Smart contracts, Ethereum

Portland State
Computer Science

Motivation

- Bitcoin
  - Distributed ledger of financial transactions (currency transfers)
  - Provides secure, immutable, global ordering of financial transactions
- What if a "transaction" were the execution of CPU instructions instead?
- What if the blockchain were treated as an execution record for a computer that includes its programs and their processes?

Goal

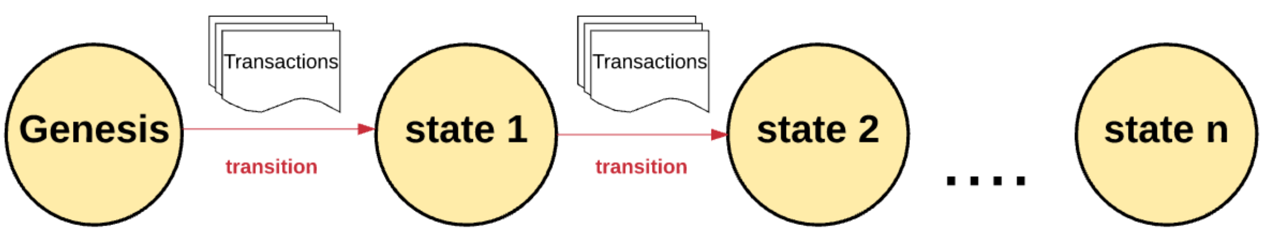
- Extend blockchain to create a replicated, distributed, state machine that can…
  - Store arbitrary data
  - Store persistent programs and their execution states
  - Support function calls from users to these programs and have results globally visible and agreed upon

Smart contract definitions

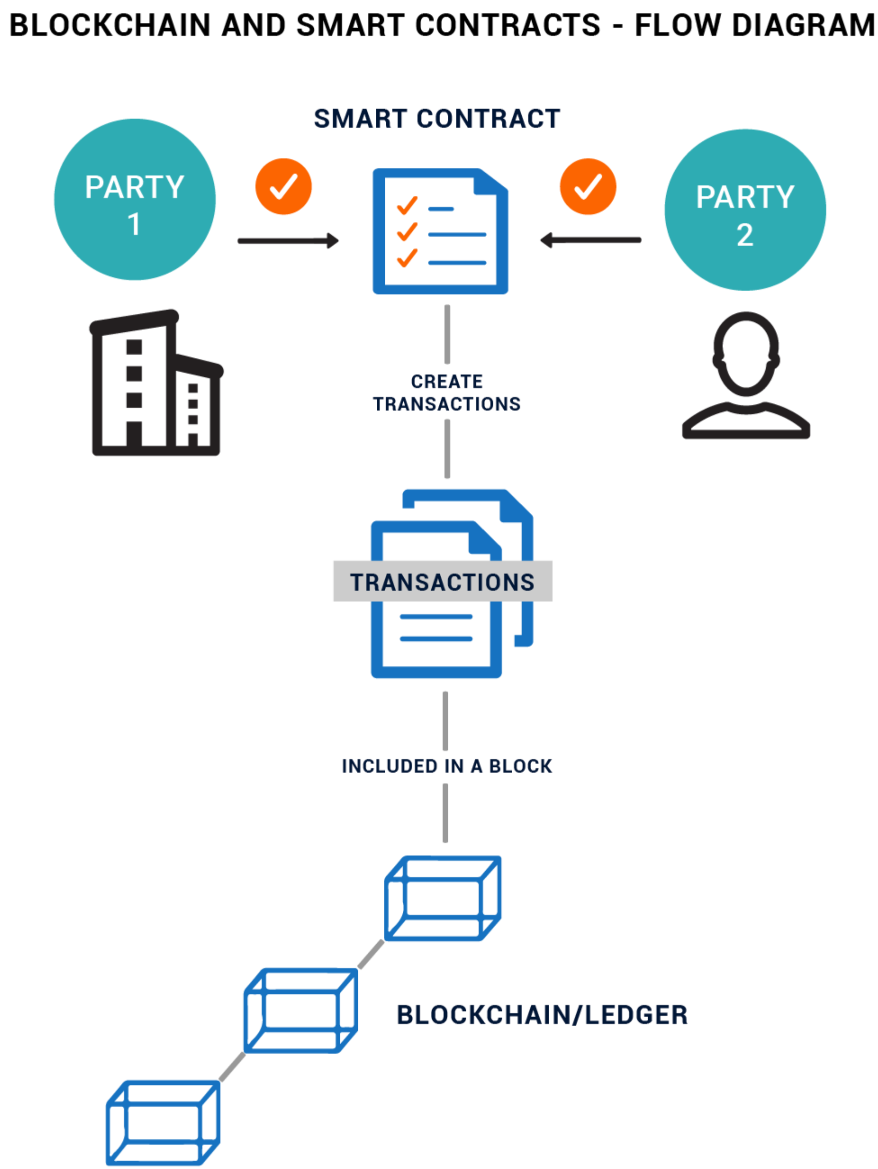
- Also known as "persistent scripts" or "stored procedures"
- #1: A computer program executed in a secure environment that directly controls digital assets
- #2: Computer program that digitally facilitates, verifies, or enforces the performance of a contract and its transactions in a trackable and irreversible manner without a third party
- Model
  - Programs first committed to blockchain
  - Receive authenticated inputs via other programs or users on the blockchain
  - Produce state changes and output based on program execution
  - Execution is duplicated and replicated across all participating nodes to maintain single global state

Operating paradigm

- Begin with "genesis state" (similar to CoinBase)
- Use distributed consensus to implement shared state machine
  - Blockchain executes transactions to move states
- Abstraction
  - Single, shared machine
  - Single shared, persistent memory storing code, execution state, and data for smart contract (akin to a persistent process)
  - Abstraction of a single, global computer with shared-state?
    - Mainframe computing model
    - Proof that everything old is new again! ☺



- Credit: LinuxFoundationX: LFS171x



Used to implement DApps (Distributed Applications)

BUT..

Immutability

- Contract code is ***immutable!***
  - Code is there to stay, permanently, on the blockchain and can never be modified or updated again once deployed
    - Code is law
    - No mechanism to patch (e.g. the opposite of CI/CD)
- Motivates…

Security

- Konstantopoulous
  - "*In a potential future where* ***whole organizations are governed by smart contract code****, there is an immense need for proper* ***security****.*
  - Must ensure your contract has no vulnerabilities *before* deployment
    - Why code audits on smart contracts matter!
    - Why program analysis and symbolic execution matter!
  - Fixes to vulnerable code require completely new contract to be deployed and users moved over to new contract address (if possible)
  - Kill switches and safety valves sometimes built into contracts
    - But, this protects contract owner at the expense of users.
    - Tension between trusting code or trusting owner of the contract
    - Buyer beware!

Classes of DApps

- Automate or streamline operation of a trusted third party (trust is expensive)
- Automate transaction processing
- Implement legal contracts with unambiguous terms that can be expressed in code of program
- Create scarcity in digital domain (e.g. currencies, coins/stock, collectibles)

Sports betting

```
if TigerWoodsWinsMasters2019() is true:
    party_A.transfer(14*bet_amount)
```

```
if TrailBlazersWinChampionship2021() is true:
    party_A.transfer(3000*bet_amount)
```

https://www.sportsbettingdime.com/nba/championship-odds/

Legal contracts

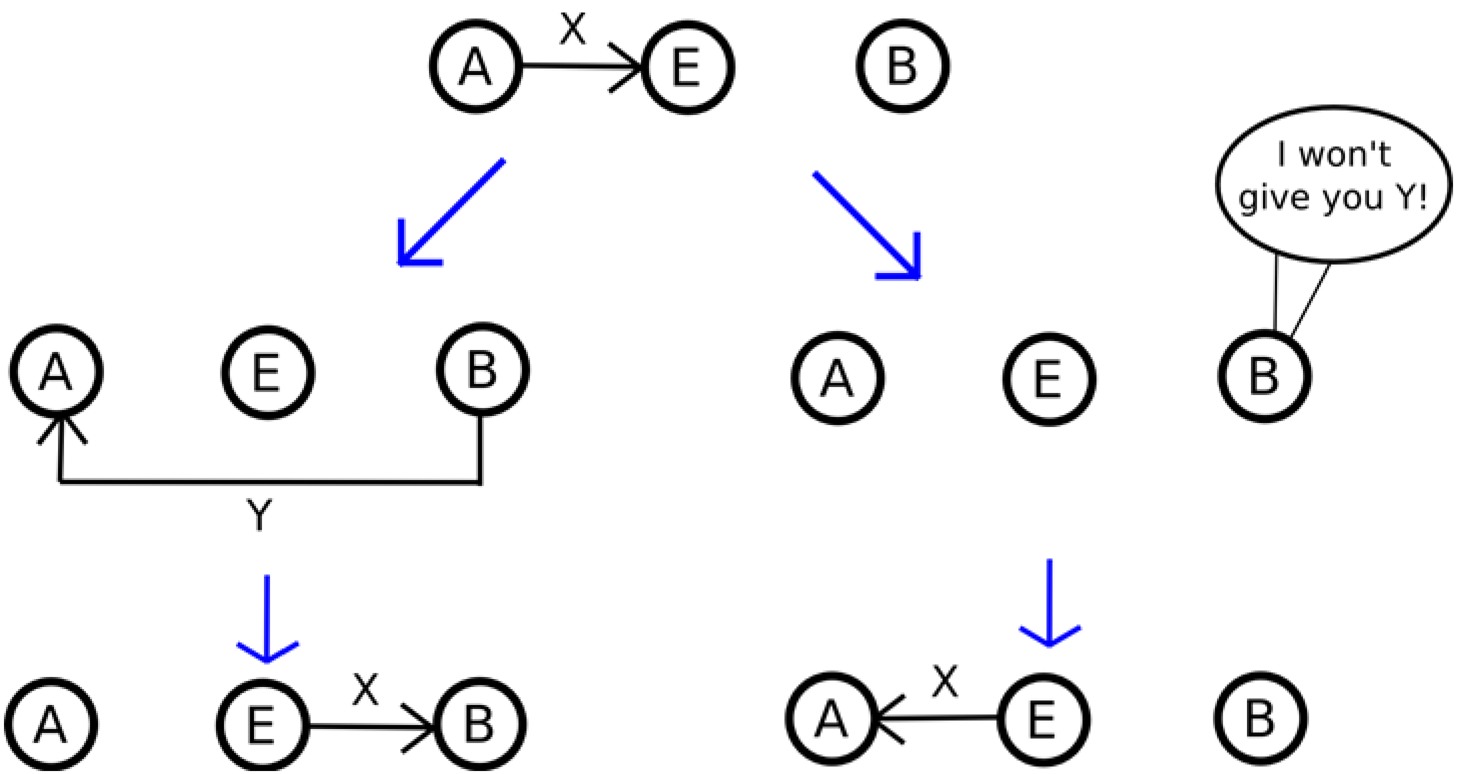
- Trust fund

```
if current_year() > 2040:
    child_A.transfer(fund.balance())
```

- Digital will
  - Dead man's switch that executes code to transfer digital assets upon owner dying
  - Private key of coroner's office signs a transaction that triggers execution of the will

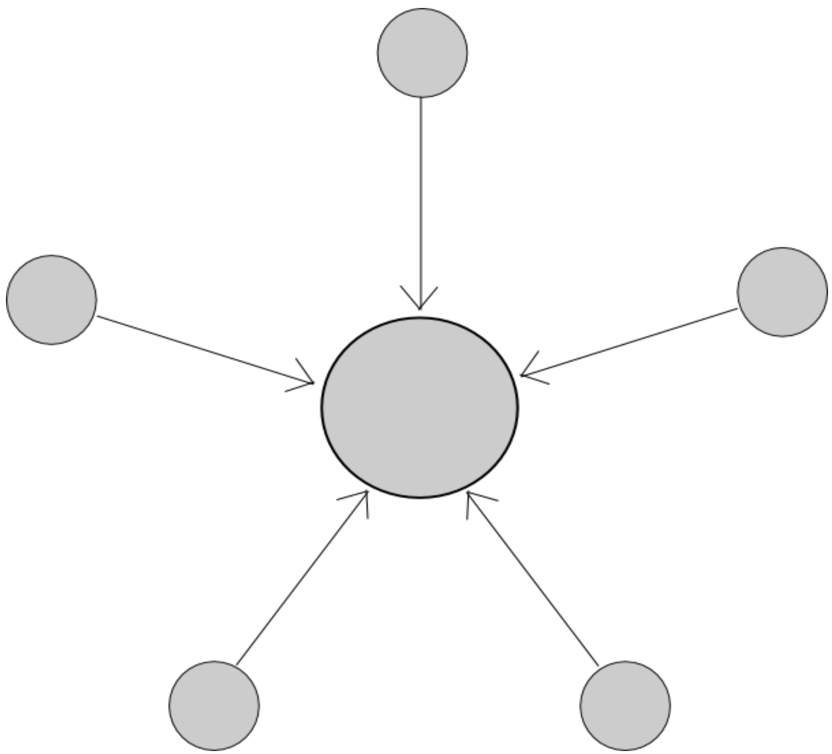
Escrow contracts

- Trustworthy asset exchange
  - A transfers X amount to E (escrow contract)
  - B transfers asset Y (e.g. digital deed) to A
  - E automatically transfers X to B upon seeing Y being transferred to A
  - If B refuses to transfer asset Y
    - E returns X amount to A after specified timeout
  - Can be done via 20 LoC, avoid paying thousands of dollars



Multi-signature, multi-party asset transfers

- Require approval of a set of individuals before executing a transfer
  - Example: Sale of a company approved by majority of stakeholders signing shares to trigger transfer



Decentralized finance applications

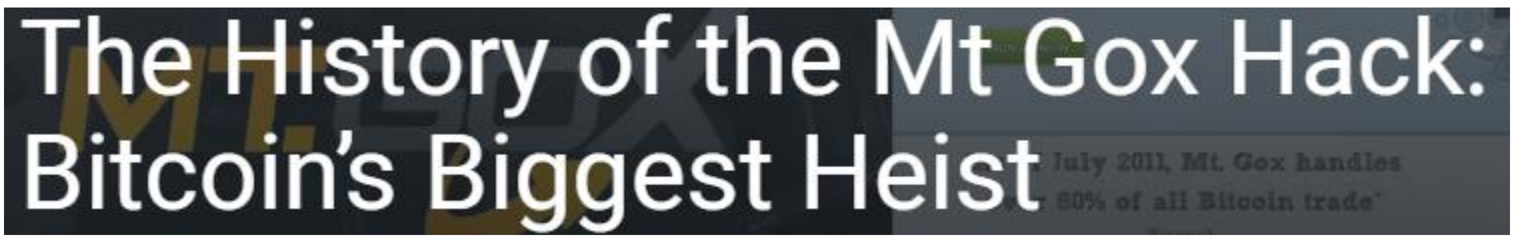
- Initial Coin Offerings selling ERC-20 tokens (more later)
  - Virtual version of IPOs selling shares of a company
- Option contracts
  - Allow a buy/sell transaction to be triggered based on date or condition (e.g. strike price) being hit
  - Executes itself according to coded terms
  - Contract can be made between parties potentially unknown to each other
  - Would afford regulators greater transparency to view and audit transactions for abuse

Decentralized finance applications

- Bootstrapping alternate networks (EOS, Tron)
  - Shares purchased via ETH
  - Shares exchanged for EOS or Tron when launched
  - https://etherscan.io/address/0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0
- Virtual crowd-source funding (Kickstarter)
  - OmiseGO https://etherscan.io/address/0xd26114cd6ee289accf82350c8d8487fedb8a0c07
- To implement "stable coins"
  - Coins pegged to real $
  - Similar to Digicash

Centralized exchanges

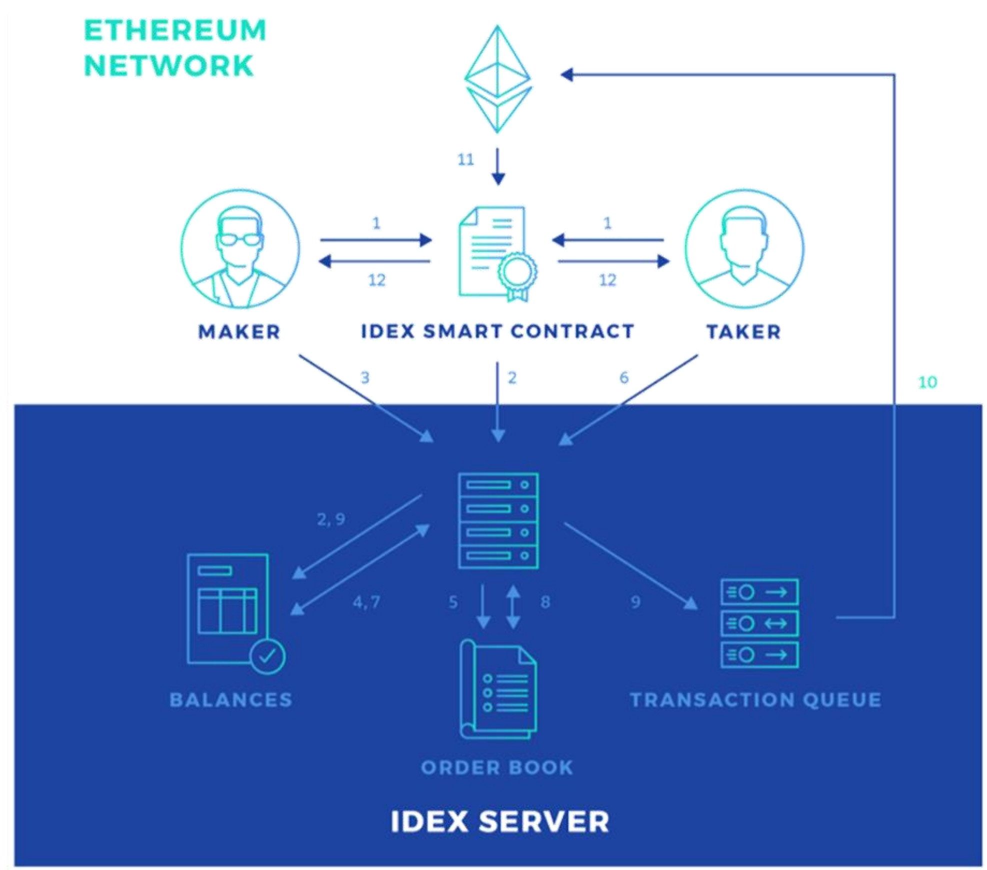
- Exchanges that hold user assets directly
  - Users deposit, withdraw, and trade ETH and ERC-20 tokens all within central contract (e.g. like E*Trade)
- Bittrex, Polonex
  - Buy, sell, trade over 100 supported ERC-20 tokens
    - https://etherscan.io/address/0x209c4784ab1e8183cf58ca33cb740efbf3fc18ef
- What if the exchange is hacked?



  - https://blockonomi.com/mt-gox-hack/
    - The victim of a massive hack, Mt. Gox lost about 740,000 bitcoins (6% of all bitcoin in existence at the time), valued at the equivalent of €460 million at the time and over $3 billion at October 2017 prices.

Decentralized exchanges

- Exchange contract does not hold user assets but instead facilitates exchange
- Users buy and sell crypto assets without an intermediary storing the assets via their private keys
- Trading ETH and ERC-20 tokens
  - EtherDelta
  - IDEX: Market making done off-chain, commit to chain via exchange



DNS

- Name to address lookups (Ethereum Name Service)
  - Can see when domain is registered! (TLS certificate transparency)

```
data domains[](owner, ip)
```

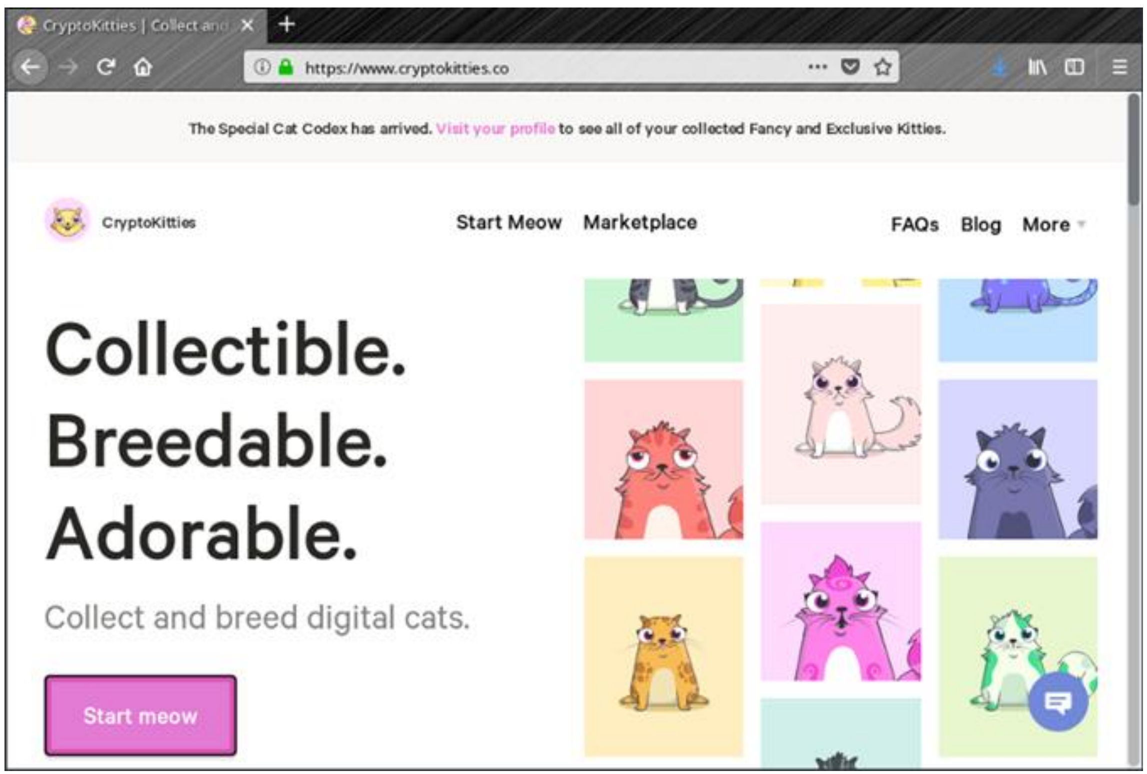
Private Storage

```
def register(name):
    if not self.domains[name].owner:
        self.domains[name].owner = msg.sender

def set_ip(name, ip):
    if self.domains[name].owner == msg.sender:
        self.domains[name].ip = ip

def get_ip(name):
    if self.domains[name]:
        return self.domains[name].ip
    else:
        return None
```

Ensure only owner can set

Collectibles

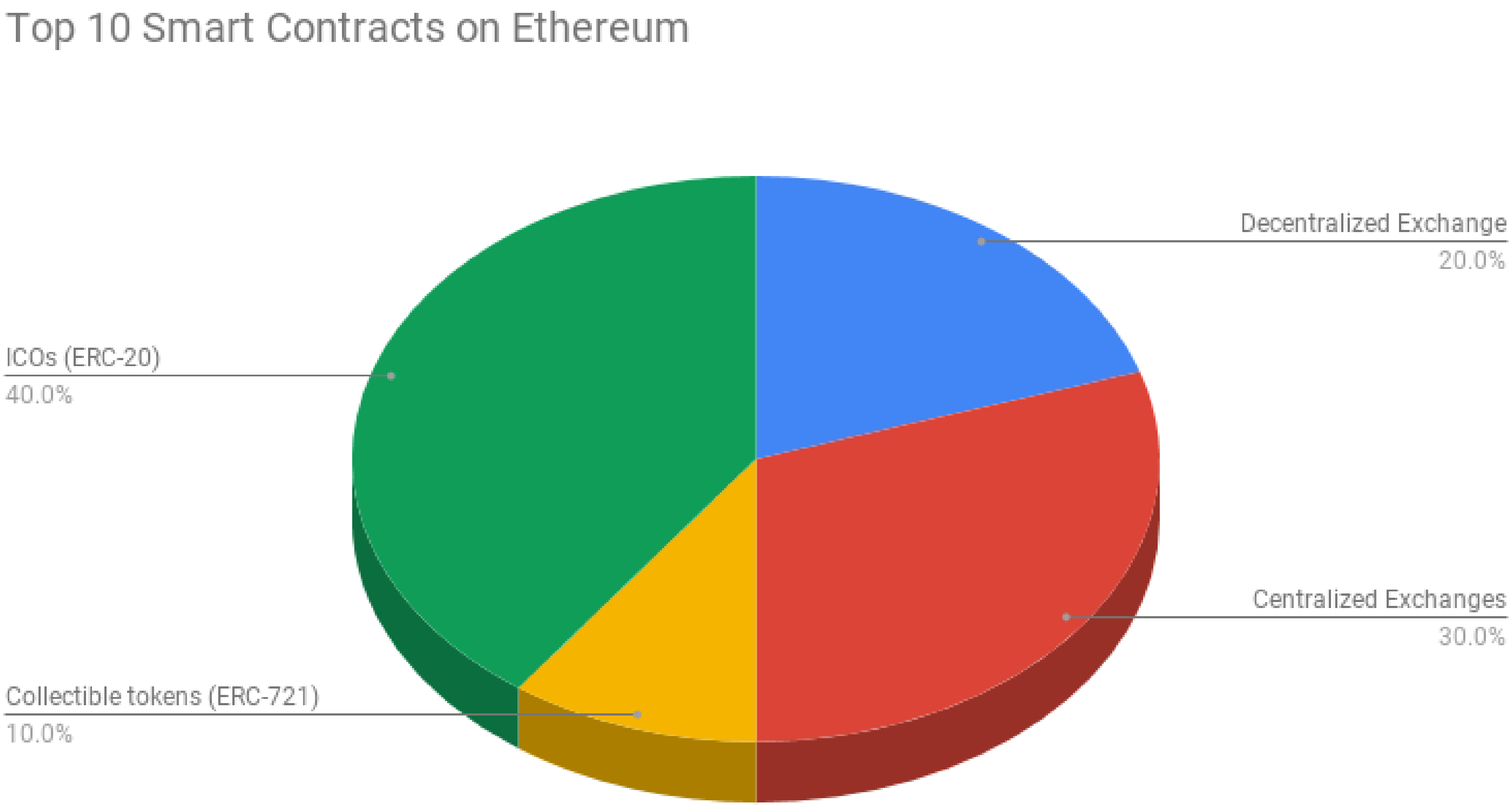
- Smart contracts for implementing ERC-721 tokens (more later)
  - Non-fungible, unique tokens that live in perpetuity (CryptoKitties)
  - Smart contract generates unique tokens that are transferred to users
  - No centralized authority to duplicate or steal kitty away
  - https://etherscan.io/address/0x06012c8cf97bead5deae237070f9587f8e7a266d



Statistics (10/2018)

What 29,985,328 Transactions Say About the State of Smart Contracts on Ethereum

Smart Contract	Transaction Count	Address	Category
EtherDelta	10354398	0x8d12a197cb00d4747a1fe03395095ce2a5cc6819	Decentralized Exchange
IDEX_1	4590376	0x2a0c0dbecc7e4d658f48e01e3fa353f44050c208	Decentralized Exchange
EOS Token	2952885	0x86fa049857e0209aa7d9e616f7eb3b3b78ecfdb0	ICO
CryptoKitties Smart Contract	2568983	0x06012c8cf97bead5deae237070f9587f8e7a266d	Collectible Token
Tron Token	1967331	0xf230b790e05390fc8295f4d3f60332c93bed42e2	ICO
Poloniex_3	1720771	0x209c4784ab1e8183cf58ca33cb740efbf3fc18ef	Centralized Exchange
Bittrex_2	1527197	0xe94b04a0fed112f3664e45adb2b8915693dd5ff3	Centralized Exchange
Bittrex Wallet	1501350	0xa3c1e324ca1ce40db73ed6026c4a177f099b5770	Centralized Exchange
BTCM	1451763	0x03df4c372a29376d2c8df33a1b5f001cd8d68b0e	ICO
OmiseGO	1350274	0xd26114cd6ee289accf82350c8d8487fedb8a0c07	ICO

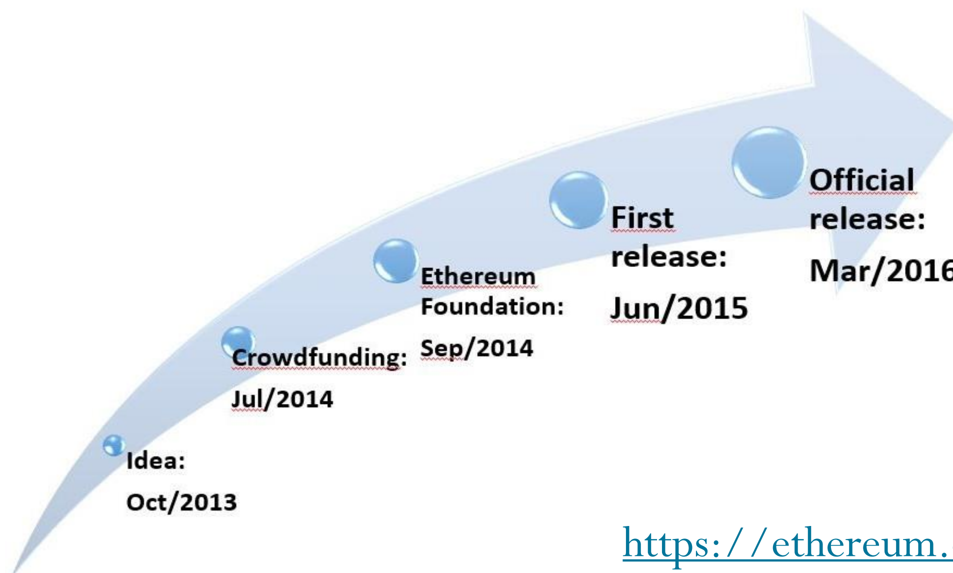

Top 10 Smart Contracts on Ethereum
Decentralized Exchange
20.0%
ICOs (ERC-20)
40.0%
Centralized Exchanges
30.0%
Collectible tokens (ERC-721)
10.0%

Ethereum

Portland State
Computer Science

History of Ethereum - Timeline

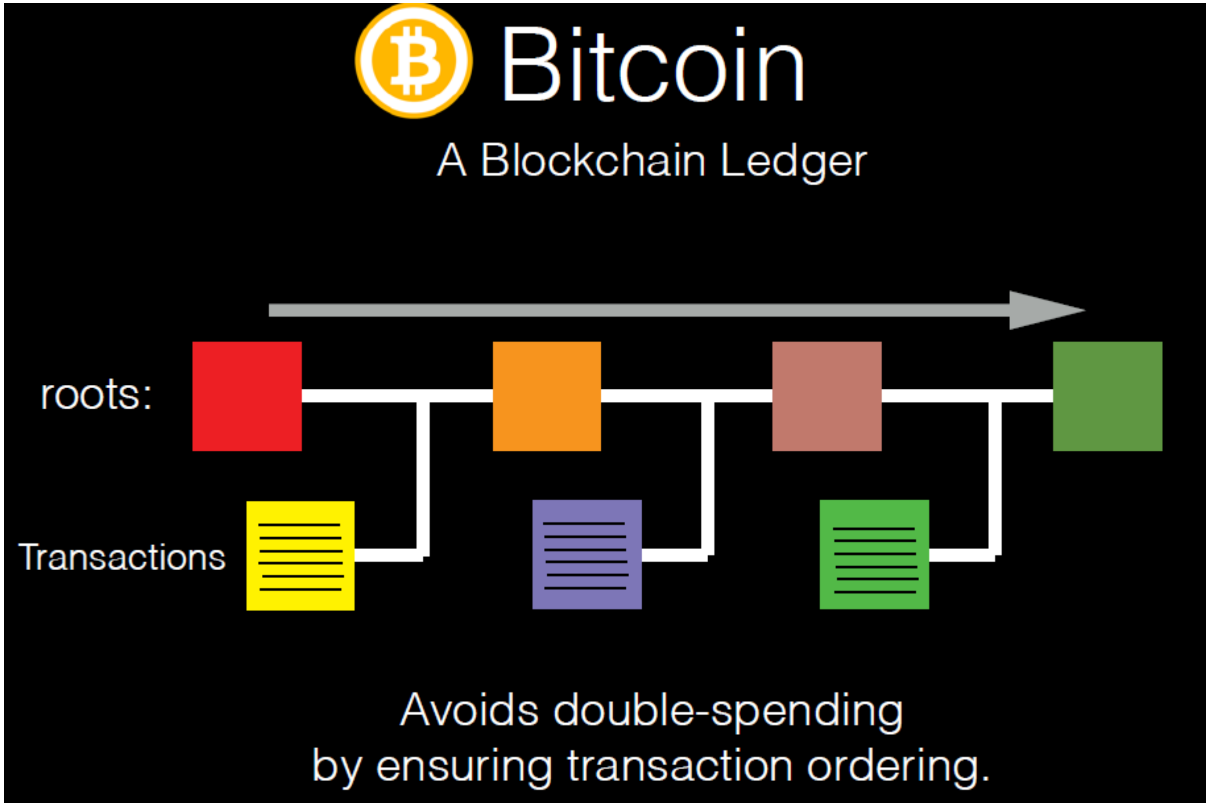
- Proposed by Vitalik Buterin in 2013 to build decentralized applications
  - Deployed in 2016
  - First blockchain to support smart contracts
  - Has a notion of storing actual state (e.g. account balance) vs. Bitcoin's UTXO where one must scan blockchain to find out balance



https://ethereum.org

Why not Bitcoin?

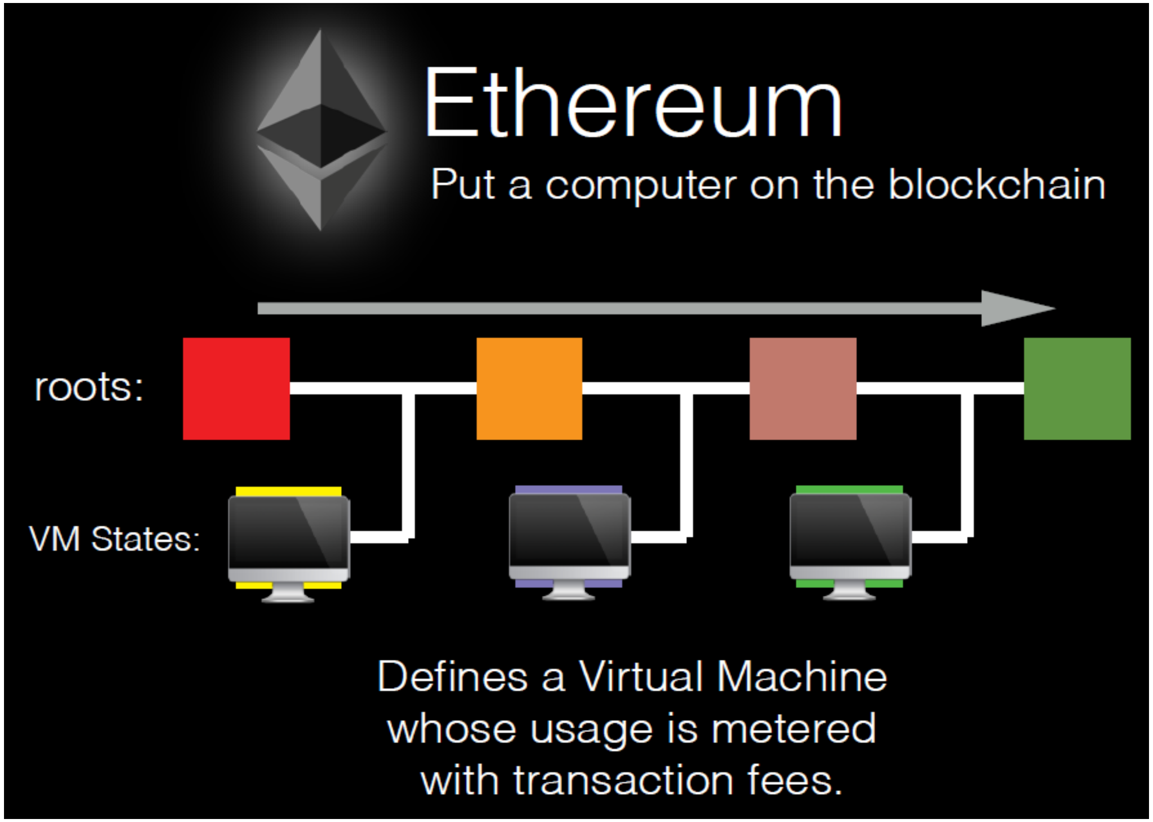
- Bitcoin with simple stack based scripts for validating properties of transfers/assets (UTXOs)



Source: MetaMask's Dan Finlay, https://github.com/MetaMask/IPFS-Ethereum-Hackathon/tree/master/slides/01_DanFinlay_intro_to_ethereum_blockchains

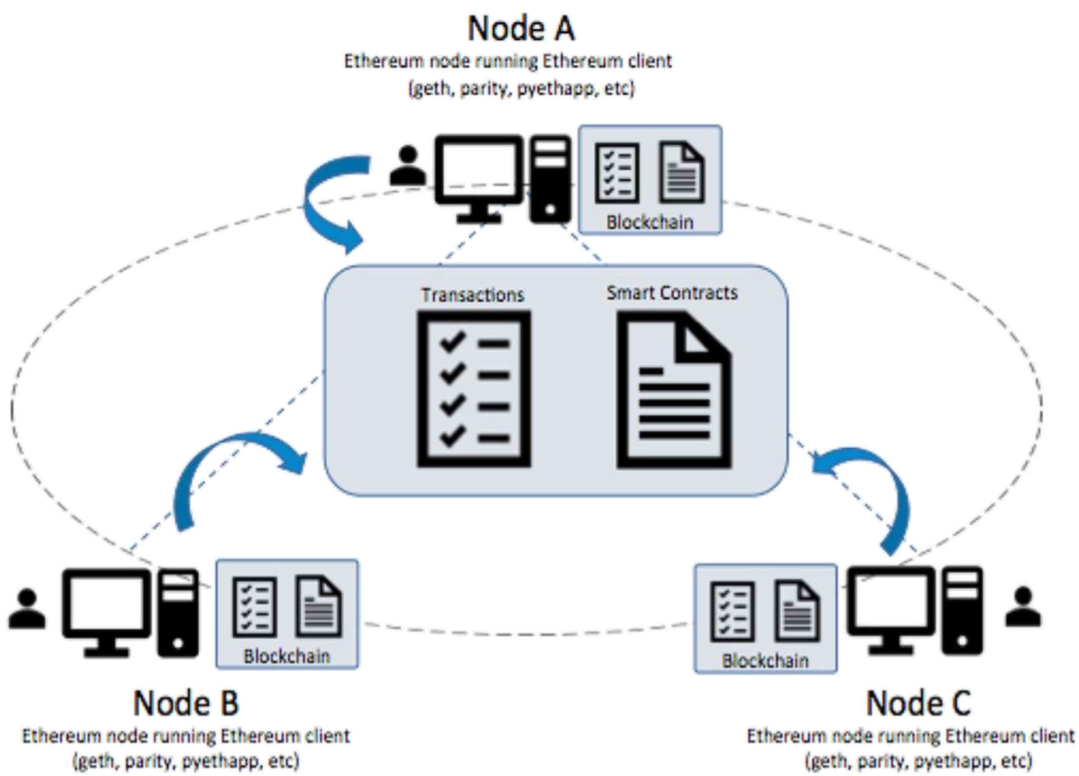
Ethereum VM (EVM)

- Turing complete run-time for computation
  - Requires a much higher transaction rate than Bitcoin as a result
  - Also requires a state-based approach for validating transactions (versus a history-based one of replaying transactions)
  - Done by adding storage to the blockchain (similar to git commits)



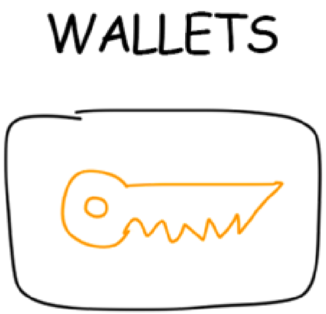
Code execution

- Every (full) node on the blockchain processes every transaction and stores entire copy of blockchain
  - e.g. the state of all contracts and accounts
  - Contract executions are redundantly performed across all nodes
- Node implemented using a secure, memory-safe language
  - e.g. Rust or Go



Accounts

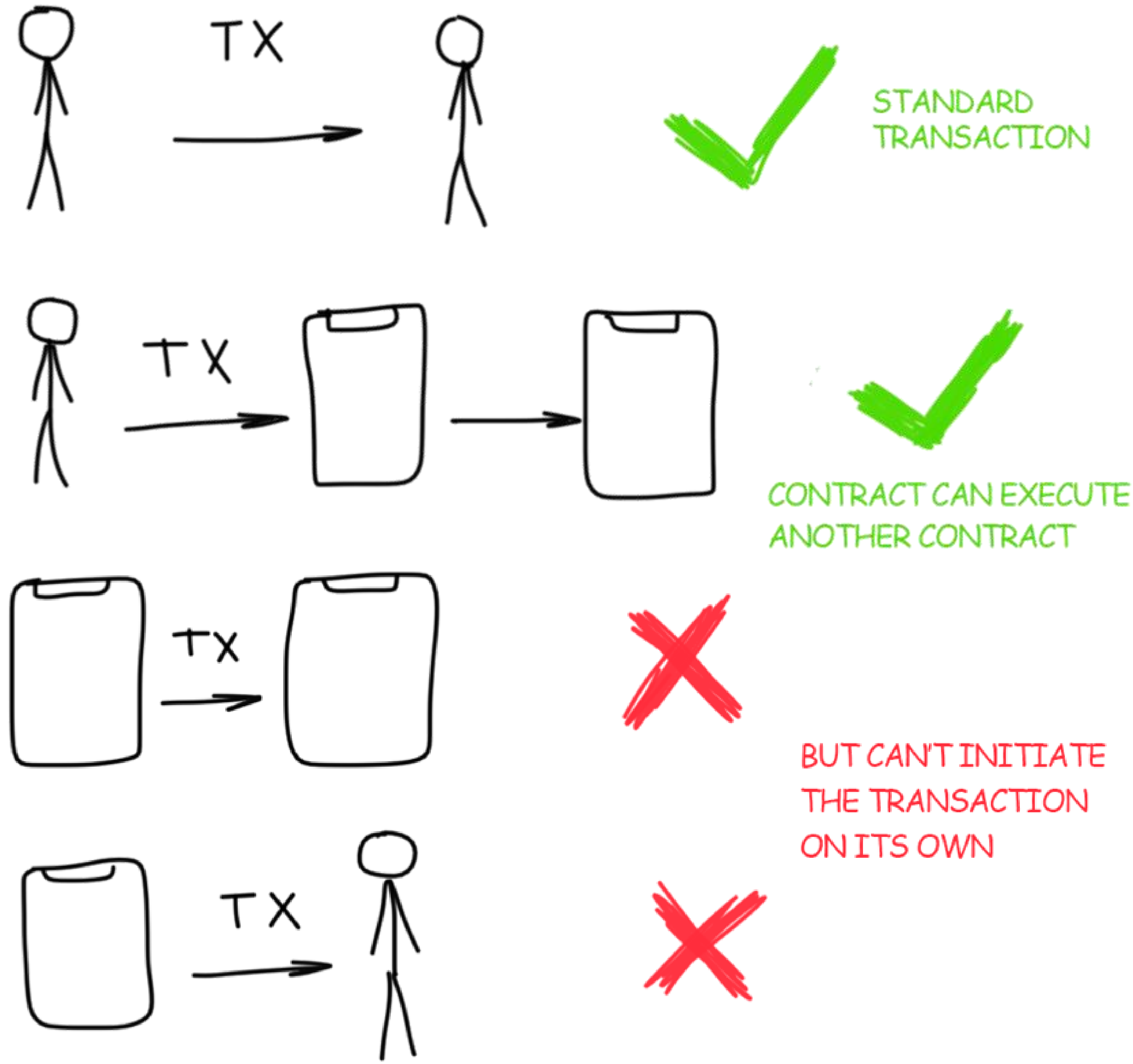
- Wallets (similar to Bitcoin)
  - a.k.a. Individual user accounts, Externally Owned Account (EOA)
  - Wallet address managed with private keys
    - Can keep a balance of ETH and send and receive it
    - Can create transactions to call code
- Smart contract account
  - Can do everything a wallet can do
    - Can hold funds (i.e. keep a balance of ETH)
    - Can send currency (ETH) to other accounts
  - But can also contain code
    - Code of smart contract stored publicly on blockchain
    - Can contain functions that may be called from wallet accounts
    - Can contain functions that may be called from other smart contracts
  - And can also store data
    - Persistent storage on blockchain that is both readable and writeable (not just UTXO)



Smart contracts can not…

- Create ETH (only mined blocks can do so)
- Query an external API (since one can not guarantee same result to all)
- Sleep (no halting of blockchain)
- Can not asynchronously call other contracts

Modes of use



TX
STANDARD TRANSACTION
TX
CONTRACT CAN EXECUTE ANOTHER CONTRACT
TX
BUT CAN'T INITIATE THE TRANSACTION ON ITS OWN
TX

Account addresses

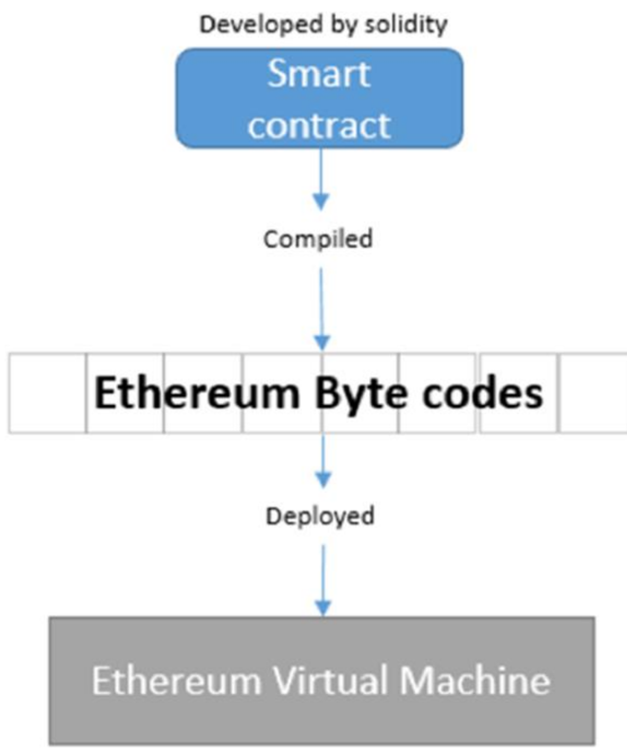
- Wallet addresses and smart-contract addresses share same format
- Private key similar to Bitcoin
  - ECDSA to digitally sign hashes of transactions/messages
- Public key (mapped directly from private)
  - Last 40 characters of the keccak-256 hash of public key

`0xA6fA5e50da698F6E4128994a4c1ED345E98Df50`

  - Note case-sensitivity
    - Done as a built-in checksum for addresses (more later)
    - https://ethsum.netlify.com/

EVM bytecode

- Each node has an EVM that executes EVM bytecode
  - Contracts compile down from higher-level language into EVM bytecode
  - Contracts typically small ~100 LoC
  - Contract compiled and executed
  - Contract can store and modify state on EVM



Example

```
contract Greetings {
    string greeting;
    function Greetings (string _greeting) public {
        greeting = _greeting;
    }

    /* main function */
    function greet() constant returns (string) {
        return greeting;
    }
}
```

What you write

What others see on the blockchain

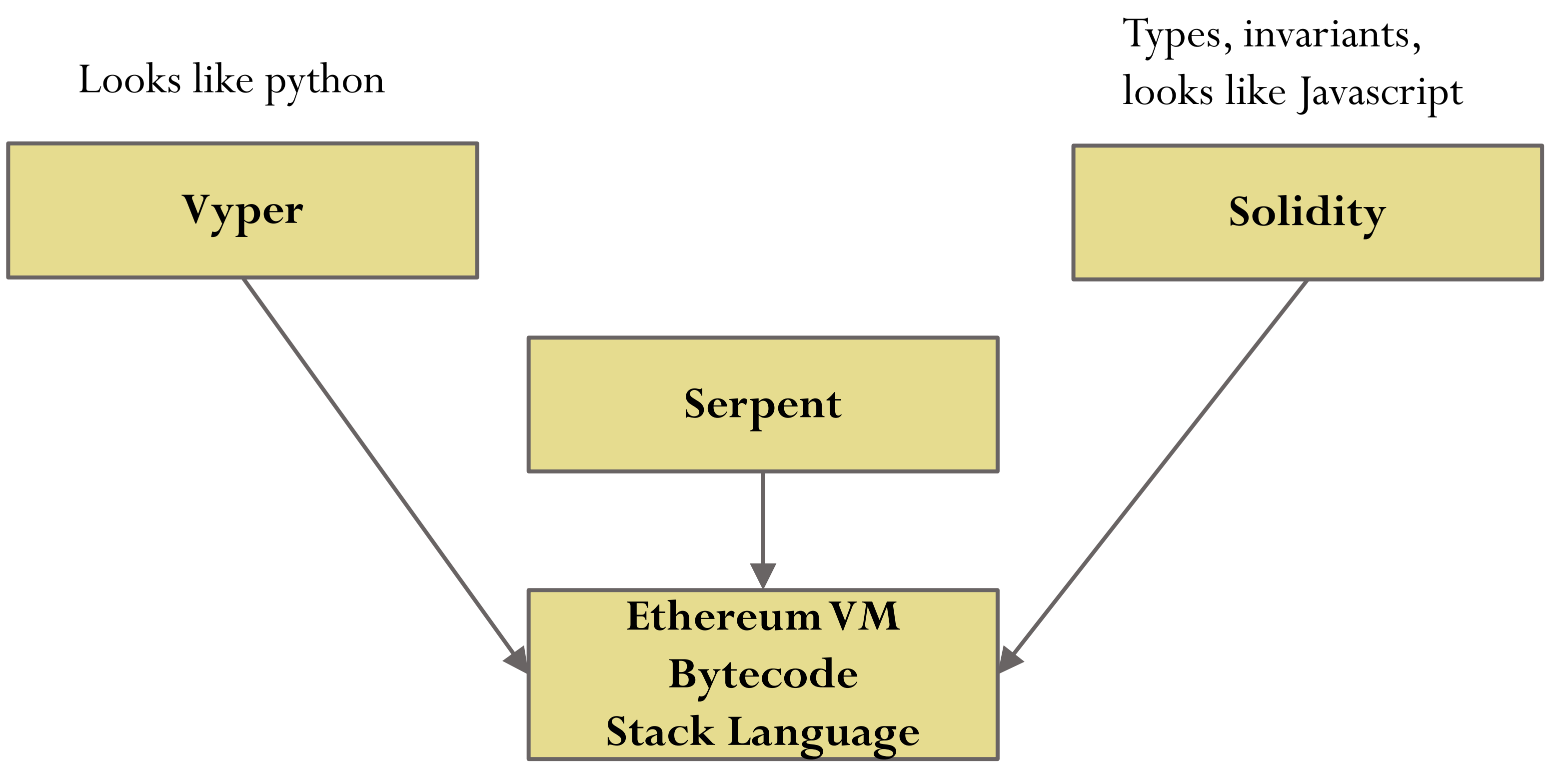
6060604052604051610250380
38061025083398101604052 8..
……

PUSH 60
PUSH 40
MSTORE
PUSH 0
CALLDATALOAD
…..

What people get from the disassembler

Multiple language alternatives

- Like LLVM, multiple languages can produce EVM bytecode
  - Must be aware of what a language provides to determine which to use
  - Initially Serpent
  - But now, most are done in Solidity
  - Vyper to potentially replace Solidity? (More later in course)



Issue

- Halting problem
  - What if I have an infinite loop in my smart contract?
  - e.g. what if a malicious account sends my EVM this program as part of a DoS attack?

```
uint i = 1;
while (i++ > 0) {
    donothing();
}
```

  - Can one tell whether or not a program will run infinitely a priori?
  - How can one limit this behavior?

Solution #1

- No loops
  - More later…

Solution #2: Gas

- Force user to supply currency (ETH) in order to execute programs and store data on EVM
  - User calling smart contract must supply $ from wallet to execute!
  - Fee charged per computational step (called "gas")
  - Fee charged per operation taking up storage
- Limits resource consumption to what sender pays for
  - Fees above paid to miners
  - Transactions specified with Gas Limit and Gas Price to estimate how much computation will cost
  - Wallet can automatically estimate both when transaction submitted
  - Creates an incentive not to use the blockchain for computation and storage that can be done off chain

Example gas charges

Operation	Gas	GasCost
PUSH1	111741	3
PUSH1	111738	3
MSTORE	111726	12
CALLDATASIZE	111724	2
ISZERO	111721	3
PUSH2	111718	3
JUMPI	111708	10

Sender pays for gas

- `gasprice`: amount of ether per unit gas
  - https://ethgasstation.info/
- `gaslimit` or `startgas`: maximum gas consumable for transaction
  - What if `gaslimit` is less than needed?
    - Out of gas exception, revert the state as if the TX has never happened
    - Sender still pays all the gas
- `transaction_fee:` total cost of transaction
  - `gasprice * consumed_gas`
- `Block Gas Limit`
  - Similar to block size limit in Bitcoin
  - Total gas spent by all transactions in a block < `Block Gas Limit`

Ethereum currency denominations

- Requires fine-grained currency
- Ethereum currency units
  - http://eth-converter.com/extended-converter.html

Multiplier	Name
10^0	Wei
10^{12}	Szabo
10^{15}	Finney
10^{18}	Ether

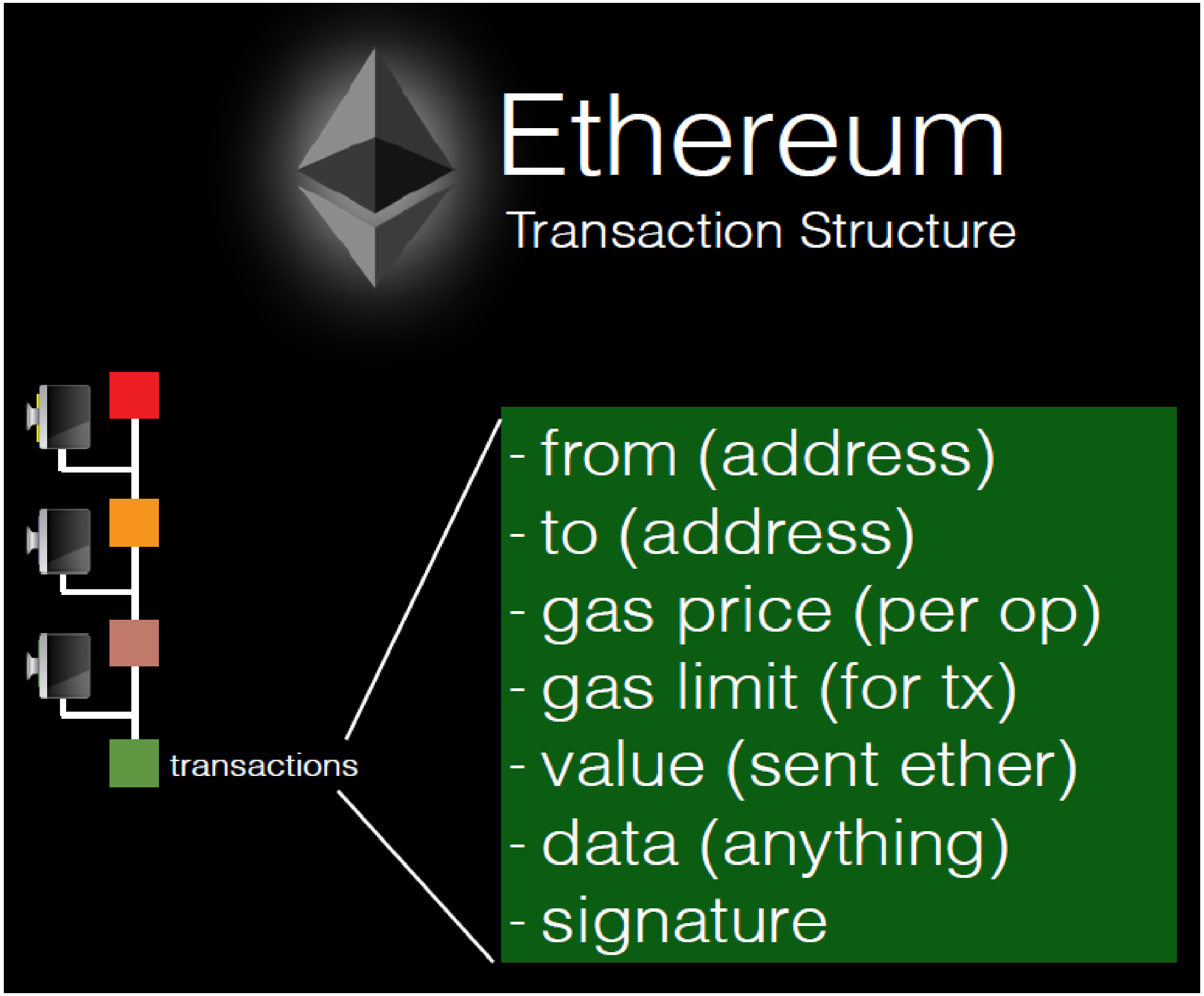
  - Wei (Dai) – author of b-money paper
  - (Nick) Szabo – BitGold
  - (Hal) Finney - RPOW

Wei	1000000000000000000
Kwei, Ada, Femtoether	1000000000000000
Mwei, Babbage, Picoether	1000000000000
Gwei, Shannon, Nanoether, Nano	1000000000
Szabo, Microether,Micro	1000000
Finney, Milliether,Milli	1000
Ether	1
Kether, Grand,Einstein	0.001
Mether	0.000001
Gether	0.000000001
Tether	0.000000000001
USD(at 158.412$ p/ ether)	158.412
EUR(at 141.230€ p/ ether)	141.230

Transactions

- Request to modify the state of the blockchain
  - Signed by originating account (either wallet or smart-contract)
- Can be of 3 types
  - Send value from one account to another (e.g. same as Bitcoin)
  - Create a smart-contract on blockchain
  - Execute smart contract code stored on blockchain

- Transactions include

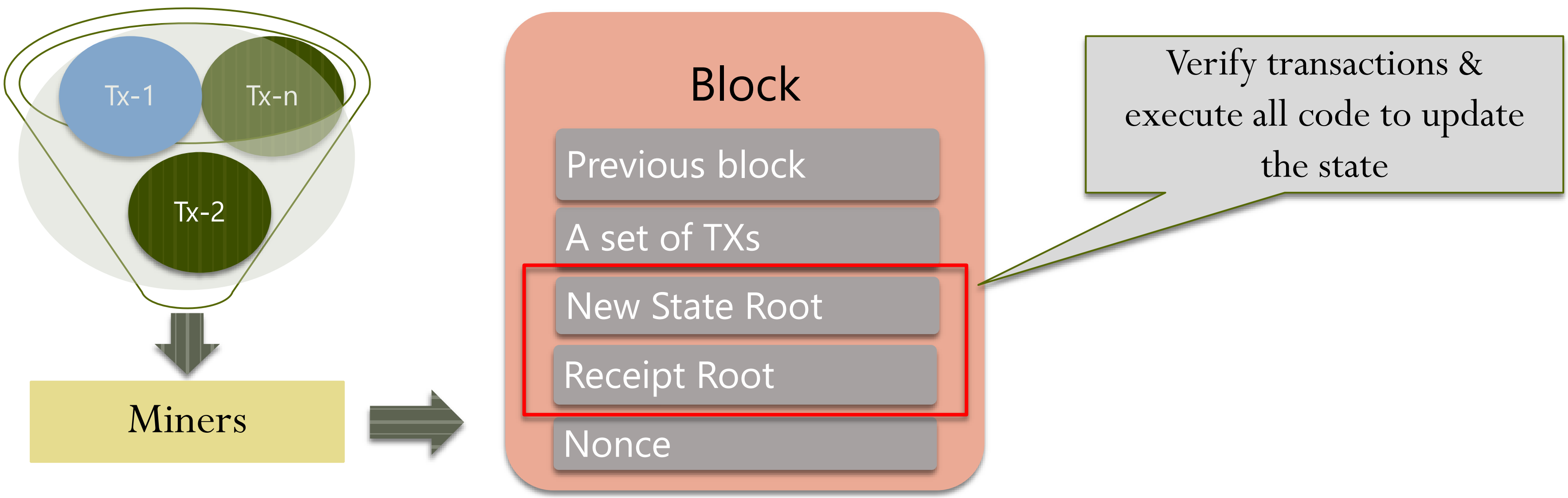


- As well as nonce (to prevent replay)

Source: MetaMask's Dan Finlay, https://github.com/MetaMask/IPFS-Ethereum-Hackathon/tree/master/slides/01_DanFinlay_intro_to_ethereum_blockchains

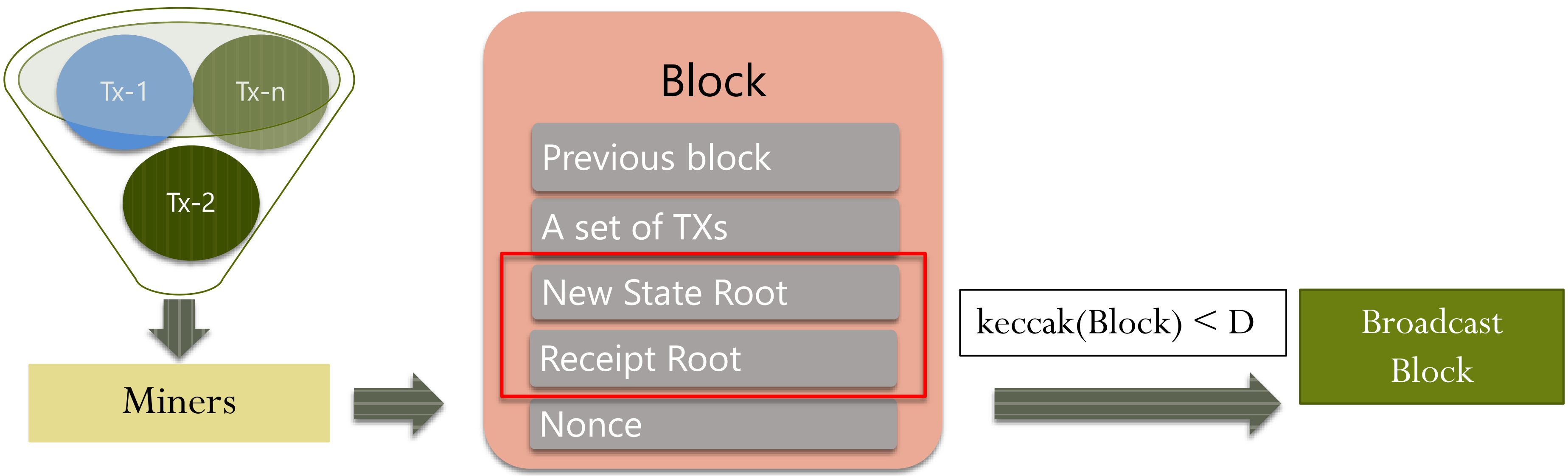
Blocks

- Ethereum uses Merkle-Patricia tries
  - 3-branch tree vs. Merkle's 2-branches
  - Flatter, wider trees requiring less hashes to validate
- Bitcoin uses SHA-256, Ethereum Keccak-256 (SHA-3) for hashes



Mining details

- Proof-of-work algorithm EthHash also uses Keccak
- Difficulty adjusted every block (instead of every 2 weeks for BTC)



EthHash and ASIC mining

- EthHash mining algorithm initially to discourage ASIC miners
  - I/O limited PoW
  - But, eventually an ASIC miner implemented
- Leads to…
  - Threat to "brick" ASIC mining via algorithm

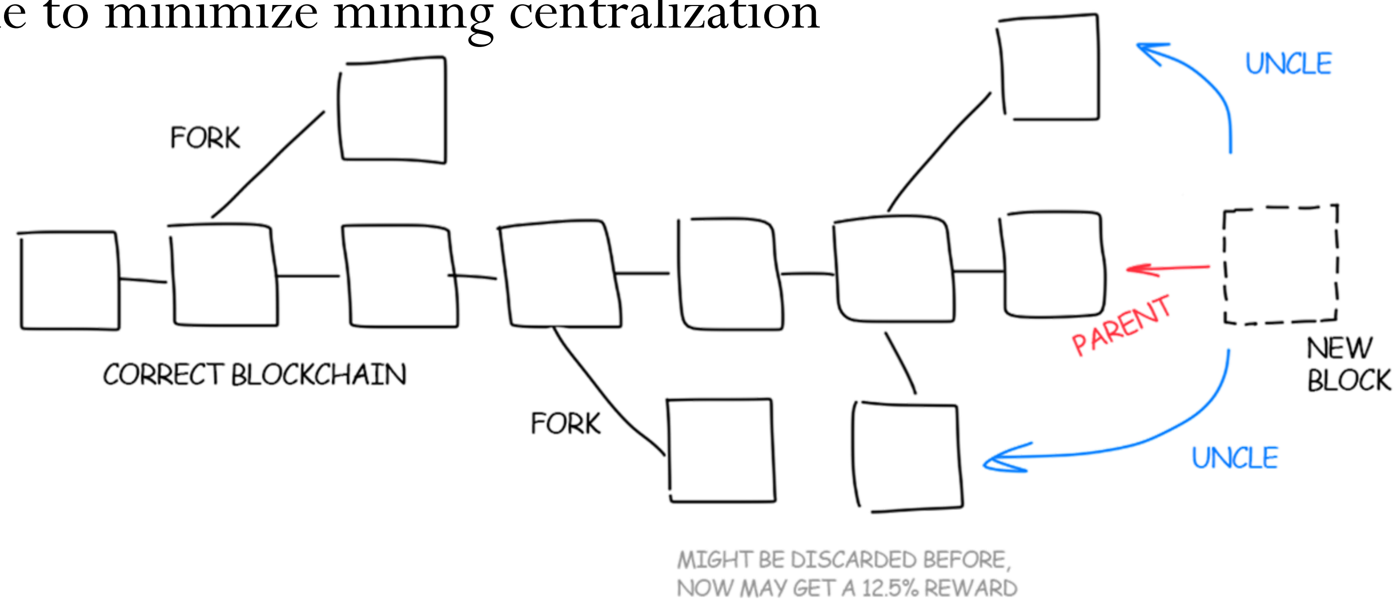
    Ethereum's Istanbul Upgrade: Understanding ProgPoW

    A handy guide to the next ETH hard fork expected in October 2019

    - Programmatic Proof-of-Work to democratize mining away from ASIC farms
    - Algorithm changes wreck ASIC investment
  - Threat of migration to Proof-of-Stake
    - Remove computation altogether

Mining details (current)

- Blocks faster than BTC (block time ~12 sec)
- Block size – (miner controlled)
- Block reward variable (inflationary) ~5 ETH
- Moved from longest-chain to a different reward protocol (GHOST)
  - Miners can make a bit more by including blocks (1/32 of an ETH each) up to maximum of two for work on side-chains eventually discarded (uncles)
  - Done to minimize mining centralization



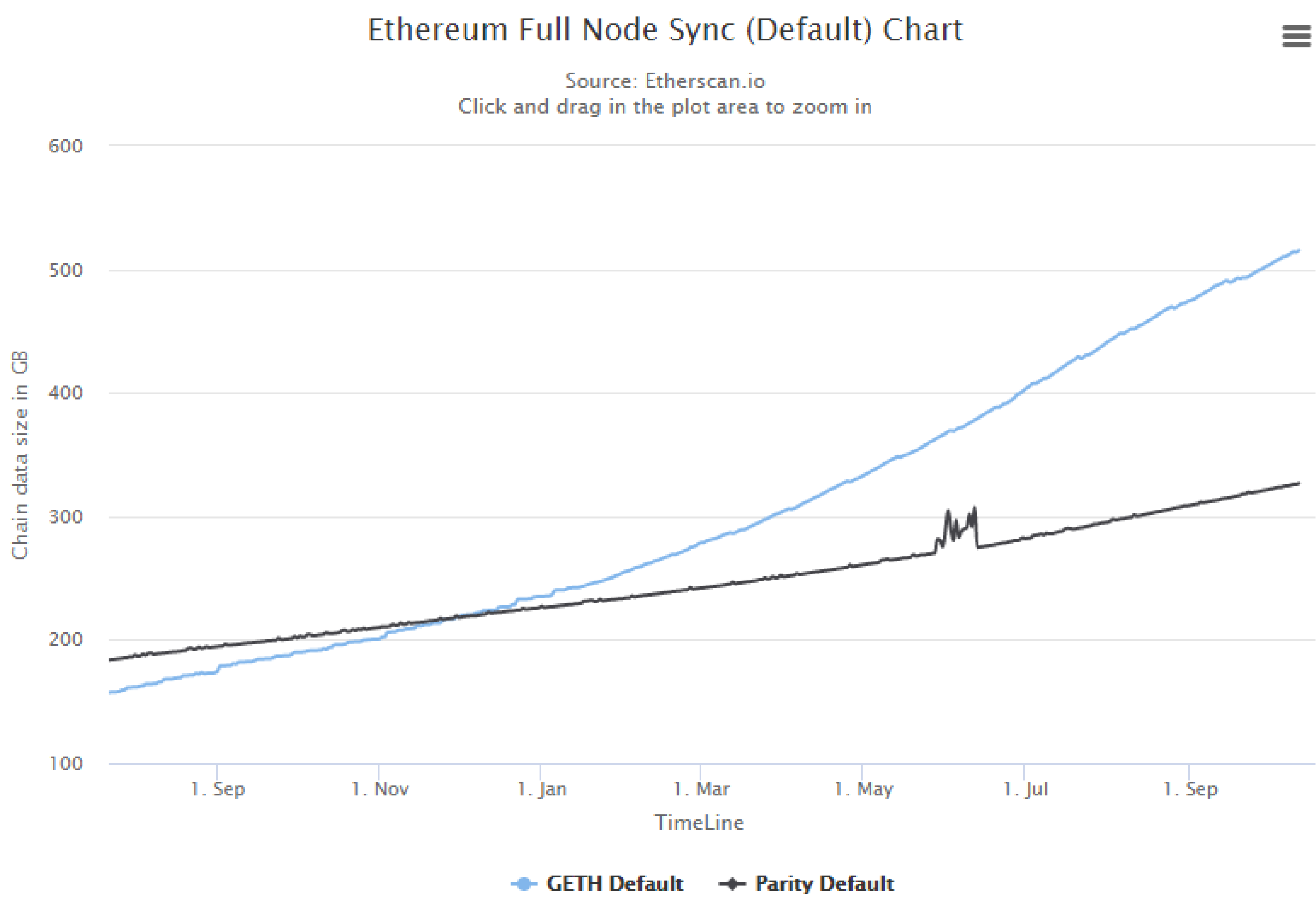
Ethereum size & archival nodes

- Archive node stores entire chain and its transactions ~4TB (4/2020)
  - Very few "archival" nodes in operation (16,650 total archival and fast nodes)
- Many archival nodes run by companies (e.g. Infura) due to resource constraints and management costs
  - Centralized, single-point of failure

Total	16650 (100%)
United States	6056 (36.37%)
China	2256 (13.55%)
Canada	919 (5.52%)
Germany	901 (5.41%)
Russian Federation	807 (4.85%)
United Kingdom	588 (3.53%)
Korea, Republic of	443 (2.66%)
Netherlands	437 (2.62%)
France	379 (2.28%)
Ukraine	255 (1.53%)

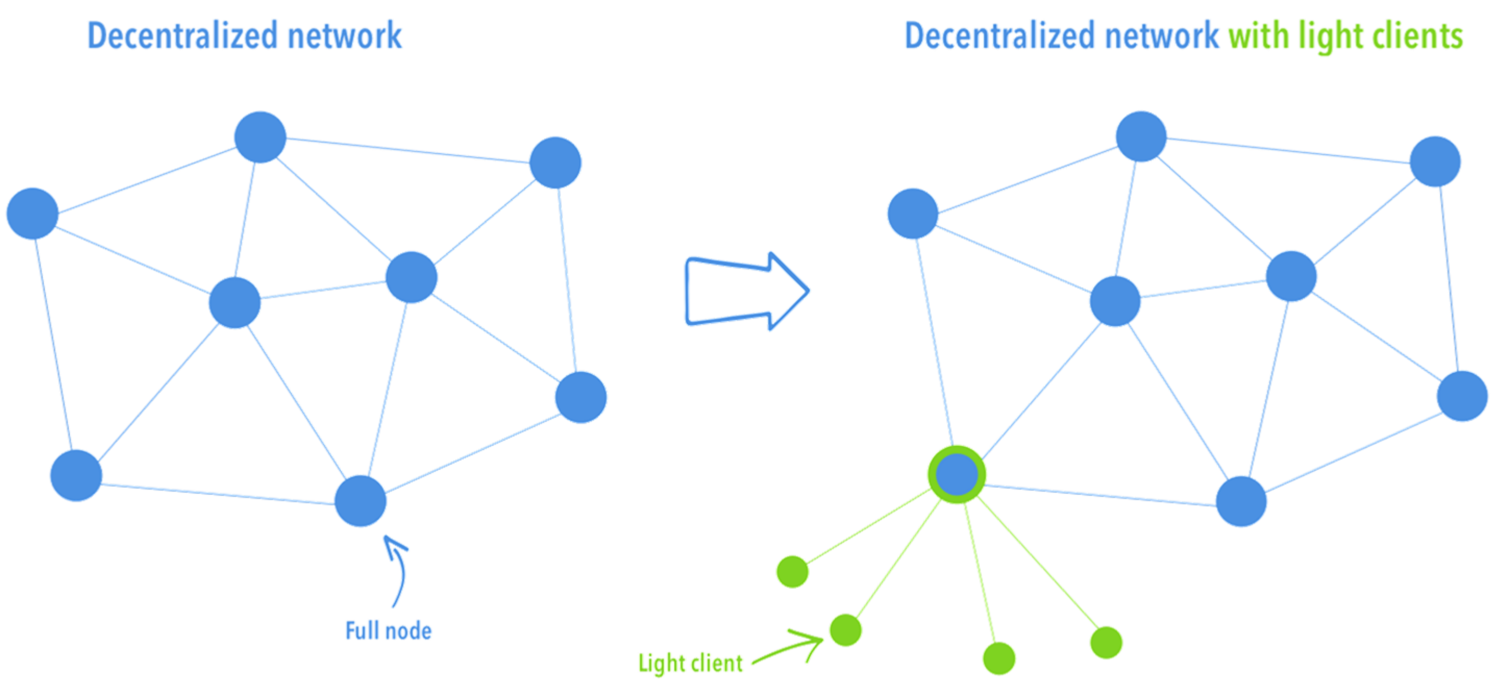
Ethereum full nodes

- Full node ~500GB (10/2020)
  - Discard unnecessary state
  - Still requires a sizeable machine and network connection to run
  - Lab 5.1



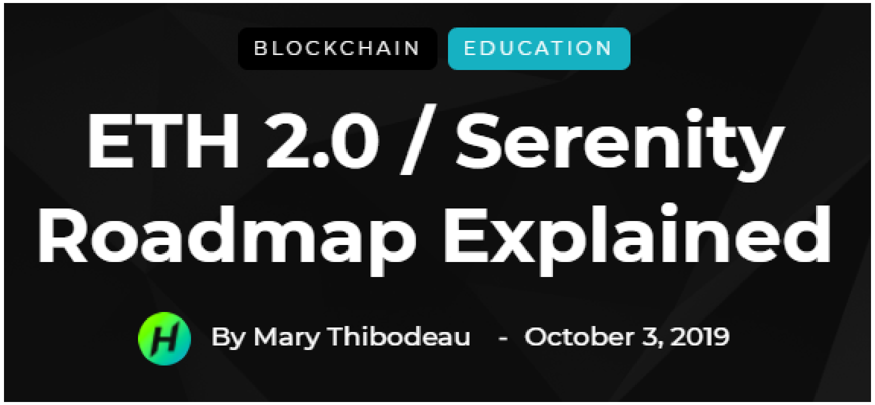
Ethereum light nodes

- Light Node (or light client) that connects to full-nodes
  - Contains all block headers (e.g. Merkle-Patricia roots) (~100MB of storage to run, 7/2018)
  - Can not execute write transactions as full-nodes do
  - Pulls block data and submits requests to a full-node when necessary
    - Requires more network resources, but less CPU/storage resources
  - Implemented and deployed in 2018 for scalability



Scheduled improvements

- ETH 2.0 (Serenity, Casper Proof-of-Stake)
  - Put security in the hands of those with the most to lose if security broken (e.g. stake-holders)
    - Beacon chain with PoS to run alongside main PoW chain
    - Eventual switchover from PoW chain if successful
  - Support for sharding to obtain scalability
    - Solve scalability via side blockchains whose state is hashed and committed to main chain periodically
  - https://media.consensys.net/the-roadmap-to-serenity-bc25d5807268
  - https://hedgetrade.com/eth-2-0-serenity-roadmap-explained/



A look at DApps on Ethereum