ETHEREUM AND SMART CONTRACTS: ENABLING A DECENTRALIZED FUTURE

Brian Ho
Gillian Chu

LECTURE OVERVIEW

1. SMART CONTRACTS
2. ETHEREUM
3. EVM (ETHEREUM VIRTUAL MACHINE)
4. USE CASES
5. STATE OF THE ECOSYSTEM

1 SMART CONTRACTS

BITCOIN REVIEW

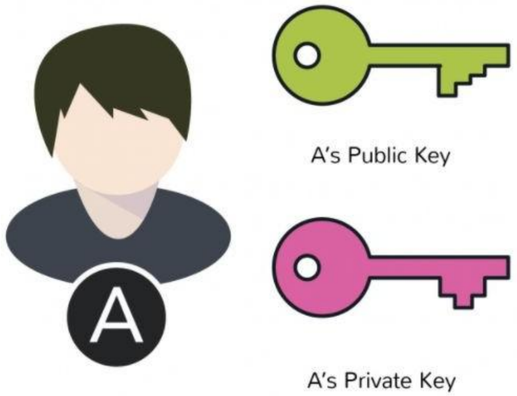
PROPERTIES OF BITCOIN

... but first, a question:

What makes Bitcoin so special?

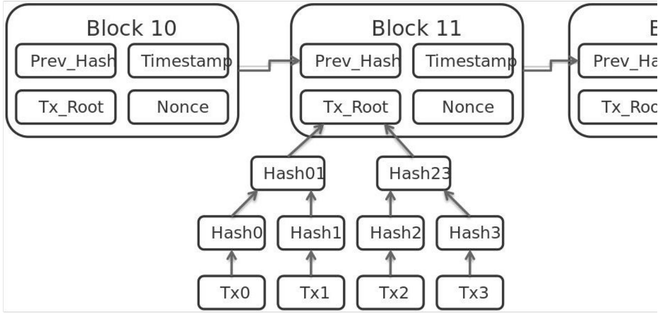
A DISTRIBUTED NETWORK

BITCOIN'S BARE BONES



Cryptographic Identities

Consensus Protocol

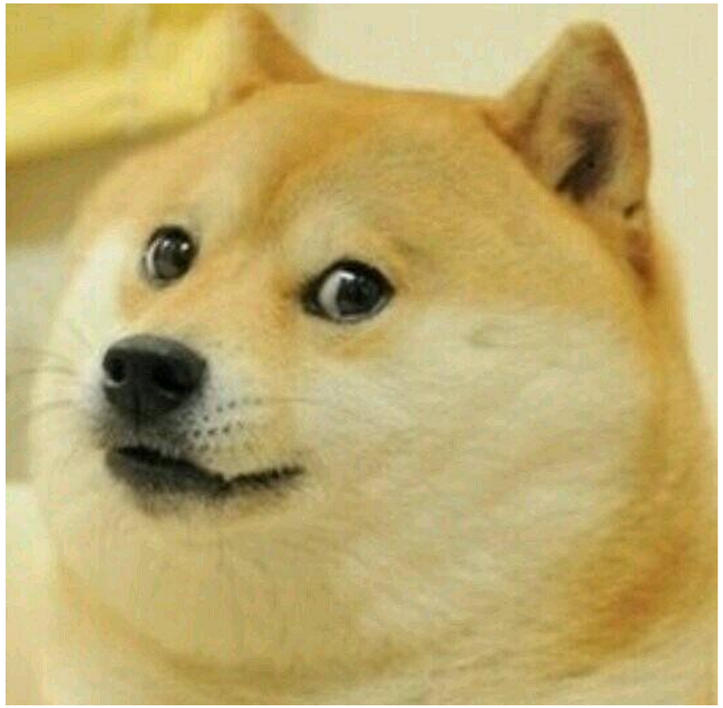


Blockchain

A DISTRUBITED NETWORK

TRANSFERRABLE BENEFITS OF BITCOIN

- Pseudonymous, cryptographic identities allow for accountability
- Democratic decisions made through consensus protocol that **doesn't require trust**
- Immutable ledger of truth
- Uncensorable, cannot be controlled by any one party
- Distributed: no central point of failure



SMART CONTRACTS

CONTRACTS

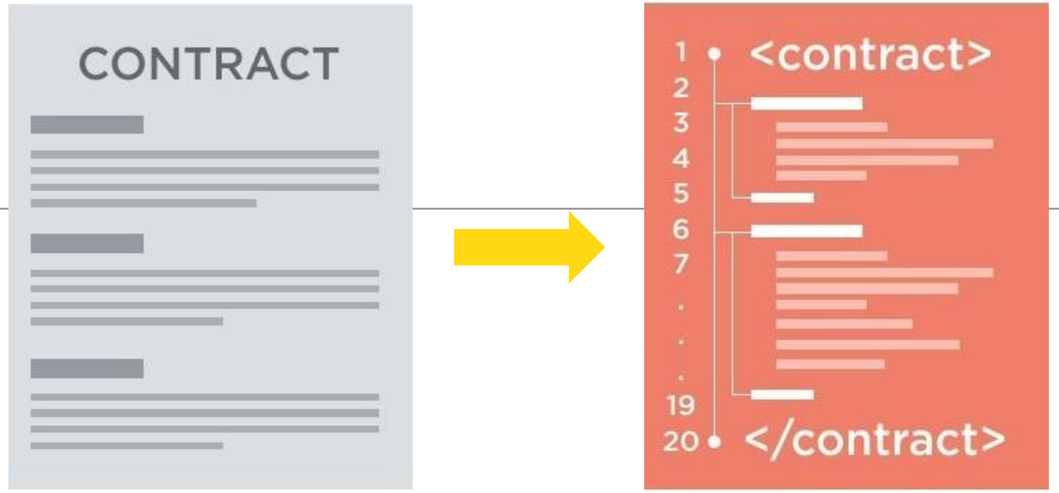
con·tract

(noun) /ˈkäntrakt/

1. a written or spoken agreement ... that is intended to be enforceable by law.

SMART CONTRACTS

CONTRACTS



smart con·tract

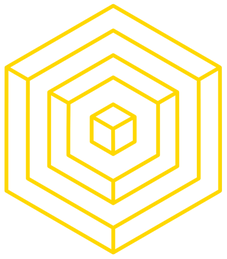
(noun) /smärt ˈkäntrakt/

1. code that **facilitates**, **verifies**, or **enforces** the negotiation or execution of a digital contract.
    a. **Trusted entity** must run this code

QUESTIONS?

2 ETHEREUM


ethereum
HOMESTEAD RELEASE
BLOCKCHAIN APP PLATFORM



WHAT IS ETHEREUM?

HIGH-LEVEL OVERVIEW

- Ethereum is a **decentralized** platform designed to run **smart contracts**
  - Like a distributed computer to execute code
  - Account-based blockchain
  - **Distributed state machine - transactions change global state**
    - transactions == state transaction function
- Ethereum has a native asset called **ether**
  - basis of value in the Ethereum ecosystem
  - needed to align incentives, given as mining rewards

WHAT IS ETHEREUM?

WHO WOULD WIN?

Bitcoin

- First successful cryptocurrency
- Trustless
- Immutable
- Uncensorable
- Pseudonymous
- No central point of failure
- One-CPU-One-Vote

1 turing-complete boi

WHAT IS ETHEREUM?

COMPARISON WITH BITCOIN

Bitcoin

- The "Gold Standard" of blockchains
- Asset: bitcoins
  - Primary purpose of the Bitcoin blockchain
- Simple and robust
- Stack-based, primitive scripting language, not Turing-complete
- UTXO-based
- Will likely remain Proof-of-Work

Ethereum

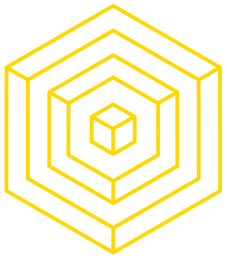
- Smart Contract Blockchain Platform
- Asset: ether
  1. Fund computation
  2. Align incentives
- Complex and feature-rich
- Turing-complete scripting language
- Account-based
- Planning to move to Proof-of-Stake

WHAT IS ETHEREUM?

COMPARISON WITH BITCOIN

Misc. Implementation Details

- Block creation time: ~15 sec vs ~10 min
- Proof-of-Work: Ethash vs SHA-256 (currently ASIC resistant)
- Exchange Rate: $841.01 (2018-02-24)



ETHEREUM ACCOUNTS

ACCOUNTS VS UTXO MODEL

Easy to make transactions and prevent double spending

Bitcoin:

Bob owns private keys to set of UTXOs

- 5 BTC ⇒ Bob
- 3 BTC ⇒ Bob
- 2 BTC ⇒ Bob

Ethereum:

Alice owns private keys to an account

address: “0xfa38b...”

balance: 10 ETH

code: c := a + b

ETHEREUM ACCOUNTS

ACCOUNTS RATIONALE

Easy to make transactions and prevent double spending

Bitcoin:

Bob owns private keys to set of UTXOs

- 5 BTC ⇒ Bob
- 3 BTC ⇒ Bob
- 2 BTC ⇒ Bob

Ethereum:

Alice owns private keys to an account

address: “0xfa38b...”

balance: 10 ETH

code: c := a + b

Space-efficient to update balances instead of storing UTXOs

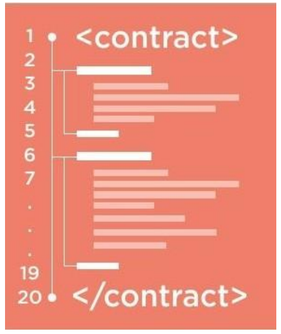
Easier to look up balance and transfer between accounts when programming

ETHEREUM ACCOUNTS

ACCOUNT TYPES

Externally Owned Accounts

- Owned by some external entity (person, corporation, etc.)
- Can send transactions to transfer ether or trigger contract code
- Contains:
  - Address
  - Ether Balance



Contract Accounts

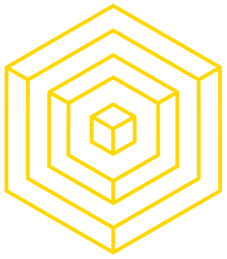
- “Owned” by contract
- Code execution triggered by transactions or function calls (msg)
- Contains:
  - Address
  - Associated contract code
  - Persistent storage

ETHEREUM SMART CONTRACTS

CONTROL

Smart Contracts in Ethereum are like autonomous agents that live inside of the Ethereum network

- React to external world when "poked" by transactions (which call specific functions)
- Have direct control over:
  - **internal ether balance**
  - **internal contract state**
  - **permanent storage**



ETHEREUM SMART CONTRACTS

SMART CONTRACTS IN ETHEREUM

Ethereum Contracts generally serve four purposes:

- **Store and maintain data**
  - Data represents something useful to users or other contracts
  - Ex: a token currency or organization's membership
- **Manage contract or relationship between untrusting users**
  - Ex: financial contracts, escrow, insurance
- **Provide functions to other contracts**
  - Serving as a software library
- **Complex Authentication**
  - Ex: M-of-N multisignature access

ETHEREUM SMART CONTRACTS

SAMPLE BETTING CONTRACT

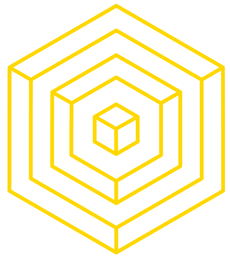
```
contract Betting {
     address public owner;
     address public gamblerA, gamblerB, oracle;
     uint[] outcomes;
     struct Bet {
         uint outcome;  uint amount;        /* Defines a Bet */
         bool initialized;
     }

     mapping (address => Bet) bets;
     mapping (address => uint) winnings;  /* Keep track of every gambler's bet */
     ...                                  /* Keep track of every player's winnings */

     function makeBet(uint _outcome) payable returns (bool) { … }
     function makeDecision(uint _outcome) oracleOnly() { … }
     function withdraw(uint withdrawAmount) returns (uint remainingBal) { … }
}
```


AUTHOR: GLORIA ZHAO & NICK ZOGHB

QUESTIONS?



3 ETHEREUM VIRTUAL MACHINE

ETHEREUM VIRTUAL MACHINE

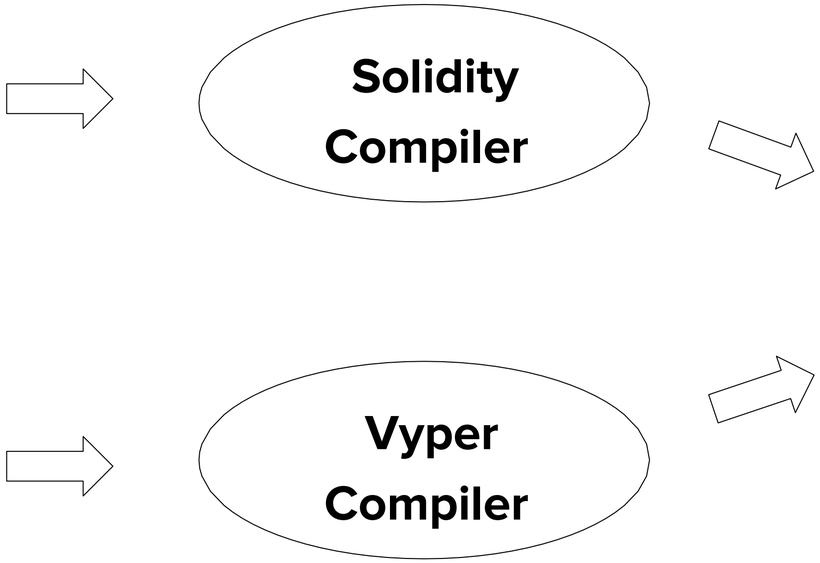
COMPILATION AND PROCESS

Solidity Contract

```
contract PhilipToken {
        /* ... */
}
```

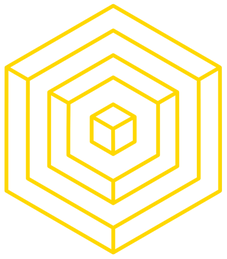
Vyper Contract

```
def register(key, value):
        /* ... */
```



EVM Code

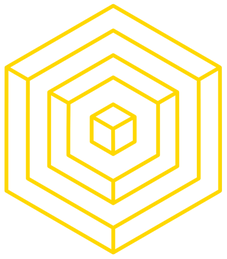
```
PUSH1 0x60  PUSH1 0x40
MSTORE  CALLDATASIZE
ISZERO
. . .
PUSH2 0x1b4c  POP
JUMP
```



ETHEREUM VIRTUAL MACHINE

DISTRIBUTED VERIFICATION & CONSENSUS

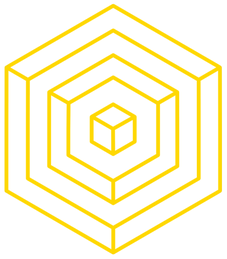
- Ethereum is a “distributed computer”: **every node** executes Ethereum smart contracts, then come to consensus on the new network state
- Ethereum’s distributed consensus protocol is **Proof-of-Work**
  - Miners competitively create blocks by executing EVM code and searching for solution the the mining puzzle
  - PoW is **competitive**, meaning only **ONE** miner is able to add block to the Ethereum blockchain and receive gas and transaction fees
  - Think of PoW as process of “randomly” selecting one node’s execution result as the correct one



ETHEREUM VIRTUAL MACHINE

HIGH-LEVEL OVERVIEW

- Every Ethereum node runs EVM as part of its block verification procedure
- Network consensus removes the need for Trusted Third Party
  - Violation of contracts requires subverting the entire network
- Secure Peer-to-Peer agreements that live on the blockchain forever
- The **EVM (Ethereum Virtual Machine)** runs contract code
- Contract code that actually gets executed on every node is EVM code
  - Our complex features are made possible by the fact that we can compile contract code into something more simple
  - **EVM code**: low-level, stack based bytecode language (i.e. JVM bytecode)



EVM GAS AND FEES

HIGH-LEVEL OVERVIEW

Immediate Issue:

What if our contract has an infinite loop?

Every node on the network will get stuck executing the loop forever!

- By the *halting problem*, it is impossible to determine ahead of time whether the contract will ever terminate
- **⇒ Denial of Service Attack!**

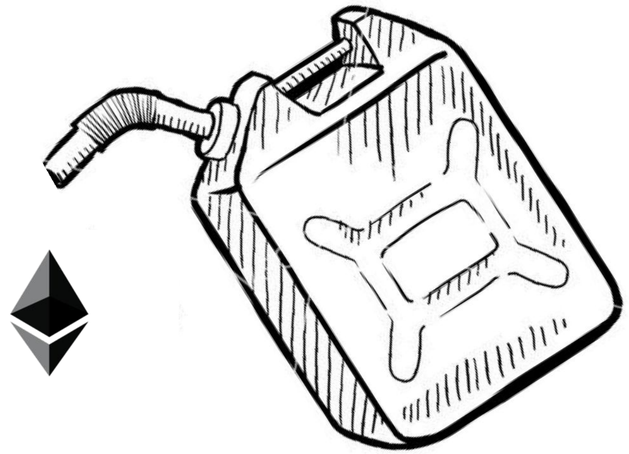
```
function foo()
{
        while (true) {
                /* Loop forever! */
        }
}
```

EVM GAS AND FEES

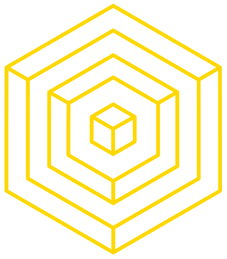
HIGH-LEVEL OVERVIEW

Ethereum's solution:

- Every contract requires "gas", which "fuels" contract execution
- Every EVM op-code requires some gas in order to execute
- Every transaction specifies:
  - the `startgas` , or the maximum quantity of gas it is willing to consume
  - the `gasprice` , or the fee in ether it is willing to pay per unit gas



EVM



EVM GAS AND FEES

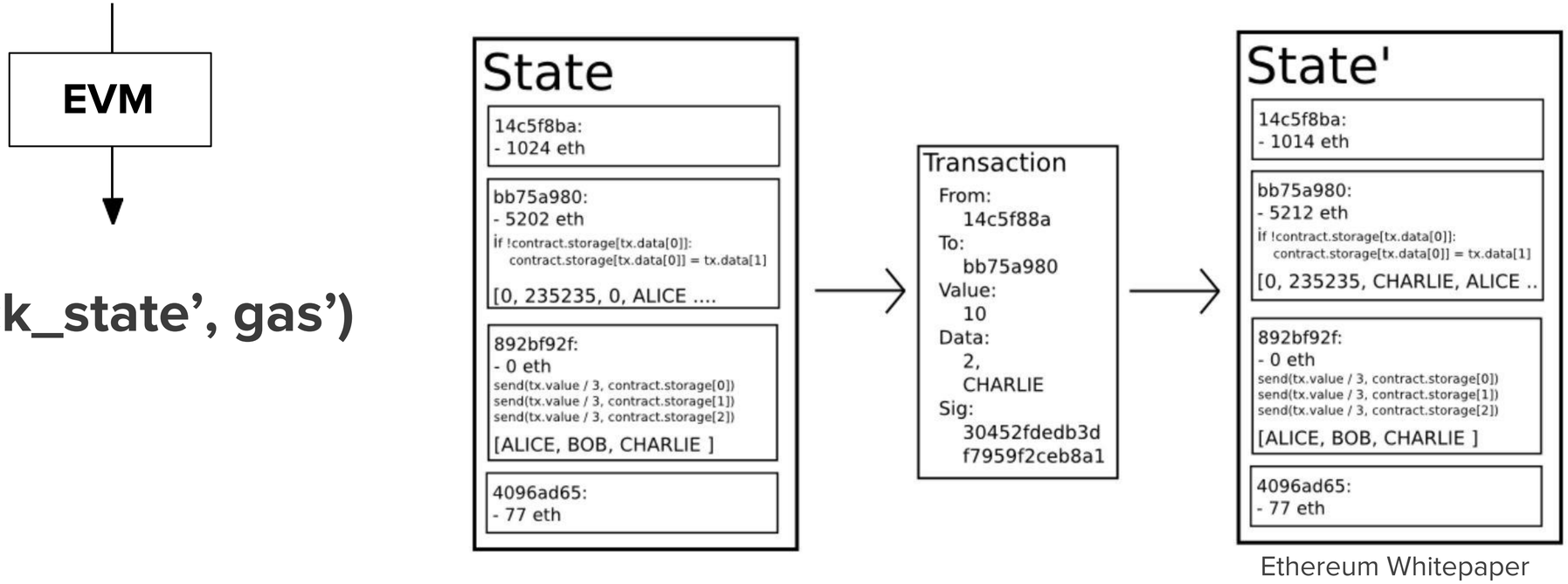
HIGH-LEVEL OVERVIEW

- At the start of the transaction
  - `startgas * gasprice` (units = ether) are subtracted from the sender's account (the one "poking" the contract)
- If the contract **successfully executes** ...
  - the remaining gas is refunded to the sender
- If the contract execution **runs out of gas** before it finishes ...
  - execution reverts
  - `startgas * gasprice` are not refunded
- Purchasing gas == purchasing distributed, trustless computational power
- An attacker looking to launch a DoS attack will need to supply enough ether to fund the attack

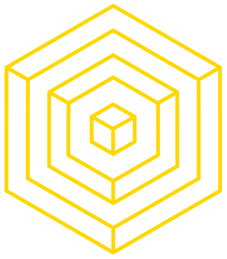
ETHEREUM NETWORK STATE

STATE TRANSITION FUNCTION

(block_state, gas, memory, transaction, message, code, stack, pc)



Ethereum Whitepaper



ETHEREUM CONCLUSIONS

IT'S NOT FOR EVERYTHING

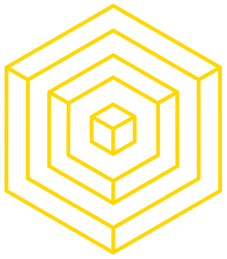
- **Ethereum is not about optimising efficiency of computation**
- Its parallel processing is **redundantly parallel**
  - **efficient way to reach consensus** on the system state without needing trusted third parties
- Contract executions are redundantly replicated across nodes
  - ⇒ expensive
  - creates an **incentive not to use the blockchain** for computation that can be done off chain

QUESTIONS?


BREAK
SECTION
BLOCKCHAIN
AT BERKELEY

4 USE CASES

4.1 BASIC USE CASES

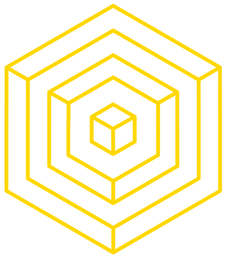


BASIC USE CASES

SMART ASSETS

- Token System Implementation
- Database with one operation
  - Ensure Alice has enough $$ and that she initiated the transaction
  - Subtract X from Alice, give X to Bob

```
def send(to, value):
    if self.storage[msg.sender] >= value:
        self.storage[msg.sender] = self.storage[msg.sender] - value
        self.storage[to] = self.storage[to] + value
```



PUBLIC REGISTRY: Namecoin

BLOCKCHAIN FUNDAMENTALS

- DNS System
  - Maps domain name to IP address
  - “gillian.chu” => “12.34.56.78”
- Easy to implement in Ethereum

```
def register(name, value):
    if !self.storage[name]:
        self.storage[name] = value
```

DOCUMENT OWNERSHIP

BLOCKCHAIN FUNDAMENTALS

“Proof-of-Existence”

- Proves ownership of a certain document without revealing it
- **Timestamps** verify ownership later

Use Cases:

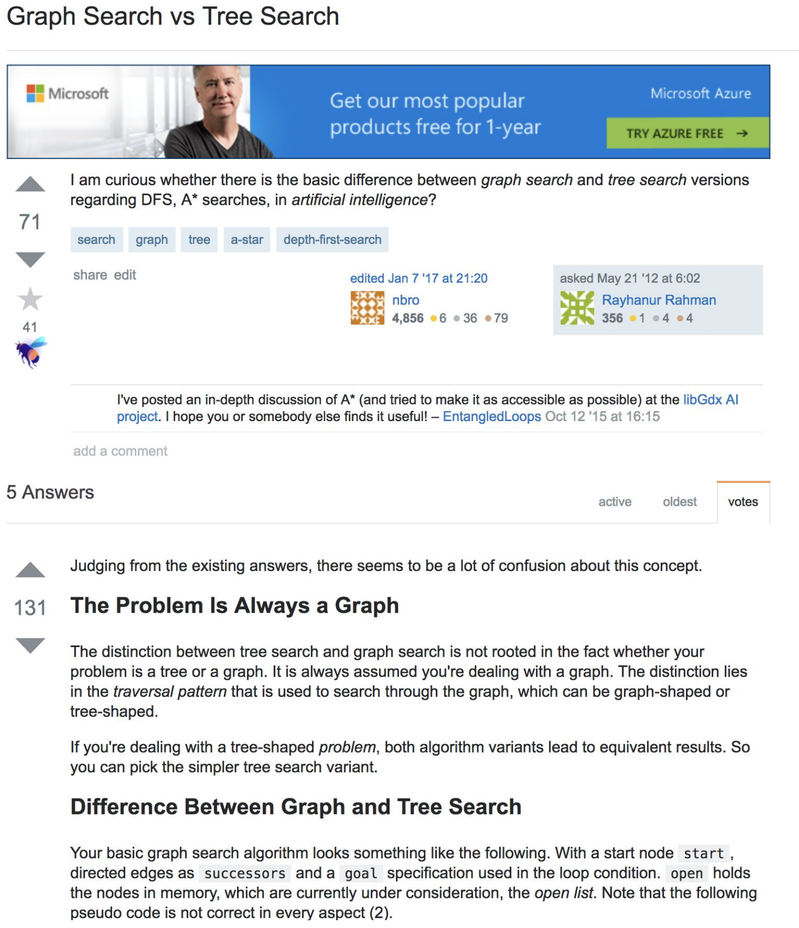
- Rent server space to store documents
  - Proof document is unmodified via hash values
  - **Integrity** guaranteed

STANDARD BOUNTIES

ETHEREUM SMART CONTRACTS

ETHDenver

- Incentivize answers to StackOverflow Q's
- Makes use of StandardBounties contract
- Uses Metamask Integration



Graph Search vs Tree Search

71

I am curious whether there is the basic difference between *graph search* and *tree search* versions regarding DFS, A* searches, in *artificial intelligence*?

search graph tree a-star depth-first-search

share edit

edited Jan 7 '17 at 21:20
nbro
4,856 ● 6 ● 36 ● 79

asked May 21 '12 at 6:02
Rayhanur Rahman
356 ● 1 ● 4 ● 4

41

I've posted an in-depth discussion of A* (and tried to make it as accessible as possible) at the libGdx AI project. I hope you or somebody else finds it useful! – EntangledLoops Oct 12 '15 at 16:15

add a comment

5 Answers

active oldest votes

131

Judging from the existing answers, there seems to be a lot of confusion about this concept.

The Problem Is Always a Graph

The distinction between tree search and graph search is not rooted in the fact whether your problem is a tree or a graph. It is always assumed you're dealing with a graph. The distinction lies in the *traversal pattern* that is used to search through the graph, which can be graph-shaped or tree-shaped.

If you're dealing with a tree-shaped *problem*, both algorithm variants lead to equivalent results. So you can pick the simpler tree search variant.

Difference Between Graph and Tree Search

Your basic graph search algorithm looks something like the following. With a start node `start`, directed edges as `successors` and a `goal` specification used in the loop condition. `open` holds the nodes in memory, which are currently under consideration, the *open list*. Note that the following pseudo code is not correct in every aspect (2).

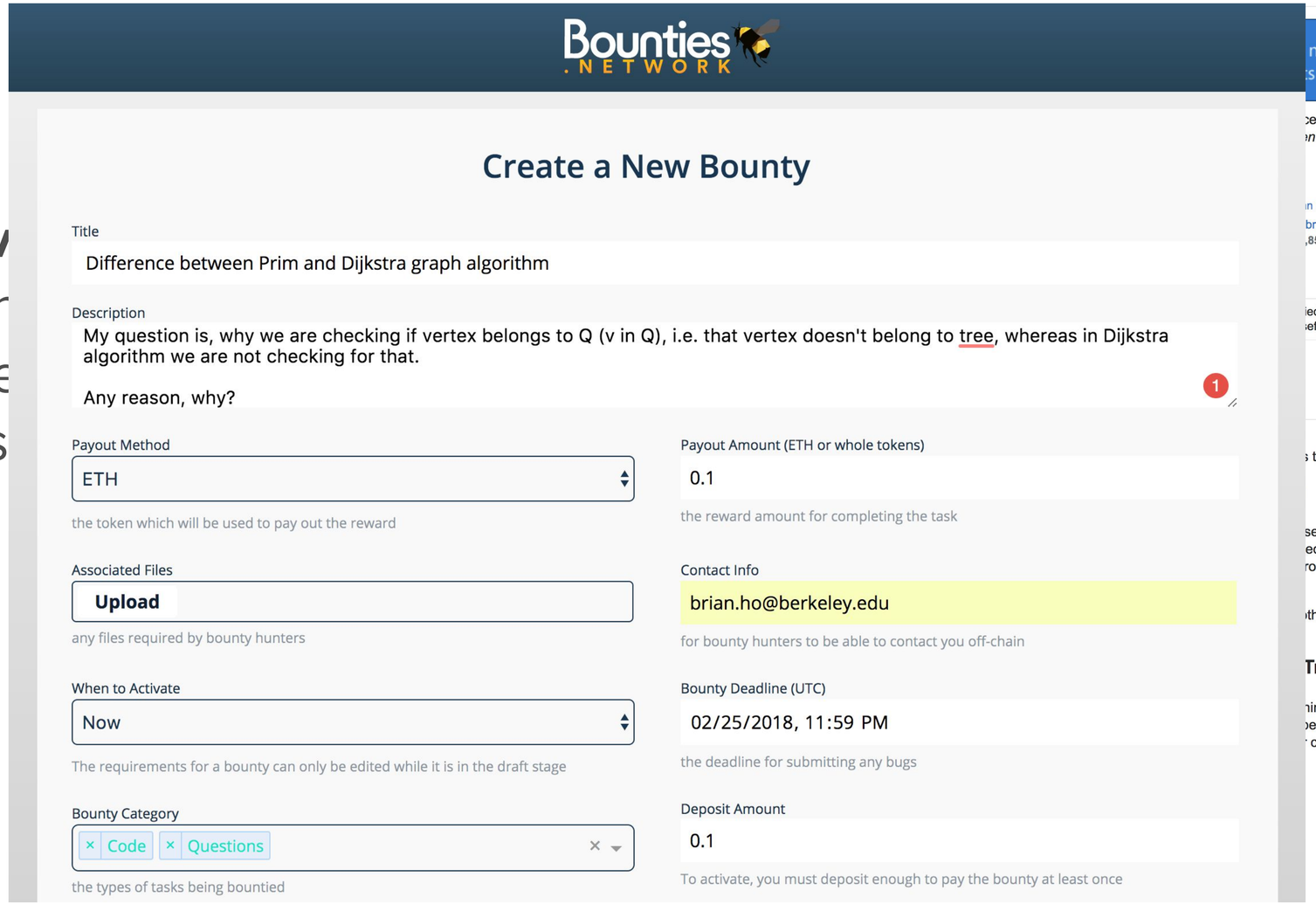
STANDARD BOUNTIES

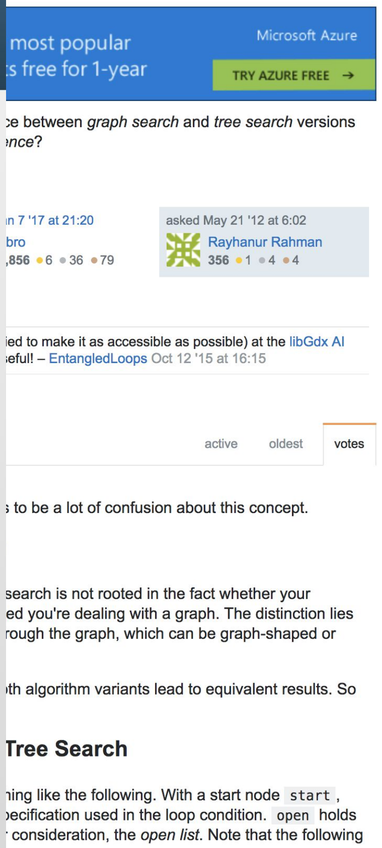
ETHEREUM SMART CONTRACTS

ETHDenv

- Incer
- Make
- Uses

Graph Search vs Tree Search



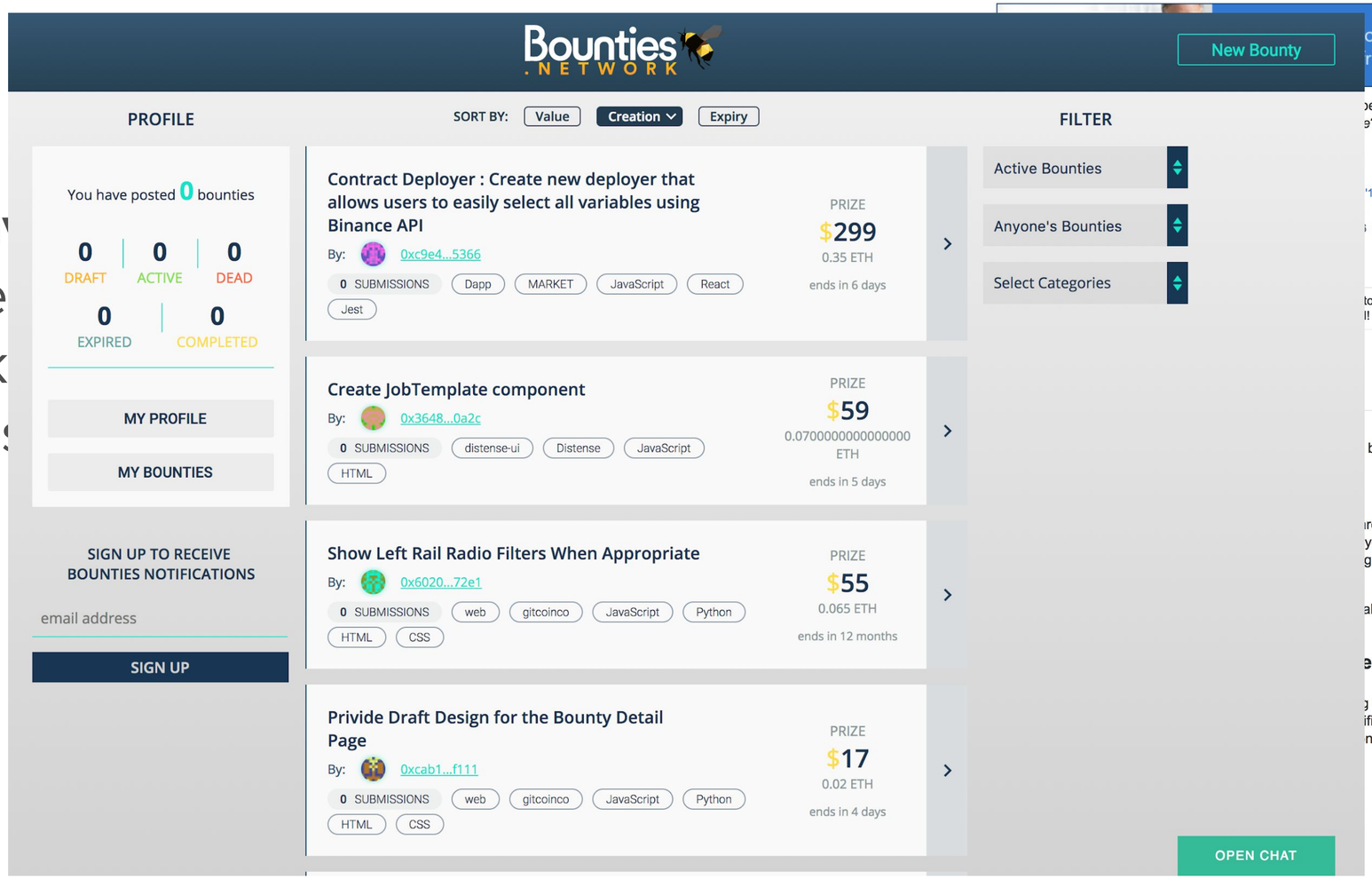


STANDARD BOUNTIES

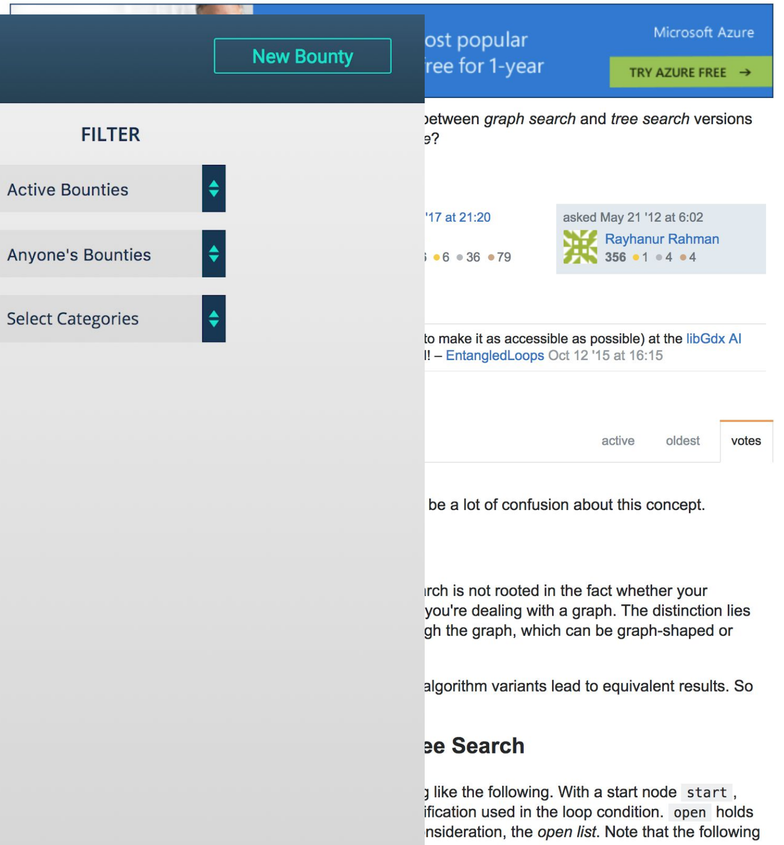
ETHEREUM SMART CONTRACTS

ETHDen

- Ince
- Mak
- Use



Graph Search vs Tree Search



4.2 ADVANCED USE CASES

DECENTRALIZED LAND TITLES

BLOCKCHAIN FUNDAMENTALS

Problem:

- Flawed paperwork, forged signatures, **unclear documents**
- Lacking **central** government authority

Pitfalls:

- Corrupt officials accepting **bribes**, **tampering** with records
- Government cannot support land record authority
- Citizens **mistrustful** of NGO/multiple NGOs

DECENTRALIZED LAND TITLES

BLOCKCHAIN FUNDAMENTALS

The Blockchain Solution:

- Hashes & Digital Signatures
- Transparency
- Immutability
- Limits Centralization

Simple mechanism to transfer ownership, like making a transaction on Bitcoin

AUTHOR: APARNA KRISHNAN
UPDATED: GILLIAN CHU

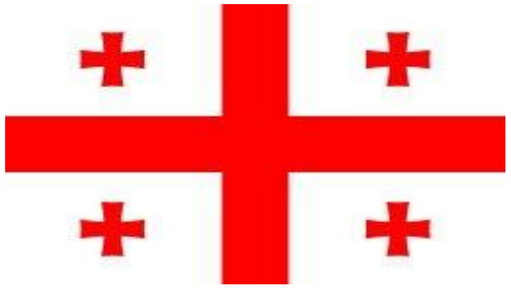
DECENTRALIZED LAND TITLES

BLOCKCHAIN FUNDAMENTALS

Caveat:

The blockchain is only as good as the information fed into it.

Countries investigating: **Georgia, Ukraine, Sweden**



AUTHOR: APARNA KRISHNAN
UPDATED: GILLIAN CHU

DECENTRALIZED LAND TITLES

BLOCKCHAIN FUNDAMENTALS

Caveat:

The blockchain is only as good as the information fed into it.

Countries investigating: **Georgia, Ukraine, Sweden**

AUTHOR: APARNA KRISHNAN
UPDATED: GILLIAN CHU

BLOCKCHAIN AT BERKELEY

COMMENTS?

PREDICTION MARKETS

A QUICK DEFINITION

Draws on the wisdom of the crowd

Ex: “Who will win the 2020 Presidential Election? Zuckerberg or Trump?”

1. Replace shares with bets
2. Random oracles report

Centralized Prediction Market

PREDICTION MARKETS

AND DECENTRALIZING THEM....

"A service that never crashes, a service that's completely transparent..."

Benefits:

- no restrictions on market creation
- shared liquidity pool
- censorship-resistant
- automatic, trustless payments

GNOSIS

augur

PREDICTION MARKETS

AND DECENTRALIZING THEM....

Use Case 1:

- Cost-effective way to “buy” information
- Bet for/against to incentivize those with insider information

GNOSIS

augur

PREDICTION MARKETS

AND DECENTRALIZING THEM....

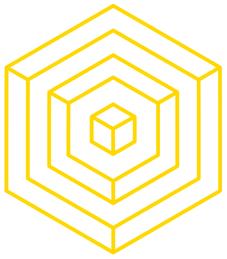
Use Case

- C
- Bet fo der
  infor

Will this movie be a flop?

GNOSIS

augur



PREDICTION MARKETS

OTHER USE CASES

- **Hedging & Insurance**
  - “Will my house burn down?”
- **Security Bug Bounty**
  - “If my company is hacked, will the hacker reveal the vulnerability in a whitehat manner?”
- **ICO Signaling**
  - “Will my ICO deliver products on time?”

PREDICTION MARKETS

FUTARCHY & CRYPTOECONOMICS

Vote Values, but Bet Beliefs

An Introduction to Futarchy

Posted by Vitalik Buterin on August 21st, 2014.

How manipulation-resistant are Prediction Markets?

Our Undertaking in Empirical Cryptoeconomics

Matt Liston Follow
Crypto native
Apr 28, 2016 · 3 min read

An Introduction to Cryptoeconomics and Futarchy experiments on Gnosis

Over the next few months we will be running several experiments on the Gnosis prediction market platform thanks to a generous Ethereum DEV grant. These experiments will test cryptoeconomic hypothesis and methods related to market manipulations and Futarchy and were inspired by Vitalik's reddit post on Empirical Cryptoeconomics. Each experiment consists of a market on

120

Empirical Cryptoeconomics self.ethereum

Submitted 2 years ago * by vbuterin Just some guy

Now that the Ethereum infrastructure is becoming increasingly mature, I thought that it would make sense to try to do some empirical tests of just how effective

QUESTIONS?

SUPPLY CHAIN AND PROVENANCE

BLOCKCHAIN FUNDAMENTALS

Problem: "Blood" diamonds

The **Kimberley Process** is a governmental effort requiring participants to certify the origin of their diamond.

- Corrupt officials take bribes to sign certifications
- Complex supply chains mask actual trails

Everledger: provenance of diamonds

AUTHOR: MAX FANG & APARNA KRISHNAN
UPDATED: GILLIAN CHU

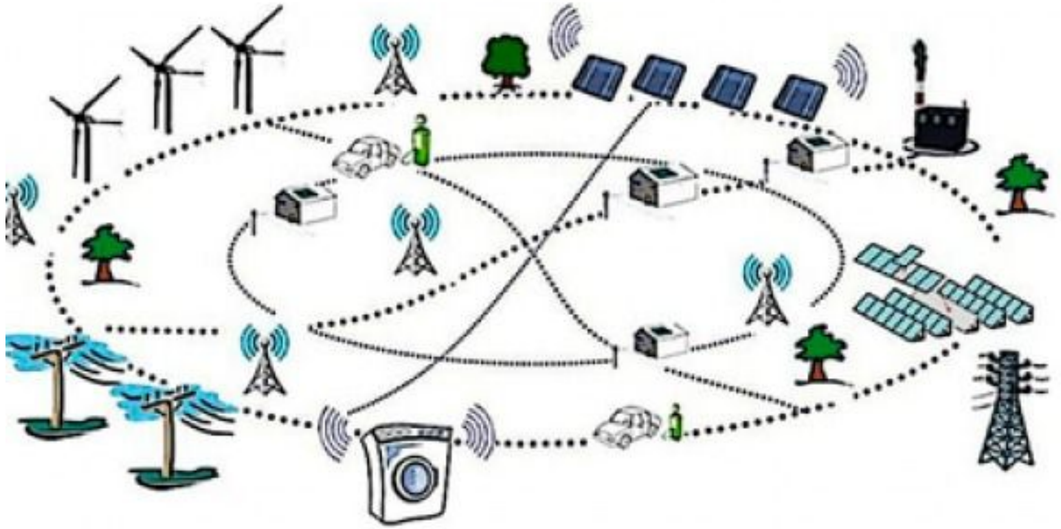
SMART ENERGY GRIDS

BLOCKCHAIN FUNDAMENTALS

Setup: Imbalances in energy across a community of houses
i.e. House A has excess heat,
House C lacks heat

Problem: Need custom infrastructure from A ⇒ B ⇒ C

Solution: Ethereum smart contract for financial commitments



AUTHOR: MAX FANG
UPDATED: GILLIAN CHU

AFTER HOURS TRADING

BLOCKCHAIN FUNDAMENTALS

Tokenizing shares of stock

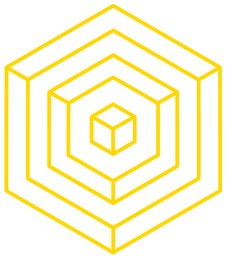
Problem: “Liquidity Problem”
Regulation restricts trading to only using limit orders

Blockchain Solution:

- Stock is subtracted from user’s account and converted into stock tokens
- Stock tokens can be traded worldwide (even after market close!)
- Redeemable via a legal contract

AUTHOR: MAX FANG
UPDATED: GILLIAN CHU

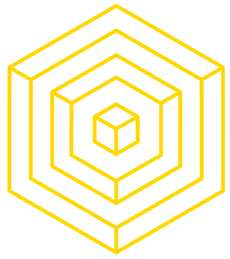
QUESTIONS?



COMMENTARY ON USE CASES

BLOCKCHAIN FUNDAMENTALS

Blockchain vs. The Internet



5 STATE OF THE ECOSYSTEM

THE COMMUNITY

LEADERS

Vitalik Buterin

Roger Ver

Marc Andreesen

Bobby Lee

Nick Szabo

Andreas Antonopoulos

CONSENSYS

BLOCKCHAIN FUNDAMENTALS

AUTHOR: GILLIAN CHU

CONSENSYS

BLOCKCHAIN FUNDAMENTALS

AUTHOR: GILLIAN CHU

HOMEWORK

READINGS

- **Finish** reading the Ethereum Whitepaper
  - after “Miscellanea and Concerns”
- Selfish Mining: https://bitcoinmagazine.com/articles/selfish-mining-a-25-attack-against-the-bitcoin-network-1383578440/
- Optional: Take a look at these cool Ethereum Applications
- Optional: Ethereum Yellow Paper

HOMEWORK

- **Come up** with a dApp (decentralized app) idea or blockchain use case! Try to be as creative and specific as possible, and justify the use of a blockchain. Due by next Saturday, March 3rd at 11:59 PM. Check Piazza for submission link.