How Ethereum Works

John Long,
Head Instructor for Blockchain at Davis

Overview

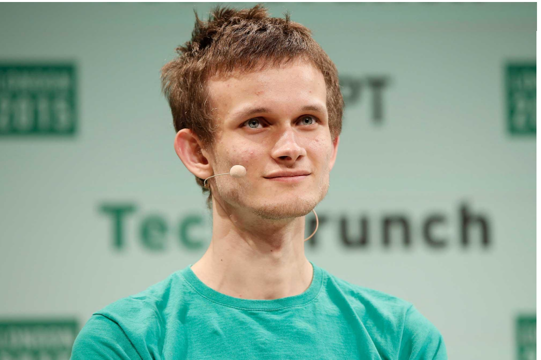
- Historical Context: Why Ethereum?
- The tech you already know (and then some)
- The “secret sauce”: What Makes Ethereum, *Ethereum*
- Shortcomings and Concluding remarks

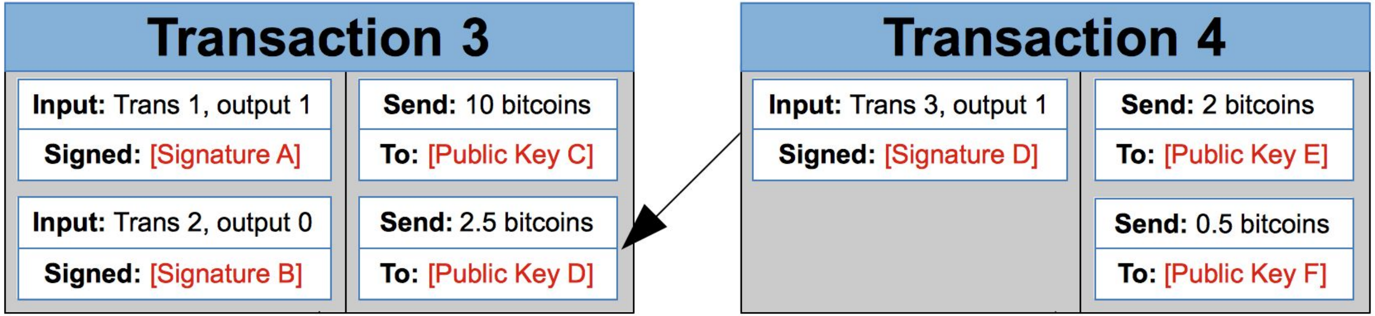
Early Cryptocurrency History

- Bitcoin launched in 2008
  - Proof that a digital, decentralized, consensus IS FEASIBLE
- Altcoin Flood
  - Alternative cryptocurrencies attempted to address issues that Bitcoin had such as…
  - Incentivization: Shouldn’t nodes be rewarded as well?
  - Centralization: more and more mining power hoarded by single individuals/groups
  - Security: What if SHA-256 is compromised?

Birth of Ethereum

- Launched in 2014 by Vitalik Buterin et al.
- Had the idea of leveraging blockchain technology for decentralized applications
  - Apps that do not need to rely on a single provider for access!
- Recognized the need for a more powerful scripting language
  - Bitcoin has a language, but is NOT TURING COMPLETE
  - Designed for complex transactions





Transaction 3

Input: Trans 1, output 1
Signed: [Signature A]

Input: Trans 2, output 0
Signed: [Signature B]

Send: 10 bitcoins
To: [Public Key C]

Send: 2.5 bitcoins
To: [Public Key D]

Transaction 4

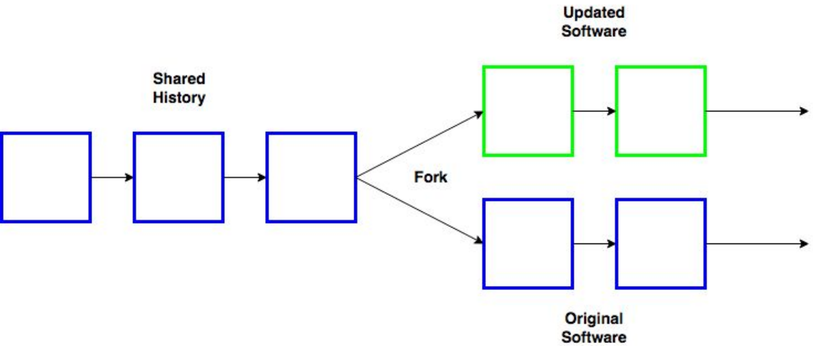
Input: Trans 3, output 1
Signed: [Signature D]

Send: 2 bitcoins
To: [Public Key E]

Send: 0.5 bitcoins
To: [Public Key F]

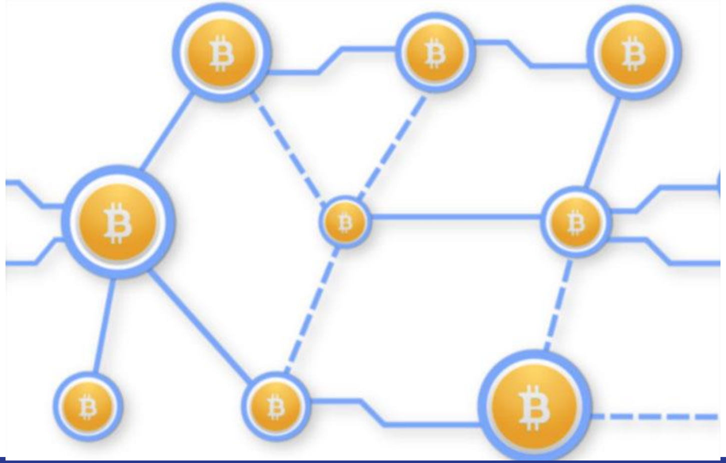
Ethereum Milestones

- Ethereum itself has a rich history
  - Frontier and Homestead - Initial release, making sure everything worked properly
  - Byzantium and Constantinople
  - “Serenity”/Ethereum 2.0 - Focus of Discussion today
- Some Updates require hard forks
  - Protocol Upgrades
  - Blockchain based on new protocol will no longer be compatible with older nodes
  - Everybody has to upgrade ASAP to do anything with the network



Similarities to Bitcoin

- Uses a Blockchain with the same hash linking technique found in Bitcoin
  - Still possible to have orphaned blocks/parallel chains
- Incentive exists for people to mine, expend computational power
- Nodes hold copies of the chain, responsible for verification and distribution of data



What Makes Ethereum *Ethereum*

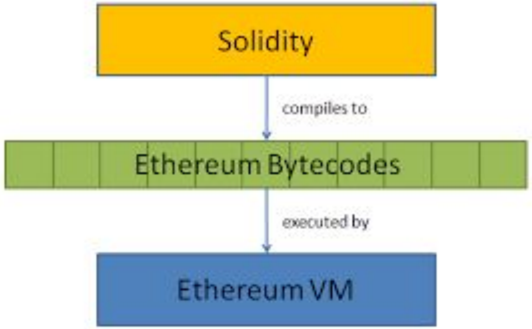
- The **Smart Contract**
- Traditional Contract is defined as…
  - Agreement between at least two people
  - Recognized by a trusted third party
- Problems
  - What if the third party is compromised?
  - Will any degree of vetting be enough?
- Idea by Nick Szabo (legal scholar/CS)
  - Have a contract that is enforced by a machine
  - Ex: vending machine
- One Step Further: Have contracts be enforced by Blockchain tech + Turing Completeness

How Smart Contracts Work in Ethereum

- Developer compiles a contract into something the nodes on the blockchain network understand (Solidity, Vyper, etc. -> EVM Opcodes)
- Compiled code piggybacks on transaction
- Nodes recognize contract is NOT a transaction
- Smart contract is given its own address (like a user!)
- All nodes WILL EXECUTE the contract!
    - Stay tuned on how Ethereum Nodes differ from Bitcoin Nodes

Ethereum Nodes

- Hold copies of the Ethereum blockchain but also execute code!
- Each node possesses the “EVM” or Ethereum Virtual Machine
  - Stack-based
  - Incredibly limited set of opcodes (opcodes usually a byte, <255)
  - Sandboxed, code run on EVM CANNOT interfere with host node
- Concurrency vs. Parallelism
  - Every node runs the contract when it goes through, must reach current contract “state”
    - Great for reliability, bad for performance
  - DO NOT CONFUSE WITH PARALLELISM!
    - Can’t take computationally expensive payload and split it

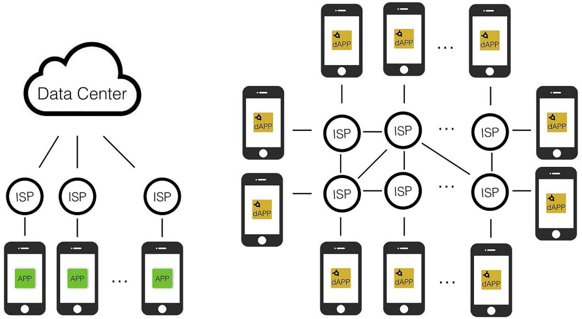


Gas and Ether

- Ether - name of 1 unit of Ethereum crypto
- Gas - unit of computation
  - Purchased via Ether (small fraction of Ether, a “gwei”)
  - Each smart contract operation costs Gas
  - Can have static vs dynamic costs (Hashing data vs math)
- Want to separate utility from market volatility
- Need gas to power a car, but you pay for gas with money

Decentralized Applications

- Smart Contracts have own address, invoked via transactions
- Smart Contracts can invoke other smart contracts
- Can have multiple smart contracts work together
  - “Spoke and Hub” model as an example
- Allows for Decentralized Applications
  - Run entirely on the Ethereum blockchain, does not require a single server



Apps

dApps

Decentralized Applications (cont.)

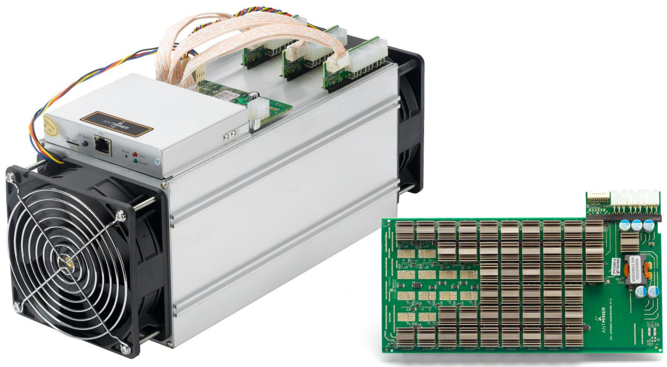
- Decentralized Autonomous Organizations (DAOs)
- Have certain organizational rules codified as smart contracts
  - Ex: Managers get paid via smart contract
  - Ex: payment conditional on some vote
- Better integration with tokens/voting

Ethereum Block Size

- NOT FIXED, dependent on “Gas Limit”: how much gas all the transactions/data in the block contains
  - Around 10 million Gas right now
- Considering that certain opcodes/instructions are costlier than others, allows for variable block sizes
- Also, miners collectively vote on what the gas limit should be
  - Reflects the current mining capabilities

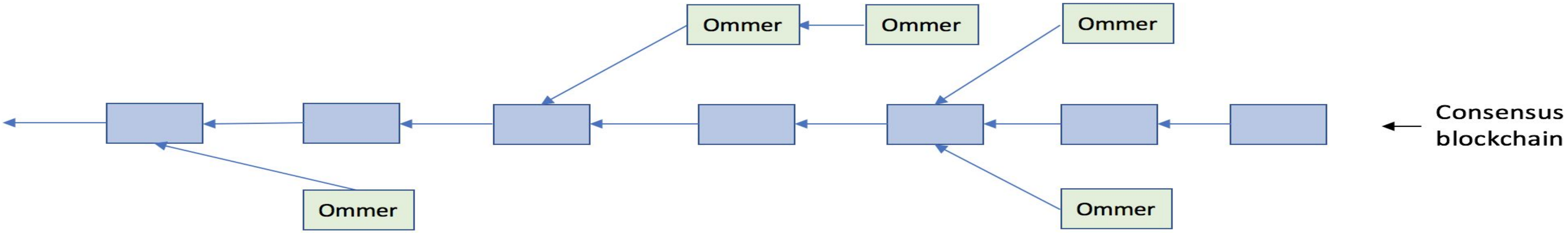
Ethereum Mining

- Currently PoW, but will transition to PoS
- SHA-3 related, but very memory intensive
  - Check out Dagger-Hashimoto if interested, designed to deter ASIC implementation
- “Difficulty Bombs”
  - Implemented to deter centralization of power
  - After fixed number of blocks, the difficulty will skyrocket for miners
  - Despite this, ASIC miner does exist, but the profit margin is too narrow for much centralization



Ethereum "Uncles/Ommers"

- Highly possible that two valid blocks will be submitted simultaneously
- There are now two valid chains
  - Longest chain will become the valid one and the other is orphaned
- Traditionally, miners were not rewarded for such things
- In Ethereum PoW orphans are known as "Uncles/Ommers"
- Reduced reward is still given in an effort to deter BTC-like pooling



Shortcomings

- Scalability Trilemma (Also from Buterin)
  - Decentralization, Security, Scalability (you can only have two out of the three)
- Speed
  - Ethereum is not designed for high-performance applications
- Immutability
  - The DAO incident
  - Contracts can only be destroyed, not replaced
- Explosive Blockchain Growth
  - Variable block size + fast generation + smart contract data ON TOP of traditional transactions allows for massive Blockchain size
  - Deters people from creating nodes
- Oracle Problem
  - Smart contracts can't get data about the outside world

The DAO Incident

- How a DAO is usually started
  - Smart contracts created with initial logic for taking Ether in exchange for tokens
  - Tokens represent voting rights for organization
  - Period of time where people buy tokens
  - Then the DAO begins operation
- “The DAO” had a flaw in exchange logic, hacker siphoned away millions of dollars worth of Ether
- Users split
  - Can leave things “as-is”
  - Attempt to undo the damage through fork
  - No way to easily undo the damage

Ethereum 2.0: What’s Next

- Codename “Serenity” update
- Full shift to Proof-of-Stake
  - People “lock up” Ether to get the chance to validate transactions
  - False transactions can cause you to lose Ether
- Shard Chains
  - Redistribute parts of main chain to other nodes for faster processing
  - Tantalizing hints of parallelism!
- Beacon Chain
  - Helps synchronize shards, allows them to reach consensus

Thank You!