

# Ethereum Solutions Developer

How to design and build great Ethereum solutions

# Sections

- The History of Ethereum
  - The Quest for Digital Money (1973-2008)
  - Bitcoin (2008-2009)
  - Ethereum (2015)
  - Hyperledger (2015)



# Sections

- How is Ethereum used?
  - Ether
  - ERC-20
  - ERC-721
  - Smart Contracts
  - Blockchain 2.0 == Decentralization + Permanence
- How does Ethereum work?
  - Mining vs non-mining nodes
  - PoW
  - PoS
  - Ethereum 2.0 Upgrade
  - Gas



# Sections

- What does an Ethereum application look like?
  - Application layers
  - Voting Demo
- Solution Design Considerations
  - IPFS, IoT
  - Deterministic Concurrency
  - Security Best Practices
  - Public vs Private network instances



# Sections

- Hands-On Labs
  - Lab environment setup and configuration
  - The Remix IDE
  - MetaMask and the Test Networks
  - Infura.io
  - Web3.js
  - Events
  - Function Modifiers
  - Mappings and Structs
  - Inheritance
  - Deploying to Live Networks
  - Creating Your Own ERC-20 Token
  - Creating Your Own ERC-721 Token
  - Bonus Lab: Working with Truffle Boxes



# The History of Ethereum

From Stagflation to Smart Contracts

# The Quest for Digital Money

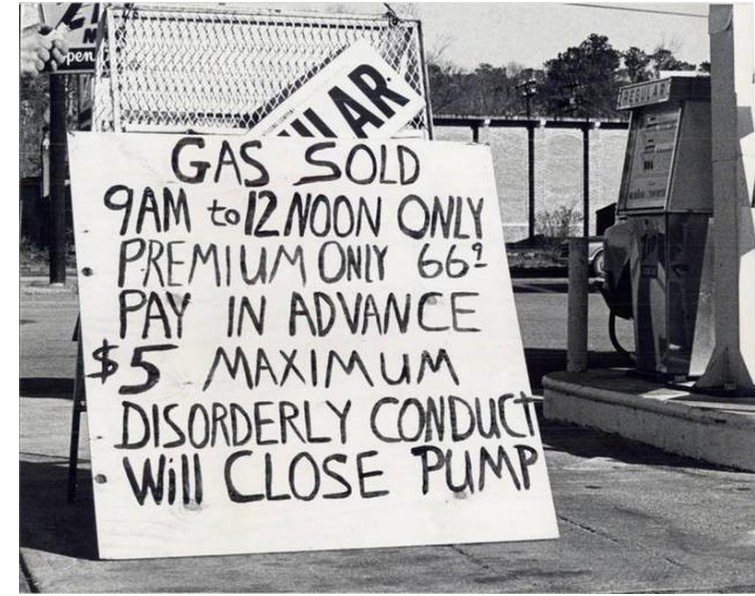
- In the 1970's the United States was experience a set of economic conditions knows as *stagflation* (high unemployment + high inflation)
  - 1971 – President Nixon imposes wage and price controls
  - 1973 – OPEC oil embargo
- Is there a solution available via fiscal policy, monetary policy, both, or something else altogether?





# The Quest for Digital Money

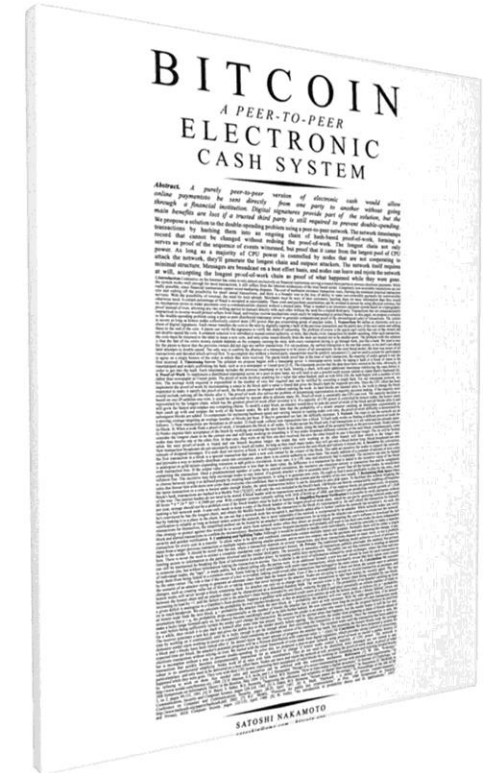
- Could 'digital money' be the answer?
  - A virtual rather than physical currency
  - Currency independent of any one central bank or nation state
- In the mid-1970's a group of hackers, political scientists, economists, computer scientists, mathematicians, and general enthusiasts began their quest.
  - How to solve the ***double spend*** problem
    - What prevents the same digital resource from being consumed more than once?





# Bitcoin

- 2008 – The Bitcoin Whitepaper is published
  - The double-spend problem has been solved!
  - <https://bitcoin.org/bitcoin.pdf>
- 2009 – Bitcoin goes live
  - The Bitcoin blockchain offers a currency, called Bitcoin
  - One single, shared ledger per network
  - Ability to record data, but not to take action on it (**Blockchain 1.0**)
  - Highly anonymous and fully-transparent
  - Primary Focus: A ledger to enable and facilitate digital payments



# Bitcoin

## Transaction View information about a bitcoin transaction

967c3d317cf2ccbee6e7f902bfc25ea0f433ec12985dd073552cf4aad0384a0e

17QrQKaWKxwauF1RsPGsMWyo4GeX1yJnRX

→

1C4iGmwpCoLL8ub57W6j3hTqSzQJUkZdrP

0.04848376 BTC

4 Confirmations

0.04848376 BTC

Summary

|                    |   |
|--------------------|---|
| Size               | 191 (bytes)                                 |
| Weight             | 764   |
| Received Time      | 2019-05-22 15:33:26                         |
| Included In Blocks | 577257 ( 2019-05-22 15:43:51 + 10 minutes ) |
| Confirmations      | 4   |
| Visualize          | <a href="#">View Tree Chart</a>             |

Inputs and Outputs

|                          |                |
|--------------------------|----------------|
| Total Input              | 0.04989428 BTC |
| Total Output             | 0.04848376 BTC |
| Fees                     | 0.00141052 BTC |
| Fee per byte             | 738.492 sat/B  |
| Fee per weight unit      | 184.623 sat/WU |
| Estimated BTC Transacted | 0.04848376 BTC |

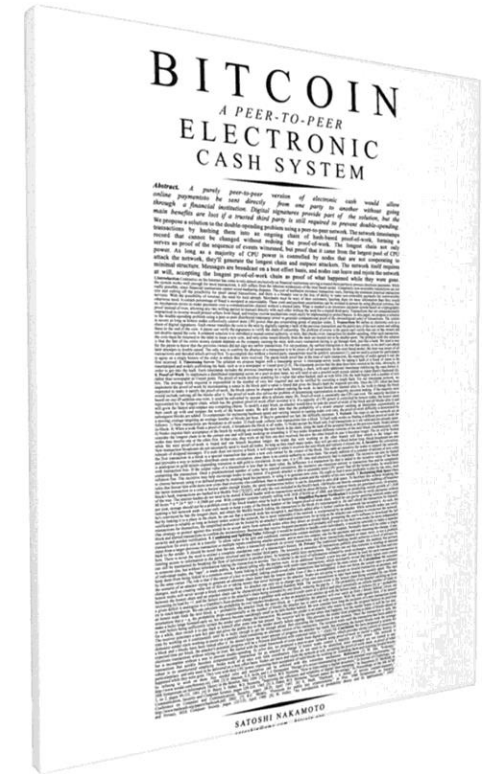
Scripts

Show scripts & coinbase



# The Reaction to Bitcoin

- *“Bitcoin 2.0” – Let’s take all the great features of Bitcoin and add to them!*
- *“Enterprise Blockchain” – Let’s take some of those blockchain ideas and concepts and use them for business solutions!*



# Ethereum


- In July 2015 the Ethereum Mainnet goes live
  - Ethereum was designed to fulfill the “Bitcoin 2.0” vision
  - Ethereum supports a currency, called *Ether*
  - The Ethereum ledger can be used to record ANY type of data, not just Ether exchanges
  - Single, shared ledger per network
  - Blockchain 2.0 – Smart Contracts and the EVM
    - Record data and programmatically respond to it
    - Blockchain as decentralized workflow
  - Highly anonymous, fully-transparent
  - Primary Focus: A decentralized application development platform




# Ethereum

**Overview**State Changes NewComments

Transaction Hash:

0x831fd26634e7dad3852797a1d3358e70619f9a0e451ce77fa3899261614d1d1d 

Status:

 Success


Block:

7810580 3 Block Confirmations


Timestamp:

🕒 1 min ago (May-22-2019 03:59:45 PM +UTC)

From:

0xc24cb5d8890d2e0bdbce4d91f73e2d243c4a890c 

To:

0x219466a5a45ada2be276e0fa3e7e2c706ca832bc 

Value:

4.90047421 Ether (\$1,257.02)

Transaction Fee:

0.000378 Ether (\$0.10)

# Ethereum

|  |  |
|--|--|
| <a href="#">Overview</a> <a href="#">Event Logs (1)</a> <a href="#">State Changes</a> <span>Now</span> |  |
| [ This is a Ropsten <b>Testnet</b> Transaction Only ]  |  |
| Transaction Hash:  | 0xbb369501ef3be2e61309b2593783558c597857ee436f0f8e17934630b9c04c59 |
| Status:  | <span>✓ Success</span>   |
| Block:   | 3878391 <span>1770659 Block Confirmations</span>                   |
| Timestamp:   | ⌚ 274 days 14 hrs ago (Aug-21-2018 01:51:41 AM +UTC)               |
| From:  | 0x1f78f7c18c63614344fd076b76b9374e993e24b7                         |
| To:  | Contract 0x522e0bdb3ca54942396a01ecb61949b1bd609ce8 <span>✓</span> |
| Value:   | 0 Ether (\$0.00)   |
| Transaction Fee:   | 0.000139774 Ether (\$0.000000)                                     |
| Gas Limit:   | 550,000  |
| Gas Used by Transaction:   | 139,774 (25.41%)   |
| Gas Price:   | 0.000000001 Ether (1 Gwei)   |

Input Data:

04Cooltest Person in Mike's ClassOksana Davis

# Hyperledger

- Hyperledger goes live in December 2015
  - Hyperledger was designed to be an Enterprise Blockchain platform
  - No native currency
  - Ledger can be used to store any type of data
  - Multiple ledger per network
  - Identity and Permissioning
  - Smart Contracts (Chaincode)
  - Primary Focus: A decentralized platform for building cross-organizational business solutions



**HYPERLEDGER**



# Hyperledger

```
1  {
2    "$class": "org.hyperledger.composer.system.AddParticipant",
3    "resources": [
4      {
5        "$class": "org.acme.vehicle.auction.Member",
6        "balance": 0,
7        "email": "alice@abc.com",
8        "firstName": "Alice",
9        "lastName": "Johnson"
10     }
11   ],
12   "targetRegistry": "resource:org.hyperledger.composer.system.ParticipantR
13   "transactionId": "f01890fd-8b01-49b3-adf9-1b3533119ecf",
14   "timestamp": "2019-05-22T17:39:52.673Z"
15 }
```

# The Future

- Greater levels of interoperability thanks to projects like the Interledger Project
  - Interledger.org
  - Hyperledger & Ethereum
    - Hyperledger Besu and Burrow
- The Ethereum 2.0 upgrade
  - More to come!
- Moving beyond cryptocurrency
  - NFTs
  - Decentralized audit trails
- A foundational layer
  - Enabling great emerging tech solutions



ethereum **2.0**

# How is Ethereum Used?

It's more than just currency...

# The Ledger

- Ethereum offers a permanent, decentralized ledger capable of managing all types of data\*
  - *\*Certain data types may work better, or be less cost-prohibitive than others*
  - This ledger can be a great resource for creating audit trails
    - Financial transaction data
    - Supply Chain / Chain of Custody data
    - Weather and Climate data
    - Public Health data
    - Economic data
    - Sports data
    - Gaming and Recreation data
    - Local, Regional, National, International data
    - Contracts, Agreements, and Legislation
    - Status information

H.G.'s Cheesecake Shop  
Cash Disbursements Journal  
March 2025

| Date | Account Debited           | Dr. # | General Debit | Amount Payable Debit | Salaries Debit | Cash Credit |
|------|---------------------------|-------|---------------|----------------------|----------------|-------------|
| 3-1  | Rent                      | 1000  | 800-          |                      |                | 800-        |
| 3-8  | Acct. Pay - Henry's Bk    | 1001  |               | 500-                 |                | 500-        |
| 3-8  | Acct. Pay - Helen's Bk    | 1002  |               | 250-                 |                | 250-        |
| 3-9  | Salaries                  | 1003  |               |                      | 350-           | 350-        |
| 3-10 | Credit Card Pay - Am Bank | 1004  | 150-          |                      |                | 150-        |
|      |                           |       |               | 750-                 | 350-           | 2050-       |

# Ether

- Ether is the native currency of the Ethereum blockchain
  - Currencies are a *native* part of the platforms that support them. They were not created *post-launch*.
- Ether can be used as a form of money, or a way to transfer value
- Ether can be stored in wallets or in Smart Contracts
  - Smart Contracts can implement payable and receivable functions
  - Wallets can be hardware-based, software-based, and paper-based
- Many people view Ether as a security
- Rules, regulations, and tax laws can vary substantially from one region to the next, and are constantly changing
  - Talk to a legal expert!



# ERC-20 Coins and Tokens

- The ERC-20 standard allows developers to create sub-classes of Ether
  - The coins/tokens are compatible with all technology that supports Ether, such as wallets, exchanges, and point of sale systems
- Equity Coins
  - Equity coins represent an *ownership interest* in some *real-world asset with underlying value*
  - Equity coins are like shares of stock
  - Own 1 self-driving car, or a 1% share of 100 self-driving cars?
- Utility Tokens
  - Utility tokens represent *credits, or usage on a platform*
  - The New York subway system, Dave & Buster's, Frequent Flyer Miles, customer loyalty points



# ERC-721 Non-Fungible Tokens

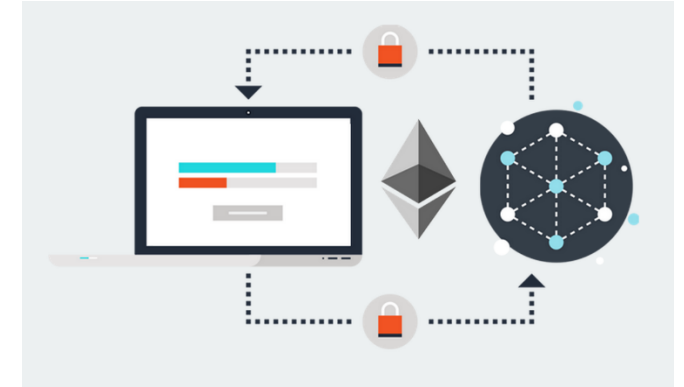
- “Fungible” = Each individual item is worth exactly as much as each other item of the same type
  - US Dollar bills are ***fungible***
    - All dollar bills are worth the same amount - \$1
  - Oranges are ***non-fungible***
    - New, fresh, ripe oranges are worth more than old, rotten, moldy oranges
- Ether and ERC-20s are fungible
  - The ERC-721 standard support the creation of ***NFTs***
- Can be made non-transferrable
  - I can give the NFT to you, but you can never give it to someone else
- Used for collectibles, certifications, awards, achievements, milestones, ownership of non-traditional assets, etc
- ERC-1155 – The Future of NFTs
  - Allows the transfer of multiple tokens at once, single Smart Contract can support multiple NFTs





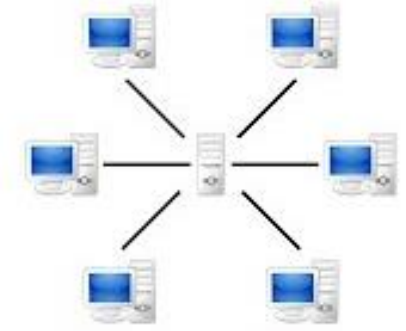
# Smart Contracts

- “*Contract*” is a terrible term...
  - There is nothing legally binding about a Smart Contract
  - It’s just some computer code!
- Smart Contracts allow for a programmatic response to transactions recorded on-chain
  - Smart Contracts make ledger data *actionable*
- Smart Contracts make blockchain an application development platform
  - Enable workflow solutions
- Remember our Politician?
  - Write a Smart Contract for campaign donations
  - Donations are released as campaign promises are fulfilled
- Selling a Used Car
  - Purchase price put into escrow contract
  - Used to fund major repairs
  - If no repairs are needed, the seller receives the full amount



# Blockchain 2.0 = Decentralization + Permanence

- Blockchain solution architecture differs from conventional solution architecture in two major ways:
  - Decentralization
    - Ethereum runs on a **Peer-to-Peer (P2P)** network architecture
      - Leaderless networks, orchestrated by a common protocol
      - All nodes are equally privileged
  - Permanence
    - Conventional Database = **Create, Read, Update, Delete**
    - Blockchain Ledger = **Create, Read, ~~Update, Delete~~**



Server-based



P2P-network

# Blockchain 2.0 = Decentralization + Permanence

- Before selecting blockchain, make sure you want both of these attributes
  - Consider blockchain social media
    - Decentralization = Beneficial, prevents censorship
    - Permanence = Great in some scenarios (politician giving a speech), undesirable in others (social media platform for children and teens)
- Permanence without Decentralization
  - Amazon QLDB, custom database solution
- Decentralization without Permanence
  - Distributed databases
  - P2P database platforms
    - OrbitDB – [orbitdb.org](https://orbitdb.org)



# How does Ethereum Work?

...and why do they call it '*mining*'?

# Mining vs Non-Mining Nodes

- **Mining** is the process of forming **group consensus** on a block of transactions
- Mining is currently very inefficient, and very expensive
- Profitable mining requires expensive and specialized hardware
  - ASIC = Application Specific Integrated Circuit
- Nodes which lack the power to profitably mine should still be able to participate on the network
  - Without losing money!
- Non-mining nodes keep a copy of the ledger, but do not participate in the group consensus process
  - No mining rewards
  - Does receive gas payouts



# Proof-of-Work

- Two-way encryption
  - Data can be encrypted and decrypted using the same algorithm
  - Public Key cryptography
    - “**Key pairs**” are created for each identity
    - Messages encrypted with a **private key** can only be decrypted using the corresponding **public key**, and vice-versa
    - Ethereum wallets store a *user’s private key*
- One-way encryption (cryptographic hashes)
  - Data can be encrypted, but never decrypted
  - Encrypted data cannot be reversed engineered
  - Each hash is unique to the input that created it
  - Hashes are always fixed length (typically 20-32 characters)
  - Used as “*digital thumbprints*”
  - zk-Snarks / Zero Knowledge Proofs
    - *Zero Knowledge Succinct Non-Interactive Argument of Knowledge*
    - Can you prove you know something without revealing what you know?





# Proof-of-Work

- Each individual block in the chain will be validated by the network
  - This process ensures all copies of the ledger are identical
- When it's time to validate a block, each mining node will attempt to find a variable piece of data (***the nonce***) which when combined with the data on the block and the ***hash of the previous block*** gives a ***hash output*** that meets the ***required difficulty*** (number of leading zeros)
  - `hashOf(Block Data + Previous Block Hash + Nonce) = hashWithDesiredLeadingZeros`
  - This is a guessing game; there is no better or faster approach than random guessing





# Proof-of-Work

- The first node to find a working nonce value will share the value with the network
  - Each node will see if the proposed nonce value works with their set of data
- The network will vote on the nonce
  - 51% or more agree – miner is rewarded
  - 50% or less agree – miner is wrong, process begins again
- Why not just compare hashes?
  - Why introduce a seemingly unnecessary piece of work into the process?
  - Generating truly random data is very difficult and expensive, generating hashes is (almost) free
  - No **rational actor** will choose incur a cost with a zero or near-zero chance of recouping it
  - The cost of the artificial work encourages honesty – forces participants to have “some skin in the game”



# Why is it called “*Mining*”?

- Block rewards are paid by minting new coins (*creating new money*)
- In traditional mining, a person goes underground and works hard. If they’re successful, they have *new money* to show for their efforts.
- In Proof-of-Work mining, a node works hard to guess the correct none. If they’re successful, they have *new money* to show for their efforts.



# Proof-of-Stake

- ***Proof-of-Stake*** is a proposed alternative to Proof-of-Work
- In ***PoS*** validation, each node that wishes to participate in the consensus process for the current block must lock up some their funds in a ***stake***.
- One node is *pseudo-randomly* chosen from those who have staked their funds to hash their copy of the current block and share the result
  - Pseudo-randomness can be influenced by things like coin balance, coin age, and more
- Each participating node will compare the hash of their copy of the block to the hash the validator has shared
  - If they are the same, that node only knows it shares its data in common with the validator node, but it doesn't know anything more



# Proof-of-Stake

- At this point, each staked node can choose whether to remain staked or not
- The validator's hash will be voted on
  - 51% or more agree – validator is rewarded, stake is returned
  - 50% or less agree – validator is wrong, no reward is earned and stake is lost
- In many PoS implementations, the *voters* are rewarded or punished as well

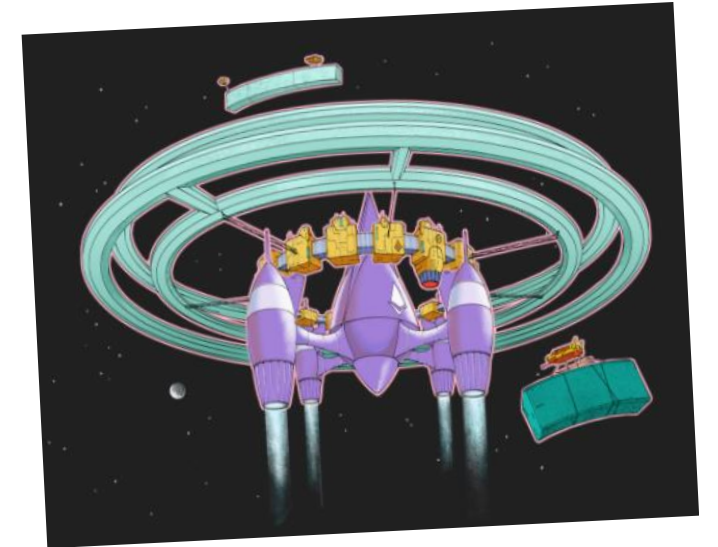


# PoW vs PoS

|                      | Proof-of-Work                   | Proof-of-Stake              |
|----------------------|---------------------------------|-----------------------------|
| Speed                | Slow                            | Fast                        |
| Security             | Excellent                       | Untested                    |
| Data Synchronization | Comparing Nonce values          | Comparing block hash values |
| Energy Consumption   | Enormous                        | Small                       |
| Investment Type      | Specialized Hardware            | Staking Funds               |
| Reward Mechanism     | Newly minted coins              | Newly minted coins          |
| Punishment Mechanism | Cost incurred to generate nonce | Losing staked funds         |

# The Ethereum 2.0 Upgrade

- Why upgrade?
  - The Ethereum 2.0 upgrade has been designed to address some significant concerns about the long-term viability of permissionless blockchains:
    - Cost
    - Environmental Sustainability
    - Ledger Size
    - Performance
    - Scalability



# The Ethereum 2.0 Upgrade

- Cost:
  - The PoW consensus mechanism has created a very expensive platform to use
  - Over time, PoW networks become extremely inefficient
    - This is largely a result of the ever-increasing hash rate of network miners.
      - Competitive ASIC miners can easily cost over \$10,000 each
  - The network as a whole becomes very secure, but extremely inefficient as hash rates increase





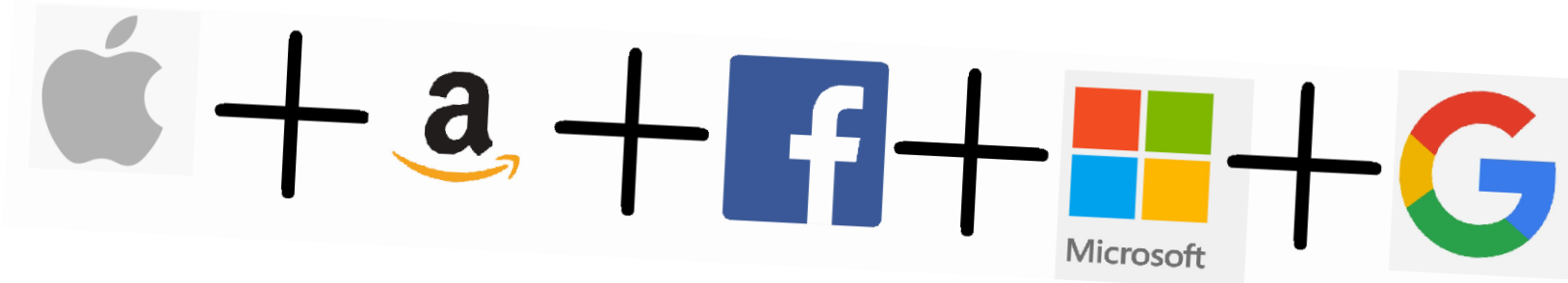
# The Ethereum 2.0 Upgrade

- Cost:
  - How bad is this problem?
    - An Amazon EC2 costs **~\$0.04 per hour**. The Ethereum Virtual Machine (mainnet) costs about **\$240,000 per hour**.
    - A 250GB hard drive costs between **\$20-\$60**. 250GB of storage on the Ethereum mainnet costs **~\$84,000,000**



# The Ethereum 2.0 Upgrade

- Environmental Sustainability:
  - PoW consensus consumes a MASSIVE amount of electricity
    - Bitcoin PoW consumes more electricity than all but the **5 largest nations on Earth**
    - Bitcoin consumes more energy than Apple, Amazon, Facebook, Microsoft, and Google **COMBINED**



# The Ethereum 2.0 Upgrade

- Environmental Sustainability:
  - PoW consensus consumes a MASSIVE amount of electricity
    - Many organizations have an interest in using cryptocurrencies to diversify their balance sheets, but this can cause a lot of problems for Enterprise Sustainability Goals or other similar environmental initiatives.
    - One can reasonably expect permissionless blockchains to grow MUCH MORE over the next 5-10 years than they have their first 5-10 years.



# The Ethereum 2.0 Upgrade

- Ledger Size
  - The Ethereum Mainnet ledger has become very large
    - Spring 2018: Ledger exceeds 1TB
    - Early 2021: Ledger 2+TB
    - One can reasonably expect permissionless blockchains to grow MUCH MORE over the next 5-10 years than they have their first 5-10 years.
  - The ledger will only continue to grow...
    - Running 'light' nodes only partially addresses this issue
    - Full nodes will still be required
    - As the network grows, more full nodes will be required



# The Ethereum 2.0 Upgrade

- Ledger Size
  - There has been disagreement over whether this is really an issue...
    - Some people believe this is unsustainable
    - Some people believe technology will always stay one step ahead
    - What do *you* think?



# The Ethereum 2.0 Upgrade

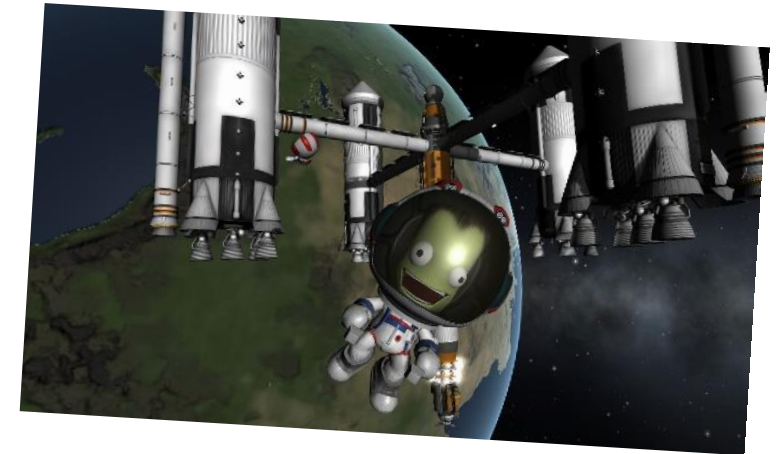
- Performance
  - Ethereum mainnet performance has been mediocre at best
    - Typical performance: 10-20 TPS\*
    - Visa: 20,000 – 70,000 TPS\*
    - Facebook: 175,000+ TPS\*
    - *\*NOTE: When comparing TPS measurements, make sure you know what defines a 'transaction'*
  - Smaller networks and private network instances don't suffer as much
  - One can reasonably expect permissionless blockchains to grow **MUCH MORE** over the next 5-10 years than they have their first 5-10 years



# The Ethereum 2.0 Upgrade

- Scalability

- The current state of the PoW mainnet does not encourage participation from 'everyday users'
  - Many feel this goes against the decentralized ethos of blockchain
- Scaling up an highly inefficient approach only **INCREASES** inefficiency
  - This quickly becomes an unsustainable problem
  - The Rocket Problem
- One can reasonably expect permissionless blockchains to grow **MUCH MORE** over the next 5-10 years than they have their first 5-10 years





# The Ethereum 2.0 Upgrade

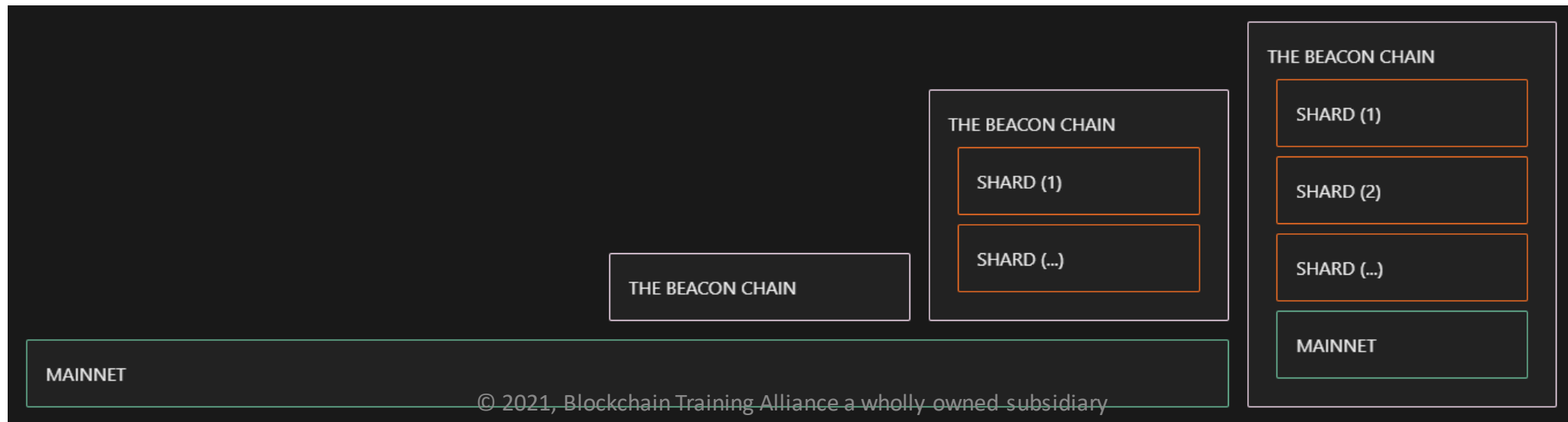
- Biggest Concern:
  - “*How do we move from PoW to PoS without alienating PoW hardware owners?*”
  - A hard cutover would upset the vast majority of the current node owners
  - Original Solution: ***The Ethereum Ice Age***
    - Gradually ramp-down PoW while ramping up PoS
    - This plan is now deprecated in favor a three phase upgrade





# The Ethereum 2.0 Upgrade

- The upgrade from Ethereum 1.0 to Ethereum 2.0 will happen in three phases:
  - Phase 1: Beacon Chain (January 2021)
    - A new PoS network goes live alongside the existing PoW network



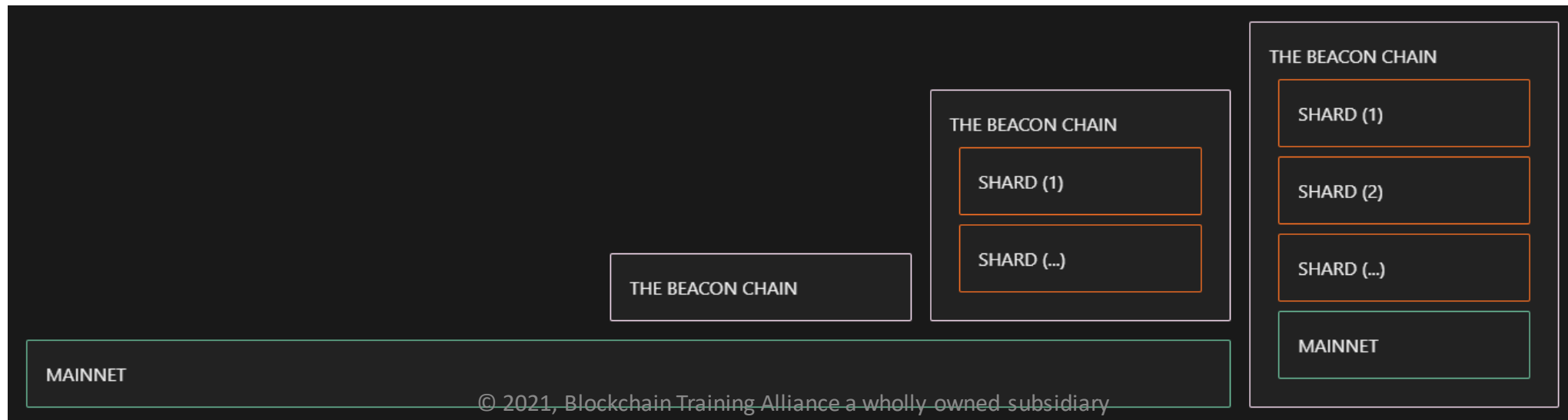
# The Ethereum 2.0 Upgrade

- The upgrade from Ethereum 1.0 to Ethereum 2.0 will happen in three phases:
  - Phase 2: New Shard Chains (sometime 2021)
    - 64 new shards will go online using PoS consensus
    - Validators will be randomly assigned (and re-assigned) to different shards
    - ‘Sharding’ is a way to split the resources of the network
    - Benefits:
      - Greater performance
      - Smaller ledger size on each node
    - Drawbacks:
      - Fewer nodes validating each block



# The Ethereum 2.0 Upgrade

- The upgrade from Ethereum 1.0 to Ethereum 2.0 will happen in three phases:
  - Phase 3: The Docking (sometime 2022)
    - PoW network will 'merge' or 'dock' with the 64 PoS shards
    - 65<sup>th</sup> PoW shard will remain PoW



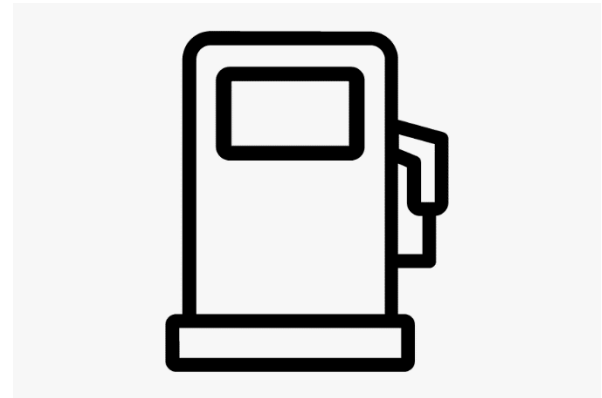
# The Ethereum 2.0 Upgrade

- What do I HAVE to do?
  - Nothing.
- What CAN I do?
  - Run a node
    - <https://ethereum.org/en/eth2/get-involved/#clients>
  - Stake your Ether
    - <https://ethereum.org/en/eth2/staking/>
  - Hunt down bugs (and earn money, up to \$50,000)
    - <https://ethereum.org/en/eth2/get-involved/bug-bounty/>



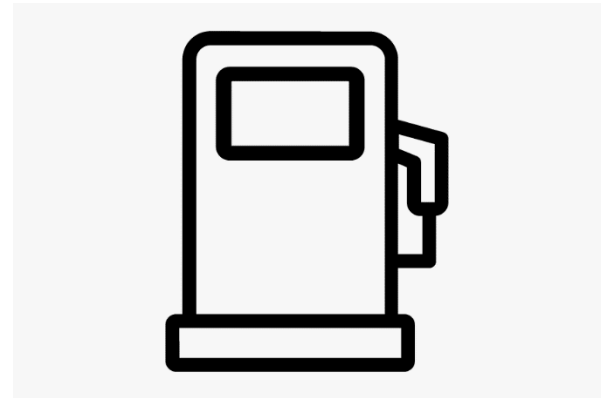
# Gas in Ethereum

- “**Gas**” is used to pay for transactions on the Ethereum network
- Remember: There are only two things you can do on the ledger:
  - Read an existing record
    - Reading never incurs a gas charge, it’s always free
    - A request for data can be serviced by one single network node
      - All copies of the ledger are identical, no need for a second node’s help or verification
  - Create a new record
    - Creating a record always incurs a gas charge, it’s never free
    - Adding a record to the ledger requires the entire network to add it to their copy of the ledger
      - All copies of the ledger must be identical, all nodes need to add the transaction



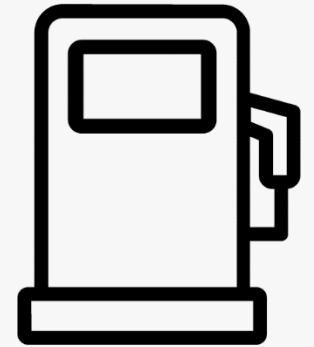
# Gas in Ethereum

- When a user submits a transaction, enough Ether is removed from their wallet to fund their desired gas purchase
  - If a transaction is not provided enough gas, execution will stop and the ledger will roll back to its previous state
    - No transaction will be recorded
    - The user will not be refunded their gas
  - If a transaction is provided more gas than it needs, unburnt gas will be converted back into Ether and put into the user's wallet
    - The transaction will be recorded successfully



# Gas in Ethereum

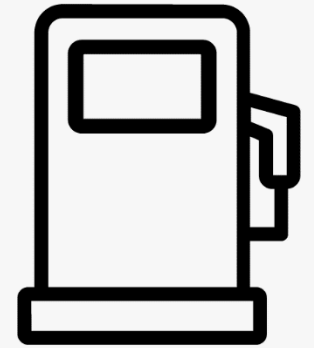
- Why not pay to use the network with Ether?
  - This is how Bitcoin works
  - If there's already a currency, why not use it?
- Charging users helps to prevent infinite loops
  - An infinite loop would require infinite gas, which would require infinite Ether, which would cost an infinite amount of money
- Cryptocurrency Price Volatility
  - Using the network at different times should always incur the same relative cost
  - Ether and Gas have an inverse price relationship, when one goes up the other goes down
    - Today:  $\$1 = 1 \text{ Ether} = 1 \text{ gas}$ 
      - Ether price =  $\$1$
      - Gas price = 1 Ether
    - Next Week:  $\$1 = \frac{1}{2} \text{ Ether} = 1 \text{ gas}$ 
      - Ether price =  $\$2$  – 200% increase
      - Gas price =  $\frac{1}{2} \text{ Ether}$  – 50% decrease
    - Gas Per Dollar remains constant!





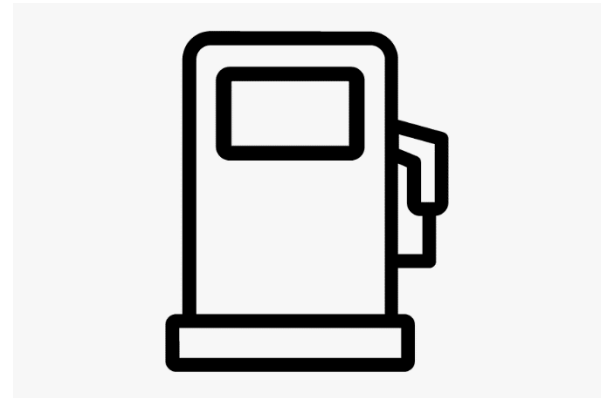
# Gas in Ethereum

- Gas Price determines how quickly transactions are processed
  - A node with multiple transactions in the queue will select the user who is willing to pay the highest gas price
  - Paying a higher gas price results in a quicker transaction
  - Gas is priced in *Gwei*
- Eth Gas Station
  - <https://www.ethgasstation.info/>



# Gas in Ethereum

- Gas cost = fixed cost for function + variable cost based on data
  - Limiting the amount of data can help reduce gas costs
- Fixed cost based on Opcodes within a function call
- To determine gas cost:
  - Test your Smart Contract in a tool such as Remix
    - Quick, easy, relatively accurate
  - Compile your Smart Contract and add up the value of all Opcodes in a function call
    - Slow, difficult, extremely precise estimates



# Gas in Ethereum

## Opcodes & Gas

|          | QUICKSTEP    | FASTESTSTEP  | FASTSTEP   | MIDSTEP | SLOWSTEP | EXTSTEP         |  |
|----------|--------------|--------------|------------|---------|----------|-----------------|--|
| Gas cost | 2            | 3            | 5          | 8       | 10       | 20              |  |
|          | ADDRESS      | DUP          | MUL        | ADDMOD  | JUMPI    | BLOCKHASH       |  |
|          | ORIGIN       | SWAP         | DIV        | MULMOD  | EXPBASE  | BALANCE         |  |
|          | CALLER       | PUSH         | MOD        | JUMP    |          | EXTCODESIZE     |  |
|          | CALLVALUE    | ADD          | SDIV       |         |          | EXTCODECOPYBASE |  |
|          | CALLDATASIZE | SUB          | SMOD       |         |          |                 |  |
|          | CODESIZE     | LT           | SIGNEXTEND |         |          |                 |  |
|          | GASPRICE     | GT           |            |         |          |                 |  |
|          | COINBASE     | SLT          |            |         |          |                 |  |
|          | TIMESTAMP    | SGT          |            |         |          |                 |  |
|          | NUMBER       | EQ           |            |         |          |                 |  |
|          | DIFFICULTY   | AND          |            |         |          |                 |  |
|          | GASLIMIT     | OR           |            |         |          |                 |  |
|          | POP          | XOR          |            |         |          |                 |  |
|          | PC           | NOT          |            |         |          |                 |  |
|          | MSIZE        | BYTE         |            |         |          |                 |  |
|          | GAS          | CALLDATALOAD |            |         |          |                 |  |
|          |              | CALLDATACOPY |            |         |          |                 |  |
|          |              | CODECOPY     |            |         |          |                 |  |
|          |              | MLOAD        |            |         |          |                 |  |
|          |              | MSTORE       |            |         |          |                 |  |
|          |              | MSTORE8      |            |         |          |                 |  |

# Units of Value in Ethereum

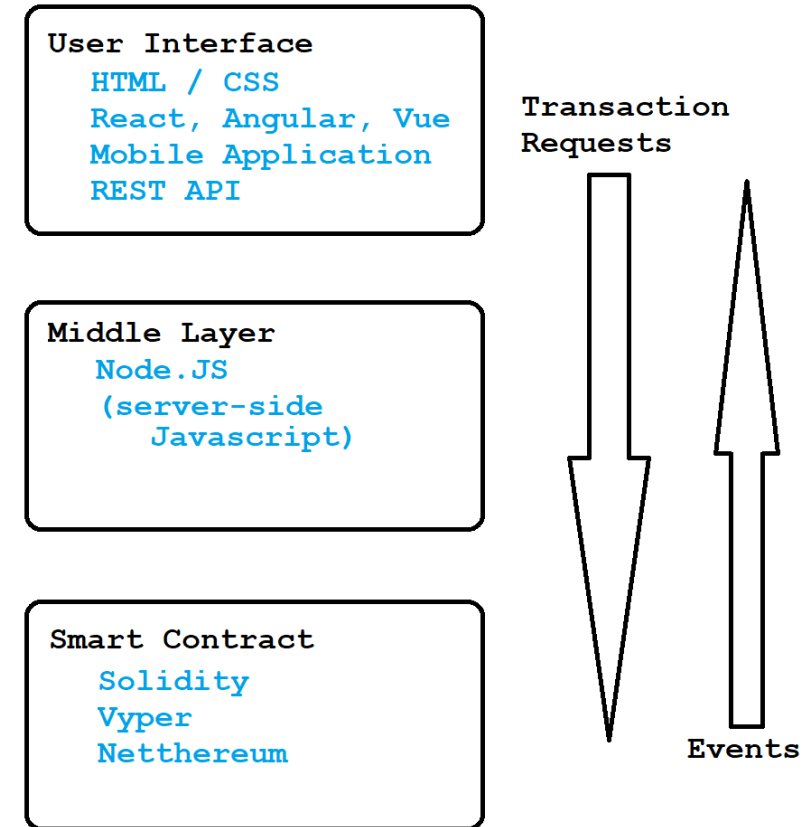
| Unit                | Wei Value | Wei                       |
|---------------------|-----------|---------------------------|
| wei                 | 1 wei     | 1                         |
| Kwei (babbage)      | 1e3 wei   | 1,000                     |
| Mwei (lovelace)     | 1e6 wei   | 1,000,000                 |
| Gwei (shannon)      | 1e9 wei   | 1,000,000,000             |
| microether (szabo)  | 1e12 wei  | 1,000,000,000,000         |
| milliether (finney) | 1e15 wei  | 1,000,000,000,000,000     |
| ether               | 1e18 wei  | 1,000,000,000,000,000,000 |

# What Does an Ethereum App Look Like?

...and what are the technologies involved?

# Blockchain Application Layers

- Each blockchain application will have three primary architectural layers
  - The User Interface Layer
    - The user interface layer provides a way for end-users (either human or machine) to interact with our Smart Contract solution
    - A blockchain solution can have multiple different user interfaces at this layer
  - The 'Middle Layer'
    - The Middle Layer is the glue to connect the User Interface Layer to the Smart Contract / Blockchain Layer
    - A blockchain solution can have multiple different middle layer implementations
  - The Smart Contract / Blockchain Layer
    - The Smart Contract layer contains all the logic that lives on the decentralized peer-to-peer blockchain network
    - A blockchain solution can have multiple linked Smart Contracts at this layer



# Blockchain Application Demo



# Solution Design Consideration

What else goes into an Ethereum application design?

# Hypermedia Content Management

- IPFS – The Interplanetary FileSystem
  - IPFS.io
  - Designed as a compliment to, as well as a replacement for, the HTTP protocol
  - IPFS content uses a content-based URL, not a location-based URL
    - The URL of an IPFS item is the hash of its contents
  - Content is requested from all nodes on the network that have all or a part of it
    - Content is not requested from a specific location
    - Content delivery speeds up as it becomes more widely-distributed
  - IPFS can be used in Ethereum solutions
    - To store large amounts of data (BLOBs, images, videos, PDFs, etc)
    - To host Middle Layer and User Interface assets
      - Create a P2P solution at every layer!
      - Incredible fault tolerance and failover capacity



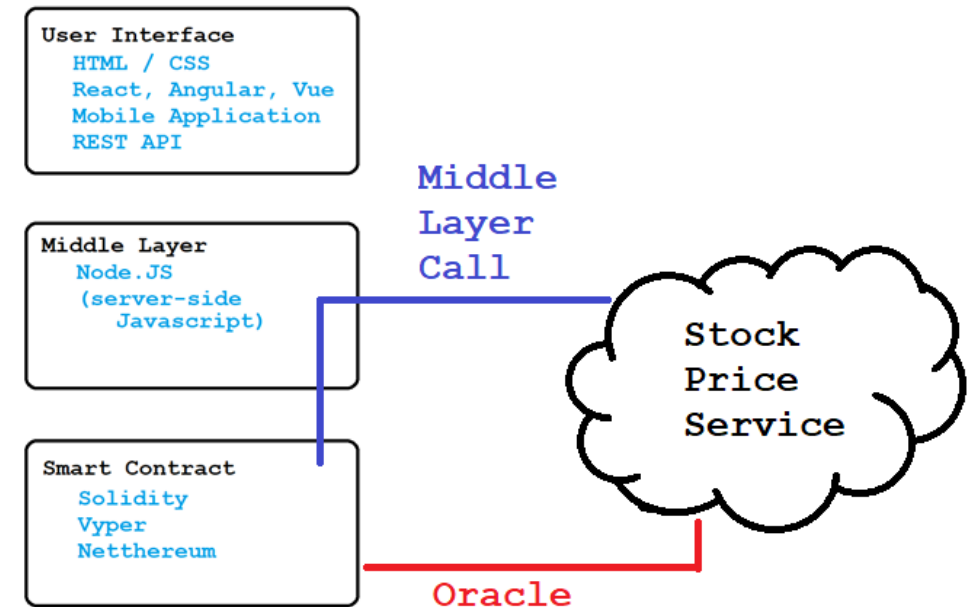
# Blockless Platforms

- IoTA
  - Decentralized ledger platform designed for IoT and Smart Device Solutions
    - Designed to handle data volume of modern IoT devices
  - Tangle consensus
    - Each transaction is verified by the submitters of the following several transactions
    - Rather than having the entire network validate a block of transactions, a small number of individual nodes validate each individual transaction
    - Greater performance, *relatively* lower level of security
  - Typically used a front-end capture layer for IoT devices
    - 'Uninteresting' transactions are managed on IoTA
    - 'Interesting' transactions are passed to an Ethereum network to be included on the permanent ledger for long-term auditing needs or requirements



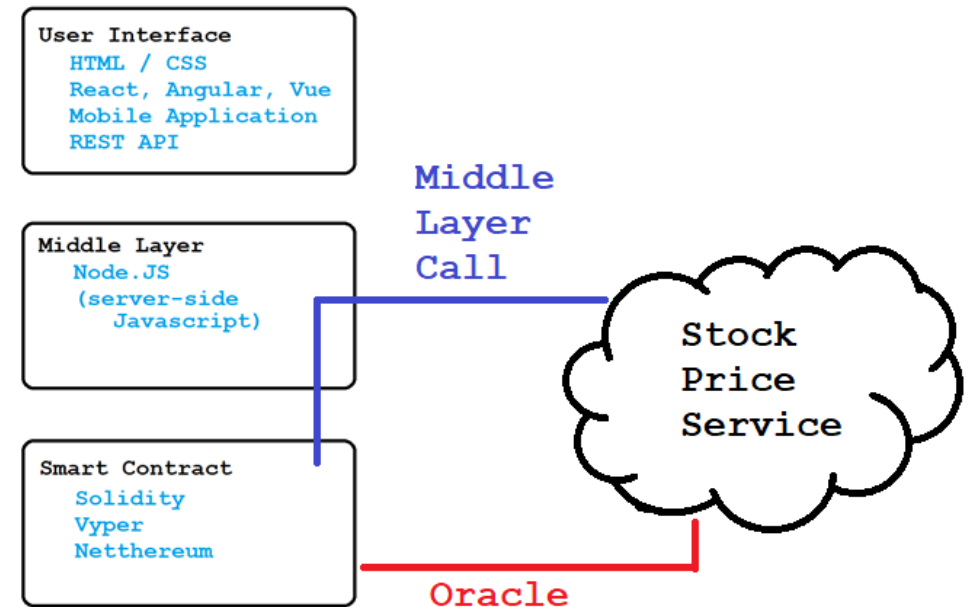
# Deterministic Concurrency

- Many solution designs that would be acceptable and safe in the client / server world aren't in the blockchain world
- The Blockchain development experience has been crafted to mirror the *classic* development experience as much as possible
  - Don't forget things are VERY different under the surface
  - You are running code on a large, leaderless, decentralized peer-to-peer networks
- Consider a solution which requires external data in order to make a decision
  - Your Ethereum application will query an external web service to get the current price of a particular share of stock
  - The stock in question is experiencing a lot of price volatility
  - The stock pricing service returns prices with a high degree of precision



# Deterministic Concurrency

- Using an **Oracle** pattern would result in:
  - Each node making the call to the external service at *slightly* different times
  - Each node potentially getting a *slightly* different price
  - Many, many variations of the current block, each with a *slightly* different price quote
  - Inability to form consensus on the block
  - Discarding the block
- Making a single Middle Layer call would result in:
  - A single price quote being passed to the Middle Layer
  - Each node on the blockchain network receiving the exact same price
  - Consensus can most likely be formed on the current block



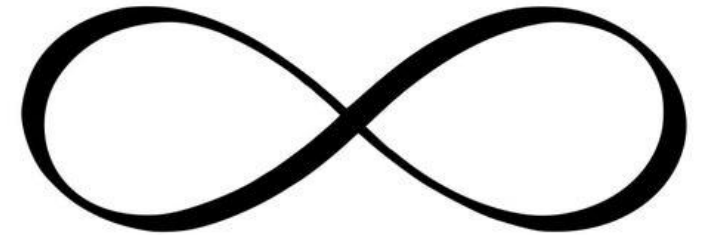
# Security Best Practices

- Consensys maintains an excellent knowledgebase of Smart Contract best practices at <https://consensys.github.io/smart-contract-best-practices>
  - The following examples (and more) can be found in the “*Known Attacks*” section



# Security Best Practices

- Reentrancy
  - One of the major dangers of calling external contracts is that they can take over the control flow, and make changes to your data that the calling function wasn't expecting.
    - This class of bug can take many forms, and both of the major bugs that led to the DAO's collapse were bugs of this sort.
  - The first version of this bug to be noticed involved functions that could be called repeatedly, before the first invocation of the function was finished.
    - This may cause the different invocations of the function to interact in destructive ways.





# Security Best Practices

- Reentrancy
  - Since the user's balance is not set to 0 until the very end of the function, the second (and later) invocations will still succeed, and will withdraw the balance over and over again.

```
// INSECURE
mapping (address => uint) private userBalances;

function withdrawBalance() public {
    uint amountToWithdraw = userBalances[msg.sender];
    (bool success, ) = msg.sender.call.value(amountToWithdraw)("");
    require(success);
    userBalances[msg.sender] = 0;
}
```

# Security Best Practices

- Forcibly Sending Ether to a Contract
  - At first, it would seem like there was no way for *somethingBad* to happen
  - However, it is possible to forcibly send Ether to a Smart Contract
    - The *balance* property could be forced above zero

```
contract Vulnerable {  
    function () payable {  
        revert();  
    }  
  
    function somethingBad() {  
        require(this.balance > 0);  
        // Do something bad  
    }  
}
```

© 2021, Blockchain Training Alliance a wholly owned subsidiary  
of The Crypto Company (CRCW)

# Security Best Practices

- Front-Running
  - All transactions are visible in the *mempool* for a brief period before being executed
    - This means observers of the network can see and react to a transaction before it is included in a block.
  - An example of how this can be exploited is with a decentralized exchange where a buy order transaction can be seen, and second order can be broadcast and executed *before* the first transaction is included.
  - Protecting against this is difficult, must happen at the contract, not the network layer



# Public vs Private Networks

- Ethereum is a network protocol
  - It can run on *any* network, from the *internet* to your local *intranet*
    - HTTP is used to serve content on the public internet as well as private intranets
    - Setting up a private Ethereum network is a quick and simple process
      - Install an Ethereum client (typically *GEth*)
      - Create a Genesis Block configuration file
        - JSON data
      - Give your network a unique ID
      - Start your node
      - Add additional nodes by using the new network ID



# Public vs Private Networks

- Ethereum is a network protocol
  - Ethereum solutions can be deployed to networks only accessible to the relevant end users
    - This is oftentimes much cheaper than running an application on the mainnet
    - You could be more vulnerable to hacks, attacks, and exploits
  - Ask Yourself:
    - Am I building an application for the world at large, or for a limited group of users?
    - Do I need to be able to identify members of the network?
    - Do I need to be able to control network access?

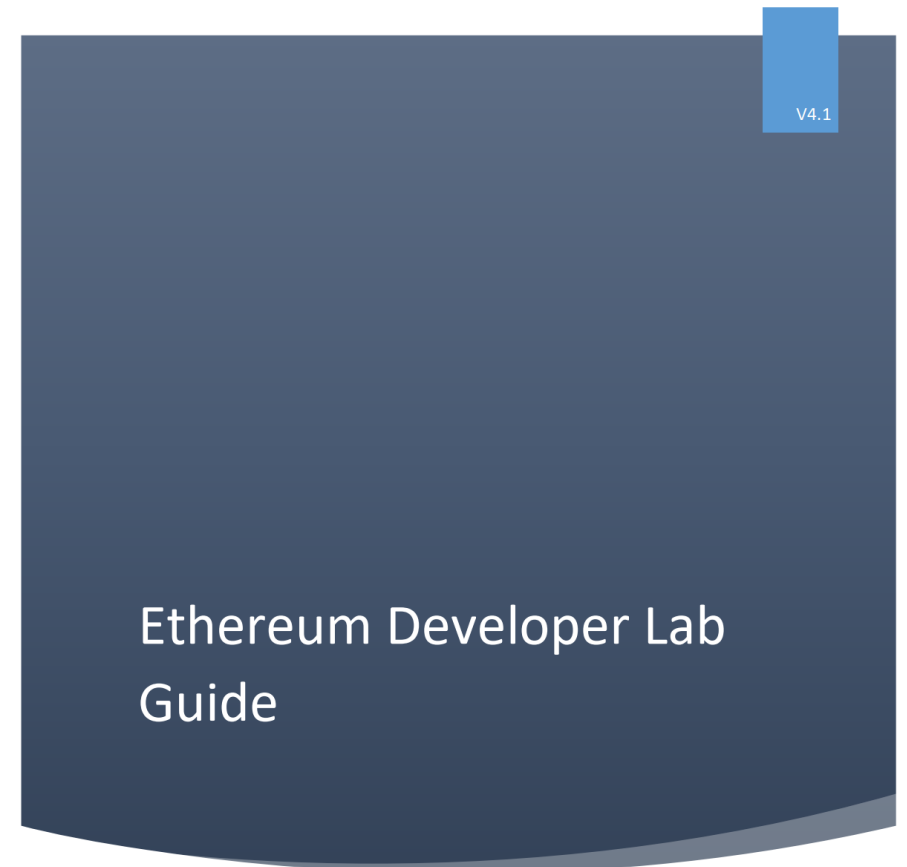


# Hands-On Labs

Time to put knowledge into action!

# Hands-On Labs

- Lab environment setup and configuration
  - How to get your lab environment up and running
  - Some configuration notes on your environment
  - How to use the lab guide and code snippets file
- Lab Exercises:
  1. The Remix IDE
  2. MetaMask and the Test Networks
  3. Infura.io
  4. Web3.js
  5. Events
  6. Function Modifiers
  7. Mappings and Structs
  8. Inheritance
  9. Deploying to Live Networks
  10. Creating Your Own ERC-20 Token
  11. Creating Your Own ERC-721 Token
  12. Bonus Lab: Working with Truffle Boxes



LAB GUIDE FOR CERTIFIED ETHEREUM DEVELOPER  
KRIS BENNETT – BLOCKCHAIN TRAINING ALLIANCE

COPYRIGHT BLOCKCHAIN TRAINING ALLIANCE | 2021-2022