The Blockchain: What It is & Why It Matters to Us

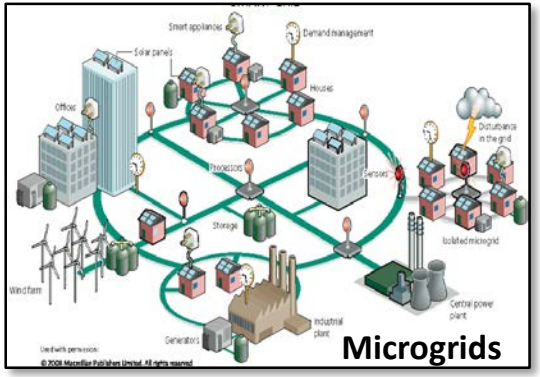
Douglas C. Schmidt & Abhishek Dubey
Vanderbilt University

This work has been funded in part by Siemens, Varian, & Accenture

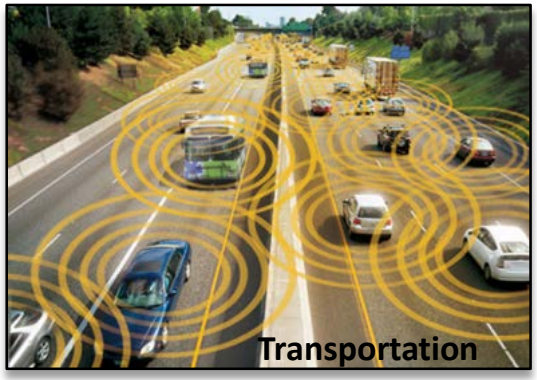
Overview of the Presentation

Overview of the Presentation

- Motivate why IoT systems need new mechanisms to decentralize control & information



Microgrids

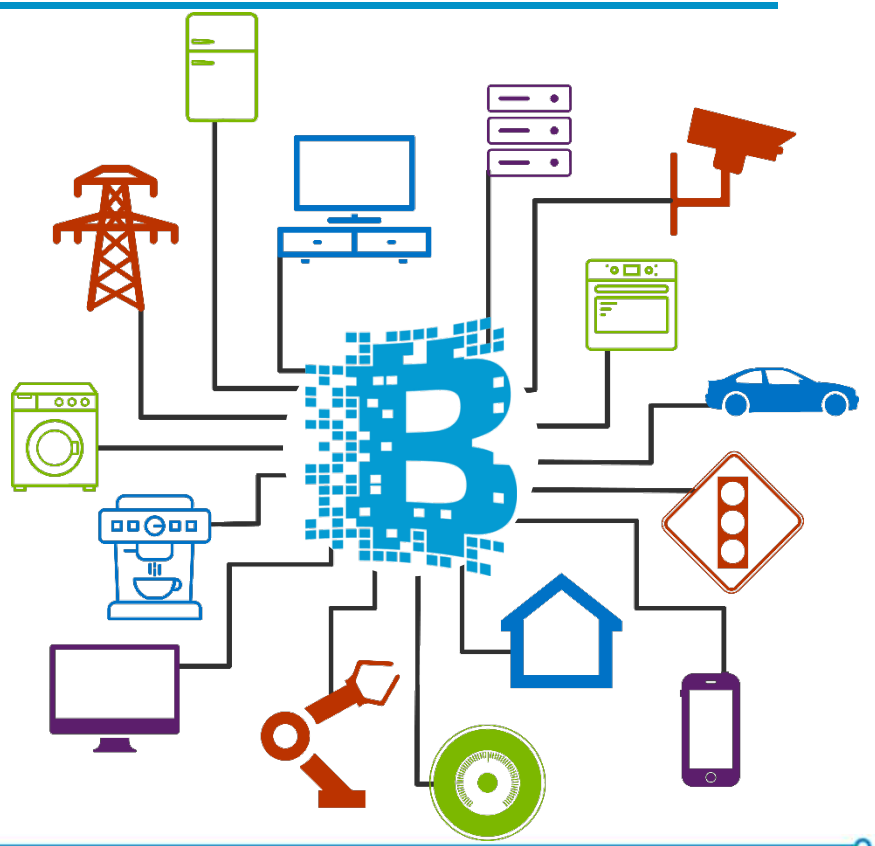

Transportation

Health Care

Smart Buildings

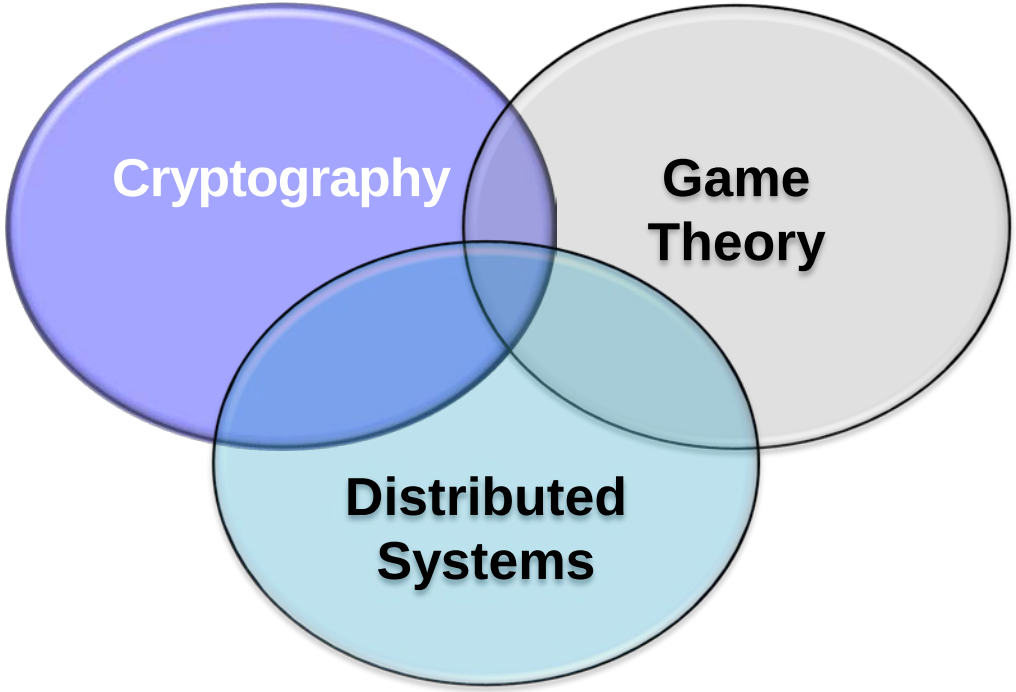
Overview of the Presentation

- Motivate why IoT systems need new mechanisms to decentralize control & information
- Provide an introduction to the blockchain & why it matters to the middleware IOT community



Overview of the Presentation

- Motivate why IoT systems need new mechanisms to decentralize control & information
- Provide an introduction to the blockchain & why it matters to the middleware IOT community
- Explain the technical foundations of blockchain



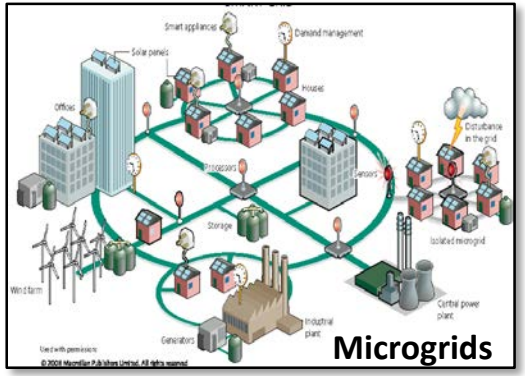
Overview of the Presentation

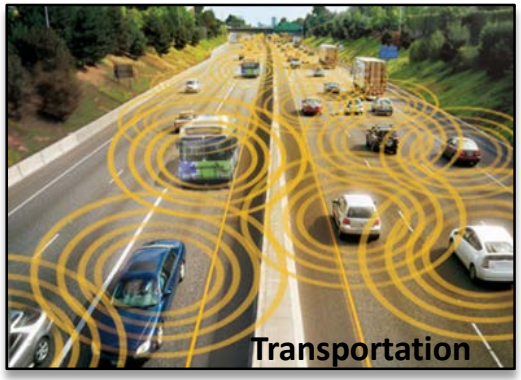
- Motivate why IoT systems need new mechanisms to decentralize control & information
- Provide an introduction to the blockchain & why it matters to the middleware IOT community
- Explain the technical foundations of blockchain
- Explore challenges to applying blockchain for various domains, including IOT & beyond

Motivation: Decentralized Control & Information in IoT Systems

Decentralized Control & Information in IoT Systems

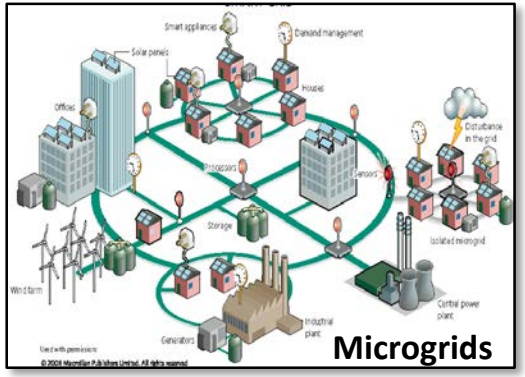
- An increasing # of IoT systems require decentralized control & information management

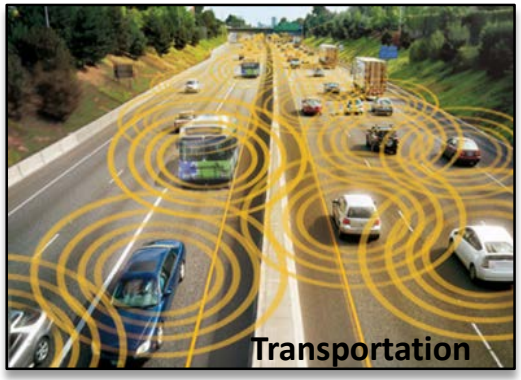




Decentralized Control & Information in IoT Systems

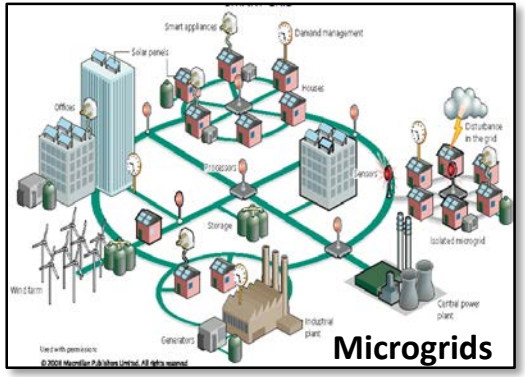
- An increasing # of IoT systems require decentralized control & information management, e.g.
  - Real-time information dissemination & time-synchronized task execution



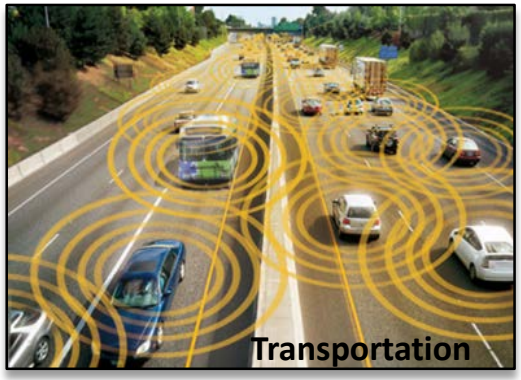


Decentralized Control & Information in IoT Systems

- An increasing # of IoT systems require decentralized control & information management, e.g.
  - Real-time information dissemination & time-synchronized task execution
  - Preserve information integrity across all actors in the system



Microgrids

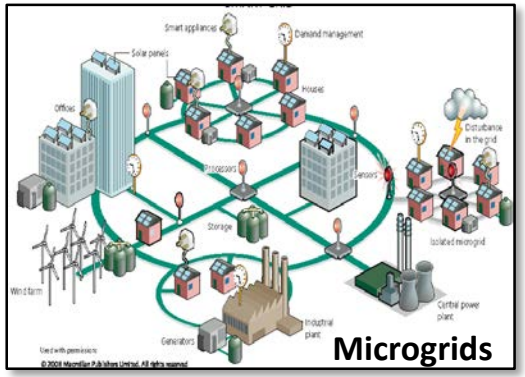

Transportation

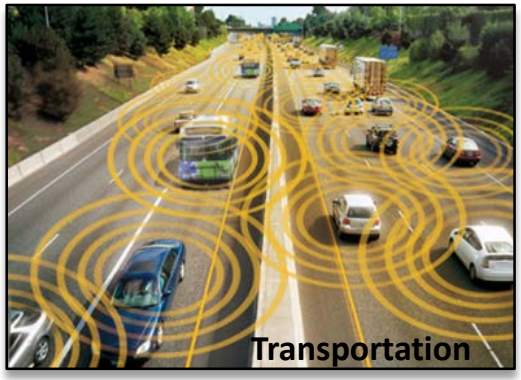
Health Care

Smart Buildings

Decentralized Control & Information in IoT Systems

- An increasing # of IoT systems require decentralized control & information management





A key challenge is how to develop these types of systems most effectively

Emerging Trend: Digital Asset Management

What is a Digital Asset?

- A digital asset is anything existing in a binary format that comes with (some) rights to use

See en.wikipedia.org/wiki/Digital_asset

What is a Digital Asset?

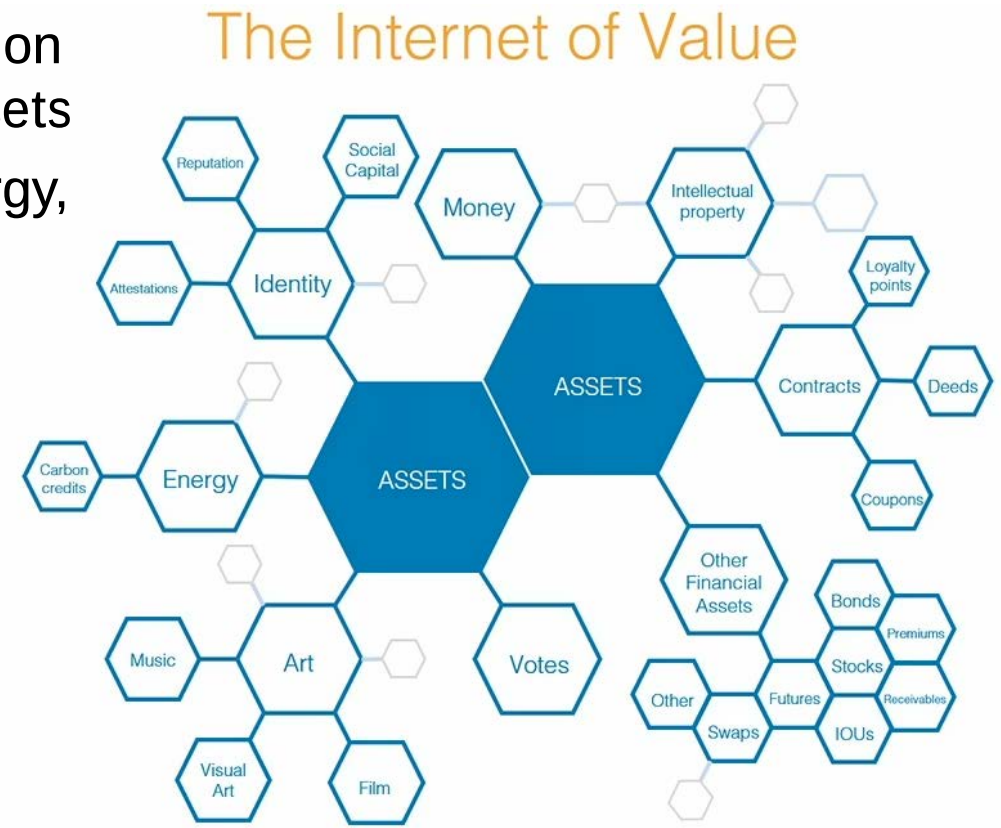
- A digital asset is anything existing in a binary format that comes with (some) rights to use
  - **Native digital assets**
    - An asset lacking physical substance that can be owned or controlled to produce value
      - e.g., digital music, images, movies, electronic funds, software, etc.

What is a Digital Asset?

- A digital asset is anything existing in a binary format that comes with (some) rights to use
  - Native digital assets
  - **Digital representations of traditional assets**
    - Assets represented by paper certificates or titles
      - e.g., property, gold, autos, stock, currency, etc.

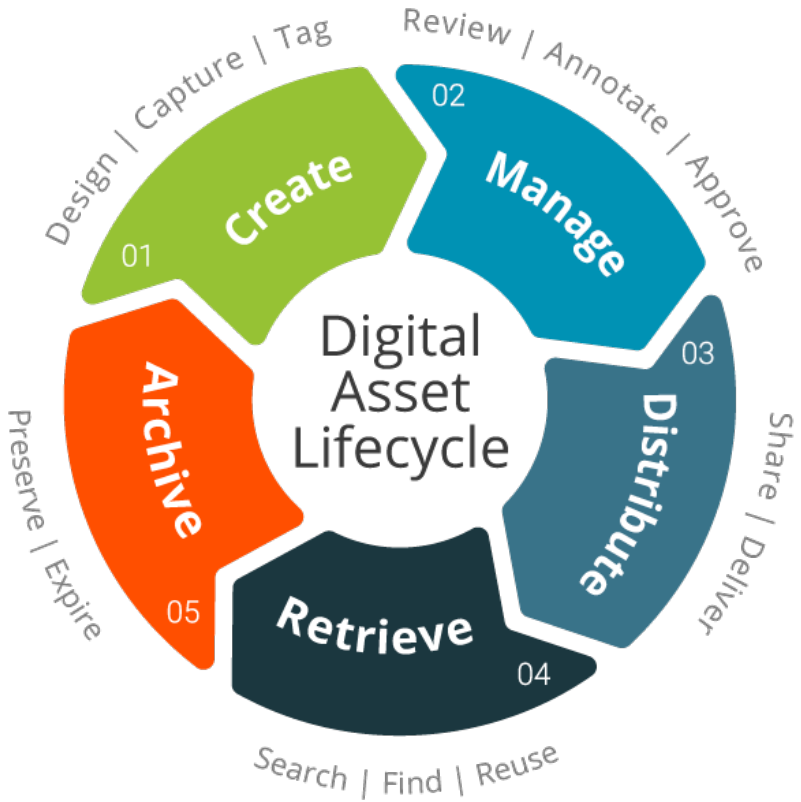
What is a Digital Asset?

- Our economy increasingly depends on effective management of digital assets
  - e.g., entertainment, finance, energy, defense, elections, reputation in social networks, etc.



What is a Digital Asset?

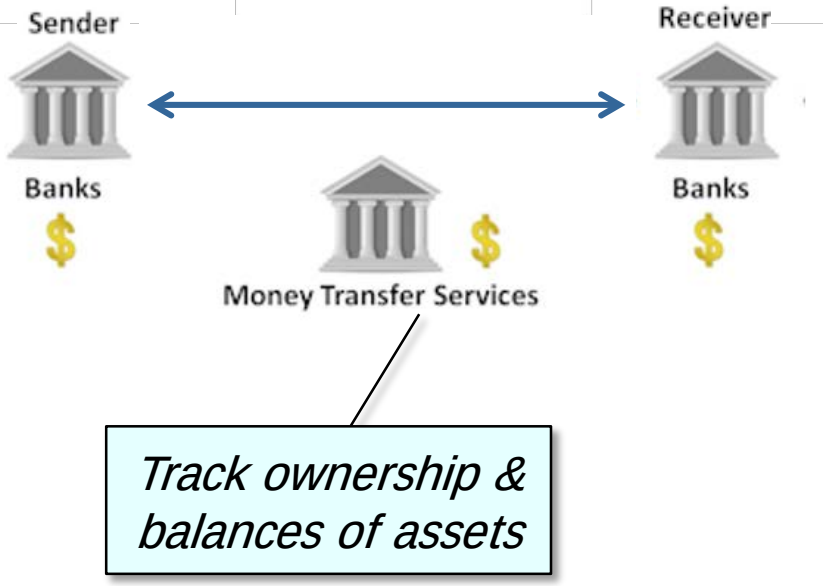
- There are some common operations on digital assets



See en.wikipedia.org/wiki/Digital_asset_management

What is a Digital Asset?

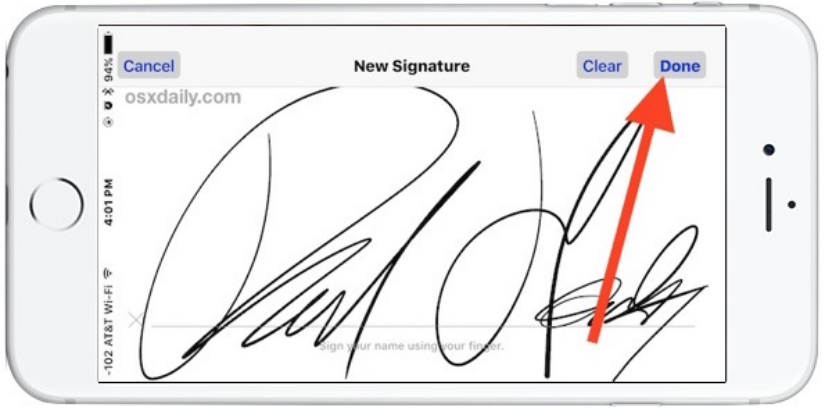
- There are some common operations on digital assets, e.g.
  - Secure online transactions
    - e.g., record & transfer digital assets between sender & receiver



See newdigitalnoise.com/ever-expanding-digital-world

What is a Digital Asset?

- There are some common operations on digital assets, e.g.
  - Secure online transactions
  - Identity attestation & provenance tracking
    - e.g., allow an identity/entity to make a claim that can be verified later



See www.accenture.com/us-en/blogs/blogs-david-treat-digital-identity

What is a Digital Asset?

- Digital assets have historically been managed by centralized/proprietary "brokers"

A broker serves as a "trusted intermediary"

What is a Digital Asset?

- Digital assets have historically been managed by centralized/proprietary "brokers", e.g.
  - Financial institutions

What is a Digital Asset?

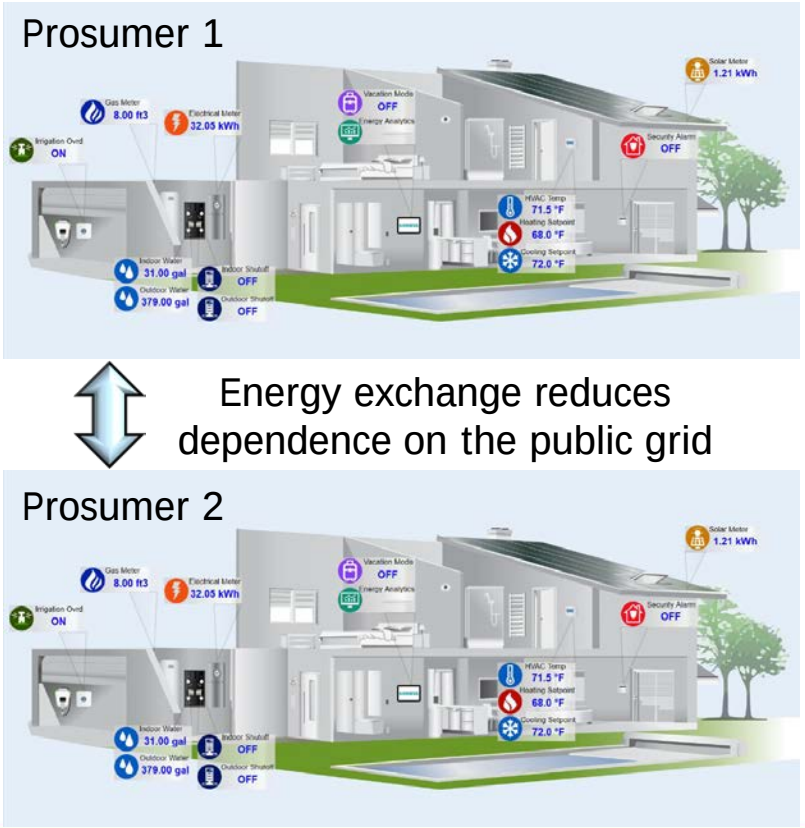
- Digital assets have historically been managed by centralized/proprietary "brokers", e.g.
  - Financial institutions
  - Credit agencies

What is a Digital Asset?

- Digital assets have historically been managed by centralized/proprietary "brokers", e.g.
  - Financial institutions
  - Credit agencies
  - Electronic medical records

What is a Digital Asset?

- Digital assets have historically been managed by centralized/proprietary "brokers", e.g.
  - Financial institutions
  - Credit agencies
  - Electronic medical records
  - Energy transactions in a "micro grid"

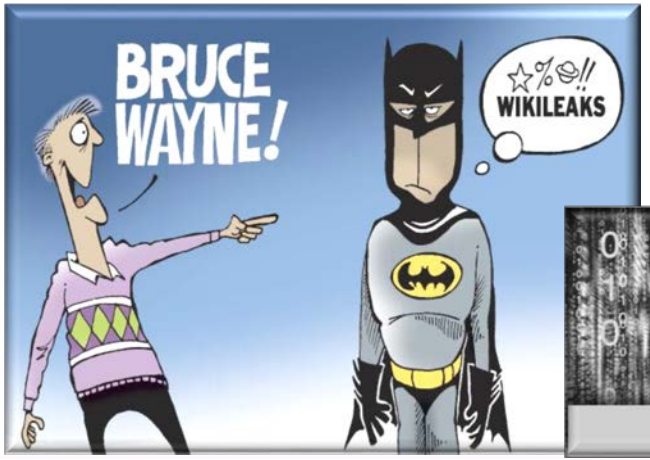


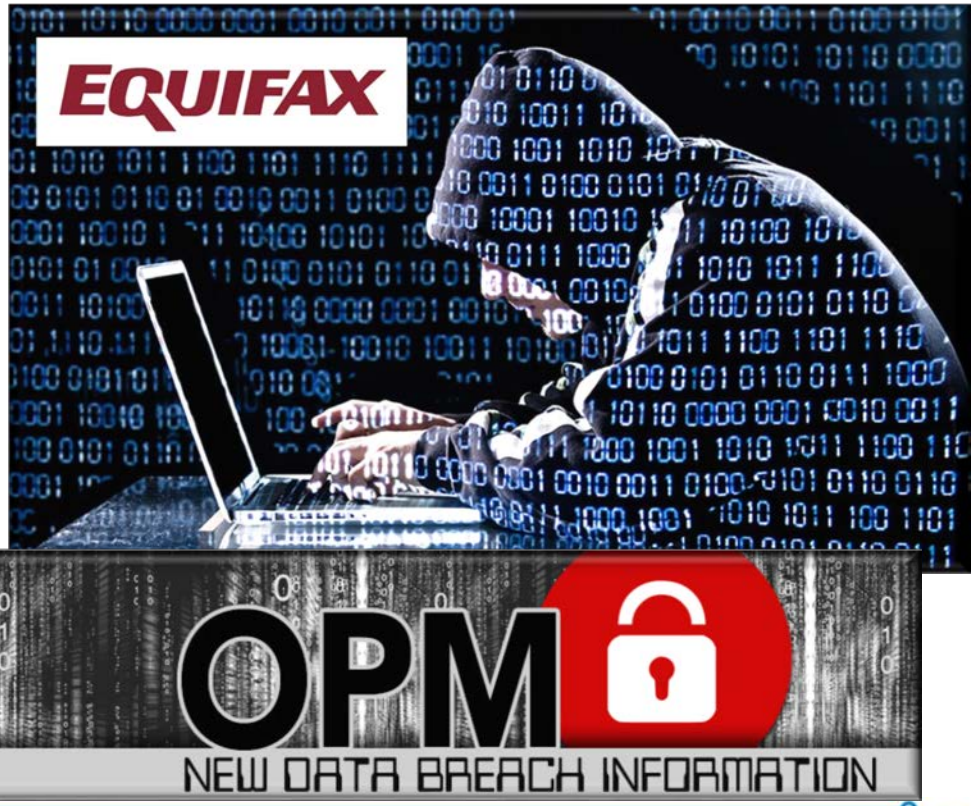
What is a Digital Asset?

- However, centralized/proprietary brokers have key limitations

What is a Digital Asset?

- However, centralized/proprietary brokers have key limitations, e.g.
  - Insecure
    - e.g., numerous data breaches of brokers in recent years





See theconversation.com/why-dont-big-companies-keep-their-computer-systems-up-to-date-84250

What is a Digital Asset?

- However, centralized/proprietary brokers have key limitations, e.g.
  - Insecure
  - Expensive
    - Brokers “take their cut,” which increases costs to providers & consumers

What is a Digital Asset?

- However, centralized/proprietary brokers have key limitations, e.g.
  - Insecure
  - Expensive
  - Vulnerable to abuse
    - Centralized brokers concentrate great power, wealth, & control into a small # of institutions

See en.wikipedia.org/wiki/Criticism_of_the_Federal_Reserve

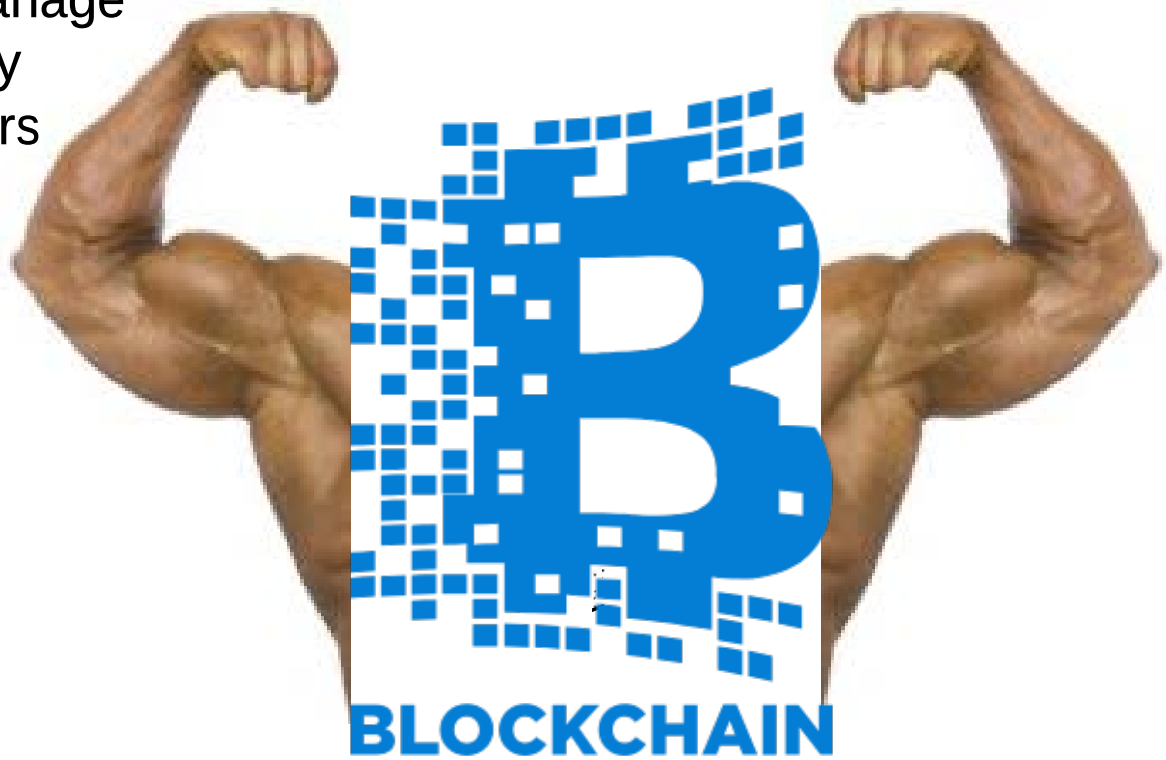
What is a Digital Asset?

- However, centralized/proprietary brokers have key limitations, e.g.
  - Insecure
  - Expensive
  - Vulnerable to abuse
    - Centralized brokers concentrate great power, wealth, & control into a small # of institutions
      - Can be both a blessing & a curse..

e.g., (questionable) transactions can be quickly & easily reversed

What is a Digital Asset?

- Blockchain is designed to manage digital assets more effectively by “disintermediating” brokers



What is a Digital Asset?

- Blockchain is designed to manage digital assets more effectively by “disintermediating” brokers

Blockchain is most useful when disintermediation is more important than strict confidentiality...

What is a Blockchain?

What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions

See en.wikipedia.org/wiki/Blockchain

What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions

Used as computational substrate for “crypto-currencies”, plus ***more***

What is a Blockchain?

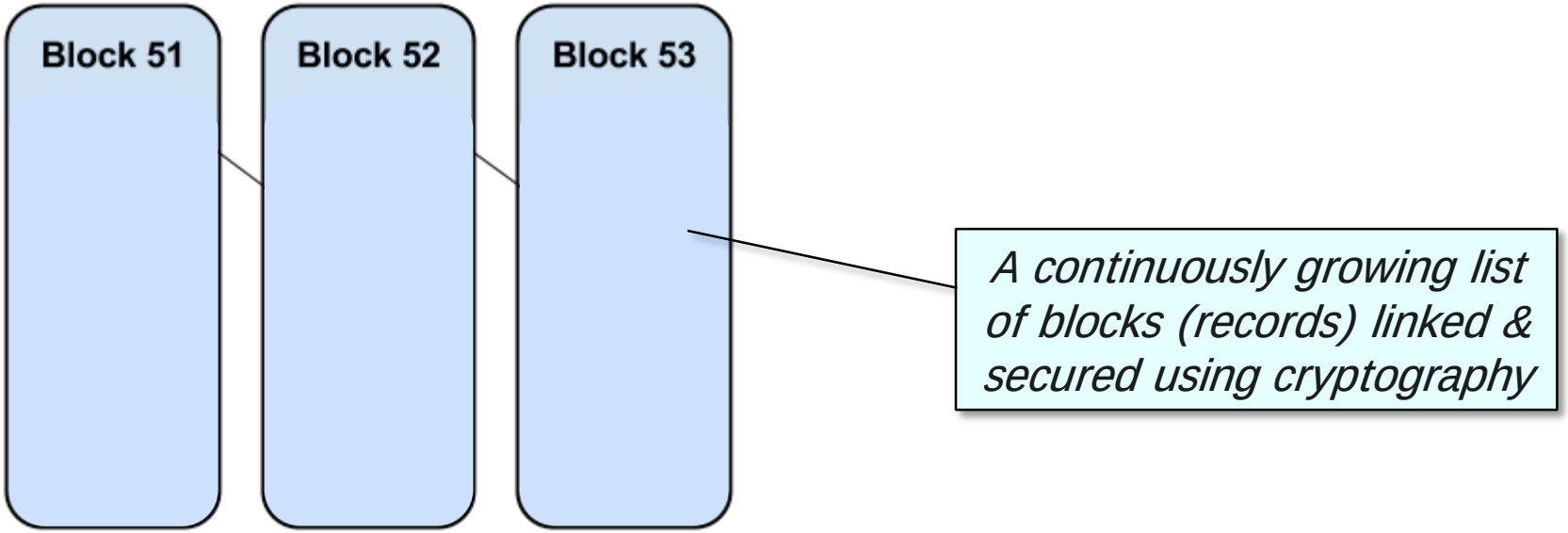
- A blockchain is a decentralized platform that supports “trustless” transactions

Used as computational substrate for “crypto-currencies”, plus ***more***

A cryptocurrency is a digital asset that uses cryptography to secure transactions, control the creation of additional units, & verify asset transfer

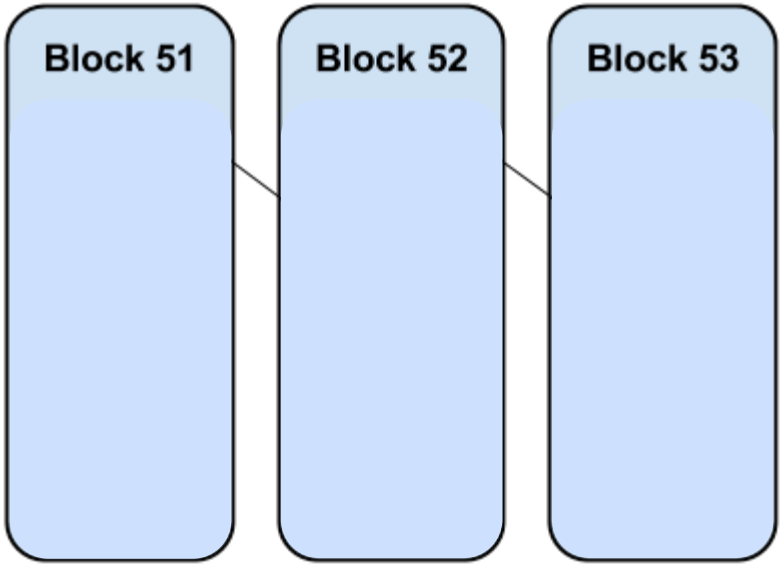
What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions



What is a Blockchain?

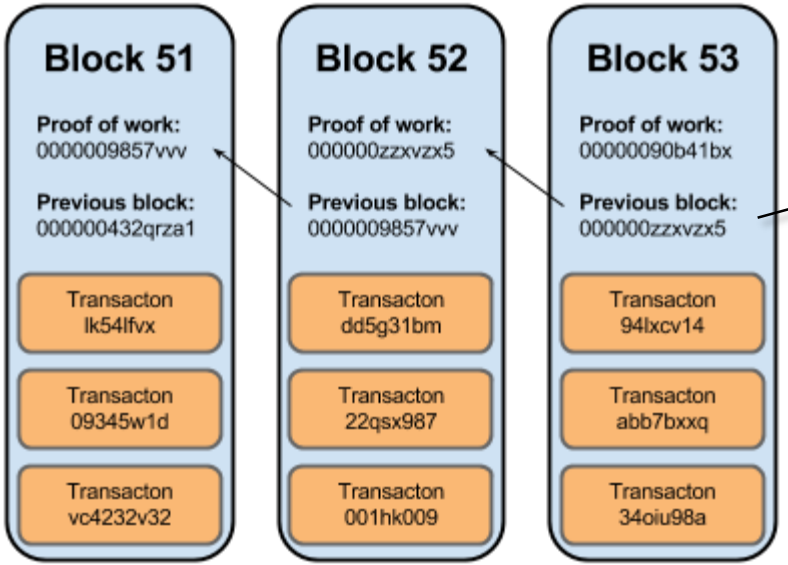
- A blockchain is a decentralized platform that supports “trustless” transactions



This chain of blocks provides an open, distributed ledger that immutably records transactions between two parties efficiently & verifiably

What is a Blockchain?

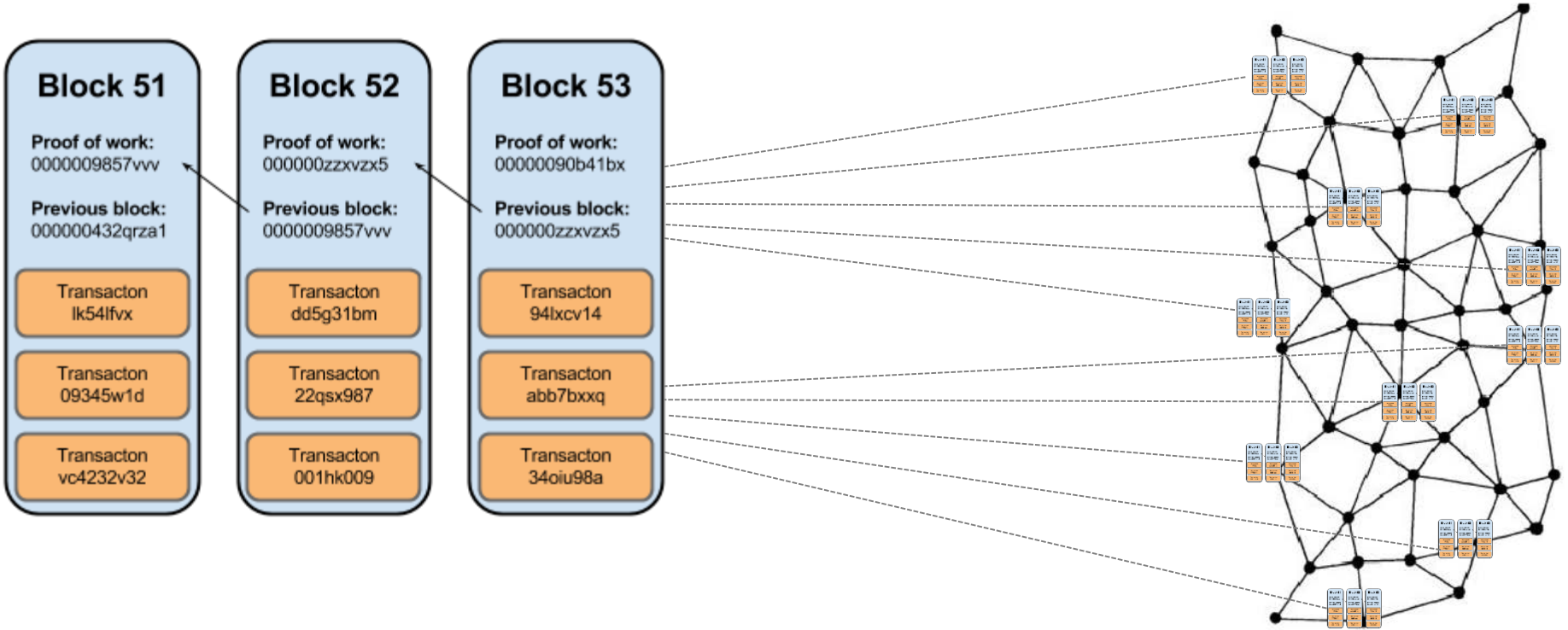
- A blockchain is a decentralized platform that supports “trustless” transactions



A block contains transaction data, a timestamp, & a hash pointer that links to the previous block (which forms the “chain” of blocks)

What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions

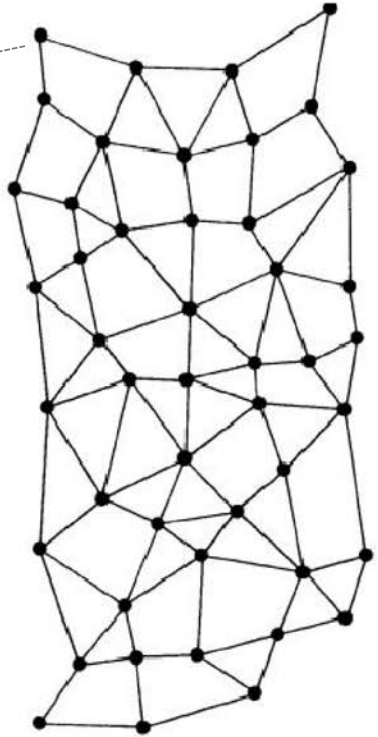


These blocks are replicated across (many) computers, rather than being stored on a central server

What is a Blockchain?

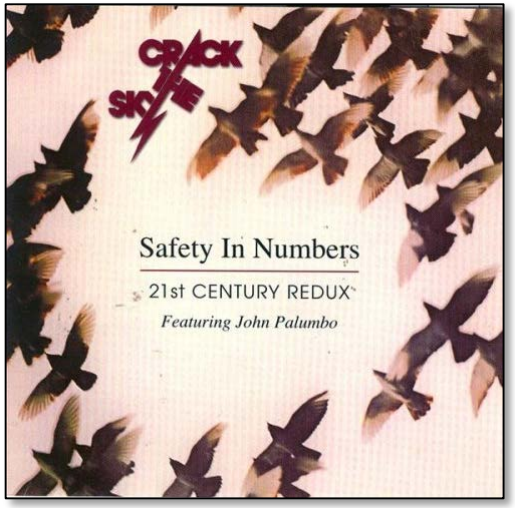
- A blockchain is a decentralized platform that supports “trustless” transactions

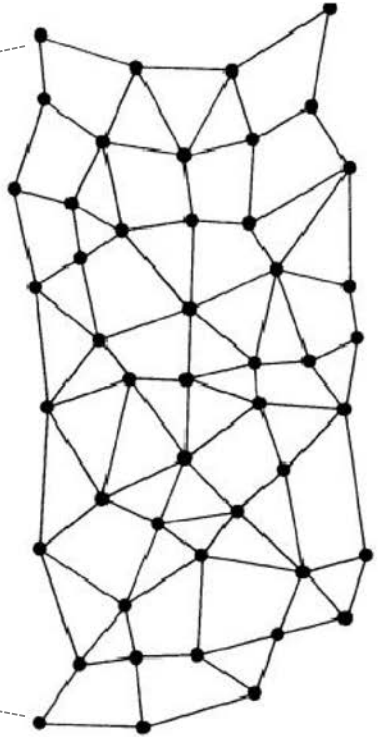
This platform may be distributed globally (public blockchain)



What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions



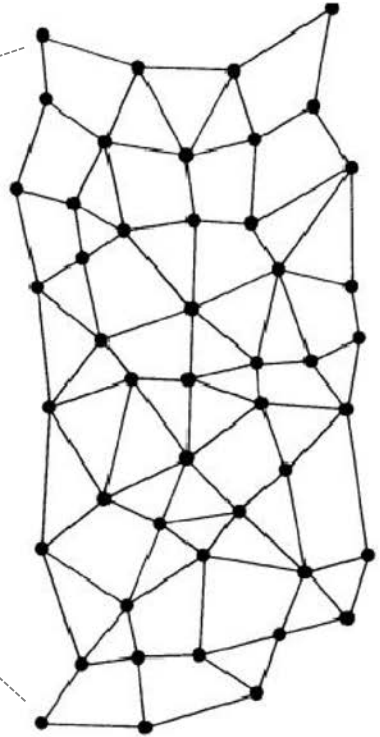


Public blockchains are less efficient, but most useful when not all participants can be trusted to behave

What is a Blockchain?

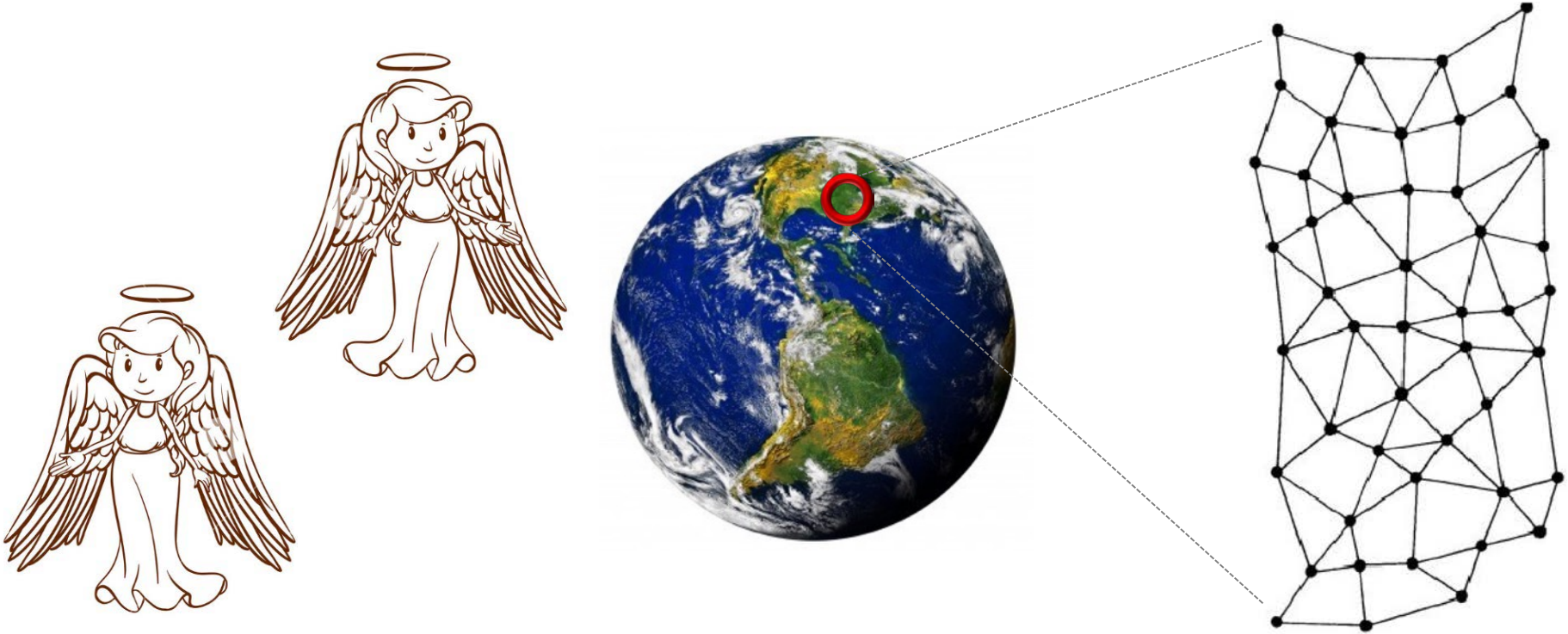
- A blockchain is a decentralized platform that supports “trustless” transactions

It could be localized to a limited group (private blockchain)



What is a Blockchain?

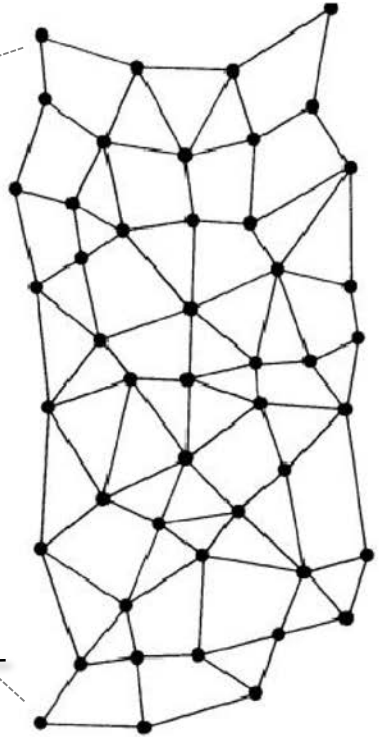
- A blockchain is a decentralized platform that supports “trustless” transactions



Private blockchains are more efficient, but often assume that participants can be trusted to behave

What is a Blockchain?

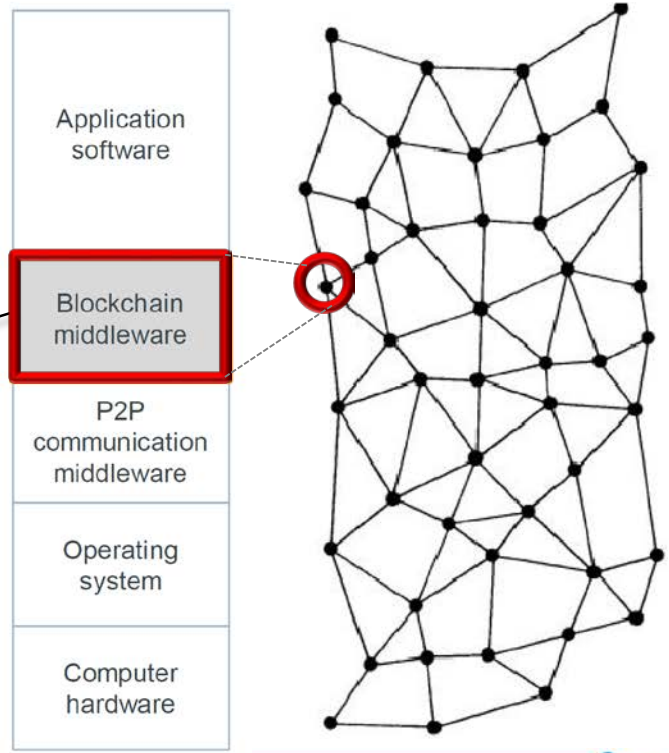
- A blockchain is a decentralized platform that supports “trustless” transactions



In either case, nodes have a high degree of independence of control & processing in a peer-to-peer (inter)network

What is a Blockchain?

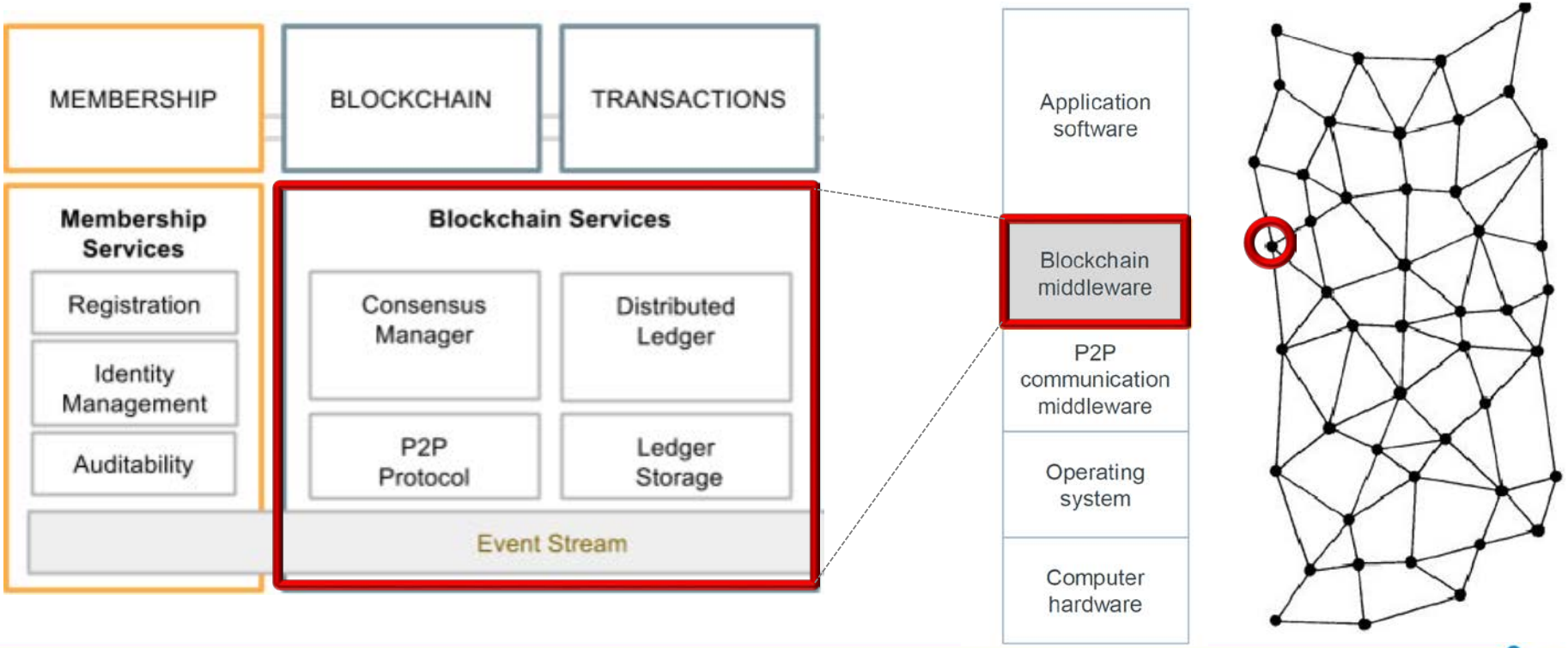
- A blockchain is a decentralized platform that supports “trustless” transactions



Each node in a blockchain network runs common middleware & all nodes have equal level of privilege & access

What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions

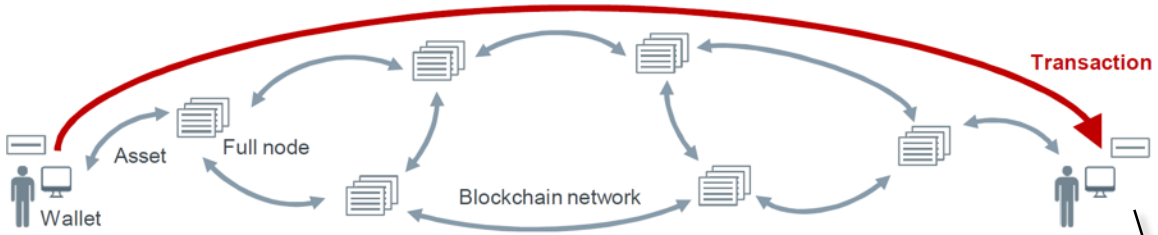


Blockchain middleware provides services to applications beyond what's provided by the OS & communication protocols

What is a Blockchain?

- A blockchain is a decentralized platform that supports "trustless" transactions

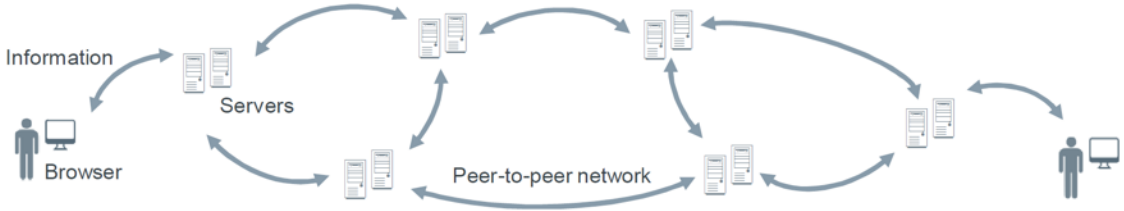
DIGITAL ASSET TRANSACTION
Record & transfer



- **Blockchain** protocol
- Immutable replicated database
- Identity, reputation

ON TOP OF

TRADITIONAL INTERNET
Store & copy

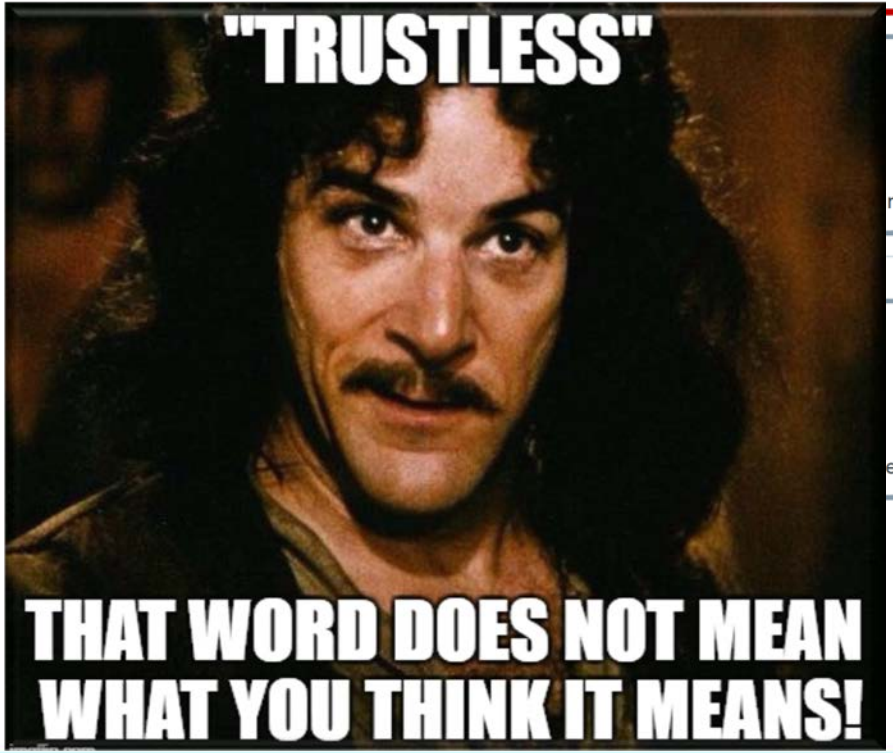


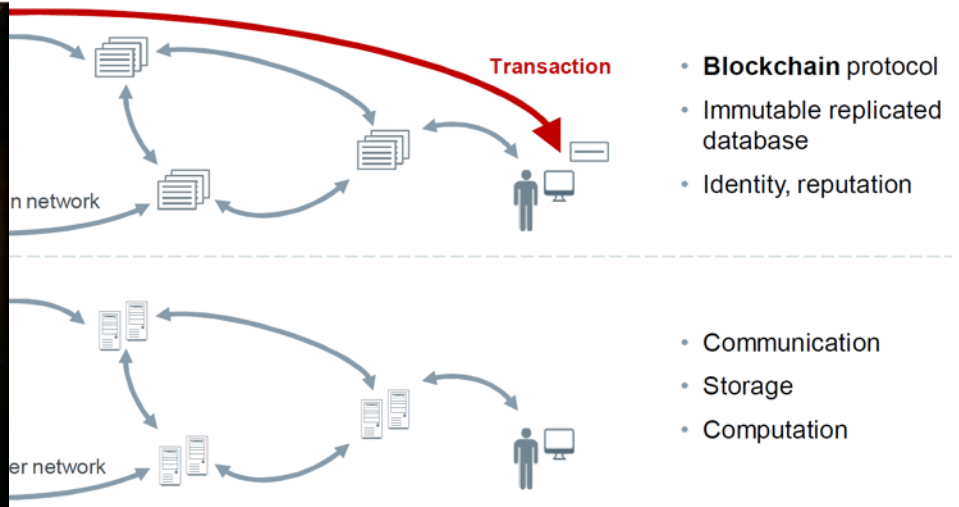
- Communication
- Storage
- Computation

Blockchain middleware enables anonymous exchange of digital assets without the need for a central authority to verify trust & transfer of value

What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions



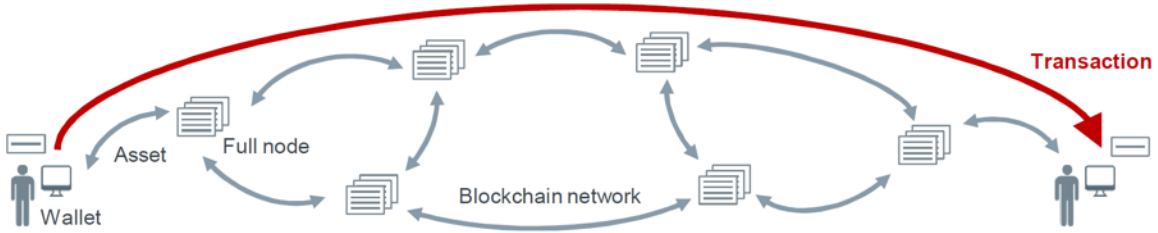


What is a Blockchain?

- A blockchain is a decentralized platform that supports "trustless" transactions

DIGITAL ASSET TRANSACTION

Record & transfer

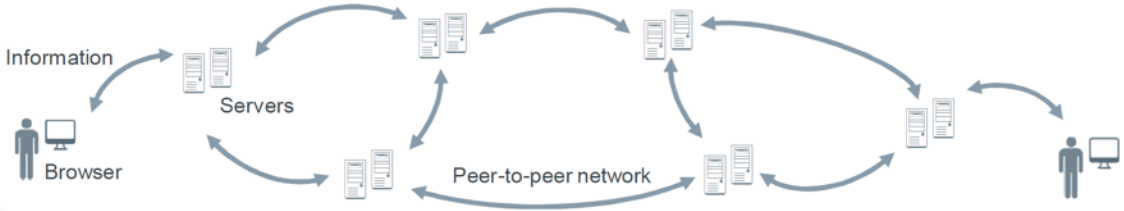


- **Blockchain** protocol
- Immutable replicated database
- Identity, reputation

ON TOP OF

TRADITIONAL INTERNET

Store & copy

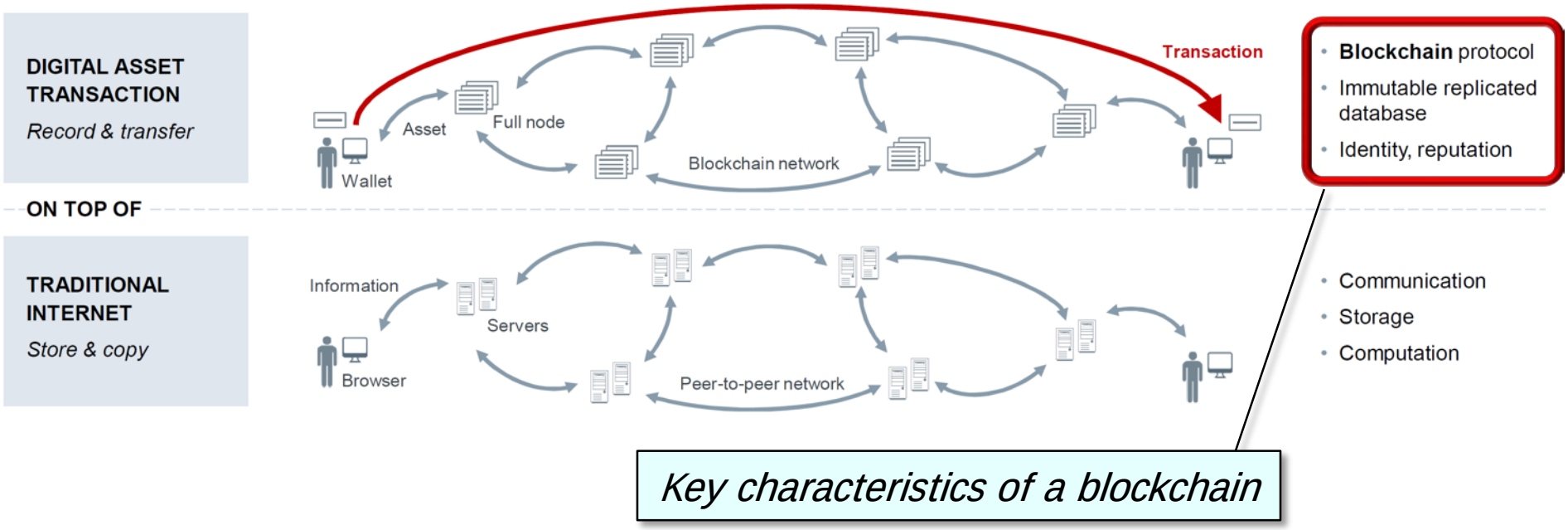


- Communication
- Storage
- Computation

Blockchain middleware enables digital asset transactions atop the traditional Internet

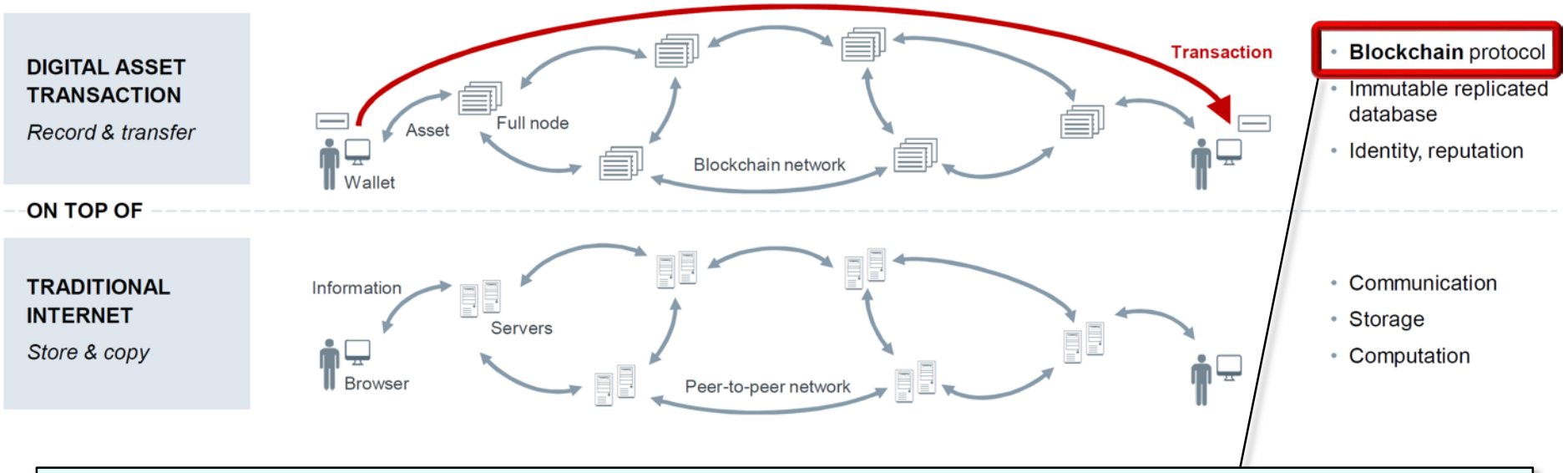
What is a Blockchain?

- A blockchain is a decentralized platform that supports "trustless" transactions



What is a Blockchain?

- A blockchain is a decentralized platform that supports "trustless" transactions

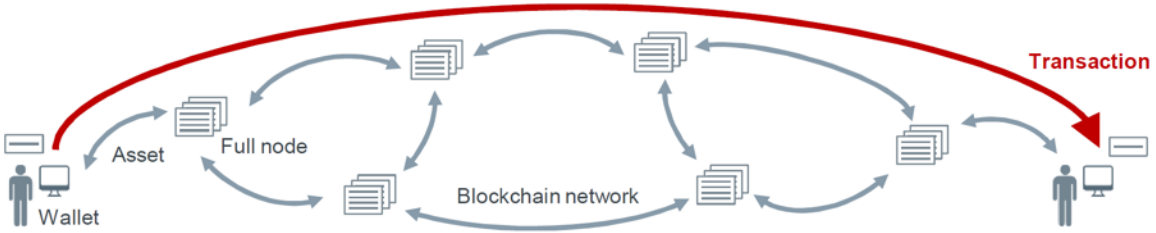


Ensures a common, unambiguous ordering of blocks & guarantees the (eventual) integrity & consistency of the blockchain across (geographically) distributed nodes

What is a Blockchain?

- A blockchain is a decentralized platform that supports "trustless" transactions

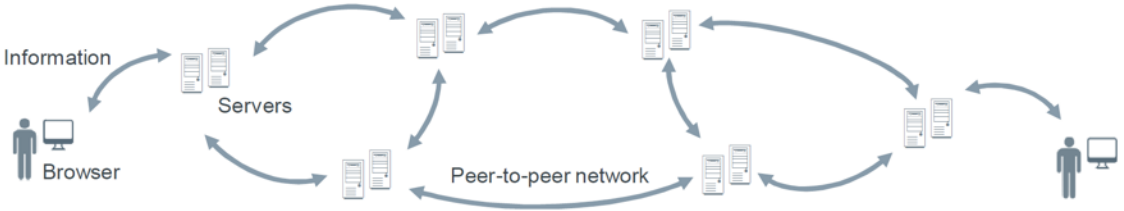
DIGITAL ASSET TRANSACTION
Record & transfer



- **Blockchain** protocol
- Immutable replicated database
- Identity, reputation

ON TOP OF

TRADITIONAL INTERNET
Store & copy



- Communication
- Storage
- Computation

A database containing immutable time-stamped information for each block that's replicated on servers (may be around the world)

What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions

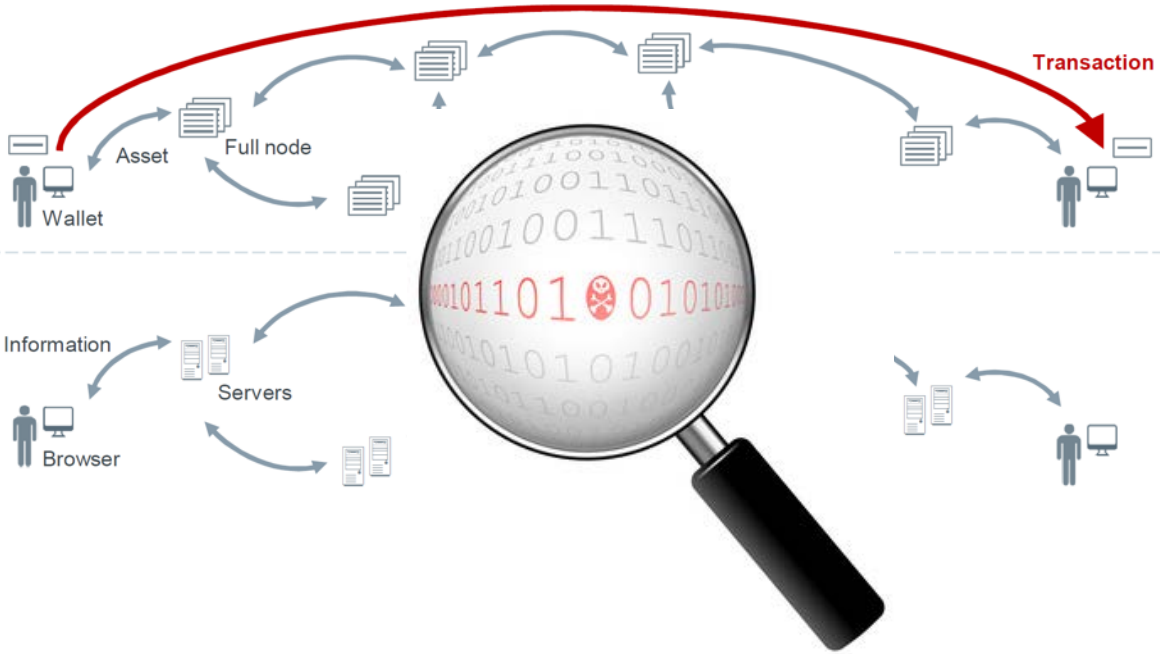
DIGITAL ASSET TRANSACTION

Record & transfer

ON TOP OF

TRADITIONAL INTERNET

Store & copy



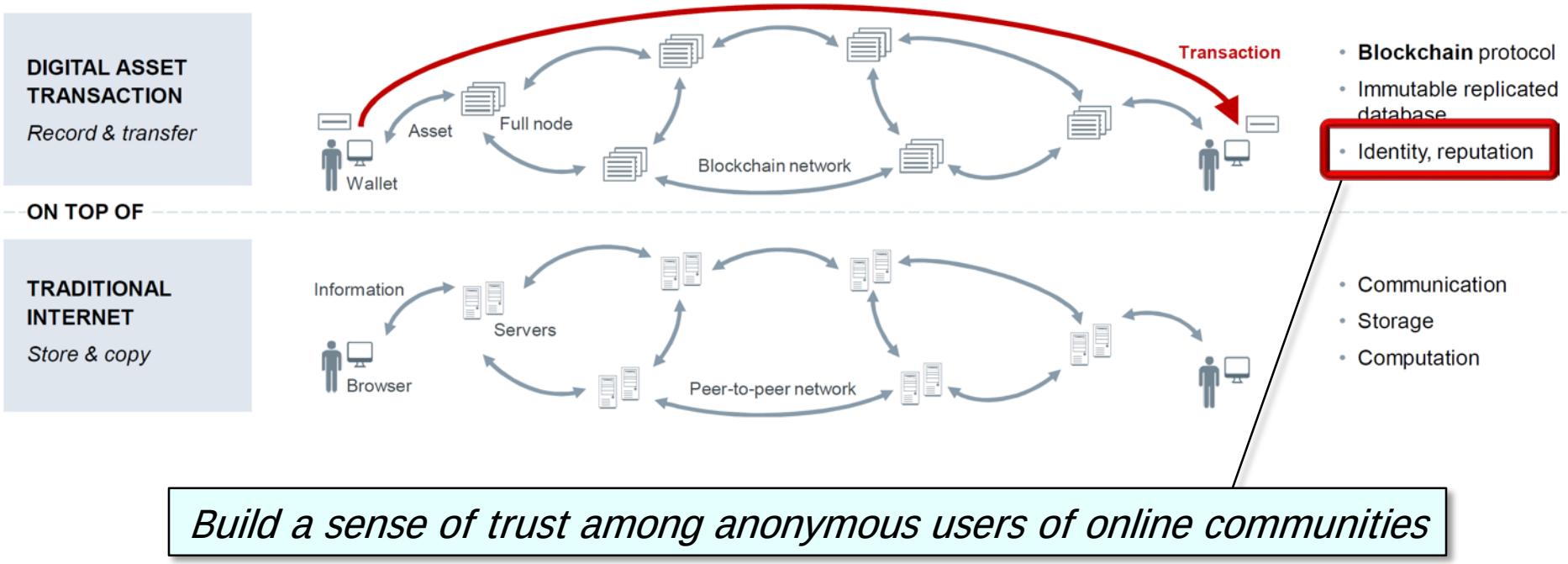
- **Blockchain** protocol
- Immutable replicated database
- Identity, reputation

- Communication
- Storage
- Computation

It’s very hard to change a blockchain without collusion & it’s very easy to detect the attempt if anyone tries

What is a Blockchain?

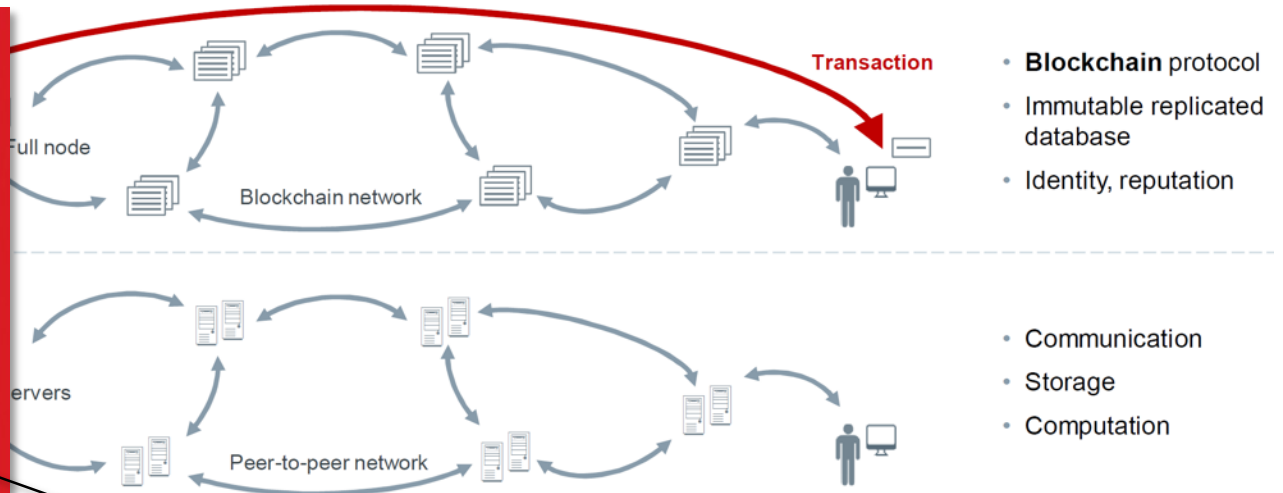
- A blockchain is a decentralized platform that supports "trustless" transactions



Build a sense of trust among anonymous users of online communities

What is a Blockchain?

- A blockchain is a decentralized platform that supports “trustless” transactions



The origins of the blockchain prototol are shrouded in mystery!

See medium.com/cryptomuse/how-the-nsa-caught-satoshi-nakamoto-868affcef595

Why Blockchain Matters

Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service"

Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service"

A key goal is to "disintermediate" centralized brokers, yet still allow multiple parties (who don't trust each other) to share a single database

Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service", e.g.
  - Payments in the financial sector
    - e.g., use on-chain tokens to represent cash, stocks, bonds, etc

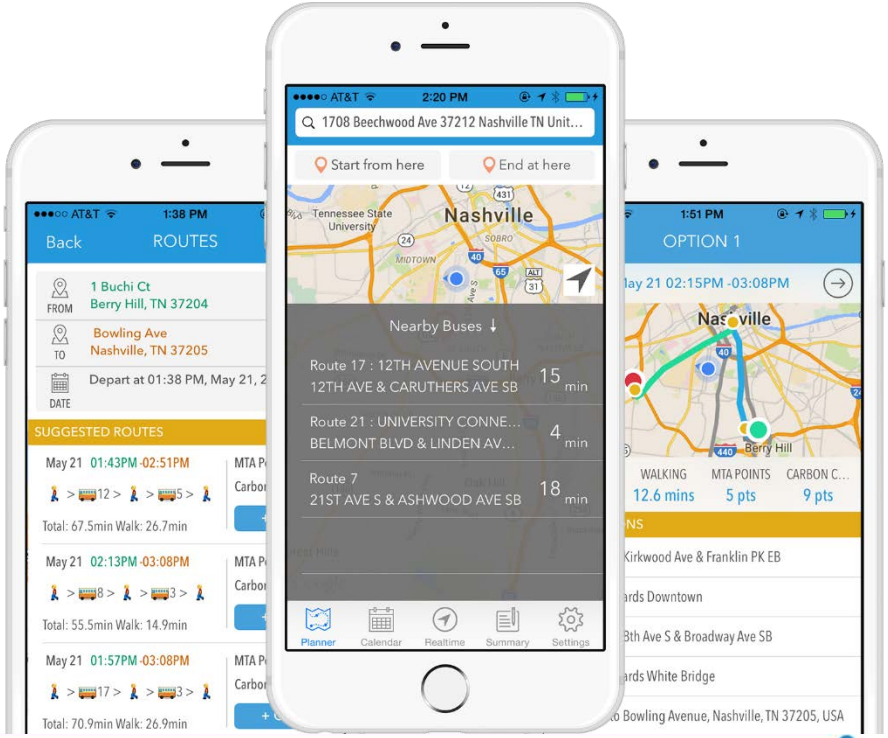
Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service", e.g.
  - Payments in the financial sector
    - e.g., use on-chain tokens to represent cash, stocks, bonds, etc

Turns out to be problematic in practice due to lack of confidentiality..

Why Blockchain Matters

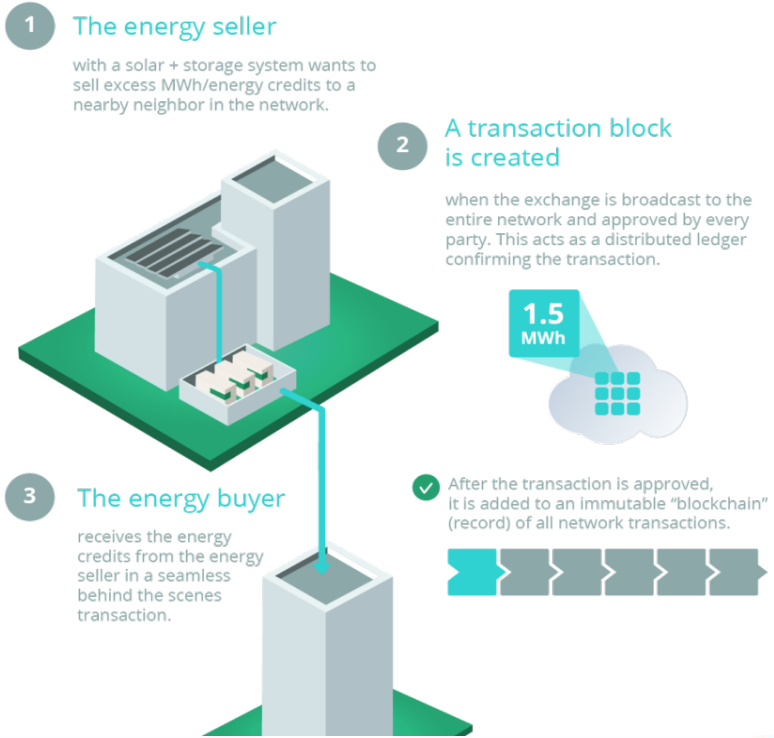
- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service", e.g.
  - Payments in the financial sector
  - Lightweight financial systems, e.g.
    - Streamline interactions between commuters & multi-modal transit



See www.vanderbilt.edu/strategicplan/undergraduate-residential-education/universitycourses-2017/smart-city-applications.php

Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service", e.g.
  - Payments in the financial sector
  - Lightweight financial systems, e.g.
    - Streamline interactions between commuters & multi-modal transit
    - Improve efficiency of energy transactions in "smart grids"



See www.dre.vanderbilt.edu/~schmidt/PDF/IOT-2017.pdf

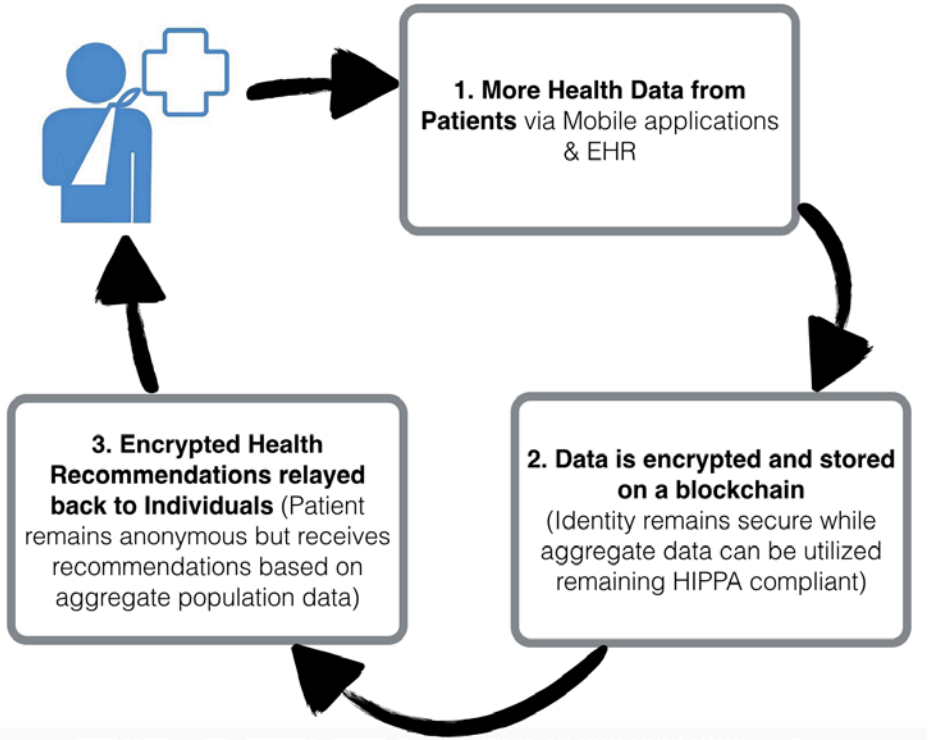
Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service", e.g.
  - Payments in the financial sector
  - Lightweight financial systems, e.g.
    - Streamline interactions between commuters & multi-modal transit
    - Improve efficiency of energy transactions in "smart grids"
    - UN provides thousands of Syrian refugees in Jordan with food, clothing, & other aid in a cost effective manner

See www.coindesk.com/united-nations-sends-aid-to-10000-syrian-refugees-using-ethereum-blockchain

Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service", e.g.
  - Payments in the financial sector
  - Lightweight financial systems
  - Interorganizational record keeping
    - e.g., provide providers, patients, (& surrogates) better access to —& control over—health info

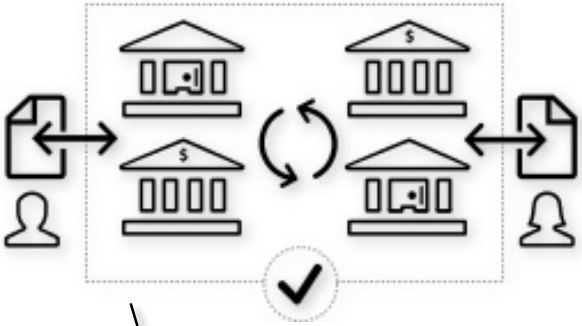


See www.dre.vanderbilt.edu/~schmidt/PDF/PLoP-2017-blockchain.pdf

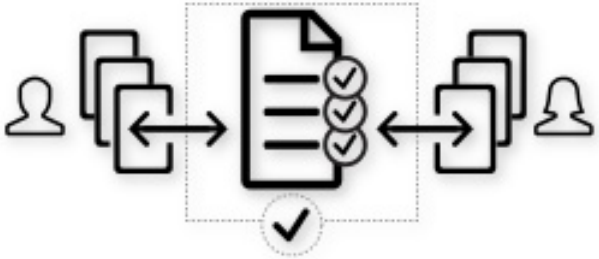
Why Blockchain Matters

- Blockchain helps address needs in various domains by providing decentralized "transactions-as-a-service", e.g.
  - Payments in the financial sector
  - Lightweight financial systems
  - Interorganizational record keeping

Centralized Broker Model



Blockchain offers an alternative to the use of centralized brokers, which may yield significant savings in hassle & cost



Disintermediated Blockchain Model

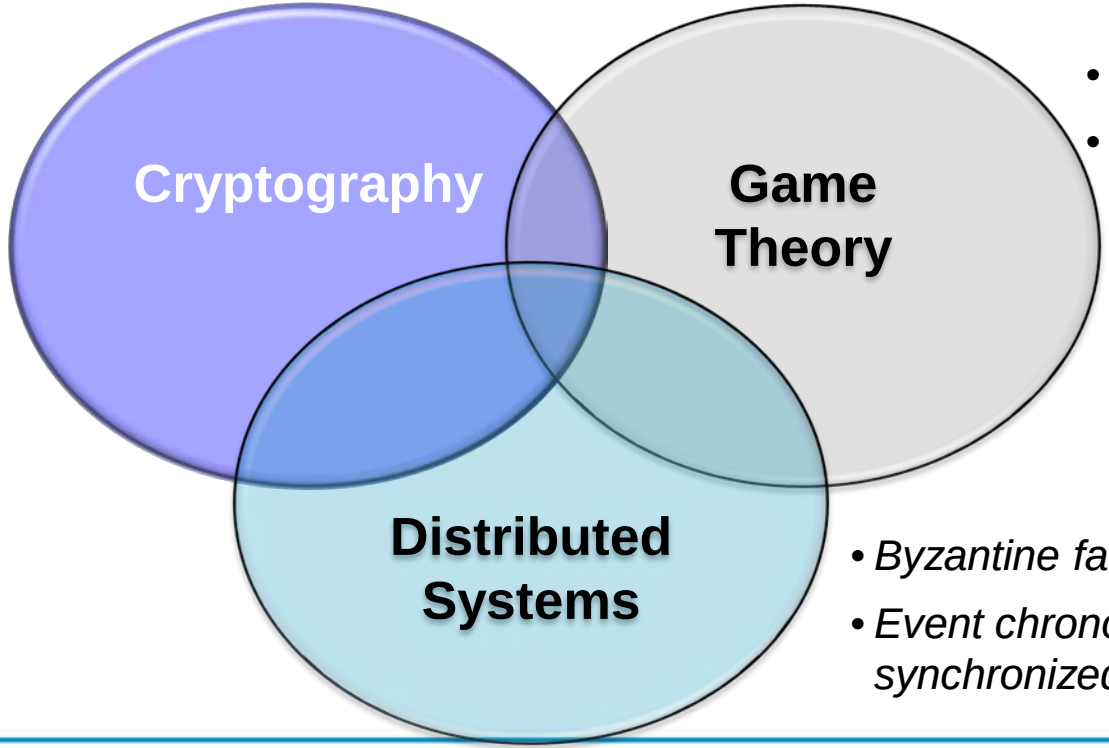
See www.multichain.com/blog/2016/05/four-genuine-blockchain-use-cases

Technical Foundations of Blockchain

Technical Foundations of Blockchain

Blockchains integrates three branches of computer science & mathematics

- *Authentication & hashing*
- *Assurance or non-repudiation*

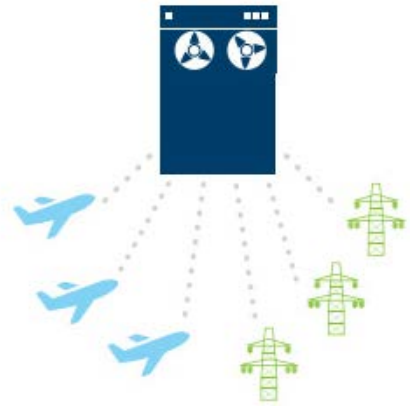


- *Incentivization*
- *Maximizing long-term payoff*

- *Byzantine fault tolerance*
- *Event chronology without synchronized clock*

Key Problem: Consistent Record Keeping in Distributed Systems

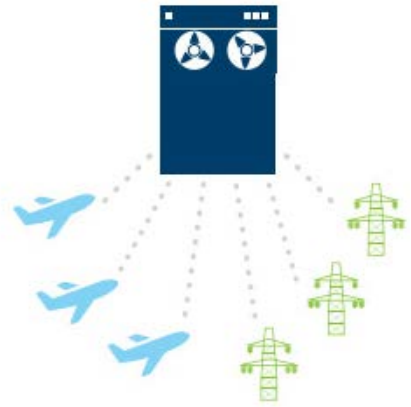
Centralized Database System



- Central authority, exclusive keeper of “ground truth”
- Maybe replicated to a redundant stand-by system for higher availability
- e.g., traditional Enterprise DB

Key Problem: Consistent Record Keeping in Distributed Systems

Centralized Database System



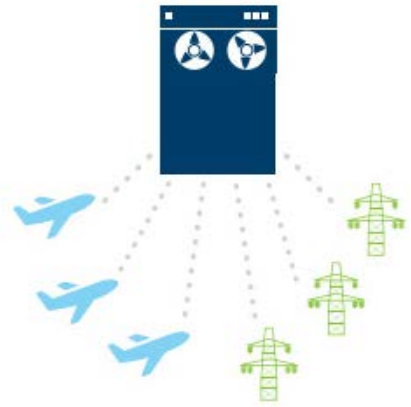
- Central authority, exclusive keeper of “ground truth”
- Maybe replicated to a redundant stand-by system for higher availability
- e.g., traditional Enterprise DB

Distributed Database System

- Consistency mechanisms for guaranteeing completeness & immutability
- Full or partial replication
- Distributed queries & transactions
- e.g., Google Spanner

Key Problem: Consistent Record Keeping in Distributed Systems

Centralized Database System



- Central authority, exclusive keeper of “ground truth”
- Maybe replicated to a redundant stand-by system for higher availability
- e.g., traditional Enterprise DB

Distributed Database System

- Consistency mechanisms for guaranteeing completeness & immutability
- Full or partial replication
- Distributed queries & transactions
- e.g., Google Spanner

Peer-to-Peer (P2P) Database System

- Decentralized Control
- Full or partial replications
- Highest reliability, worst latency
- e.g., Torrents, Napster

Key Problem: Consistent Record Keeping in Distributed Systems

The UN is using Ethereum's blockchain technology to provide aid to 1,000s of Syrian refugees in Jordan

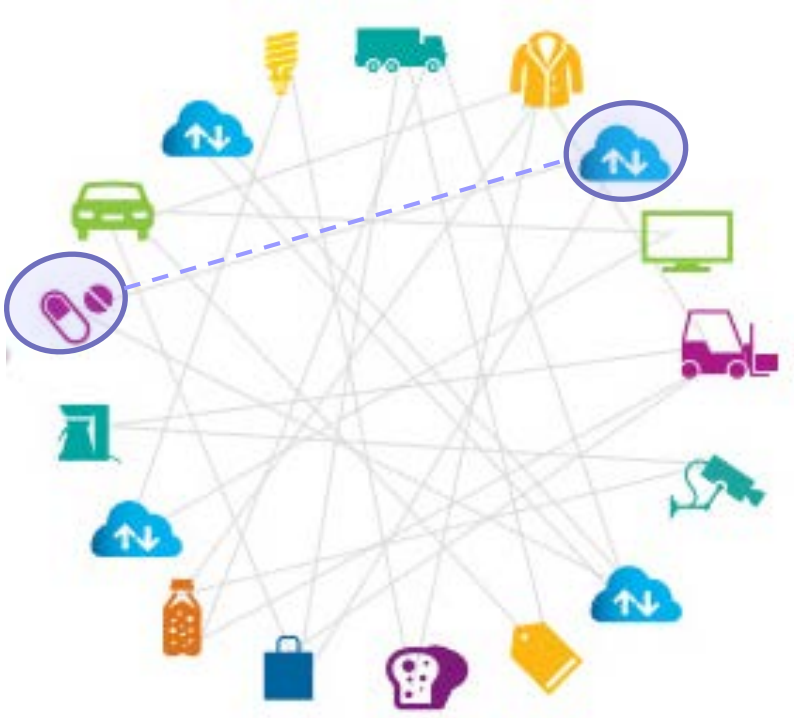
This "disintermediated" design is easier to configure & administer for an international relief agency

Peer-to-Peer (P2P) Database System

- Decentralized Control
- Full or partial replications
- Highest reliability, worst latency
- e.g., Torrents, Napster

Looking Deeper in to P2P systems

- Network of **asynchronously** computing nodes
  - Asynchrony implies responses are not guaranteed, e.g.
    - Communication can occur at any time & at irregular intervals
    - Communication delay is not bounded



Looking Deeper in to P2P systems

- Network of **asynchronously** computing nodes
  - Asynchrony implies responses are not guaranteed
  - In contrast, synchronous networks have bounded delay
    - i.e., message lost can be detected quickly & reliably

Looking Deeper in to P2P systems

- **Mutual Interest**
  - Consistent copy of database
  - Any database changes (*transactions*) *must* be recorded consistently
  - There should be no conflicting outcomes
    - *Outcome* = *transaction* applied to *database*

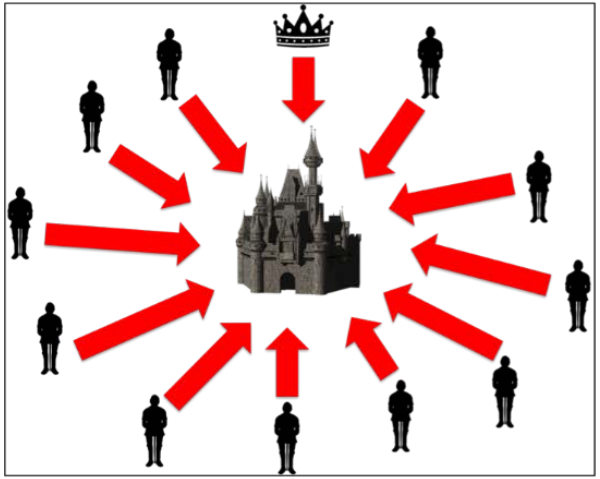
Stronger property: Everyone should agree on the sequence of transactions

e.g., milk storage & grocery store must both agree that a new bottle has been ordered & delivered

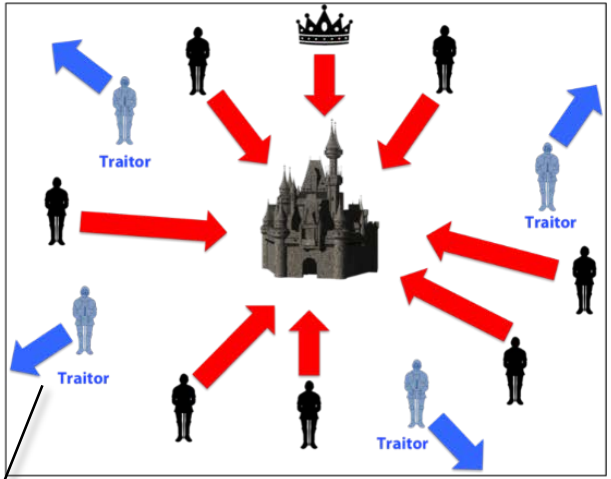
Looking Deeper in to P2P systems

- **Consensus**
  - Assume 11 generals
  - 5 want to attack & 5 want to retreat based on voting
  - The 11th general says “attack” to one group & “retreat” to other group
  - The problem is ensuring all loyal generals can agree on the strategy

Byzantine Generals Problem [Lamport 1982]



Coordinated Attack Leading to Victory

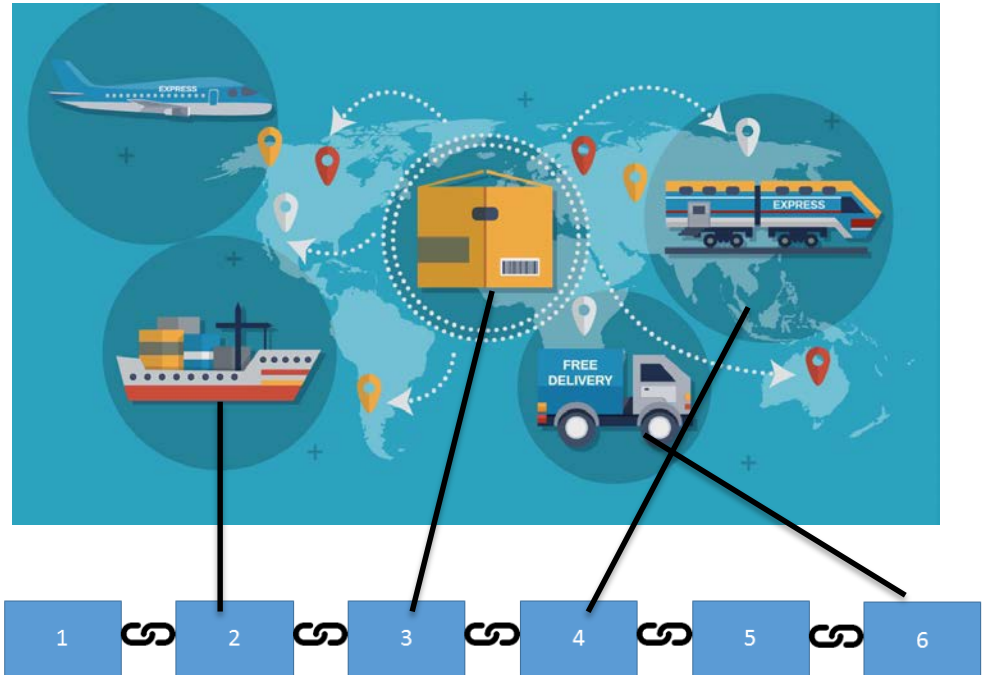


Uncoordinated Attack Leading to Defeat

A small # of traitors should not cause loyal generals to adopt a bad plan

Looking Deeper in to P2P systems

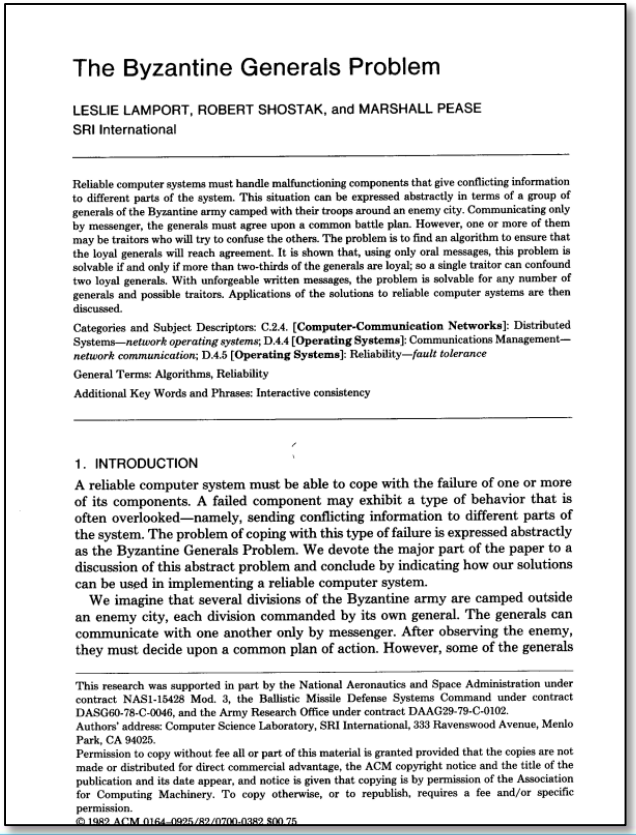
- **Consensus**
  - The problem is also valid in databases
    - e.g., how does everybody in network agree that a particular transaction occurred correctly?
  - The problem also extends to supply chains & products delivered to factories



Immutably linking a product & its subparts to its history

Byzantine General (Consensus) Approaches

- **Synchronous case**
  - Deterministic solution only if there are more than 3*m* generals & at most *m* traitors [Lamport et al 1982]
    - i.e., at-least 3 generals are required to be part of the group
  - Intuitively if 1 general is faulty you need input from 2 others to vote him/her out



The Byzantine Generals Problem

LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE
SRI International

Reliable computer systems must handle malfunctioning components that give conflicting information to different parts of the system. This situation can be expressed abstractly in terms of a group of generals of the Byzantine army camped with their troops around an enemy city. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach agreement. It is shown that, using only oral messages, this problem is solvable if and only if more than two-thirds of the generals are loyal; so a single traitor can confound two loyal generals. With unforgeable written messages, the problem is solvable for any number of generals and possible traitors. Applications of the solutions to reliable computer systems are then discussed.

Categories and Subject Descriptors: C.2.4. [**Computer-Communication Networks**]: Distributed Systems—*network operating systems*; D.4.4 [**Operating Systems**]: Communications Management—*network communication*; D.4.5 [**Operating Systems**]: Reliability—*fault tolerance*

General Terms: Algorithms, Reliability

Additional Key Words and Phrases: Interactive consistency

1. INTRODUCTION

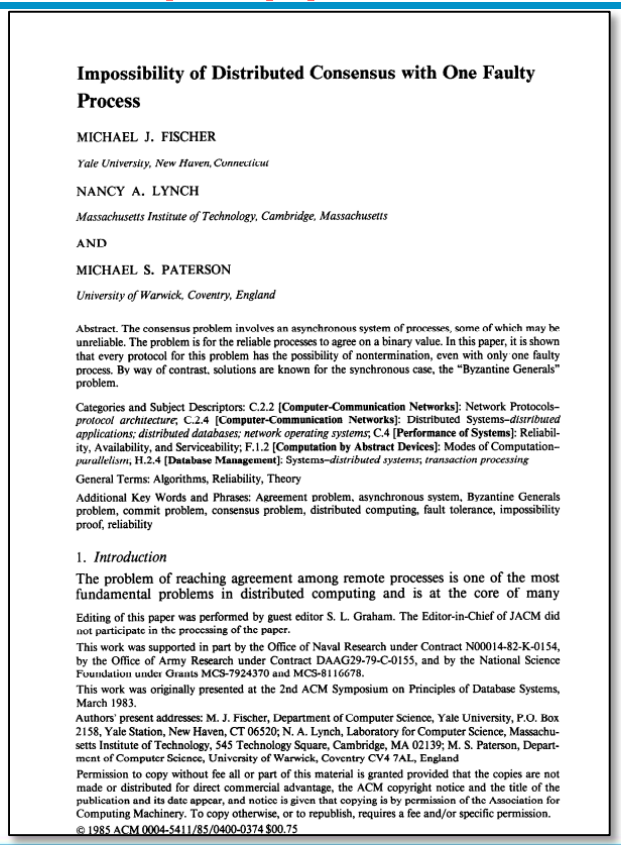
A reliable computer system must be able to cope with the failure of one or more of its components. A failed component may exhibit a type of behavior that is often overlooked—namely, sending conflicting information to different parts of the system. The problem of coping with this type of failure is expressed abstractly as the Byzantine Generals Problem. We devote the major part of the paper to a discussion of this abstract problem and conclude by indicating how our solutions can be used in implementing a reliable computer system.

We imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals

This research was supported in part by the National Aeronautics and Space Administration under contract NAS1-15428 Mod. 3, the Ballistic Missile Defense Systems Command under contract DASG60-78-C-0046, and the Army Research Office under contract DAAG29-79-C-0102.
Authors' address: Computer Science Laboratory, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025.

Byzantine General (Consensus) Approaches

- **Asynchronous case**
  - No deterministic solution of distributed consensus with one faulty process [Lynch et al 1985]



Impossibility of Distributed Consensus with One Faulty Process

MICHAEL J. FISCHER

Yale University, New Haven, Connecticut

NANCY A. LYNCH

Massachusetts Institute of Technology, Cambridge, Massachusetts

AND

MICHAEL S. PATERSON

University of Warwick, Coventry, England

Abstract. The consensus problem involves an asynchronous system of processes, some of which may be unreliable. The problem is for the reliable processes to agree on a binary value. In this paper, it is shown that every protocol for this problem has the possibility of nontermination, even with only one faulty process. By way of contrast, solutions are known for the synchronous case, the "Byzantine Generals" problem.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols–*protocol architecture*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems–*distributed applications; distributed databases; network operating systems*; C.4 [**Performance of Systems**]: Reliability, Availability, and Serviceability; F.1.2 [**Computation by Abstract Devices**]: Modes of Computation–*parallelism*, H.2.4 [**Database Management**]: Systems–*distributed systems; transaction processing*

General Terms: Algorithms, Reliability, Theory

Additional Key Words and Phrases: Agreement problem, asynchronous system, Byzantine Generals problem, commit problem, consensus problem, distributed computing, fault tolerance, impossibility proof, reliability

1. *Introduction*

The problem of reaching agreement among remote processes is one of the most fundamental problems in distributed computing and is at the core of many

Editing of this paper was performed by guest editor S. L. Graham. The Editor-in-Chief of JACM did not participate in the processing of the paper.

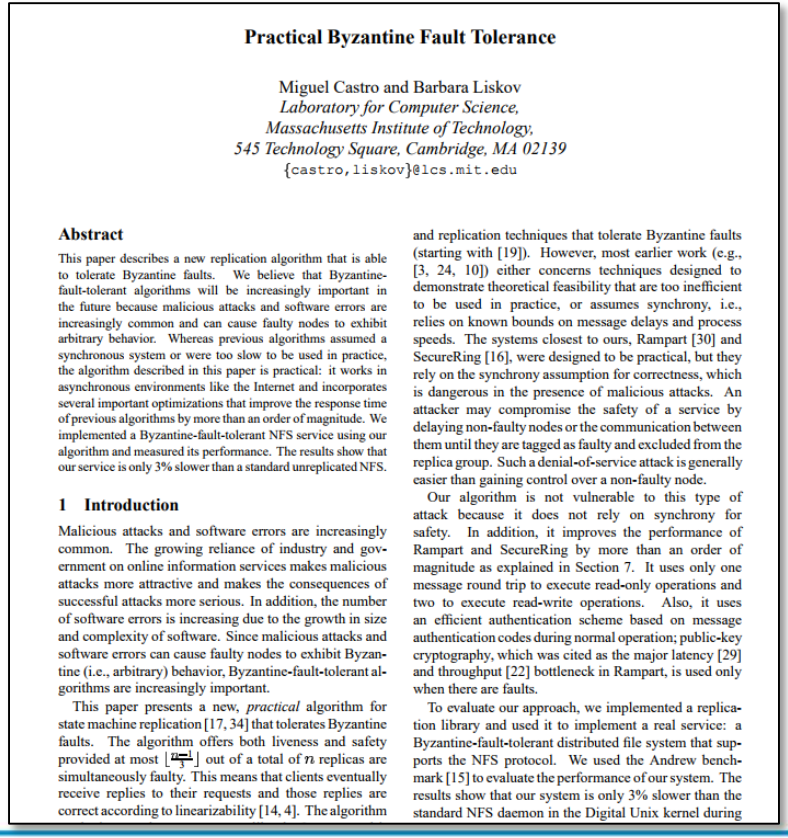
This work was supported in part by the Office of Naval Research under Contract N00014-82-K-0154, by the Office of Army Research under Contract DAAG29-79-C-0155, and by the National Science Foundation under Grants MCS-7924370 and MCS-8116678.

This work was originally presented at the 2nd ACM Symposium on Principles of Database Systems, March 1983.

Authors' present addresses: M. J. Fischer, Department of Computer Science, Yale University, P.O. Box 2158, Yale Station, New Haven, CT 06520; N. A. Lynch, Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139; M. S. Paterson, Department of Computer Science, University of Warwick, Coventry CV4 7AL, England

Byzantine General (Consensus) Approaches

- **Partially synchronous systems**
  - Practical Byzantine Fault Tolerance [Liskov et al 1999]
  - Advantage
    - Better Performance
    - Each participant in the group only waits to hear back from $2m+1$ other participants
      - m is the number of traitors



Practical Byzantine Fault Tolerance

Miguel Castro and Barbara Liskov
Laboratory for Computer Science,
Massachusetts Institute of Technology,
545 Technology Square, Cambridge, MA 02139
{castro,liskov}@lcs.mit.edu

Abstract

This paper describes a new replication algorithm that is able to tolerate Byzantine faults. We believe that Byzantine-fault-tolerant algorithms will be increasingly important in the future because malicious attacks and software errors are increasingly common and can cause faulty nodes to exhibit arbitrary behavior. Whereas previous algorithms assumed a synchronous system or were too slow to be used in practice, the algorithm described in this paper is practical: it works in asynchronous environments like the Internet and incorporates several important optimizations that improve the response time of previous algorithms by more than an order of magnitude. We implemented a Byzantine-fault-tolerant NFS service using our algorithm and measured its performance. The results show that our service is only 3% slower than a standard unreplicated NFS.

1 Introduction

Malicious attacks and software errors are increasingly common. The growing reliance of industry and government on online information services makes malicious attacks more attractive and makes the consequences of successful attacks more serious. In addition, the number of software errors is increasing due to the growth in size and complexity of software. Since malicious attacks and software errors can cause faulty nodes to exhibit Byzantine (i.e., arbitrary) behavior, Byzantine-fault-tolerant algorithms are increasingly important.

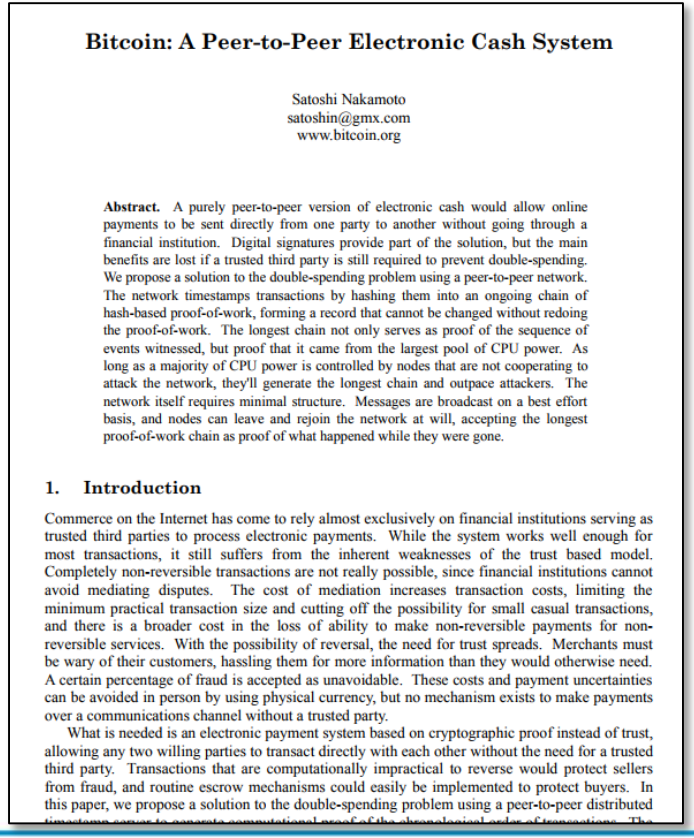
This paper presents a new, *practical* algorithm for state machine replication [17, 34] that tolerates Byzantine faults. The algorithm offers both liveness and safety provided at most $\lfloor \frac{n-1}{3} \rfloor$ out of a total of n replicas are simultaneously faulty. This means that clients eventually receive replies to their requests and those replies are correct according to linearizability [14, 4]. The algorithm and replication techniques that tolerate Byzantine faults (starting with [19]). However, most earlier work (e.g., [3, 24, 10]) either concerns techniques designed to demonstrate theoretical feasibility that are too inefficient to be used in practice, or assumes synchrony, i.e., relies on known bounds on message delays and process speeds. The systems closest to ours, Rampart [30] and SecureRing [16], were designed to be practical, but they rely on the synchrony assumption for correctness, which is dangerous in the presence of malicious attacks. An attacker may compromise the safety of a service by delaying non-faulty nodes or the communication between them until they are tagged as faulty and excluded from the replica group. Such a denial-of-service attack is generally easier than gaining control over a non-faulty node.

Our algorithm is not vulnerable to this type of attack because it does not rely on synchrony for safety. In addition, it improves the performance of Rampart and SecureRing by more than an order of magnitude as explained in Section 7. It uses only one message round trip to execute read-only operations and two to execute read-write operations. Also, it uses an efficient authentication scheme based on message authentication codes during normal operation; public-key cryptography, which was cited as the major latency [29] and throughput [22] bottleneck in Rampart, is used only when there are faults.

To evaluate our approach, we implemented a replication library and used it to implement a real service: a Byzantine-fault-tolerant distributed file system that supports the NFS protocol. We used the Andrew benchmark [15] to evaluate the performance of our system. The results show that our system is only 3% slower than the standard NFS daemon in the Digital Unix kernel during

Byzantine General (Consensus) Approaches

- **Blockchains (Proof of work)**
  - Integration of *Incentives & Game Theory* to reduce the messaging overhead of practical byzantine fault tolerance [Nakamoto 2008]



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third 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 a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

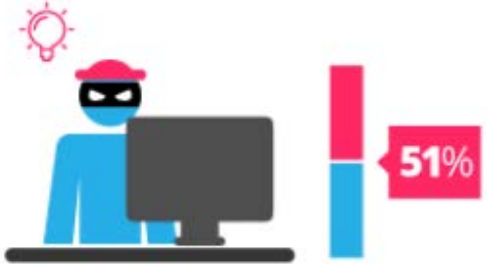
What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed

Byzantine General (Consensus) Approaches

- **Blockchains (Proof of work)**
  - Integration of *Incentives & Game Theory* to reduce the messaging overhead of practical byzantine fault tolerance [Nakamoto 2008]
  - Participants issue responses once they validate transaction & group them into a block
    - Must complete a “hard” puzzle
  - By investing energy in solving the puzzle, it’s not in their interest to lie

Proof of work requires participants to perform an expensive calculation

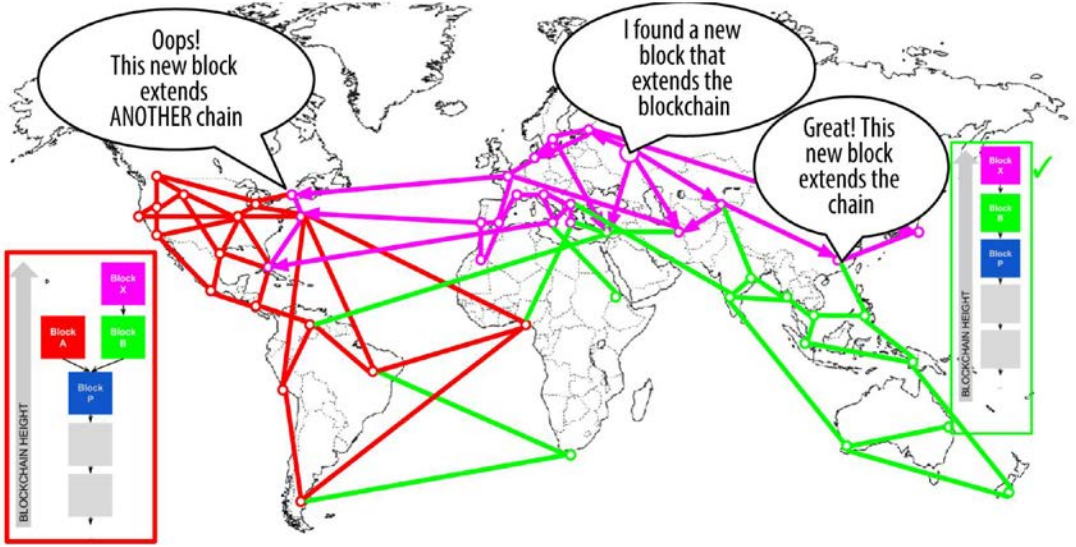
A reward is given to the first miner who solves the problem of a block (recorded in the block)



Everyone competes to solve the mathematical problem first

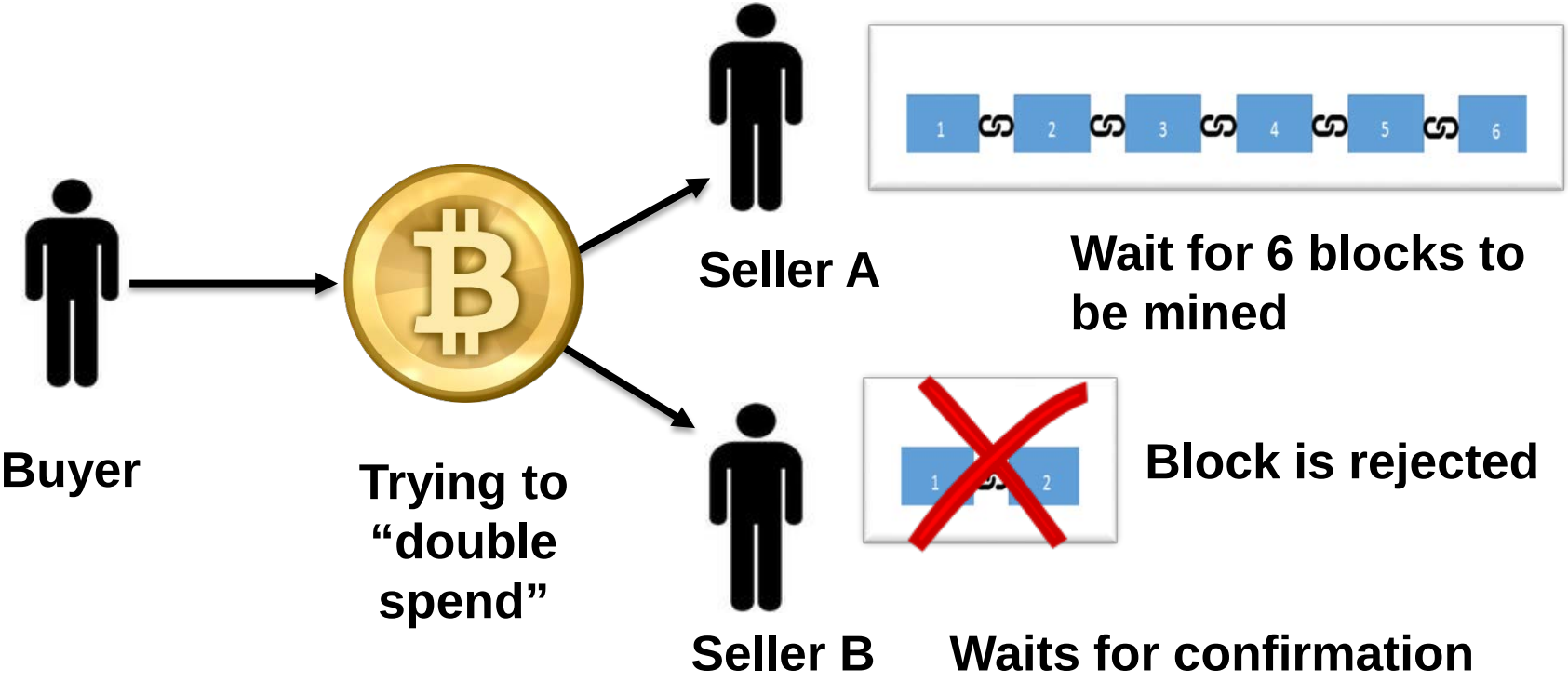
Byzantine General (Consensus) Approaches

- **Blockchains**
  - Event sequence is maintained by including a cryptographic hash of previous block into current block
  - In most cases, HASH(A) == HASH(B) implies A==B
  - Divergence is possible



Conflicts are broken with the "longest chain" rule

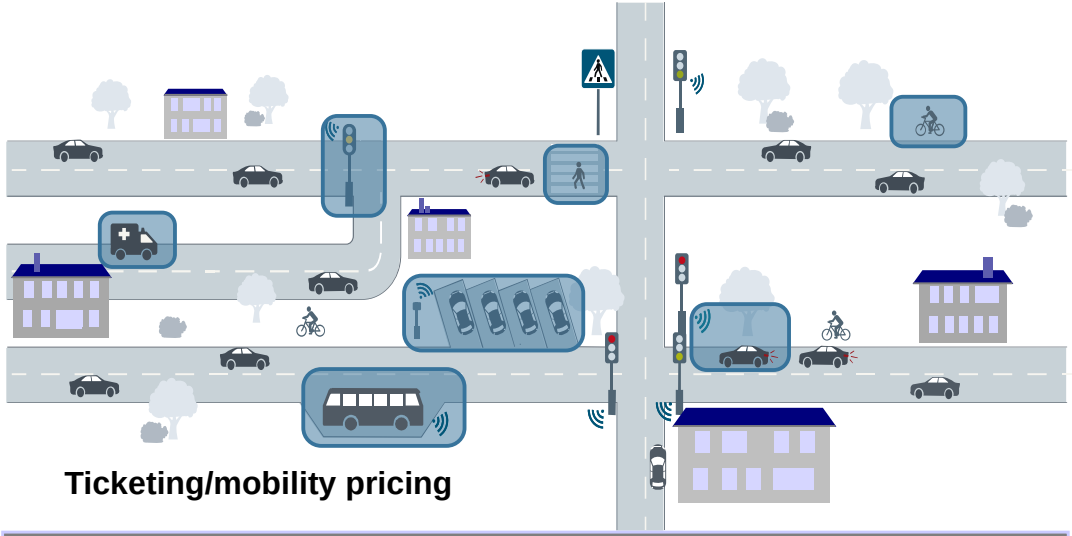
How Blockchain Helps Ensure Decentralized Transactions



Blockchains & the Internet of Things (IoT)

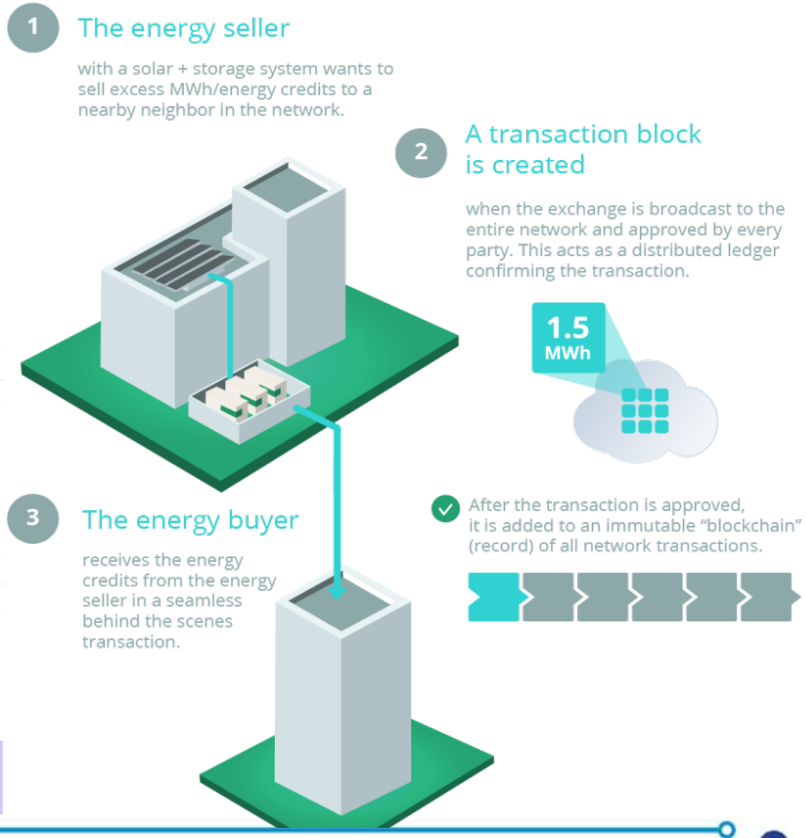
Blockchains Increasingly are Being Used in the IoT

- The notion of decentralized computation & trustless computing provides opportunities & challenges for the IoT

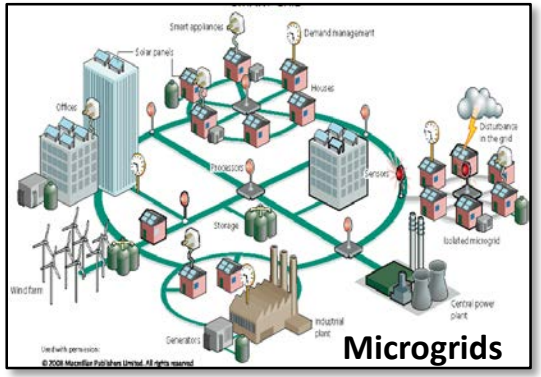


Ticketing/mobility pricing

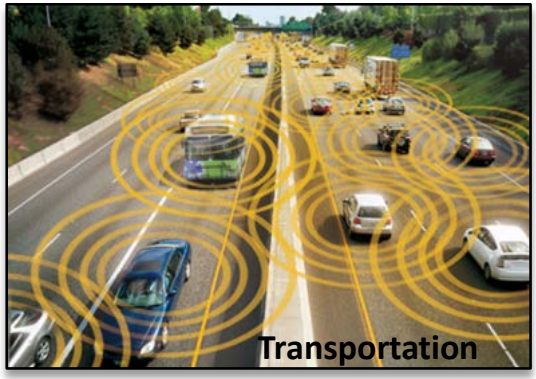
Blockchain Middleware



The Reason is the Focus on Decentralization



Microgrids


Transportation

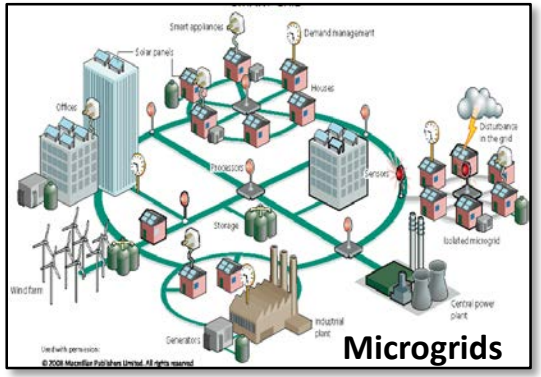
Health Care

Smart Buildings

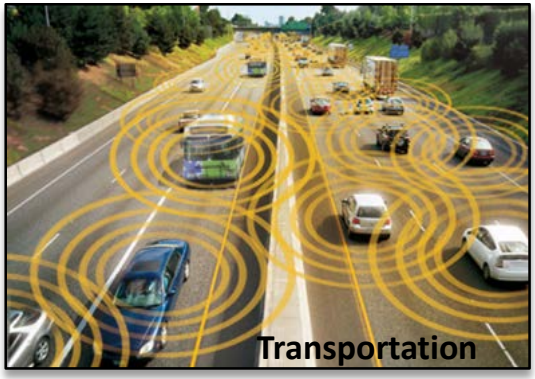
Decentralized Control

- Compute control actions using distributed averaging consensus within a specific time limit
- Requires real-time information dissemination & time-synchronized task execution

Key Requirements For Decentralized IoT



Microgrids


Transportation

Health Care

Smart Buildings

Decentralized Information

- Preserve information integrity across all actors in the system
- Support for information aggregation & transactions
- Requires consensus & distributed ledger

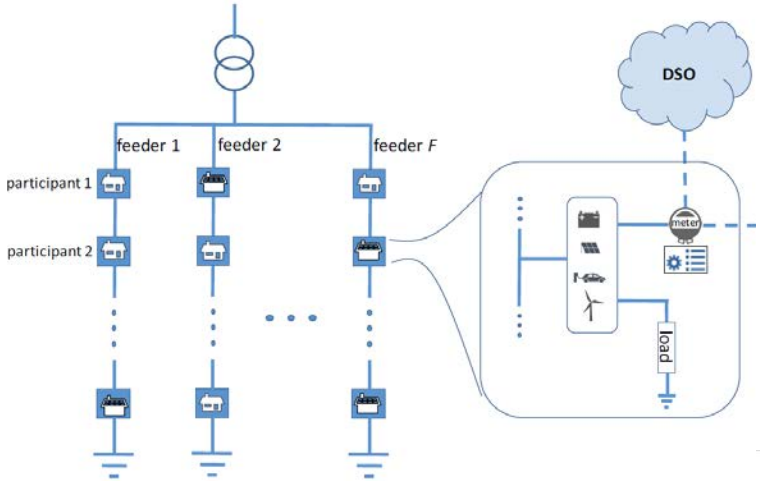
Example of Decentralized Control & Information: Transactive Energy Systems

Transactive Energy: Smart Homes → Smart Prosumers

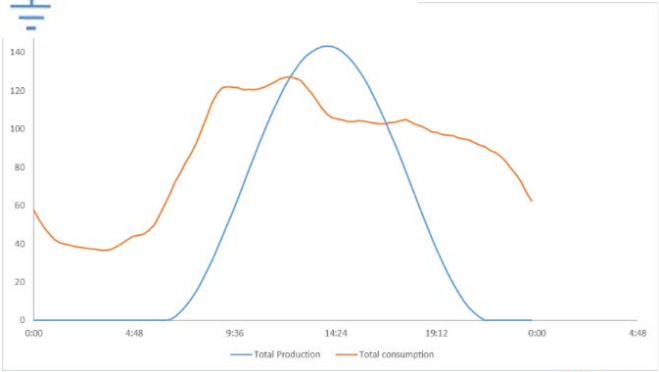
Prosumer 1

Energy exchange reduces dependence on the public grid

Prosumer 2



Typical generation & consumption profile

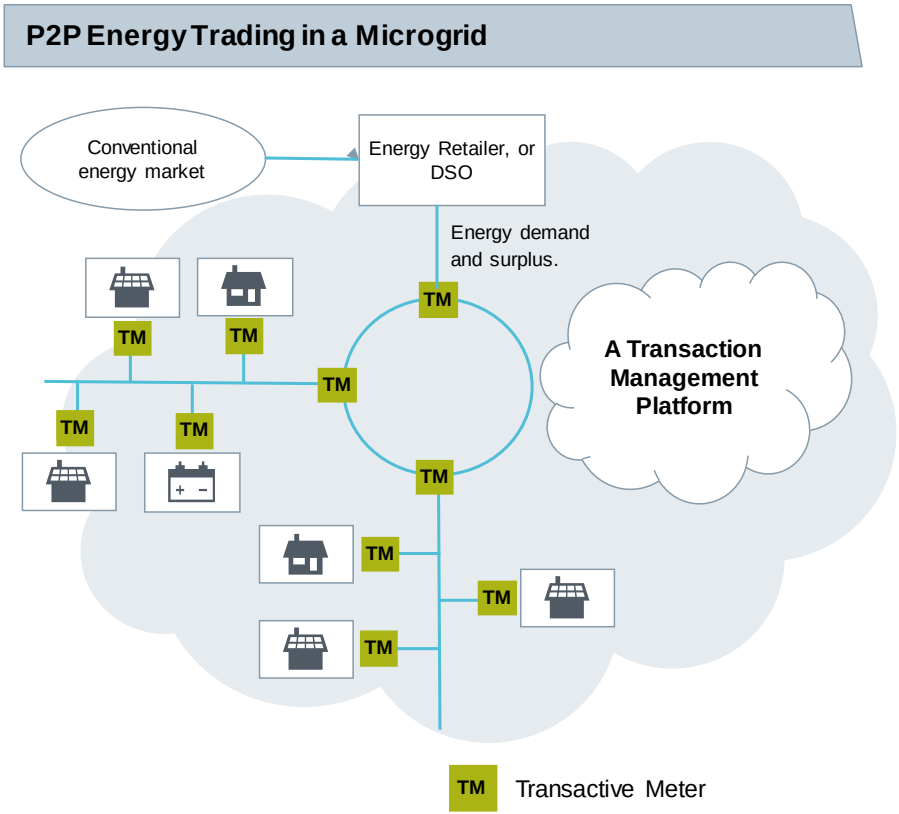


Transactive Energy: Smart Homes → Smart Prosumers

Energy exchange reduces dependence on the public grid

- Local energy market safety & efficiency
  - A decentralized control problem
  - Time-synchronized data exchange
- Integrity & auditability of finalized transactions
  - Decentralized management problem
  - Ideally suited for blockchains
- Concerns
  - Information privacy for prosumers

Transactive Energy: Smart Homes → Smart Prosumers



Privacy-Preserving Platform for Transactive Energy Systems

Karla Kvaternik
Siemens Corporate Technology
karla.kvaternik@siemens.com

Aron Laszka
Vanderbilt University
aron.laszka@vanderbilt.edu

Michael Walker
Vanderbilt University
michael.a.walker.1@vanderbilt.edu

Douglas Schmidt
Vanderbilt University
d.schmidt@vanderbilt.edu

Monika Sturm
Siemens Corporate Technology
monika.sturm@siemens.com

Martin Lehofer
Siemens Corporate Technology
martin.lehofer@siemens.com

Abhishek Dubey
Vanderbilt University
abhishek.dubey@vanderbilt.edu

Abstract

Transactive energy systems (TES) are emerging as a transformative solution for the problems faced by distribution system operators due to an increase in the use of distributed energy resources and a rapid acceleration in renewable energy generation. These, on one hand, pose a decentralized power system controls problem, requiring strategic microgrid control to maintain stability for the community and for the utility. On the other hand, they require robust financial markets operating on distributed software platforms that preserve privacy. In this paper, we describe the implementation of a novel, blockchain-based transactive energy system. We outline the key requirements and motivation of this platform, describe the lessons learned, and provide a description of key architectural components of this system.

Keywords Transactive energy platforms, blockchain, privacy, security, safety, smart contracts

ACM Reference format:
Karla Kvaternik, Aron Laszka, Michael Walker, Douglas Schmidt, Monika Sturm, Martin Lehofer, and Abhishek Dubey. 2017. Privacy-Preserving Platform for Transactive Energy Systems. In *Proceedings of ACM/IFIP/USENIX Middleware conference, Las Vegas, Nevada USA, December 2017 (Middleware'17)*, 6 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 Introduction

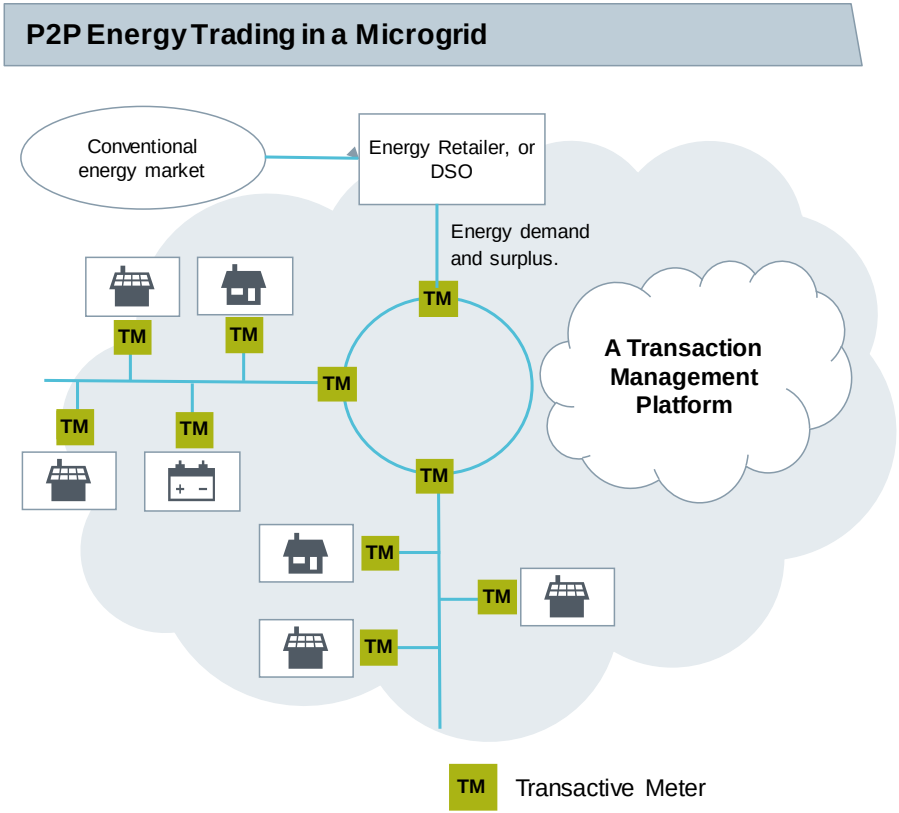
Emerging Trends: Transactive energy systems (TES) have emerged as an anticipated outcome of the shift in electricity industry, away from centralized, monolithic business models characterized by bulk generation and one-way delivery, toward a decentralized model in which end users play a more active role in both production and consumption [10] [24]. In this paper, we consider a class of TES that operates in grid-connected mode. The main actors are the consumers, connection of the network. Such installations are equipped with an advanced metering infrastructure consisting of TE-enabled smart meters. In addition to the standard functionalities of smart meters: i.e. the ability to measure line voltages, power consumption and production, and communicate these to the distribution system operator (DSO); TE-enabled smart meters are capable of communicating with other smart meters, have substantial on-board computational resources, and are capable of accessing the Internet and cloud computing services as needed. Examples of such installations include the well-known Brooklyn Microgrid Project, [3] and the Sterling Ranch learning community (currently under development) [12]. A key component of TES is a transaction management platform (TMP), which handles all market clearing functions in a way that balances supply and demand in the local market.

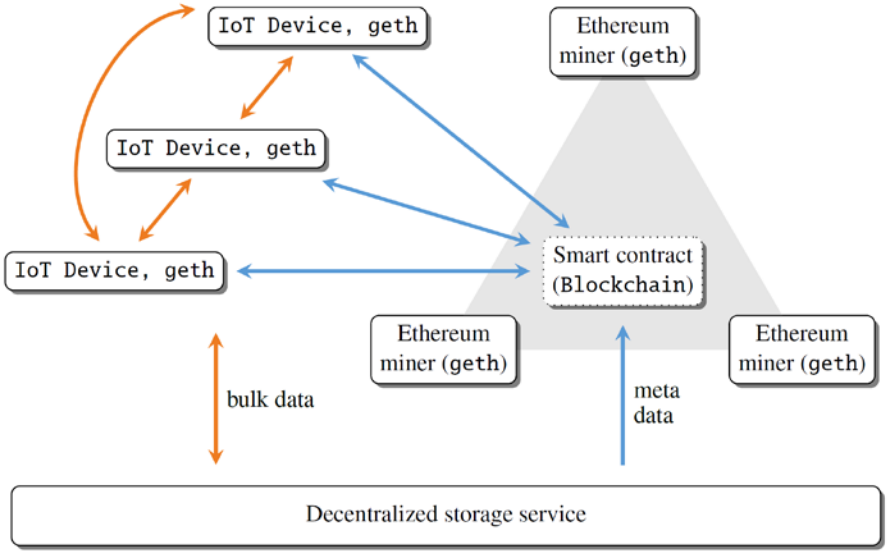
Why Blockchains?: The capabilities of TE-enabled meters allow them to form a blockchain (BC) based TMP executing a market mechanism, using smart contracts [29]. Examples of BC systems capable of executing smart contracts include Ethereum [7] and Hyperledger Fabric [9]. There are a number of appealing properties of BC systems that motivate their use in a TMP. Firstly, BC technology enables the digital representation of energy and financial assets, and their secure transfer from one set of parties to another. By design, the security of this value transfer is guaranteed by the interaction protocol itself and obviates the need for trusted transaction intermediaries. Secondly, the execution of smart contracts (i.e. code that captures the market logic and participant roles) is automated and guaranteed. Thirdly, the blockchain constitutes an immutable, complete, and fully auditable record of all transactions that have ever occurred in the BC system. These properties ensure market transparency, as well as the availability of a detailed market load profile, and grid utilization data. Thus, [1, 4, 27] have already considered such implementations.

Open challenges: Existing initiatives such as [1, 4, 27] do not

Enabling safe & private interactions with blockchains in smart grid
(see arxiv.org/abs/1709.09597?context=cs.DC)

The Key is the Transactive Platform Pattern called PETra





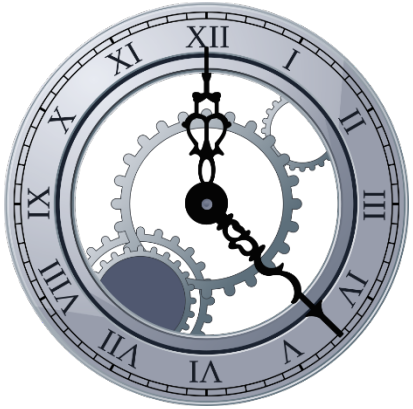
Enabling safe & private interactions with blockchains in smart grid
(see arxiv.org/abs/1709.09597?context=cs.DC)

Blockchain Challenges in the Internet of Things (IoT)

Time is a Key Challenge with Blockchains in the IoT

- Time is often critical when interacting with the physical world







Time is a Key Challenge with Blockchains in the IoT

- Time is often critical when interacting with the physical world

In 10 minutes the electrical network can finish 36,000 cycles - a fault can occur in any cycle

UBER

In 10 minutes your Uber driver can travel 6.66 miles traveling at 40 miles per hour

Alternatives to Proof of Work for IoT

- **Proof of Stake**
  - A person mines or validate block transactions according to how many coins they hold
  - A miner who owns 3% of the Bitcoin available can theoretically mine only 3% of the blocks (locked in a deposit)
  - Miners only get transaction fees

With proof of stake the attacker needs 51% of the cryptocurrency to carry out a 51% attack

Alternatives to Proof of Work for IoT

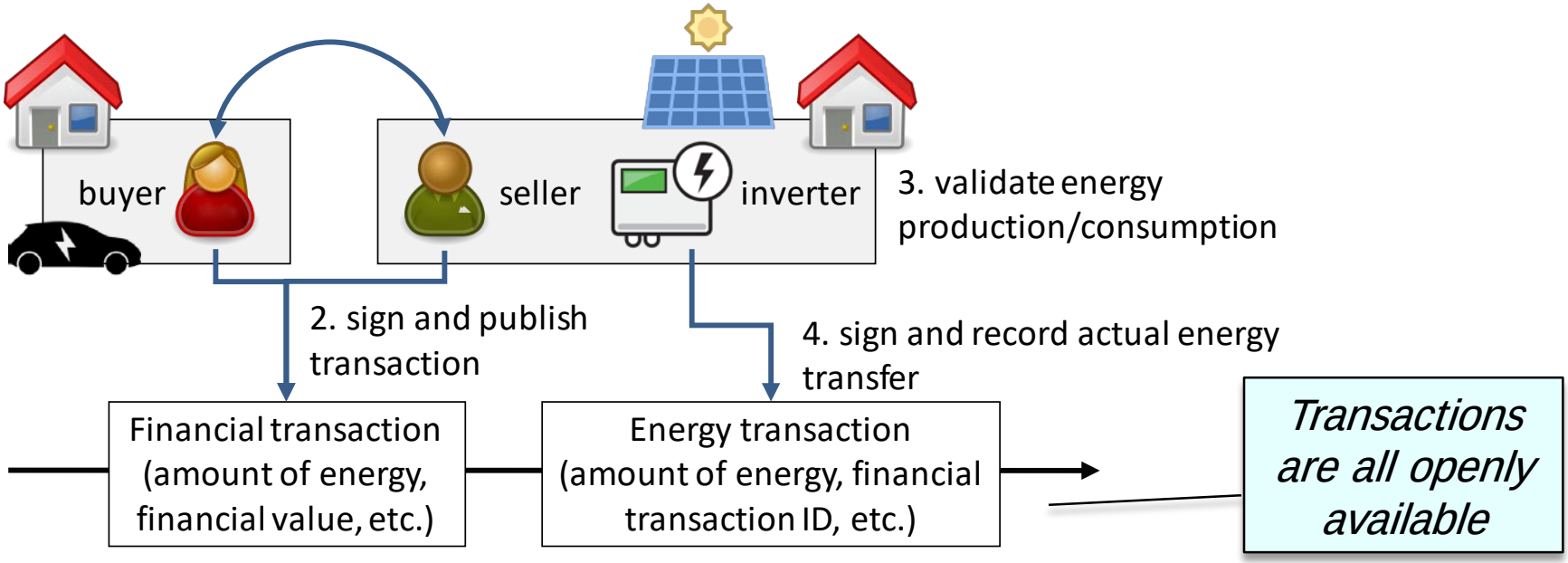
- **Proof of Authority**
  - All servers vote on the valid transactions
  - Only transactions receiving > 80% of votes are added to the ledger

Alternatives to Proof of Work for IoT

- **Permissioned Networks**
  - Permissioned networks are private networks that strictly control who can participate

Other Challenges in Applying Blockchains to IoT

- Ensuring privacy is another key problem in current blockchain technology



A. Laszka, A. Dubey, M. Walker, & D. Schmidt. Providing Privacy, Safety, & Security in IoT-Based Transactive Energy Systems using Distributed Ledgers. IoT, sep 2017.

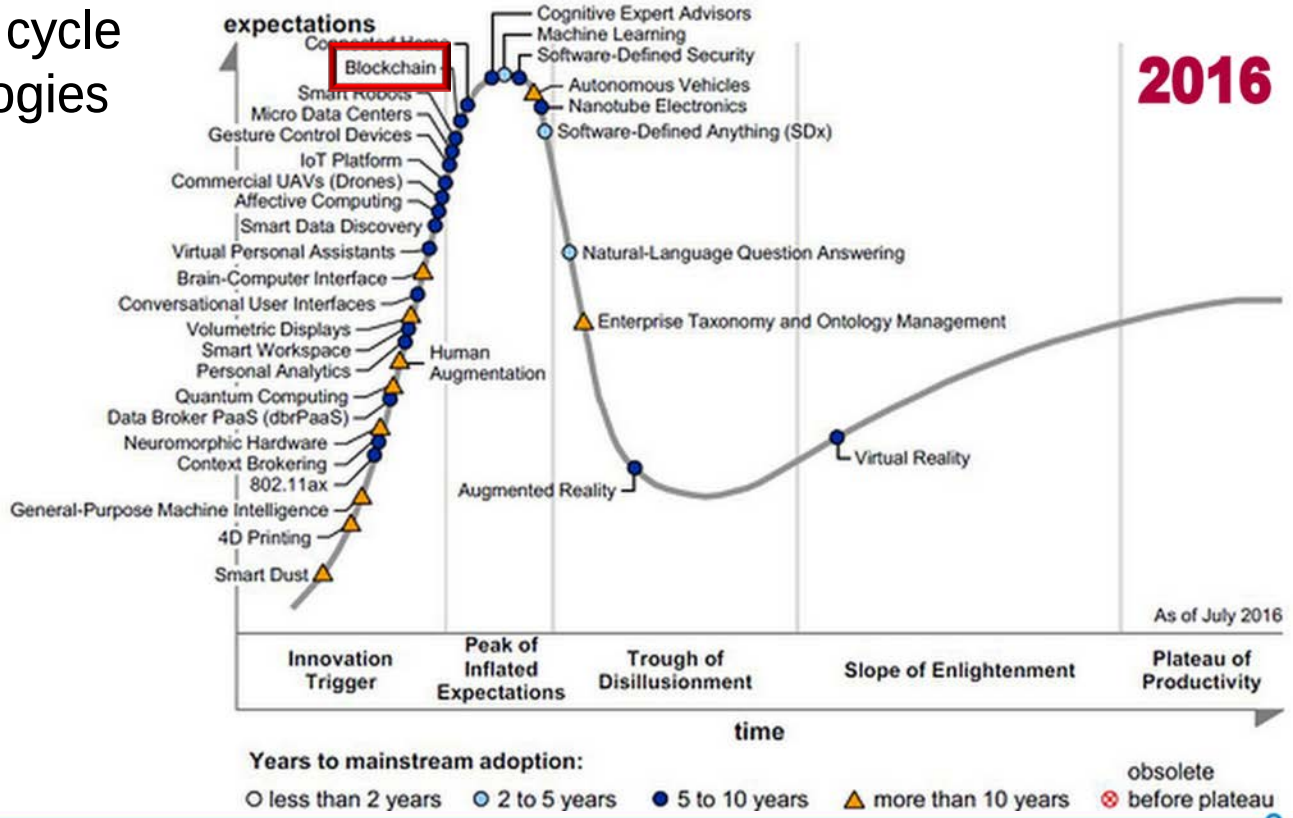
Blockchain represents a trade-off in which disintermediation is gained at the cost of confidentiality

Wrapping Up


I THINK WE SHOULD BUILD A BLOCKCHAIN
UH-OH
S. Adams E-mail: SCOTTADAMS@AOL.COM
DOES HE UNDERSTAND WHAT HE SAID OR IS IT SOMETHING HE SAW IN A TRADE MAGAZINE AD?
© 1995 United Feature Syndicate, Inc.(NYC)
WHAT COLOR DO YOU WANT THAT BLOCKCHAIN?
I THINK MAUVE HAS THE MOST RAM.

Where Blockchain is in the Hype Cycle

- Gartner's 2016 hype cycle of emerging technologies



Thanks for Attending!

At Vanderbilt we are conducting research to fill in all these gaps with blockchain in collaboration with our sponsors & industry partners

You can contact us at:

- Dr. Abhishek Dubey – abhishek.dubey@vanderbilt.edu
- Dr. Douglas C. Schmidt – d.schmidt@vanderbilt.edu
- Dr. Jules White – jules@dre.vanderbilt.edu