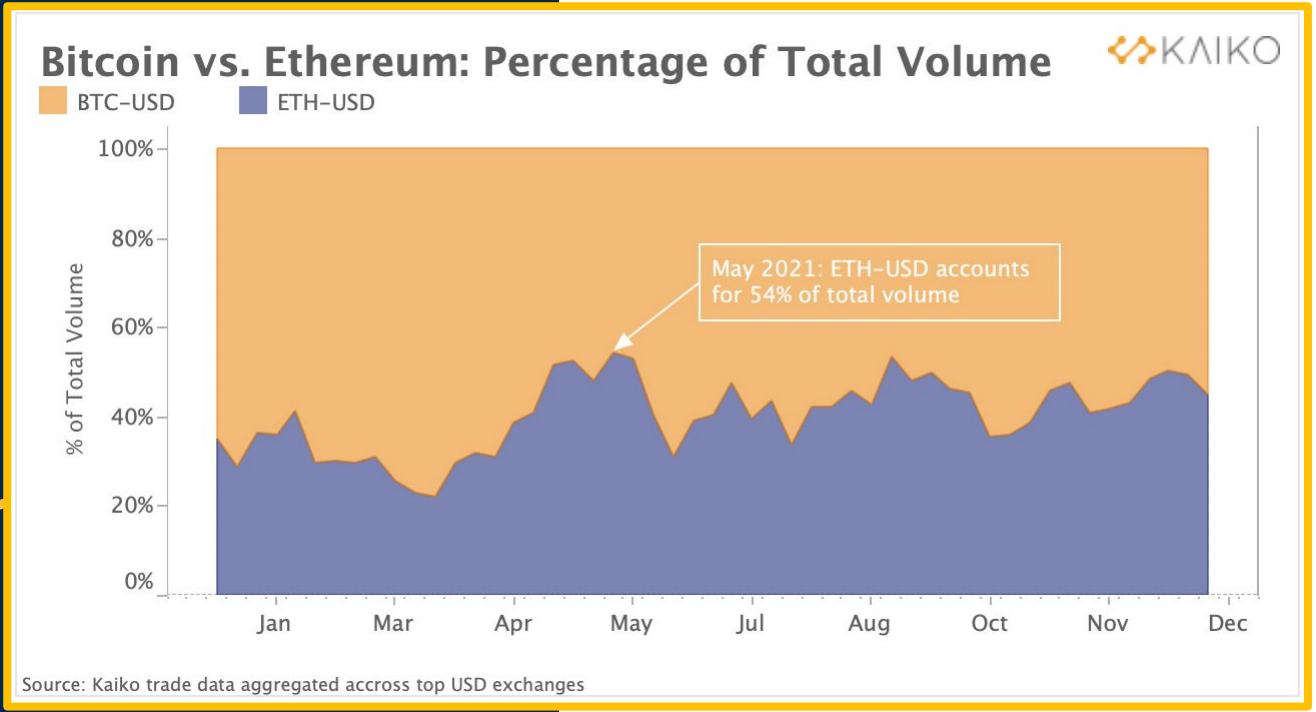
What is Ethereum?



"Ethereum is a technology that lets you send cryptocurrency to anyone for a small fee. It also **powers applications that everyone can use and no one can take down**."

– Ethereum.org

Why are we talking about ETH?



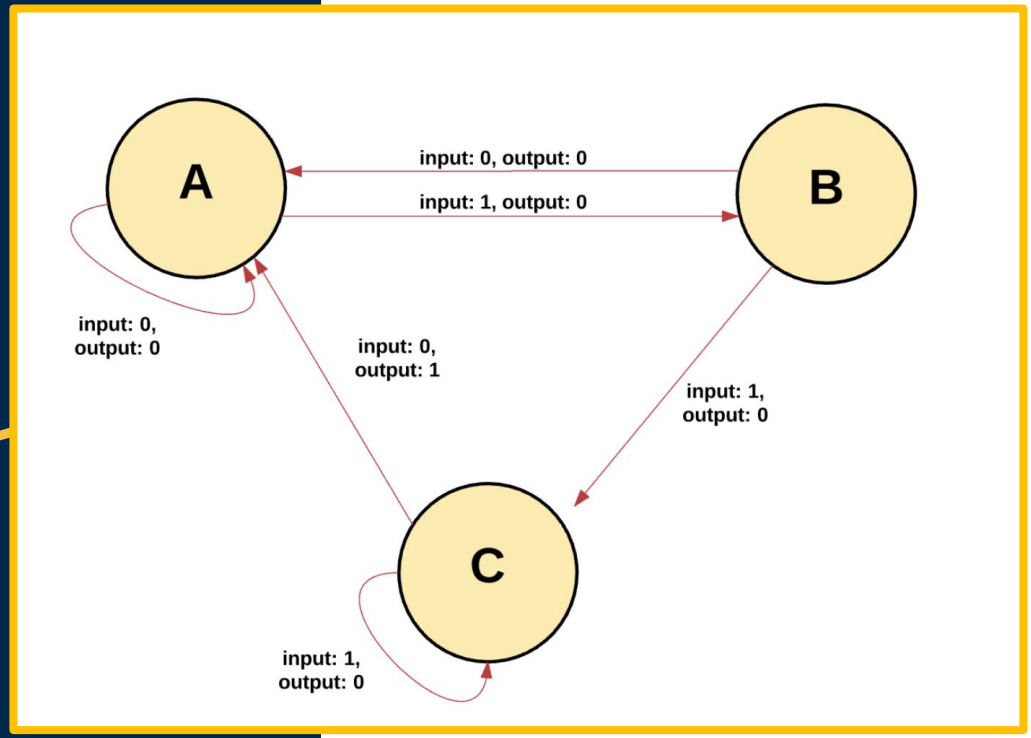
History of Ethereum

- **2013**
  - Vitalik Buterin releases the Ethereum White Paper
- **2014**
  - Dr. Gavin Wood releases the Ethereum Yellow Paper (Technical Doc)
  - First Ether sale is conducted
- **2016**
  - The DAO Fork leads to the creation of Ethereum Classic
- **2017**
  - Changes made to the network to secure against Denial of Service Attacks
- **2020**
  - Staking, Sharding, and other efficiencies are introduced via the Beacon Chain

Now, let's get to business

From Ledger to State Machine

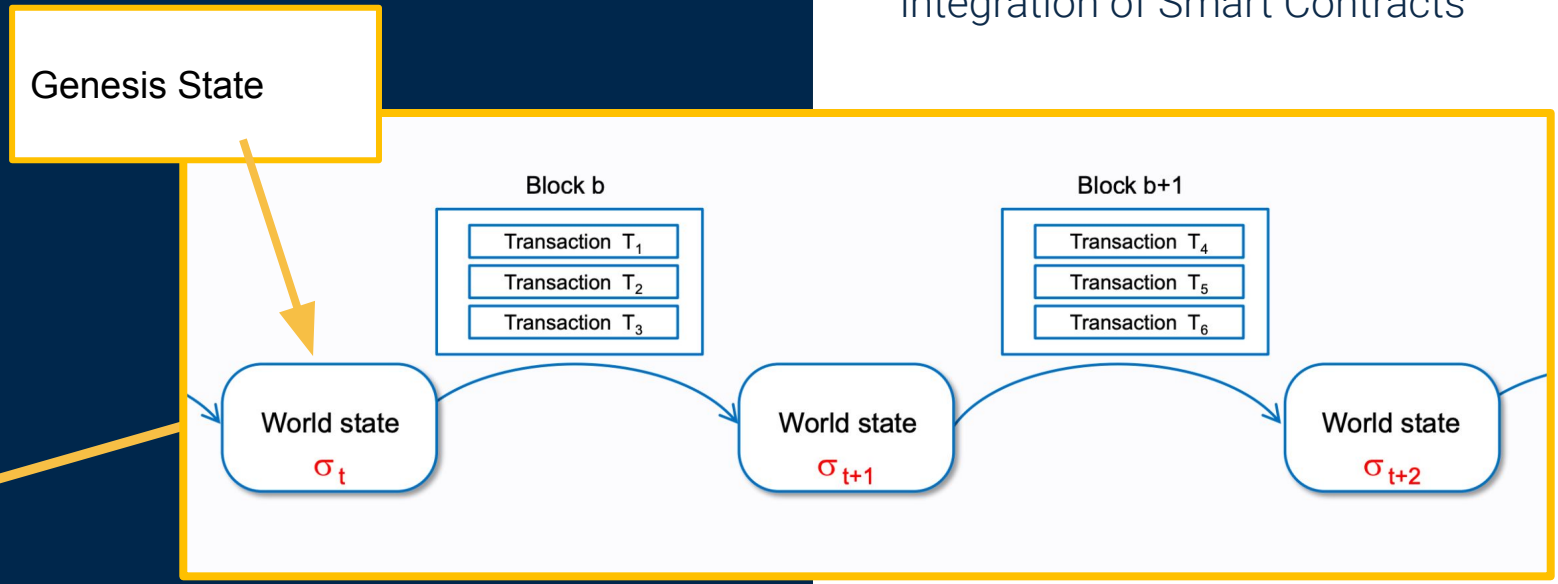
A collection of “states” that are transitioned between each other based on inputs.



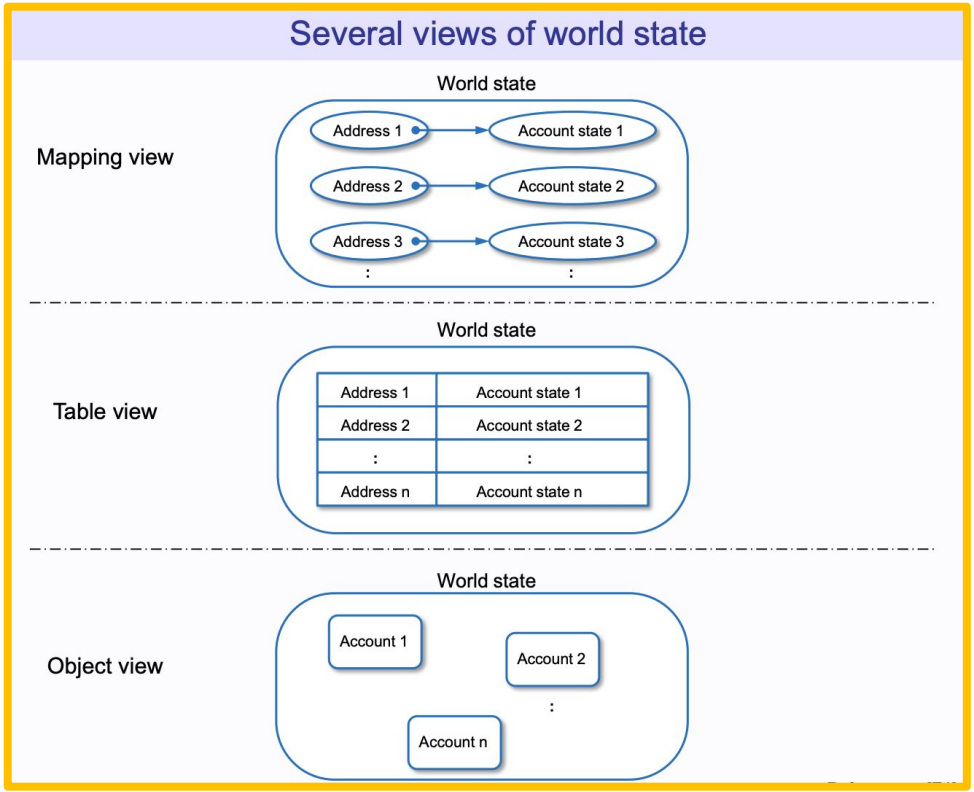
From Ledger to State Machine

The Ethereum “Blockchain” can be considered both as a blockchain and as a state machine.

This is helpful when considering Ethereum’s integration of Smart Contracts

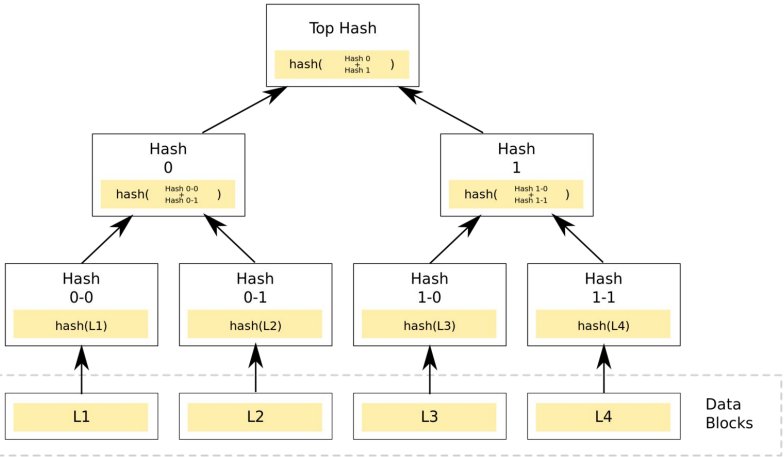


From Ledger to State Machine



Questions?

Merkle Trees



(reference)

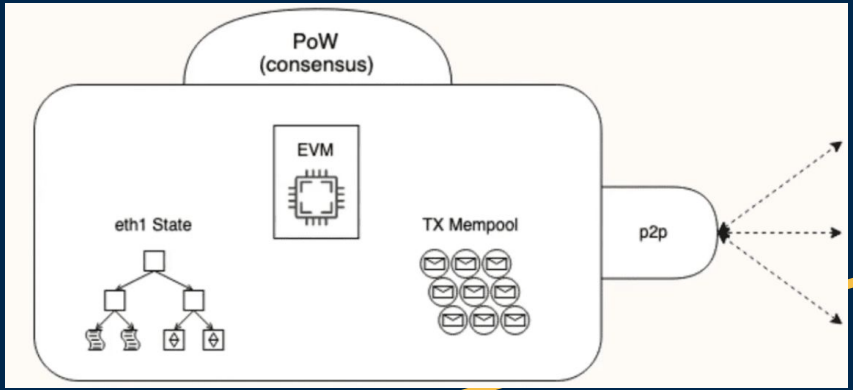
Merkle Trees

A Certified Digital Signature, Ralph Merkle (1989)

Tree: Data structure made of nodes and edges

Merkle Tree: A type of tree that uses hashes to "summarize" large amounts of data efficiently

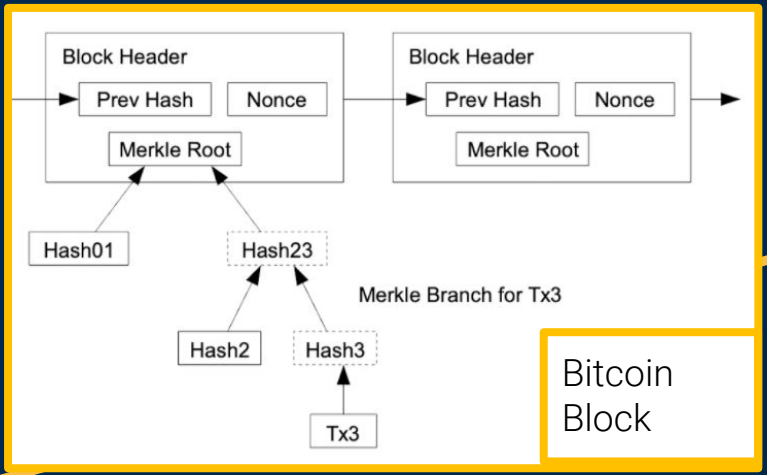
Nodes



Nodes run Ethereum Client software

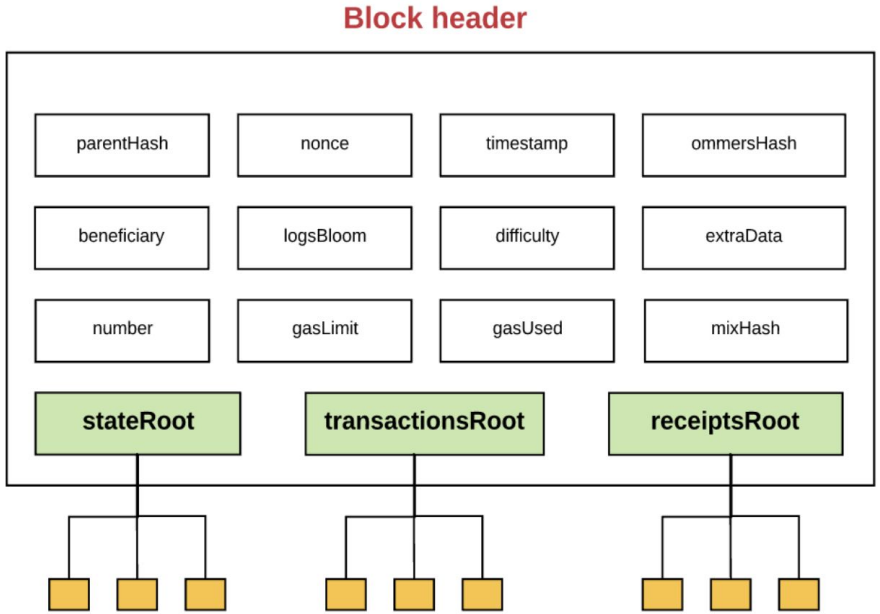
- Full Nodes
  - Stores full blockchain data.
  - Can validate all blocks.
  - All states can be derived from it.
- Light Nodes
  - Stores header chain.
  - Can verify validity by comparing against a block's state root.
- Archive Nodes
  - Stores everything that full nodes store plus historical state information.
  - Data stored can be terabytes in size.

Blocks



Bitcoin Block

Ethereum's Block

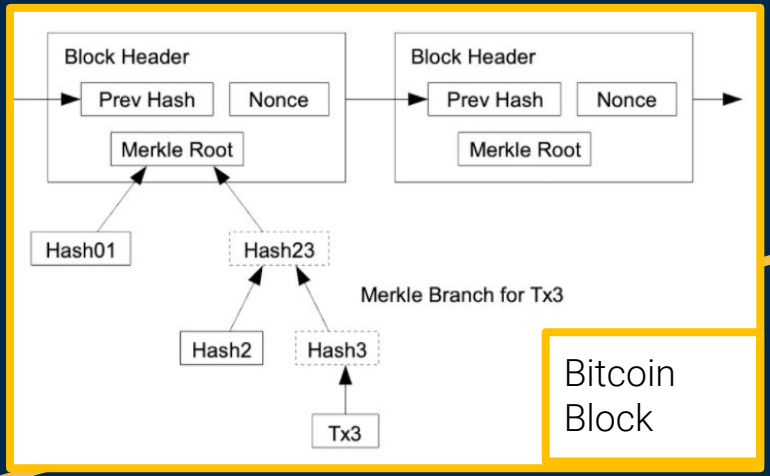


(reference)

Target size of 15M gas units

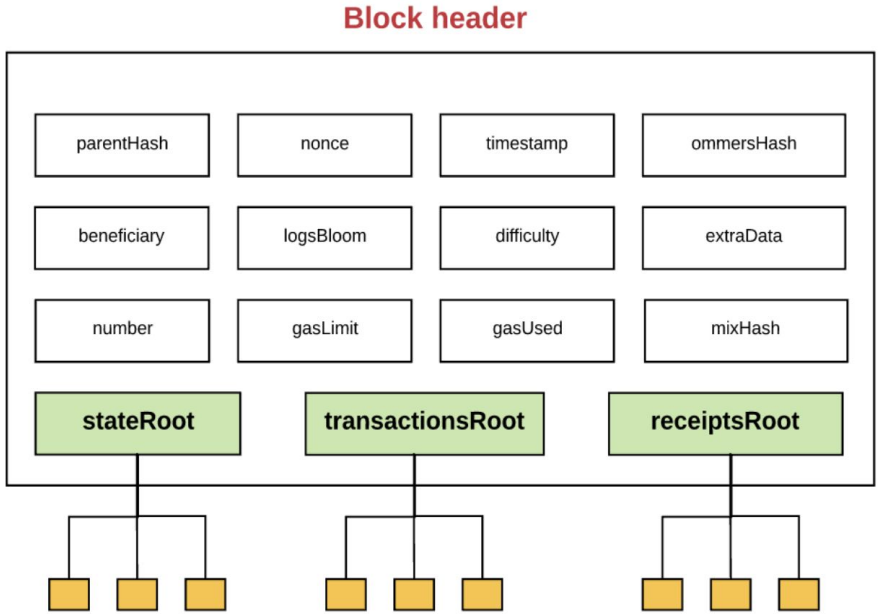
Blocks

Way more Complex



Bitcoin Block

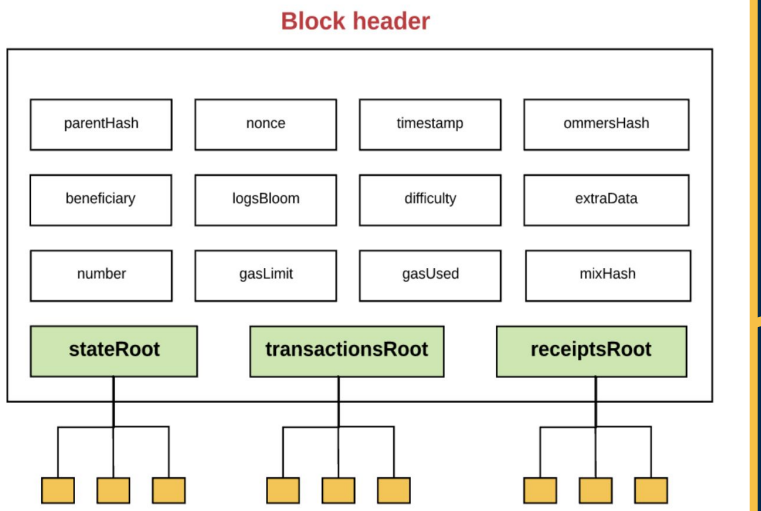
Ethereum's Block



(reference)

Blocks

Ethereum's Block



(reference)

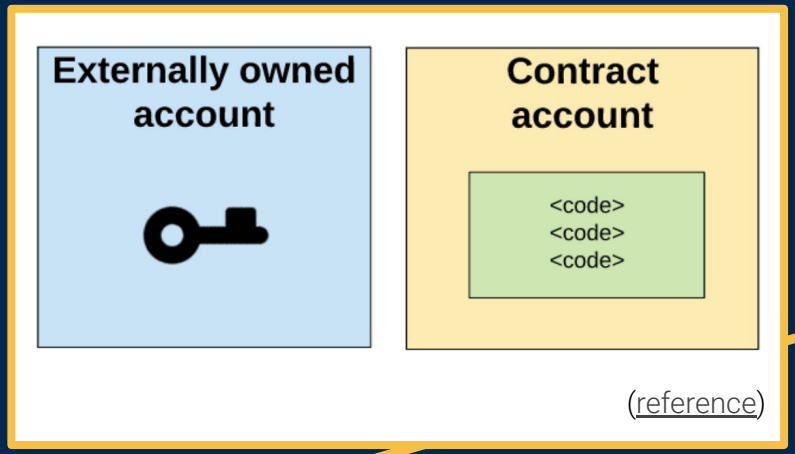
Logs: Information of specific **events** that were carried out regarding different transactions. (**events** can be emitted by Smart Contracts)

Transaction Receipts: Contains the logs specific to different transactions within the block. (Gas used, block #, block hash, etc)

Block Difficulty: Block difficulty changes over time depending on a variety of factors in order to maintain consistent block production time. This directly affects the **nonce** value.

Questions?

Accounts



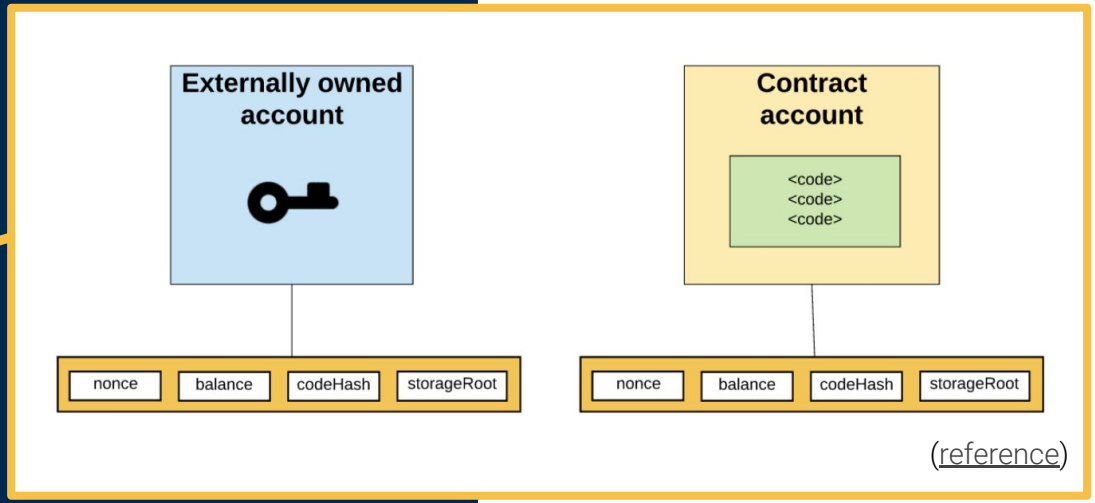
Two Types of Accounts

- Externally Owned Accounts
  - Can send messages to **both** Externally Owned Accounts and Contract Accounts
  - Controlled by their **private key** with **no code** associated with them.

- Contract Accounts
  - Cannot initiate messages on their own. Only sends messages in response to incoming messages/transactions.
  - Controlled by their **contract code** and have **code** associated with them.

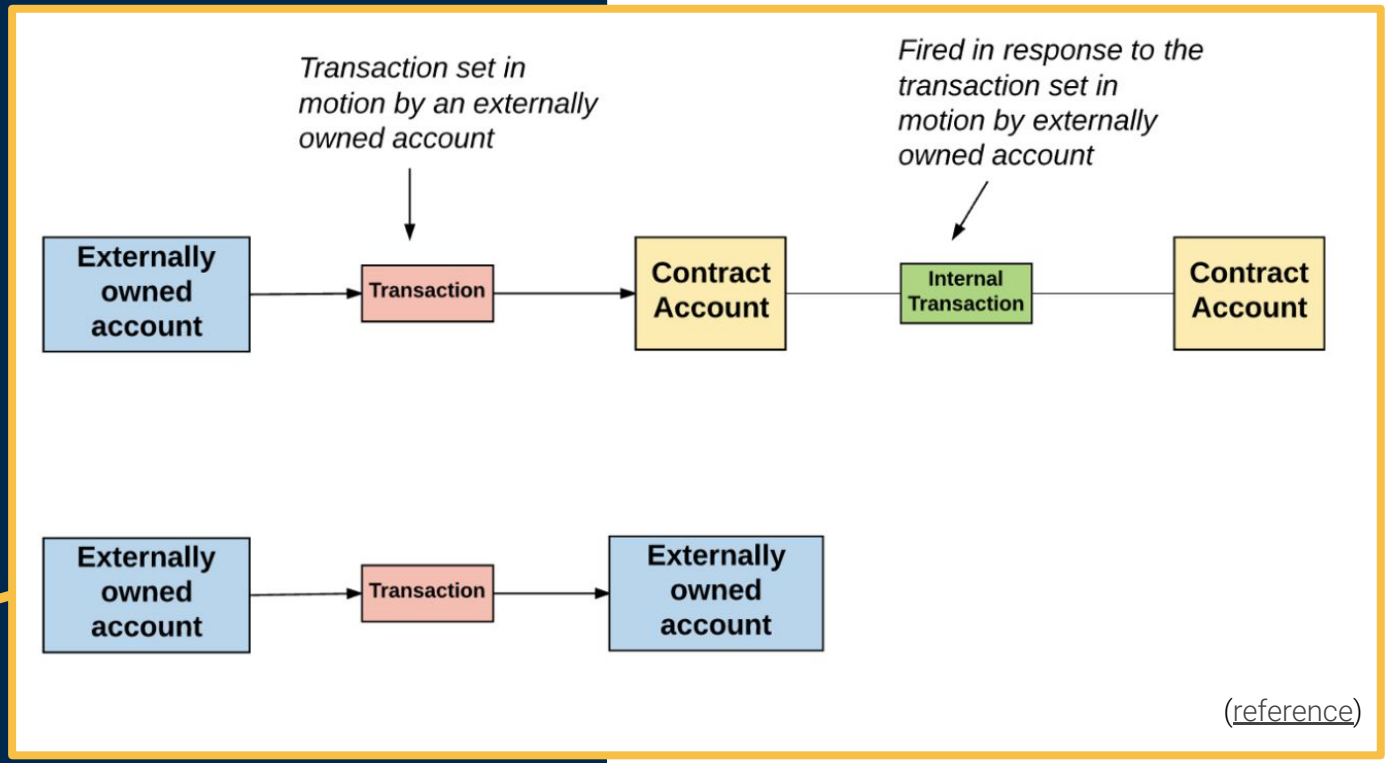
Accounts

- **nonce:** # of transactions on this acc
- **balance:** The number of Wei owned by this address. There are 1e+18 Wei per Ether.
- **storageRoot:** Merkle tree of account contents
- **codeHash:** The hash of the EVM code of this account.



(reference)

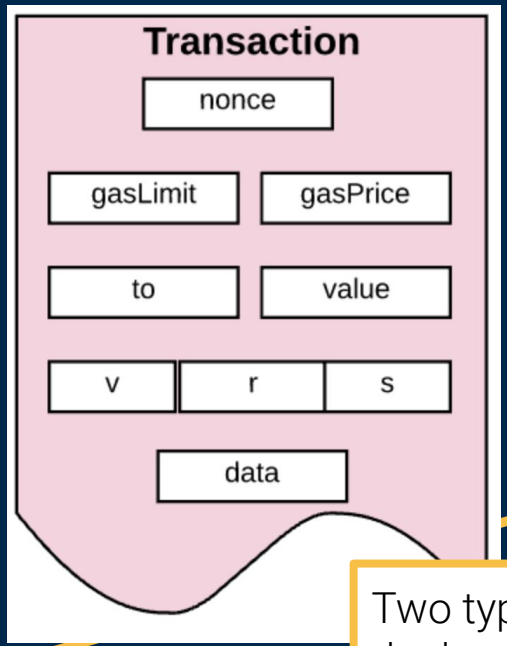
Accounts



(reference)

Questions? + 5 min break

Transactions



Two types: Contract deployment, and regular transactions.
(reference)

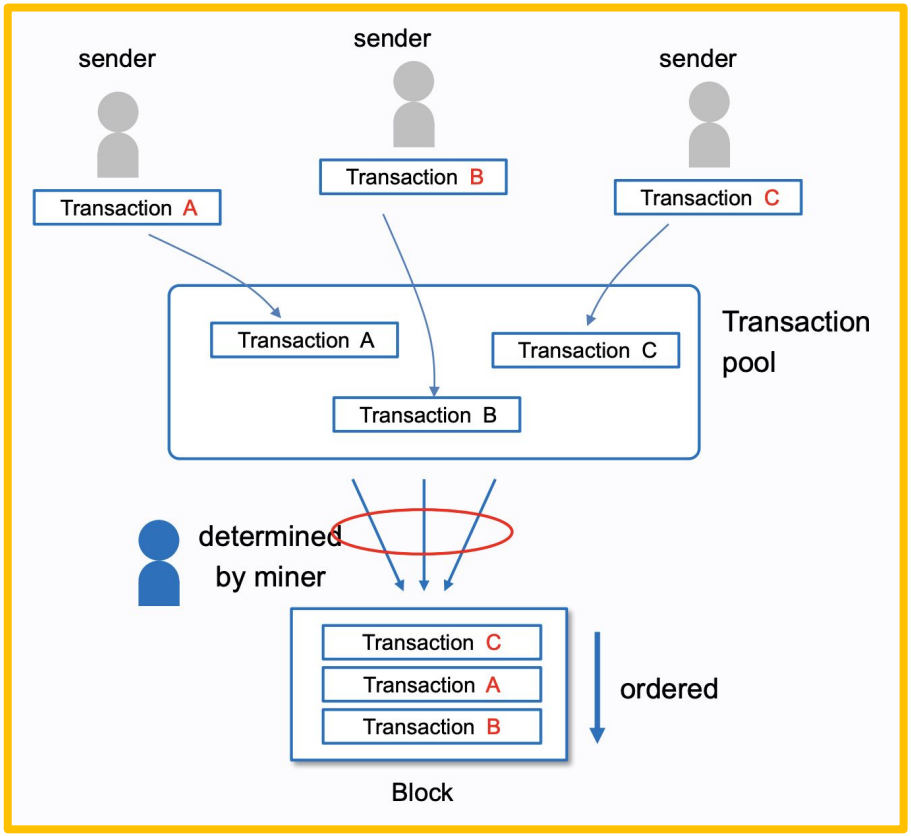
Once you send a transaction, cryptography generates a transaction hash.

The transaction is then broadcast to the network and included in a pool with lots of other transactions.

A miner must pick your transaction and include it in a block in order to verify the transaction and consider it "successful".

Your transaction will receive "confirmations". The number of confirmations is the number of blocks created since the block that included your transaction (**GHOST**). The higher the number, the greater the certainty that the network processed and recognized the transaction.

Transactions



Gas and Fees

1 billion Gwei to 1 Ether

There is an auction system implemented where **senders** will **tip (bid)** the **validators** in order to get them to add that transaction to the next block

Calculating the total transaction fee works as follows:
Gas units (limit) * (Base fee + Tip)

Let's say Oh Jun has to pay Ashish **1 ETH**. In the transaction, the gas limit is **21,000 units** and the base fee is **100 gwei**. Oh Jun includes a tip of **10 gwei**.

Using the formula above we can calculate this as 21,000 * (100 + 10) = 2,310,000 gwei or 0.00231 ETH.

Base fee is burned, and the tip (0.00021 ETH) goes to the validator (Oleg), plus the normal mining reward.

What about Orphaned Blocks?

Orphaned blocks are a much bigger problem in Ethereum because the mining time is approx. 15 seconds (compared to Bitcoin's 10 minutes time frame)

Orphan Blocks: A block who shares the same parent as another block but was not integrated into the "canonical" chain

Problem: These blocks are still valid, and need to be added at some point.

Ethereum calls Orphaned blocks that are later added to the chain: **Ommers**.

Ommers provide a smaller reward than other blocks but the incentive is there nonetheless.

Gas and Fees

Why are fees important?

Why is it a feature to burn the base fee and reward validators with transaction tips?

Gas and Fees

Why are fee's important?

Fees discourage attackers from overtaxing the network by associating costs for doing so.

Why is it a feature to burn the base fee and reward validators with transaction tips?

Gas and Fees

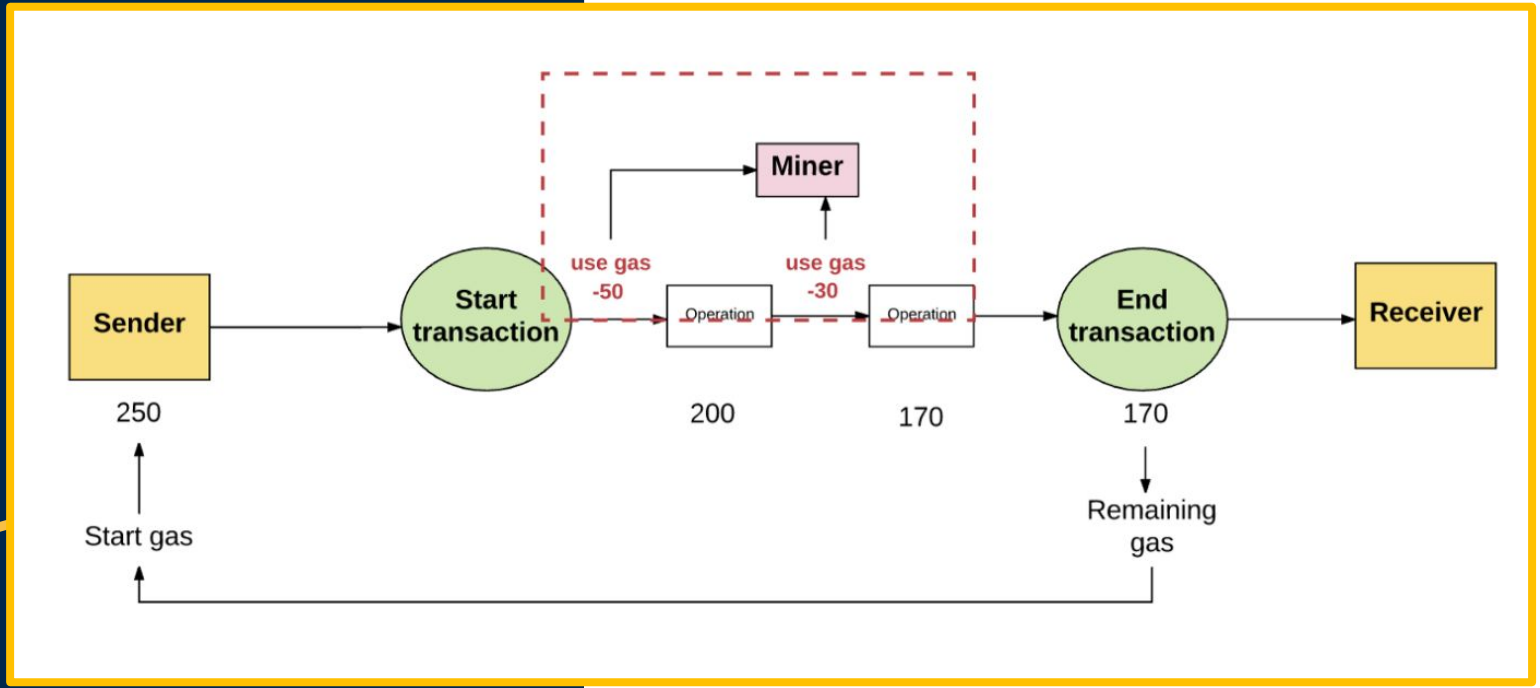
Why are fee's important?

Fees discourage attackers from overtaxing the network by associating costs for doing so.

Why is it a feature to burn the base fee and reward validators with transaction tips?

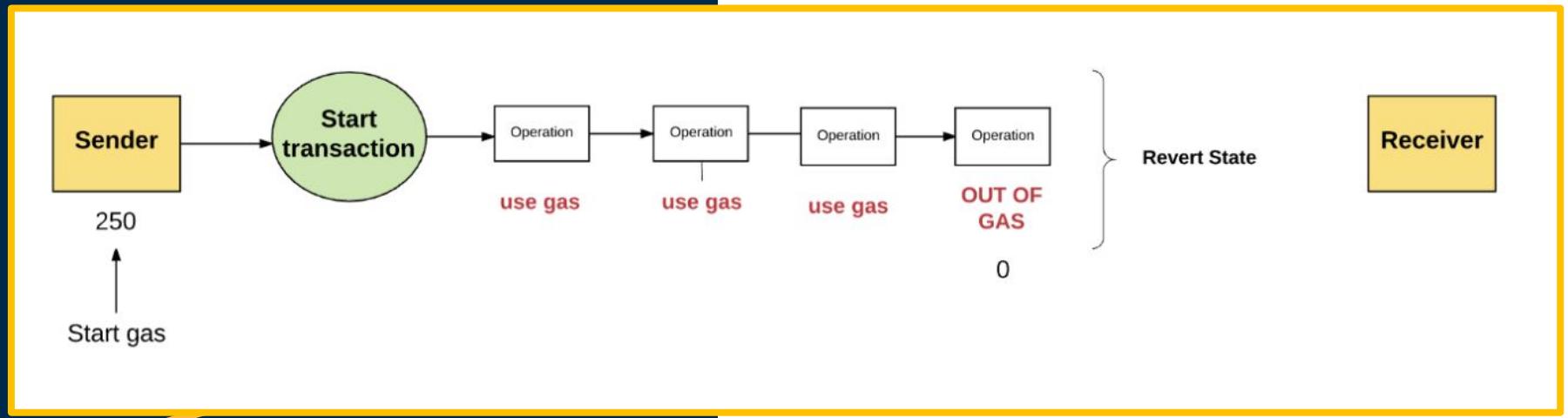
Discourages validators from minting empty blocks to be rewarded with the base fee. Encourages adding transactions to the blocks.

Typical Transaction Cycle



Running Out of Gas (OOG Error)

No Refunds! But why?



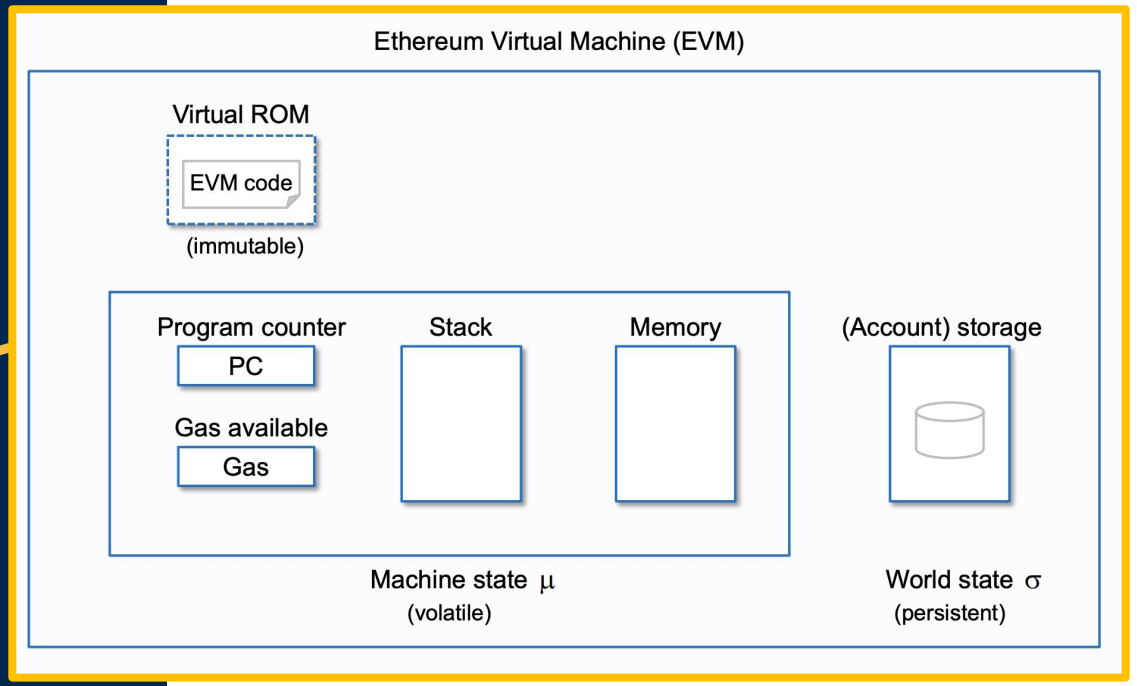
Questions?

Ethereum Virtual Environment (EVM)

EECS 370 vibes

The EVM executes as a stack machine with a depth of **1024 items**. Each item is a **256-bit word**

(reference)



Ethereum Virtual Environment (EVM)

Some EVM Opcodes **XOR**, **AND**, **ADD**, **SUB**, etc.
Blockchain specific: **ADDRESS**, **BALANCE**, **BLOCKHASH**, etc.
(reference)

Machine State

- gas available
- memory
- # of words in mem
- stack

(reference)

Contract Code is compiled into **Ethereum Bytecode** that the EVM can understand (similar to machine code).

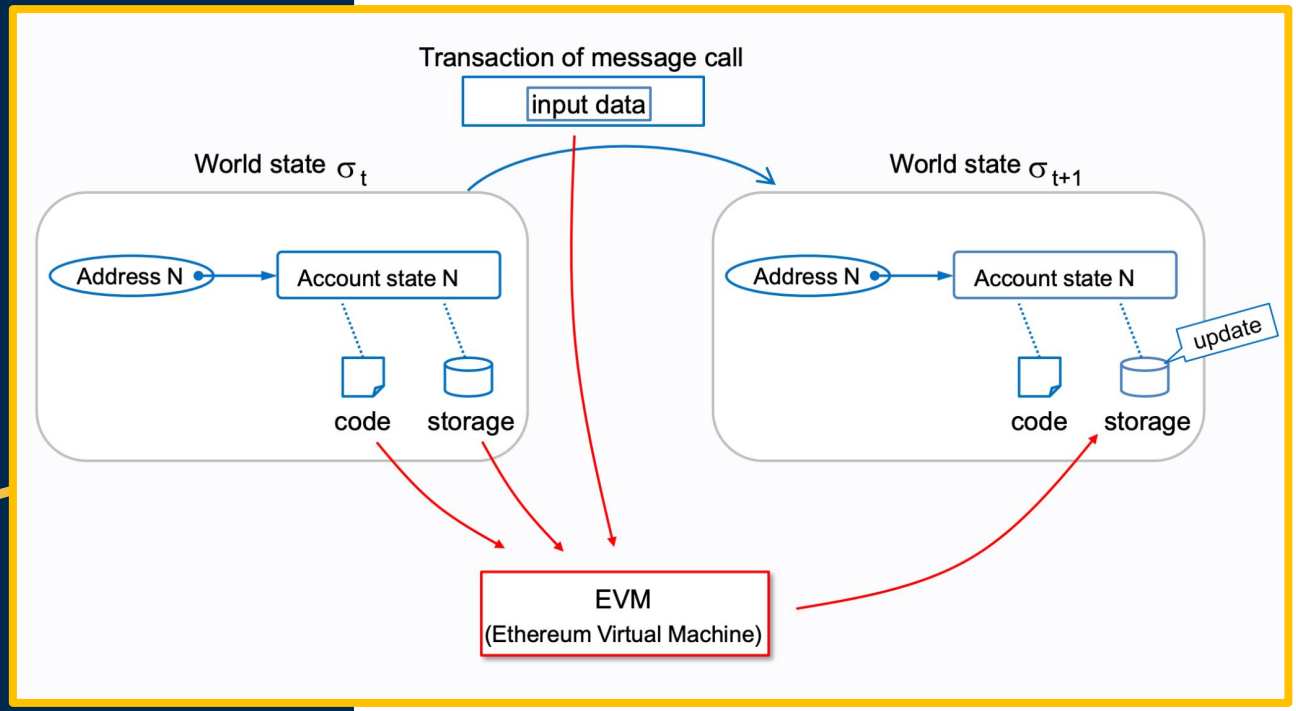
The EVM supports many custom opcodes (commands) that enable smart contract development.

The EVM executes recursively, computing the **system state** and **machine state**.

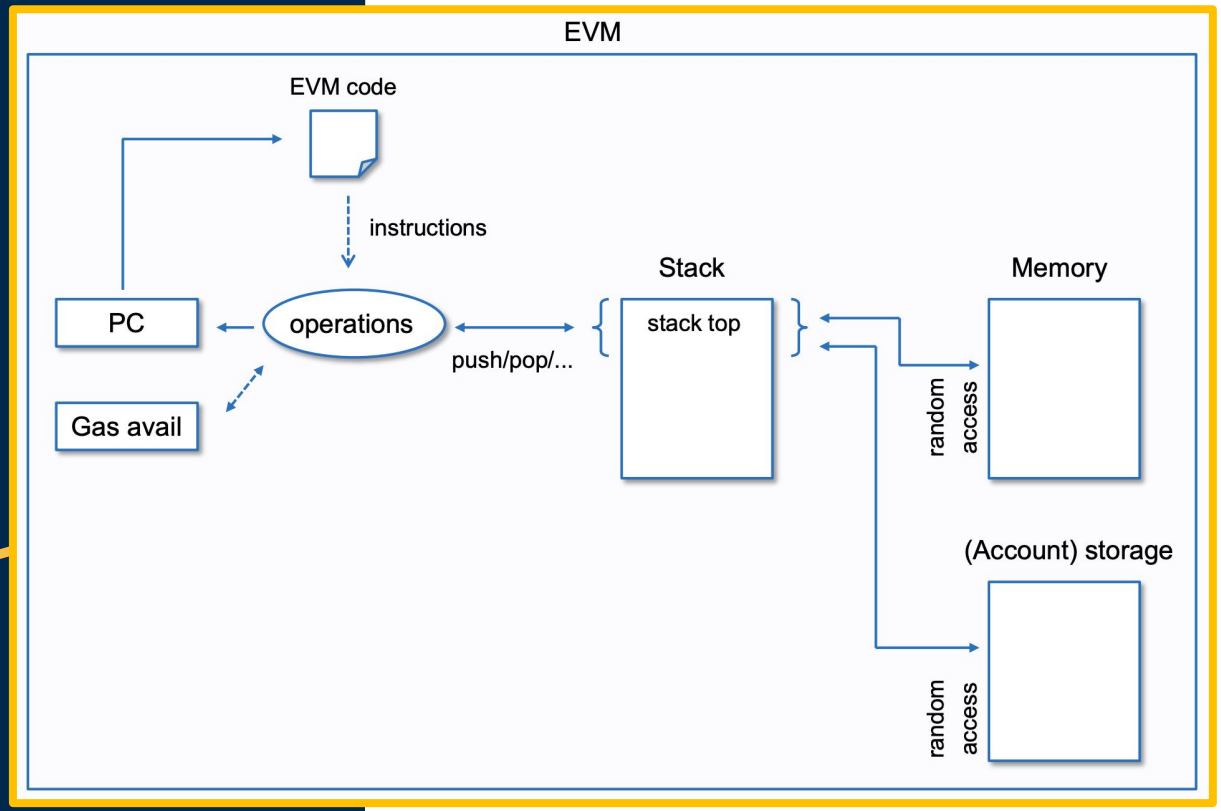
The EVM does one of the following each loop:

- machine hits an exception.
- machine ends normally.
- continues execution.

EVM for State Transition



EVM Execution Model



Current EVM Implementations

Py-EVM - Python

evmone - C++

ethereumjs-vm - JavaScript

eEVM - C++

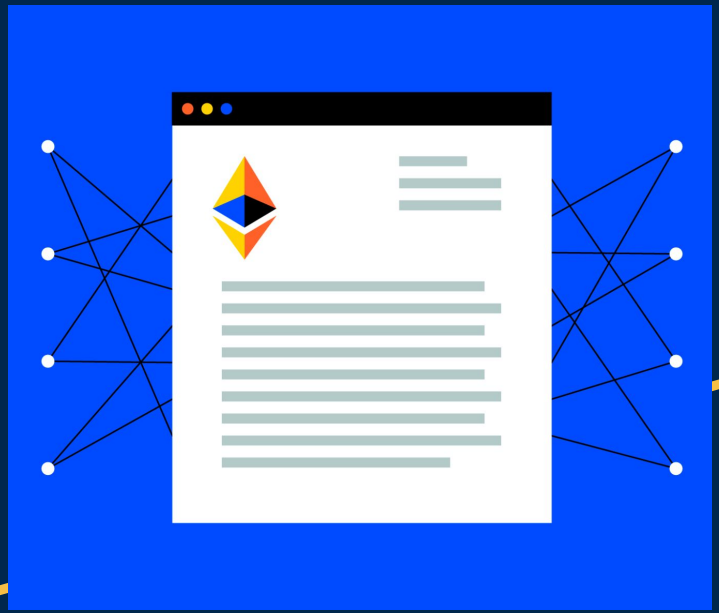
Hyperledger Burrow - Go

hevm - Haskell

Open source!!

Questions?

Smart Contracts



Typically written in Solidity or Vyper

Permissionless

- Anyone can create a smart contract.

Composability

- Contracts are Basically APIs
- Meaning, contracts can call other contracts or even deploy other contracts.

Limitations

- Smart contracts cannot access outside information (no HTTP requests allowed). It's a security thing.

dApps

OpenSea

UNISWAP

Decentralized - operate on open public decentralized platforms

Deterministic - dapps perform the same function irrespective of the environment in which they get executed.

Turing complete - dapps can perform any action given the required resources.

Isolated - dApp functionality is isolated to its own virtual environment, so bugs in a dApp don't affect the rest of the network.

Questions?

The Future of Ethereum: The Merge

The eventual planned merge between the Beacon Chain (which contains a lot of efficient new features) and the mainnet which is in widespread use.

These changes will drive transactions fees down, and increase throughput on the network.

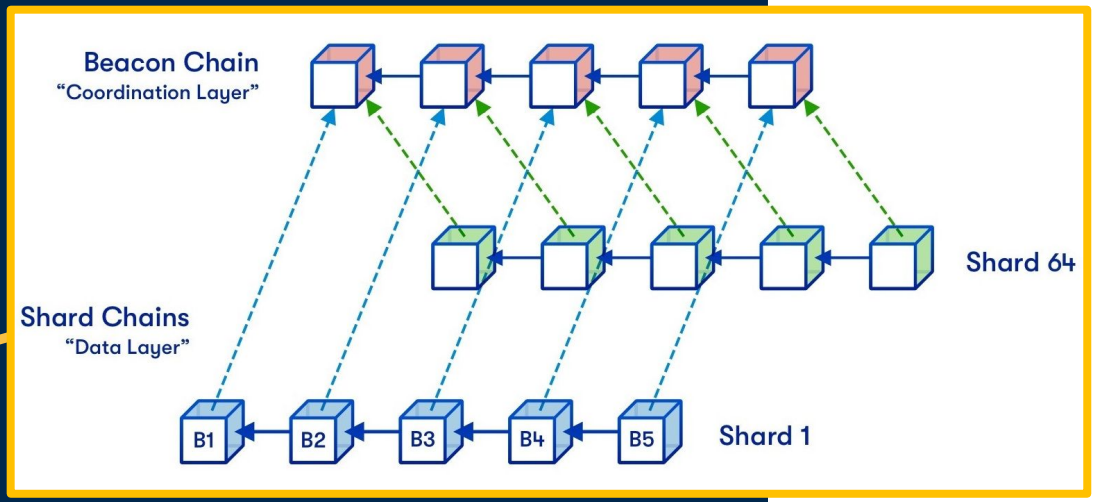
Staking (PoS revisit)

What are the benefits of proof of stake?

Sharding

Essentially, splitting the chain up into multiple smaller chains that each account for a separate portion of the network.

What benefit would arise from this?



Questions?

see u next week

More text

Readings:

1. The Ethereum White Paper (Link), by Vitalik Buterin
2. Check out the Ethereum Website

*There will be a discussion next session