ETHEREUM AND SMART CONTRACTS: ENABLING A DECENTRALIZED FUTURE

Bitcoin P2P e-cash paper

Satoshi Nakamoto satoshi at vistomail.com
Fri Oct 31 14:10:00 EDT 2008

- Previous message: Fw: SHA-3 lounge
- **Messages sorted by:** [date] [thread] [subject] [author]

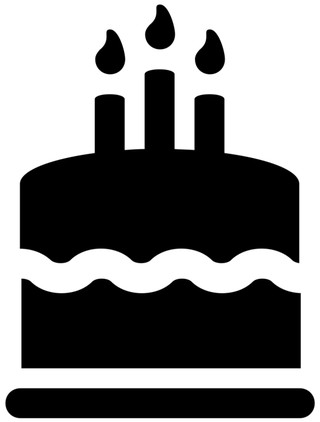
I've been working on a new electronic cash system that's fully
peer-to-peer, with no trusted third party.

The paper is available at:
http://www.bitcoin.org/bitcoin.pdf

The main properties:
 Double-spending is prevented with a peer-to-peer network.
 No mint or other trusted parties.
 Participants can be anonymous.
 New coins are made from Hashcash style proof-of-work.
 The proof-of-work for new coin generation also powers the
    network to prevent double-spending.

Bitcoin: A Peer-to-Peer Electronic Cash System

Abstract. A purely peer-to-peer version of electronic cash would
allow online payments to be sent directly from one party to another
without the burdens of going through a financial institution.
Digital signatures provide part of the solution, but the main
benefits are lost if a trusted party is still required to prevent
double-spending. We propose a solution to the double-spending
problem using a peer-to-peer network. The network timestamps
transactions by hashing them into an ongoing chain of hash-based
proof-of-work, forming a record that cannot be changed without
redoing the proof-of-work. The longest chain not only serves as
proof of the sequence of events witnessed, but proof that it came
from the largest pool of CPU power. As long as honest nodes control
the most CPU power on the network, they can generate the longest
chain and outpace any attackers. The network itself requires



Oct 31, 2008 – Birthday of Bitcoin

11th Birthday

Dün akşam saatlerinde gerçekleşen 1000 dolarlık artışın ardından, gecenin ilerleyen saatlerinde Bitcoin’de 1500 dolarlık bir artış daha meydana geldi.

Saatler içerisinde %35 artan Bitcoin, 10 bin dolar sınırını aştı.

Sabah saatleriyle birlikte inişe geçen Bitcoin şu an 9390 dolar seviyesinden işlem görüyor.

+ %35 increase in one day!!!

LECTURE OVERVIEW

1. SMART CONTRACTS
2. ETHEREUM
3. EVM (ETHEREUM VIRTUAL MACHINE)
4. USE CASES

1 SMART CONTRACTS

BITCOIN REVIEW

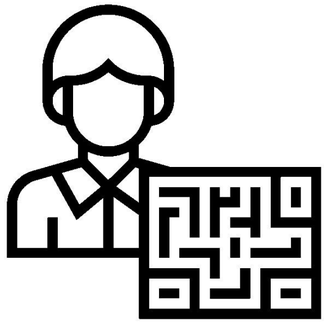
PROPERTIES OF BITCOIN

… but first, a question:

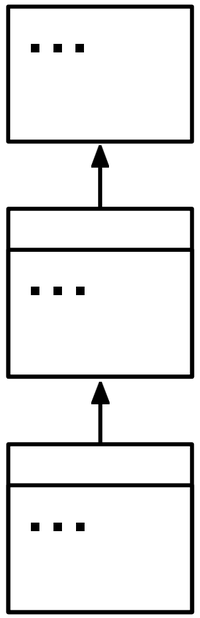
What makes Bitcoin so special?

A DISTRIBUTED NETWORK

BITCOIN'S BARE BONES



cryptographic identities



blockchain

trustless consensus

A DISTRIBUTED 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

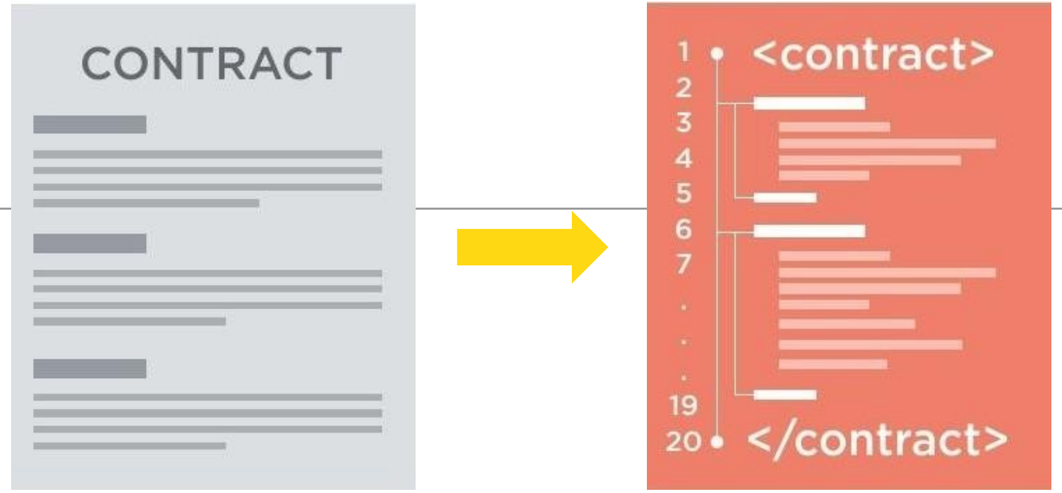
con·tract

(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

2 ETHEREUM


ethereum
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
  - **Distributed state machine - transactions change global state**
    - transactions == state transaction function
- Ethereum has a native asset called **ether**
  - it is a cryptocurrency

WHAT IS ETHEREUM?

WHO WOULD WIN?

Bitcoin

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

1turing-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
- Uses a blend of PoW and PoS

WHAT IS ETHEREUM?

COMPARISON WITH BITCOIN

Misc. Implementation Details

- Block creation time: ~15 sec vs ~10 min
- Proof-of-Work: Ethash (currently ASIC resistant) vs SHA-256
- Exchange Rate: $183,99 (2019-10-30)

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

“CLICKER” QUESTION:

DESIGN CHOICES

What are strengths of the UTXO Model?

Select all that apply.

A) Verifying transaction validity and checking double-spending

B) Space efficiency

C) Looking up account balances quickly

D) Ease in programming

“CLICKER” QUESTION

DESIGN CHOICES

What are weaknesses of the UTXO Model?

Select all that apply.

A) Verifying transaction validity and checking double-spending

B) Space efficiency

C) Looking up account balances quickly

D) Ease in programming

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

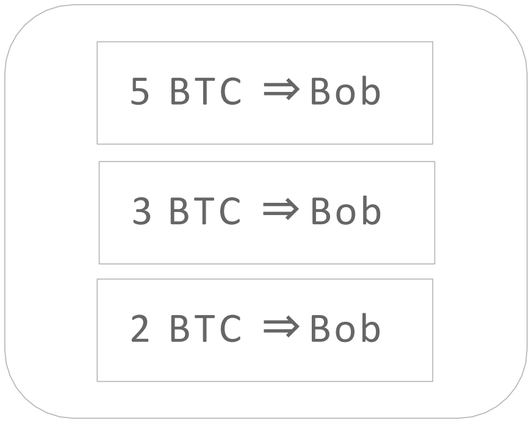
ETHEREUM ACCOUNTS

ACCOUNTS RATIONALE

Easy to make transactions and prevent double spending

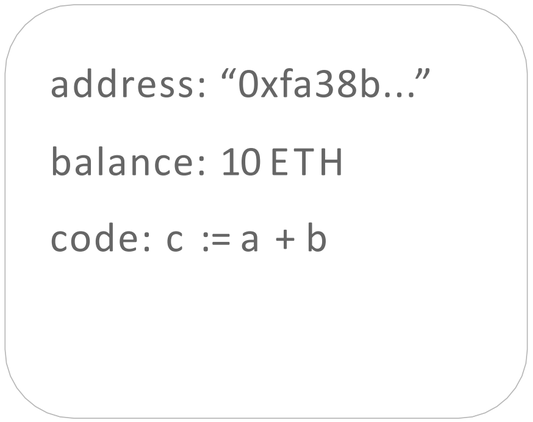
Bitcoin:

Bob owns private keys to set of UTXOs



Ethereum:

Alice owns private keys to an account



Space-efficient to update balances instead of storing UTXOs

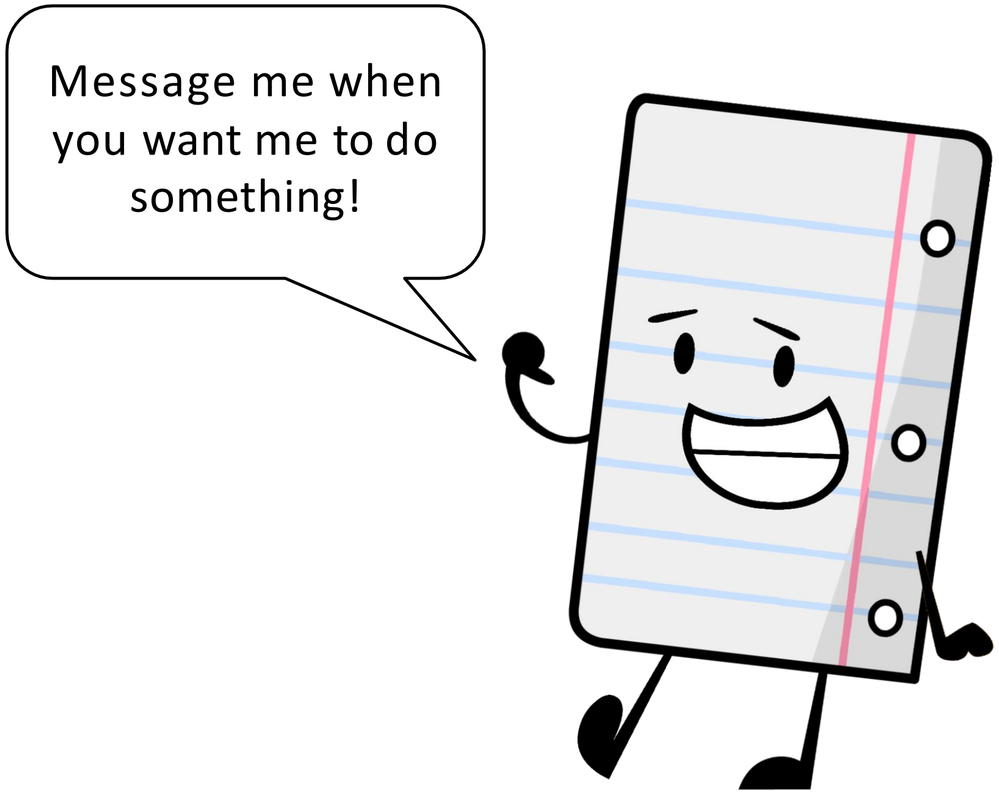
Easier to look up balance and transfer between accounts when programming

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**



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) { … }

}
```

RECIPE FOR MINING: ETHEREUM

SIMILAR TO BITCOIN

Image source: http://www.coindesk.com/information/how-to-set-up-a-miner/

A full-fledged Ethereum miner must:

1. **Download** the entire Ethereum blockchain
2. **Verify** incoming transactions and **Run** Smart Contract code invoked by transactions
2. **Create** a block
3. **Find** a valid nonce
4. **Broadcast** your block
5. **Profit**!

THE DISTRIBUTED COMPUTER

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 code and searching for solution the the mining puzzle
- 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

3 ETHEREUM VIRTUAL MACHINE

ETHEREUM VIRTUAL MACHINE

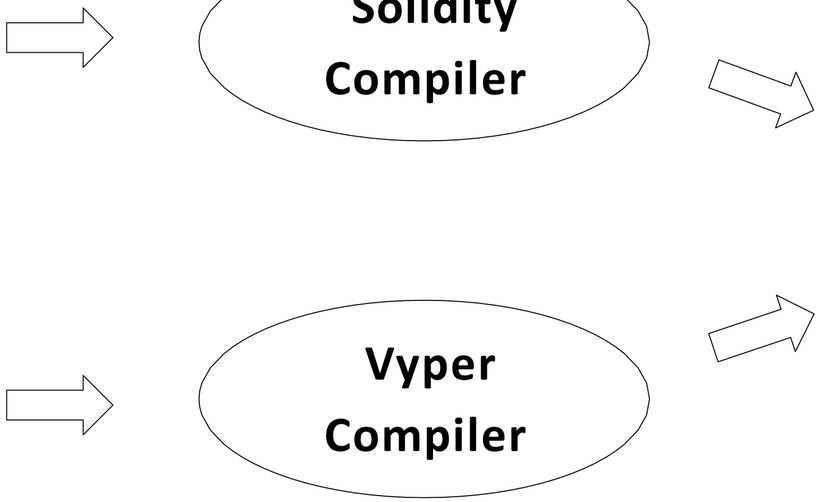
COMPILATION AND PROCESS

Solidity Contract

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

Vyper Contract

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



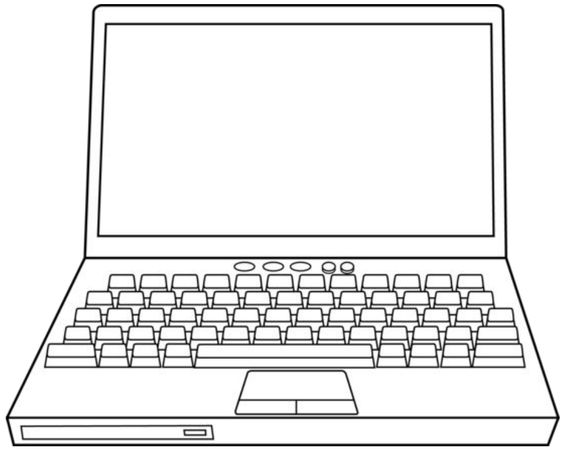
EVM Code

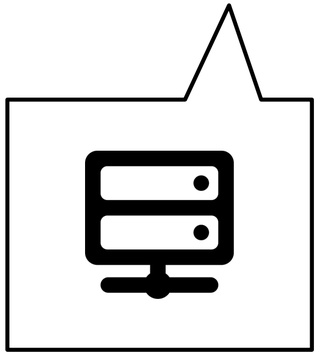
```
PUSH1 0x60 PUSH1
0x40 MSTORE
CALLDATASIZE ISZERO
. . .
PUSH2 0x1b4c
POP JUMP
```

ETHEREUM VIRTUAL MACHINE

HIGH-LEVEL OVERVIEW

- The **EVM (Ethereum Virtual Machine)** is a "mini computer" that runs contract code
- Contract code that actually gets executed on every node is EVM code
  - **EVM code**: low-level, stack based bytecode language (i.e. JVM bytecode)
- Every Ethereum node runs EVM





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!**

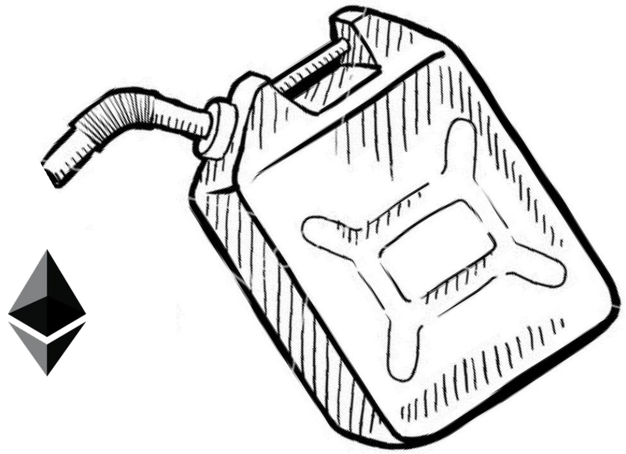
```
function foo()
{
        while (true) {
                /* Loop forever! */
        }
}
```

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

“CLICKER” QUESTION

UNDERSTANDING THE DISTRIBUTED COMPUTER

Recall that gas is used to pay for your smart contract to be validated by this “distributed computer.” **How much gas do you need to supply?**

A) Enough for 1 person to run the code

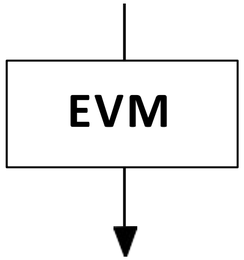
B) Enough for everyone on the network to run the code

C) Enough for the delegated validators to run the code

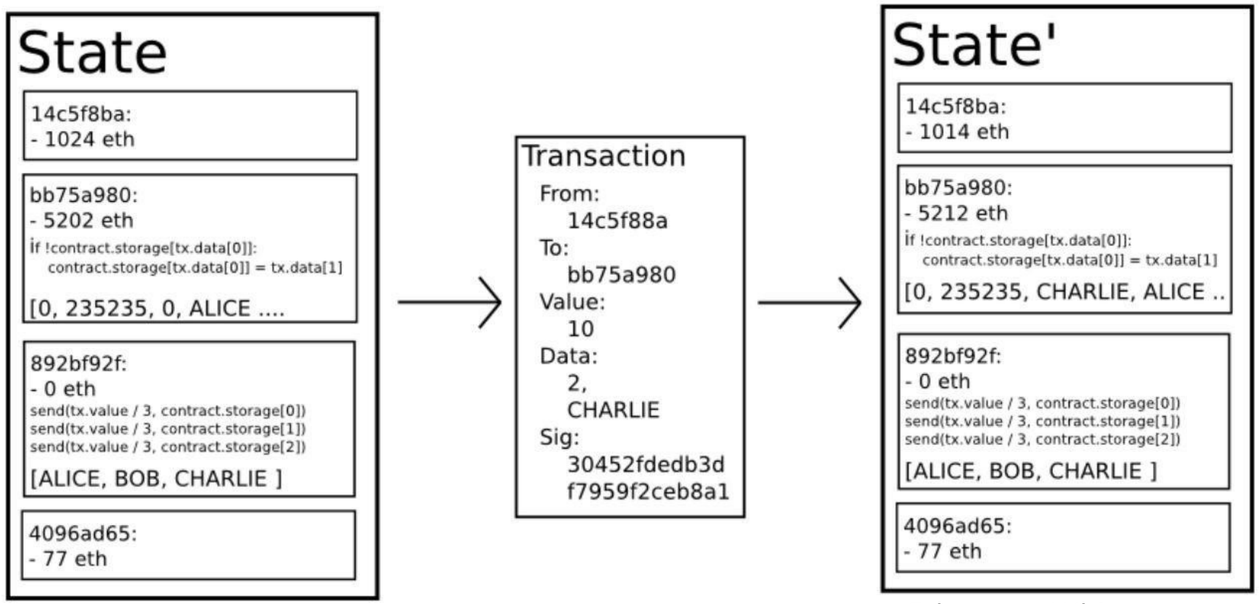
ETHEREUM NETWORK STATE

STATE TRANSITION FUNCTION

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



(block_state', gas')



Ethereum Whitepaper

ETHEREUM CONCLUSIONS

IT'S NOT FOR EVERYTHING

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

ETHEREUM CONCLUSIONS

IT'S NOT FOR EVERYTHING

How would you decide between using a centralized and decentralized solution?

ETHEREUM CONCLUSIONS

IT'S NOT FOR EVERYTHING

Use Blockchain:

- Need for a **shared database** with multiple writers
- Parties **cannot trust** one another, and no trusted third party or authority is available
- Interested in fault-tolerance, data immutability or censorship resistance

Use Centralized Database:

- Database does not need to be shared, or is shared by parties who trust one another
- Must keep data **confidential**
- Must handle **complex** and/or large amounts of data
- Need to be able to edit data
- Interested in cost-effectiveness, speed or **efficiency**

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:

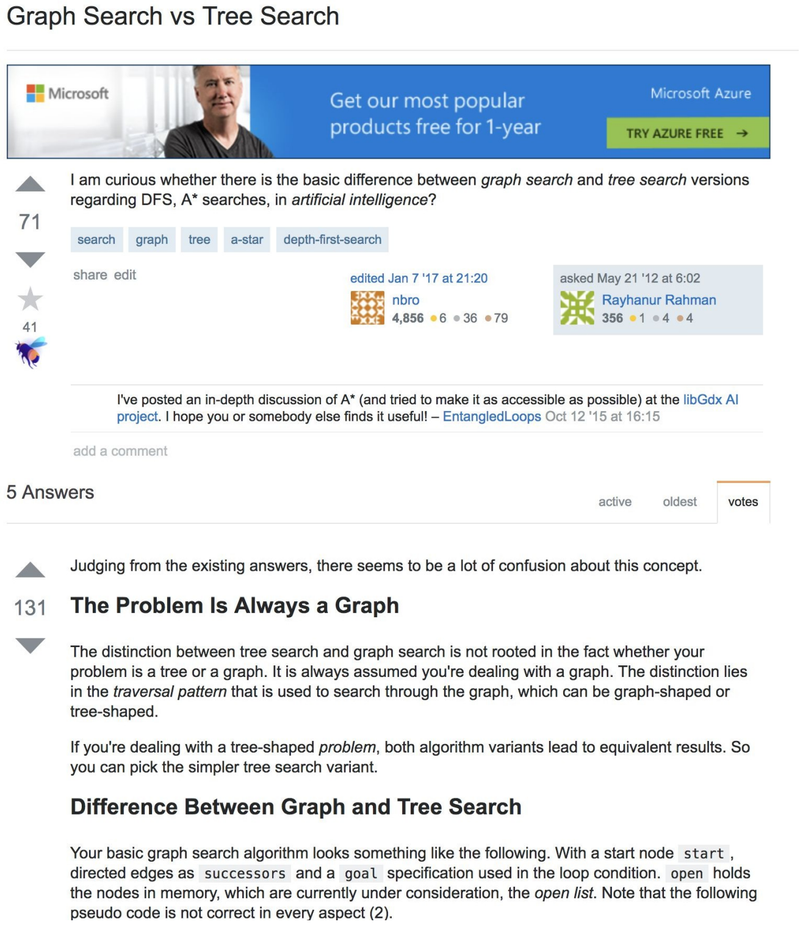
- Rent server space to store documents
  - Proof document is unmodified via hash values
  - **Integrity** guaranteed

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

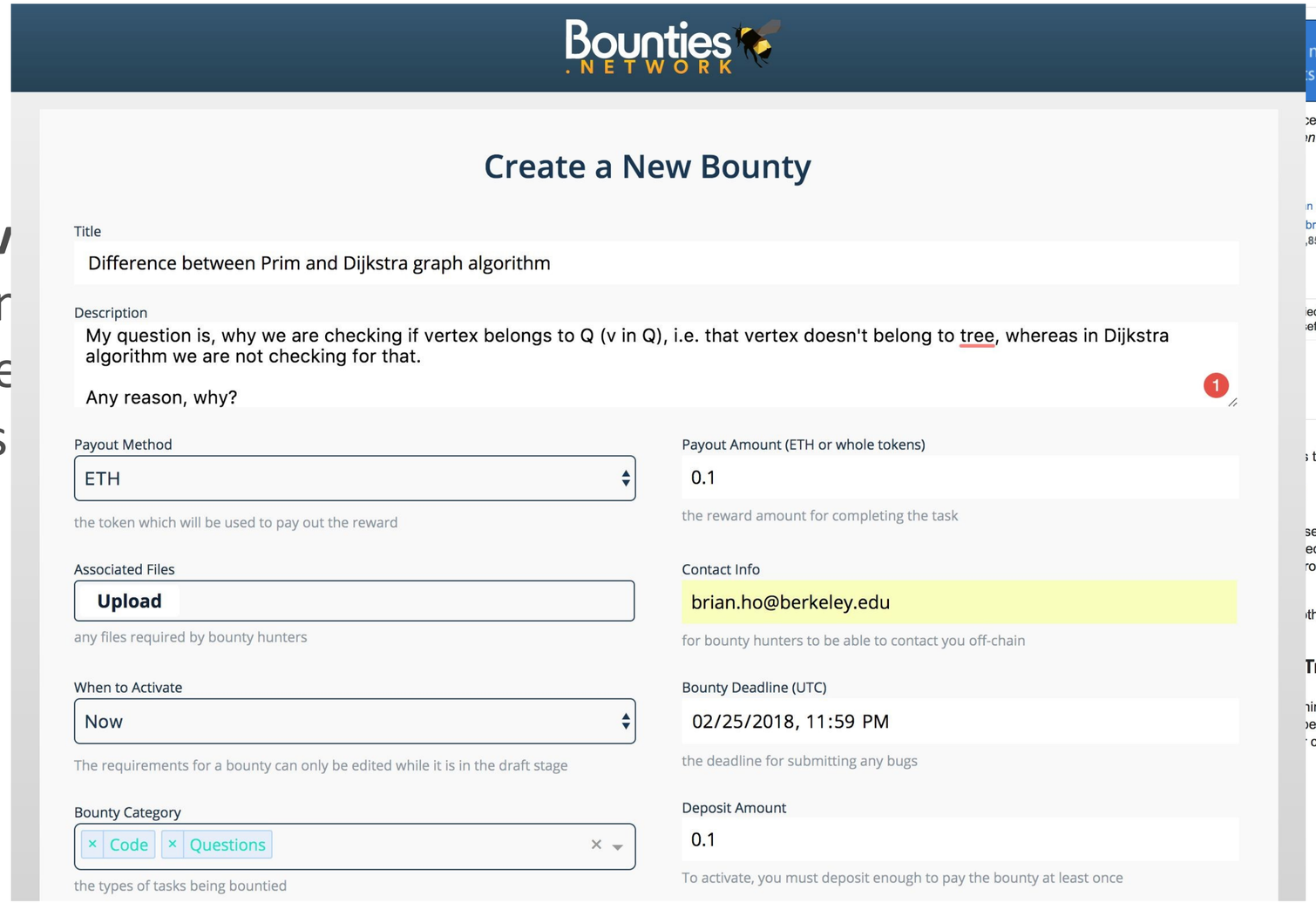
Your basic graph search algorithm looks something like the following. With a start node `start`, directed edges as `successors` and a `goal` specification used in the loop condition. `open` holds the nodes in memory, which are currently under consideration, the *open list*. Note that the following pseudo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

STANDARD BOUNTIES

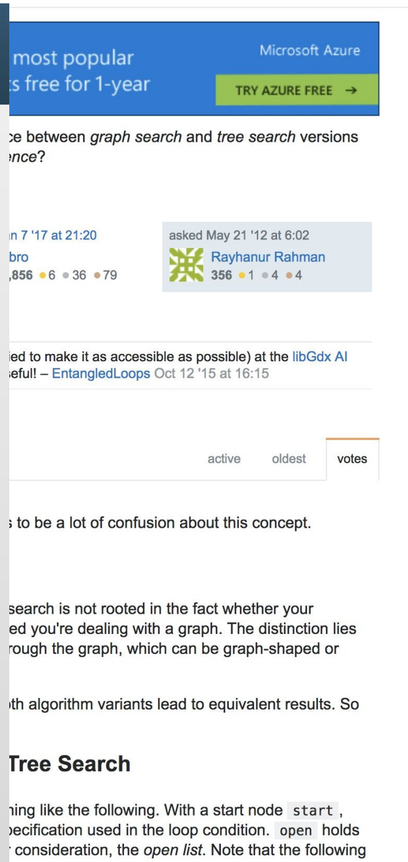
ETHEREUM SMART CONTRACTS

ETHDenv

- Incer
- Make
- Uses



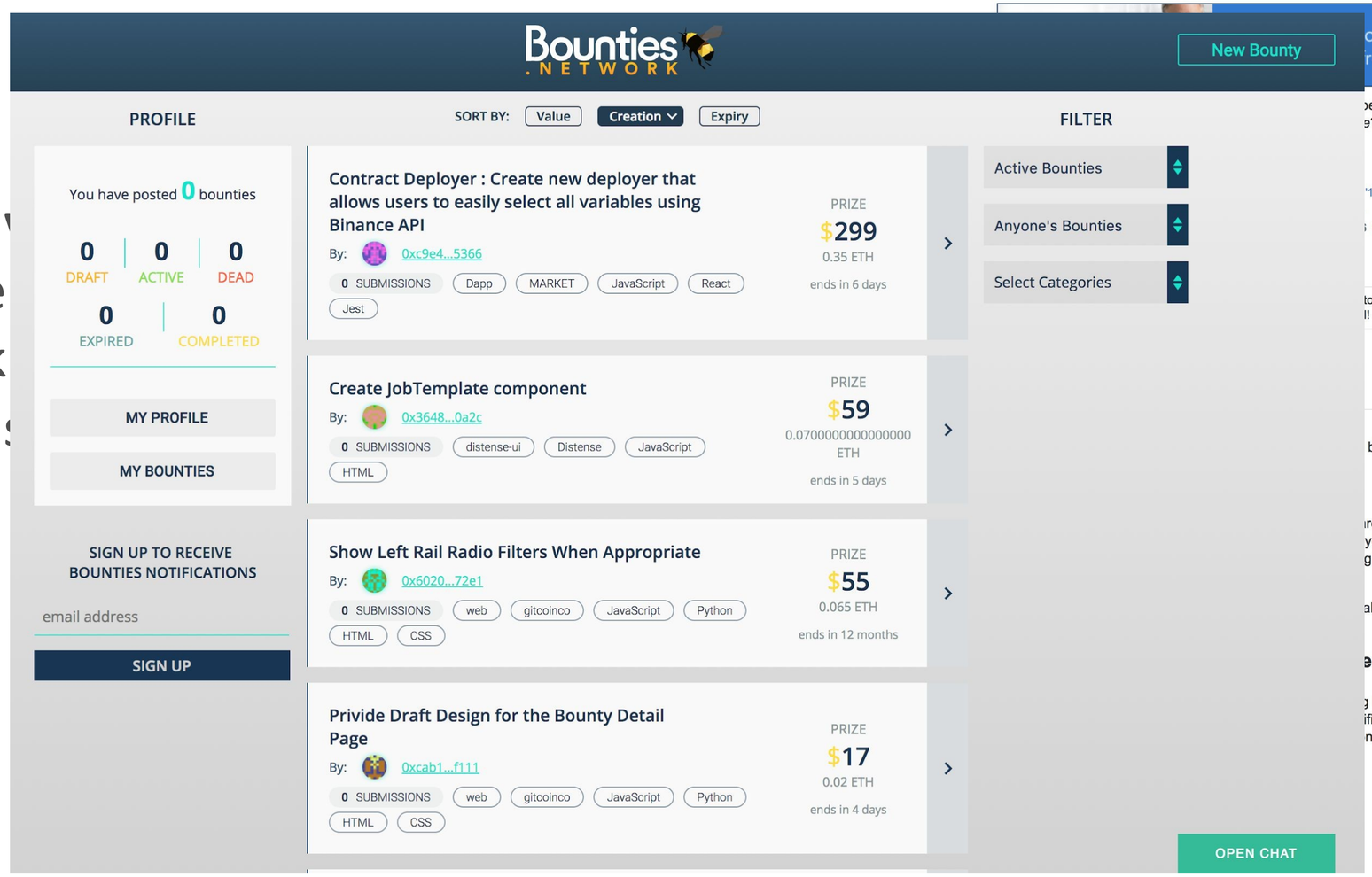
Graph Search vs Tree Search

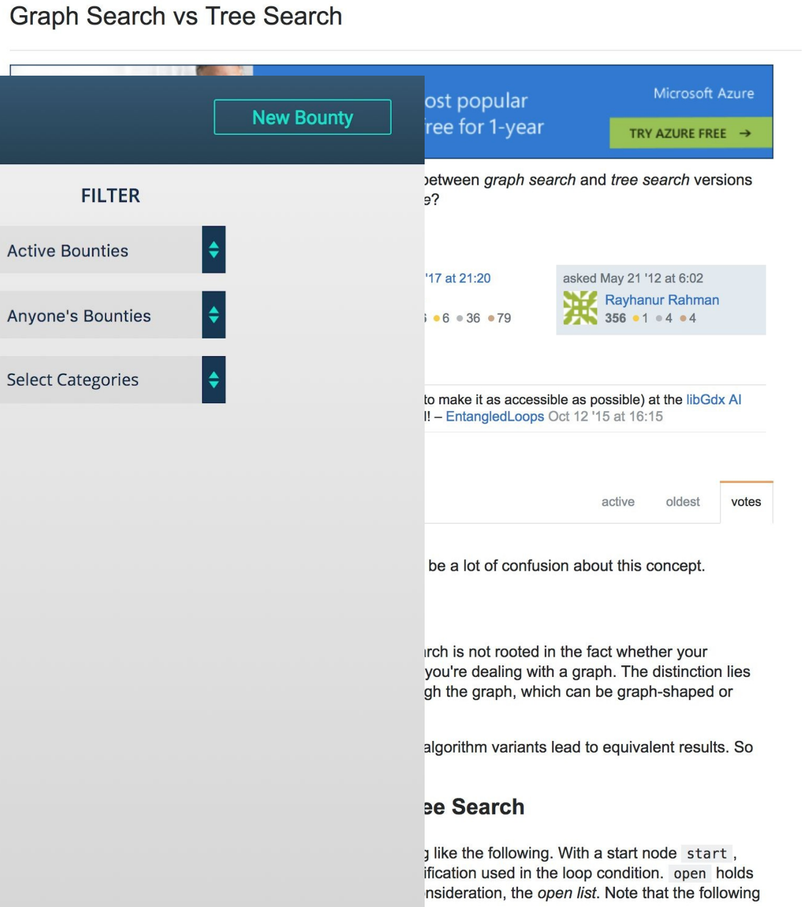


STANDARD BOUNTIES

ETHEREUM SMART CONTRACTS

- ETHDen
- Ince
- Mak
- Use





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

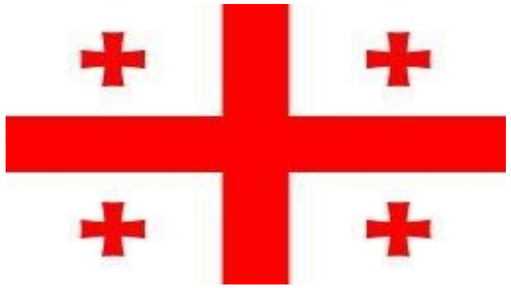
DECENTRALIZED LAND TITLES

BLOCKCHAIN FUNDAMENTALS

Caveat:

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

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



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

- Co…
- Bet for/ag… insider
  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

OTHER USE CASES

Also…

“I bet $1 million that Brian will be alive on October 4.”

-> Assassination Market

SUPPLY CHAIN AND PROVENANCE

BLOCKCHAIN FUNDAMENTALS

The idea: track supply chain information through a blockchain (transparent, immutable ledger)

Businesses: Efficient transactions through smart contracts, easy tracking and recalls

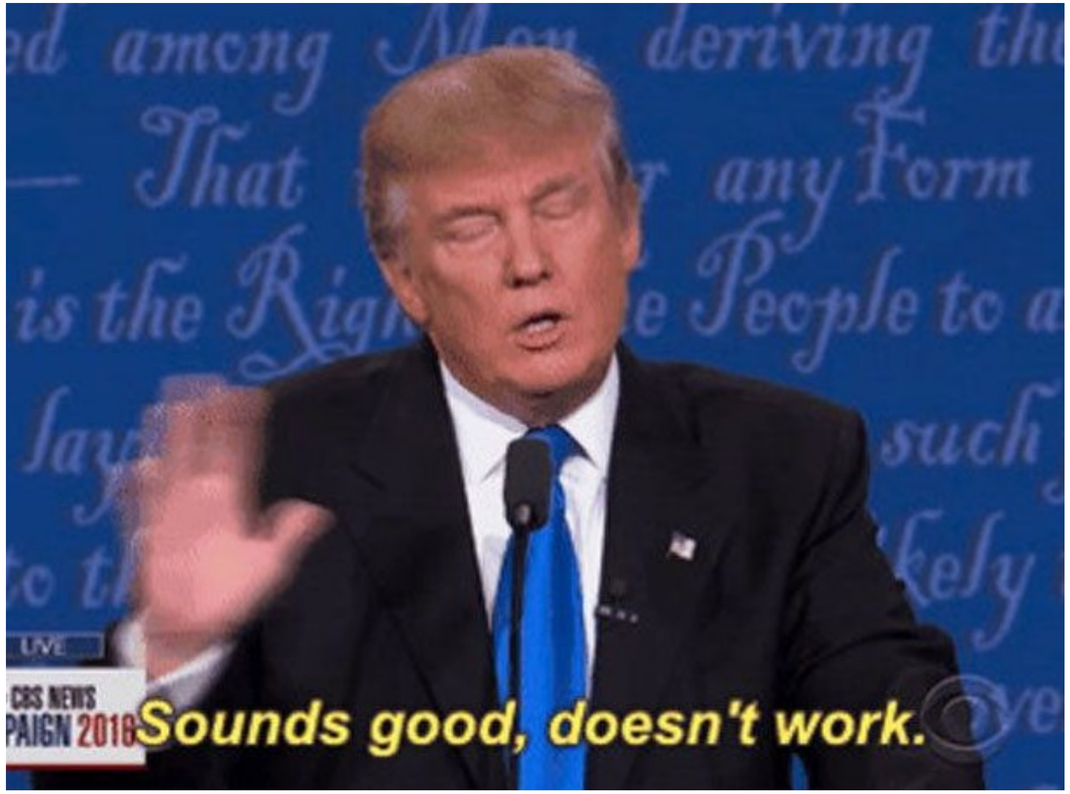
Consumers: Transparent record of products, applications in fair trade, sustainability, and ethical consumerism

SUPPLY CHAIN AND PROVENANCE

BLOCKCHAIN FUNDAMENTALS

Problem: Need to trust data input. There is no secure way to bridge the physical and digital worlds.

We could end up with Transparent, Immutable, Integrous... Garbage.



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

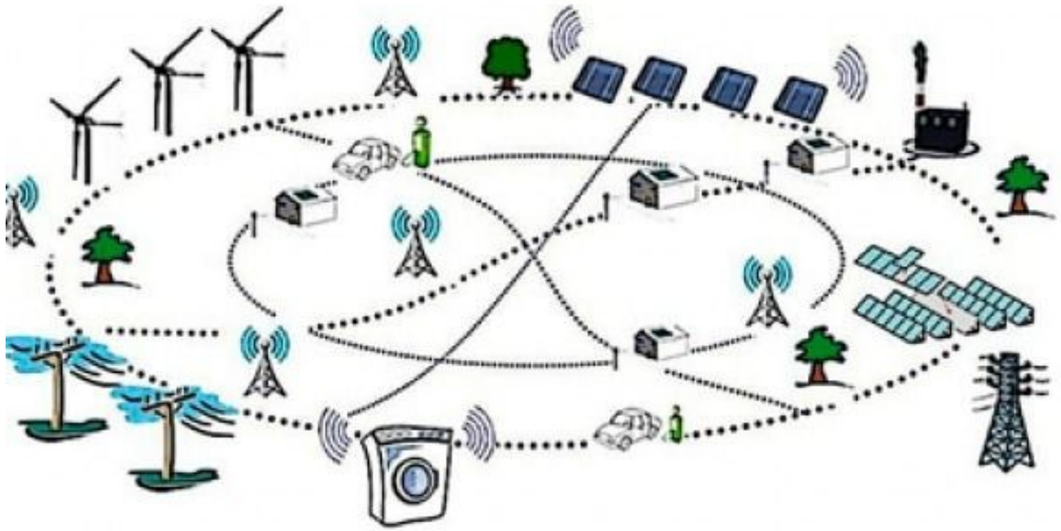
SMART ENERGY GRIDS

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



Blockchain vs. The Internet

READINGS

- **Finish** reading the Ethereum Whitepaper
- Take a look at these cool Ethereum Applications
  - https://www.stateofthedapps.com/

References

Slides mainly adopted from

- Blockchain @ Berkeley : https://blockchain.berkeley.edu/
- Blockchain @ Princeton : http://bitcoinbook.cs.princeton.edu/