CMSC 398F
Week #7
PoS, ETH, and Solidity Intro
...

Announcements

- Project 1 due date extended to Sunday
  - Questions regarding project?
- Midterm next Friday (on what we have covered so far)
  - Will include information from today
  - Multiple Choice, Short Answer, and Fill-in-the-Blank
- Slides will be updated after class today

From Last Week

- Wallets help manage their keys
  - They can send and receive to anyone else on the same network.
- There are different types of wallets
  - Hot wallets
  - Cold wallets
- Recovery Phrases
- Altcoins
  - Eth and more

Proof-Of-Stake

- Proof-of-stake underlies certain consensus mechanisms used by blockchains to achieve distributed consensus. In proof-of-work, miners prove they have capital at risk by expending energy
- Staked ETH then acts as collateral that can be destroyed if the validator behaves dishonestly or lazily
- The validator is then responsible for checking that new blocks propagated over the network are valid and occasionally creating and propagating new blocks themselves

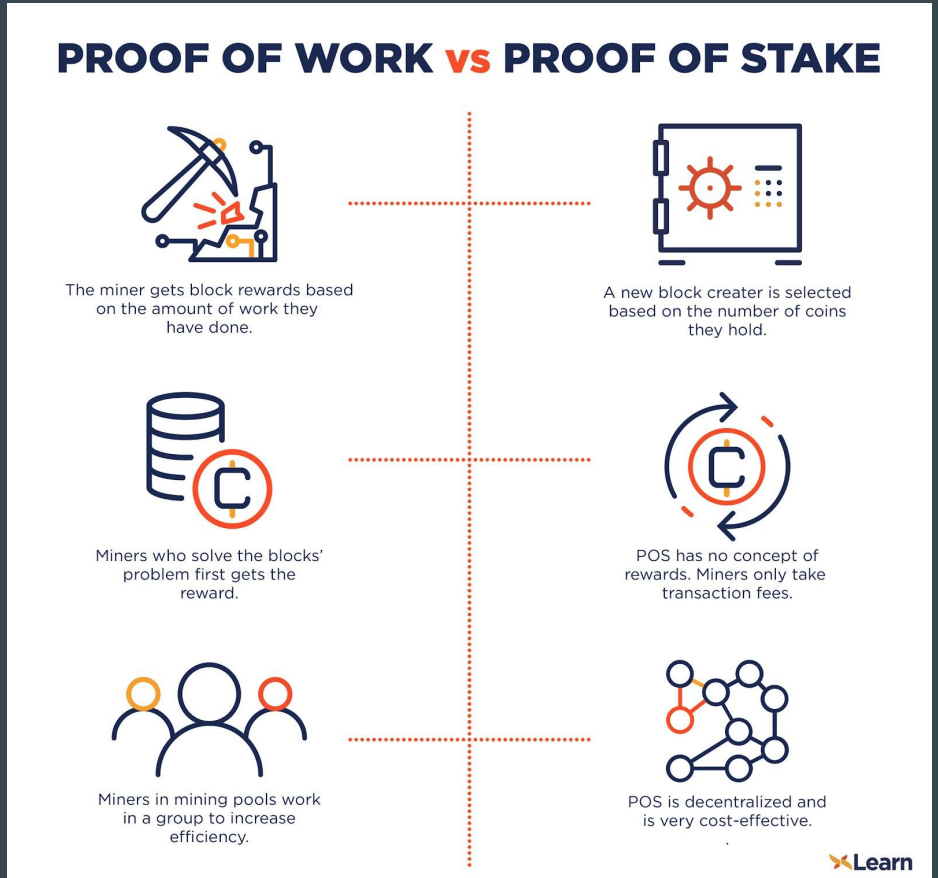
Why POS?

- Ethereum switched on its proof-of-stake mechanism in 2022 because it is more secure, less energy-intensive, and better for implementing new scaling solutions
- Staking makes it easier for individuals to participate in securing the network, promoting decentralization. validator node can be run on a normal laptop.
- Staking pools allow users to stake without having 32 ETH.
- The threat of a 51% attack still exists on proof-of-stake as it does on proof-of-work, but it's even riskier for the attackers. An attacker would need 51% of the staked ETH.
- https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/attack-and-defense/

Other POS networks

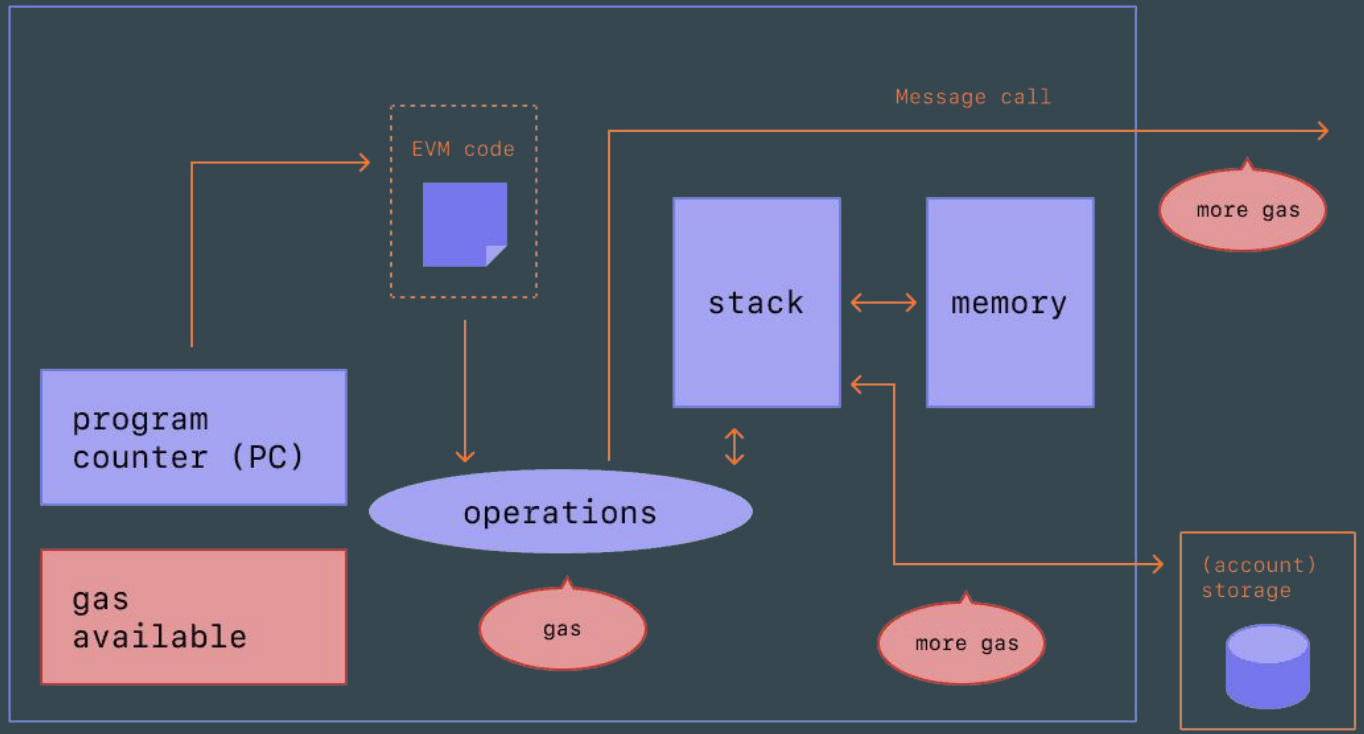
- Solana: Solana runs on an innovative hybrid consensus model comprising proof-of-stake (PoS) and proof-of-history (PoH), a proof for verifying order and passage of time between events. Due to this unique protocol design, Solana can handle up to 3,000 transactions per second.
- Cardano:
- Avalanche:Avalanche can handle a whopping 4,500 transactions per second, thereby positioning itself as a serious rival to Ethereum.Avalanche is built around a system of three interoperable blockchains: Exchange Chain (X-Chain), Platform Chain (P-Chain) and Contract Chain (C-Chain).

PoS vs PoW


PROOF OF WORK vs PROOF OF STAKE
The miner gets block rewards based on the amount of work they have done.
A new block creater is selected based on the number of coins they hold.
Miners who solve the blocks' problem first gets the reward.
POS has no concept of rewards. Miners only take transaction fees.
Miners in mining pools work in a group to increase efficiency.
POS is decentralized and is very cost-effective.
Learn

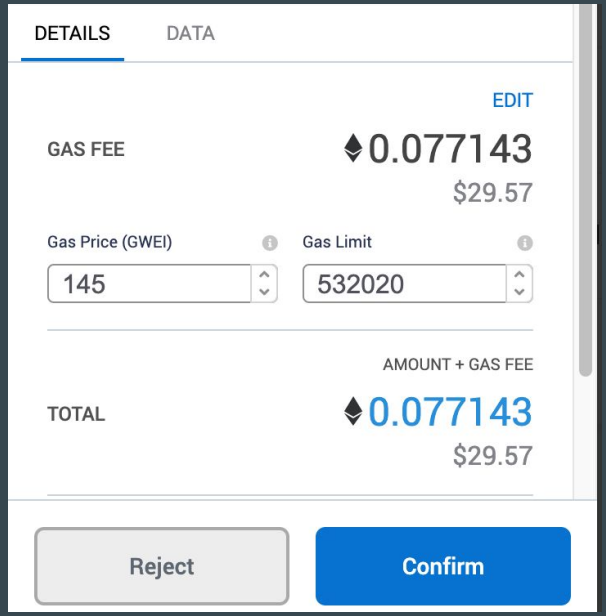
Ethereum Virtual Machine (EVM)

- Instead of a distributed ledger, Ethereum is a distributed state machine
- Ethereum's state is a large data structure which holds not only all accounts and balances, but a machine state, which can change from block to block according to a pre-defined set of rules, and which can execute arbitrary machine code.
- The EVM behaves as a mathematical function would: Given an input, it produces a deterministic output.


Message call
EVM code
more gas
stack
memory
program
counter (PC)
operations
gas
available
gas
more gas
(account)
storage

Gas and Fees

- Gas refers to the unit that measures the amount of computational effort required to execute specific operations on the Ethereum network.
- When you pay gas to submit a transaction, you are paying for the computational energy needed to power the validation of that transaction on Ethereum.



Introduction to Smart Contracts

- A "smart contract" is simply a program that runs on the Ethereum blockchain.
- Smart contracts digitize agreements by turning the terms of an agreement into computer code that automatically executes when the contract terms are met
- The code itself is replicated across multiple nodes of a blockchain and, therefore, benefits from the security, permanence and immutability that a blockchain offers
- actual tasks that smart contracts are performing are fairly rudimentary
    - automatically moving an amount of cryptocurrency from one party's wallet to another

Why are Smart Contracts Important

- Speed, efficiency, and accuracy
  - Once a condition is met, the contract is executed swiftly. No paperwork to process and no time spent reconciling errors that often result from manually filling in documents.
- Trust and transparency
  - There is no third party involved, and contracts on the blockchain are immutable.

Applications of Smart Contracts in real-world

- The applications of smart contracts are endless
- Banks
  - Issue loans or automatic payments
- Insurance companies
  - Use it to process claims
- Postal companies
  - Process payment upon delivery
- Increasing trust in retailer supplier relationships
  - The Home Depot use smart contracts to resolve disputes with vendors
- IBM Blockchain
  - Making global trade faster and more efficient

How to develop Smart Contracts - Solidity

- Solidity is a type of Object-oriented programming language that is developed specifically for smart contracts on the Ethereum blockchain
  - Used to implement smart contract features on various blockchain platforms
- Influenced by C++, Javascript, and Python and is statically-typed
- Solidity uses the EVM to function
  - EVM offers a runtime environment for smart contracts to execute
  - Solidity is compiled to bytecode that is executable on the EVM

Different ways of setting up a Solidity Compiler

- Remix
  - an application that provides plugins and a development environment for smart contracts. Users can use this application online without installing any software for the environment
- Node.js/npm
  - you will npm install a solidity compiler called solc-js. However, solc-js offer limited functionalities for accessing the compiler
- Docker Image
  - Docker image offers simple steps in setting up the environment. Docker images offer a template to build a container for running the Solidity compiler
- Binary Packages
  - Binary packages are archive files that will have all directories and files for installing the Solidity compiler on your device

Solidity Syntax

- Pragma
  - In Solidity, a pragma language will specify how the compiler will process any type of input
  - Typically, the first line of code in Solidity based smart contracts contains the pragma to specify the Solidity version
    - pragma solidity ^0.8.7;
- Contract
  - A Solidity contract is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum Blockchain
- File Importing
  - Solidity offers similar support for file import systems like JavaScript.
    - import "filename";

Data types in Solidity

- 8 bits to 256 bits Unsigned integer
- 8 bits to 256 bits Signed integer
- Boolean
- String

```
1   pragma solidity ^0.8.7;  //Pragma is necessary
2
3   contract MyContract {
4       //Different types of data type values
5       string str_value = "MyValue";
6       bool bool_value = true;
7       int256 int_value256 = -1000;
8       int8 int_value8 = -1;
9       uint256 uint_value256 = 1000;
10      uint8 uint_value8 = 1;
11  }
12
```

Variables

- State Variable
  - Users can locate the state variables within the contract storage where values are permanently stored
- Local Variable
  - Users can find the value of any local variables within the defining function. This value is not permanently stored
- Global Variables
  - Global variables help in fetching data or information from the blockchain platform and any associated transaction processes
  - Common Global variables that might be used:
    - msg.sender
    - msg.value
    - this
    - block.timestamp
    - block.difficulty

Visibility Modifiers

- Visibility modifiers define the visibility of state variables or functions
- There are 4 modifiers that Solidity has:
  - External
  - Public
  - Internal
  - Private
- We will talk about these in more depth next week!

https://cryptozombies.io/

- This is a free bootcamp for you guys to learn Solidity
- We will be teaching some important things as we continue, but not everything
  - But we expect you guys to do to assigned CryptoZombies.io each week
  - The reason for this is for you guys to get your own hands-on experience with Solidity
- Quizzes each week on Solidity from the assigned CryptoZombies.io lesson
- Next Week: Finish Lesson 1: “Making the Zombie Factory”