MTR180046
MITRE TECHNICAL REPORT

MITRE

Blockchain Technology for Government

Project No.: 10MSRF18-AA

Authors: Dave Bryson
Dave Penny
David C. Goldenberg
Gloria Serrao

December 2017

Approved By

Kris Rosfjord, Tech Futures Innovation Area Leader

03-23-18
Date

This page intentionally left blank.

Abstract

This document presents an introduction to blockchain technology with a specific focus on the core technologies, platforms, and applications that may be beneficial to MITRE's government sponsors. The document is intended to introduce MITRE and its sponsor to blockchain technology, and establish a base of knowledge upon which to further explore MITRE sponsor blockchain use cases and research. An introduction to blockchain and its critical components including cryptography, consensus, and distributed ledgers is provided. Public and permissioned blockchains are compared, and a framework is provided that outlines when it is beneficial to use blockchain solutions. Use cases applicable to MITRE sponsors such as healthcare, identity, supply chain, and the Internet of Things (IoT) are considered. A survey of leading permissioned blockchains such as Ethereum and Tendermint is presented, and important emerging features such as private transactions and state channels that strengthen enterprise blockchains are discussed.

This page intentionally left blank.

Executive Summary

Blockchain technology is evolving at a very rapid pace. Understanding the core components of the technology and how they work together is critical to tracking the state of the art. While each component plays a critical role in the technology stack, consensus is at the heart of the system and important to understand. Carefully choosing the right consensus algorithm based on the desired level of trust and security will be critical to a successful blockchain application.

While public blockchains provide the most security as they are designed to operate in a trust-less environment, government users will be most interested in a permissioned blockchain. However, the nature of a permissioned blockchain requires careful planning and governance to establish the parties participating in the consensus process. Without proper governance, there may be a possibility of politically centralizing some of the key functionality of the blockchain, limiting its capabilities, and providing a false sense of security.

As more look to permissioned blockchains to modernize traditional applications, there are several requirements that need to be addressed. For example, privacy and confidentiality on the blockchain, transaction scalability, and blockchain-to-blockchain connectivity. While there's ongoing active research in these areas across several open-source communities, permissioned blockchains need to further evolve to fully meet the needs of the government user.

Table of Contents

List of Figures

List of Tables

1 Background

In 2009, a developer (or developers) by the name of Satoshi Nakamoto created an electronic payment system named Bitcoin (Nakamoto, 2009). The goal of Bitcoin was simple: allow two parties to directly exchange a digital currency without the need for a trusted 3rd party (e.g., a bank) to mediate the transfer. Over the past eight years, Bitcoin has grown to be one of the most successful cryptocurrencies. It has also proved to be one of the first public-facing, fault-tolerant distributed systems capable of operating in an adversarial environment.

As of 2017, there are over 1000 different cryptocurrencies, many based on Bitcoin, with a combined market capitalization approaching $1 trillion. (CoinMarket Cap, 2017). The success of Bitcoin and the emergence of new platforms such as Ethereum that offer programmable smart contracts, is pushing the technology beyond the cryptocurrency use case and driving the growing interest in blockchain technology.

The level of interest is especially high in enterprise applications. The recently formed Ethereum Enterprise Alliance (Enterprise Ethereum Alliance, 2016) and the open-source Hyperledger project (Hyperledger, 2017) now have hundreds of members across a broad spectrum of Fortune 500 companies. Virtually every major software vendor is offering services and consulting on blockchain technology.

2 What is a Blockchain?

A blockchain can be compared to a bank ledger containing transactions. It provides information about the date, time, and amount of money or other property of interest changing ownership. Once transactions are written to the ledger, they are permanent; they can't be changed or deleted. Transactions are bundled into blocks and these blocks are linked to form the ledger, which is called blockchain. A blockchain is distributed over multiple nodes using an underlying peer-to-peer (P2P) network protocol for node discovery and communication.

The components of blockchain are discussed in detail in Section 3, but are briefly described below for context.

- Cryptography: hash functions that link blocks together providing integrity of the chain and digital signatures providing integrity for the transactions.
- Consensus Algorithm: The process by which parties to a blockchain decide on the ordering and presence of transactions on the ledger.
- Distributed Ledger: A distributed, replicated, representation of all transactions
- P2P Protocol: The protocol that manages the peer nodes of the network that support blockchain. Performs communication between nodes, flow control, node discovery, and framing.
- Smart Contracts: business rules or logic that can extend the functionality of a blockchain

2.1 What a Blockchain Does

At its core, a blockchain enables a network of peer computers (or nodes) to validate, settle, and agree on a record of transactions. It establishes a form of trust between parties that may not otherwise trust each other, and does so without relying on traditional centralized services, or trusted 3rd parties. This new form of decentralized trust has generated interest with users across a wide range of domains. Companies are investing research and development efforts into

blockchain technology, with the hopes of revamping existing processes to reduce cost while improving security, accountability, and transparency.

2.2 How it Operates

A blockchain is a replicated state machine. A given state is synchronized across a set of machines, or nodes, such that these nodes function as a single machine, despite the potential that some will fail either through normal faults or malicious activity.

The state of a blockchain is driven by incoming transactions, each transaction causing a state transition. What the state represents depends on the goal of the blockchain. In the case of cryptocurrency, the state is the balance of accounts, while in an enterprise blockchain, the state may represent other forms of information.

A transaction can be thought of as the fundamental unit of work in a blockchain. It is the input to the system, an atomic operation, that drives a state transition. It either succeeds and updates the state of the system, or it fails and is ignored. A transaction is redundantly verified by every blockchain node in the network and multiple transactions are batched into blocks for efficient processing.

The consensus protocol enables all nodes to agree on the transactions in a block and the order of the blocks ensuring identical copies of the blockchain are stored on every machine. These new blocks are cryptographically linked to the prior block to prevent them from being altered once agreed upon. The combination of a tamper-proof transaction log and a deterministic state transition ensures that all machines can compute the same state given the same transaction log.

3 Key Components of a Blockchain

3.1 Cryptography

A blockchain relies on two cryptographic primitives for many of its security properties: cryptographic hash functions and digital signatures.

3.1.1 Hash Function

A cryptographic hash function is a mathematical function that takes in an input string of any size and produces a fixed sized output. Cryptographic hash functions have two main properties that are used to secure blockchains.

1. It is possible to efficiently compute the hash function in the forward direction; however, it's nearly impossible to compute the inverse of the hash function. In other words, it's infeasible to learn any valid input from the output hash.
2. Small changes in the input to a hash function result in large and unpredictable changes in the output. As an example of the above two properties, using the SHA-256 hash function, we can create a unique fingerprint of the word "HELLOWORLD":

   ```
   sha256("HELLOWORLD") =
   '0x0b21b7db59cd154904fac6336fa7d2be1bab38d632794f281549584068cdcb74'
   ```

   The hash of the word results in a 256-bit value and it will always be the same value given the same input. However, given the 256-bit output value, it is computationally infeasible to discover the original input "HELLOWORLD" in our example. Furthermore, changing

even one character (dropping the ‘H’) in the original input results in a completely different hash value:

```
sha256(“ELLOWORLD”) =
'0x7d26a27cec234907afe7ce36266858446f4b8eebb7026982c8713d3c5d1c100e'
```

This makes it easy for hash functions to detect even the slightest change in an input string. Although it’s easy to detect changes in our simple example by looking at the text, hash functions can help to guarantee the authenticity of more complex content where changes may not be as easily noticeable.

Hash functions are used to both ensure an individual block of transactions cannot be altered and that the order of blocks in the overall blockchain remains consistent. Once a block is created it cannot be altered, and one cannot remove blocks or insert blocks into the middle of the blockchain. This is further explained in Section 3.3.

3.1.2 Digital Signature

Another cryptographic primitive used by blockchain technology is the digital signature. Cryptographic digital signatures are based on public key cryptography. They are the digital equivalent of a traditional signature but are much more secure. Essentially a digital signature provides a way for anyone to mathematically verify that a party is willing to attest to some digital content.

A digital signature requires a cryptographic key pair. The key pair consist of a private key, sometimes referred to as the signing key, and a public key, also known as the verification key. The signing key is a securely generated random value, while the verification key is generated from the signing key. The math behind the signature scheme ensures that it is computationally infeasible to reverse the process–you cannot learn the signing key from the verification key. The signing key should be securely controlled by the owner and never shared. On the other hand, the verification key can be shared with anyone and is used along with the signed content to verify the validity and authenticity of the signature.

For example, Bob is the owner of a key pair and uses his private key to sign the cryptographic hash of some content to generate a signature on that content[1]. Given the corresponding public key, the content, and the generated signature, anyone can verify that Bob (uniquely) signed the content.

In the context of the blockchain, signature verification of a transaction is a critical step. If the signature is invalid, the transaction is rejected. Signatures are primarily used to ensure that all data and state on the blockchain cannot be illegitimately modified.

Most blockchain implementations use Elliptic Curve Digital Signature Algorithm (ECDSA); ECDSA is a U.S. government standard. The algorithms behind ECDSA have undergone considerable cryptographic analysis and are considered secure and more efficient than other perhaps better known cryptographic algorithms such as RSA/Digital Signature Algorithm (DSA). (Narayanan, Bonneau, Felten, Miller, & Goldfeder, 2016)

[1] Quite often for efficiency a cryptographic hash of the content is signed, and not the content itself. If correctly implemented this does not introduce any weaknesses in security.

3.2 Consensus

Consensus makes it possible for a decentralized network of machines to agree upon and share the state of the system. It is critical in ensuring participants can trust the transactions processed on the blockchain–even when they may not trust each other. For example, say Alice transfers 10 digital tokens to Bob. What prevents Alice from transferring the same 10 digital tokens to Carl? In the cryptocurrency world, this is known as the Double Spend Problem (Double Spend Problem, 2017). Somehow the system must reconcile and share account information across a network of independent nodes to ensure Carl is not cheated out of his payment from Alice.

Before Bitcoin, it was impossible to electronically transfer digital money without relying on a centralized authority to manage the state of the system. Bitcoin cleverly solved the double spend problem and eliminated the need for a middleman by simply distributing a copy of the ledger to every node on the network, so anyone can check the state of an account.

However, this creates a new problem. How does a blockchain node know it's being sent valid information? What if messages are lost? What if a malicious actor is falsifying transactions or blocks? This problem is best illustrated by the well-known Byzantine Generals Problem (Byzantine Fault Tolerance, 2017).

In the Byzantine Generals Problem, there is a group of generals surrounding a city. The generals have the ability to conquer the city if they coordinate the time of attack. However, if they do not attack at the same time, they risk being defeated by the enemy. The generals can only coordinate through messengers with no way to verify the authenticity of the message. Messengers may be captured, preventing the delivery of a message. And traitors among the generals may deliberately send bad messages to disrupt coordination. How can this group successfully coordinate the attack without relying on a centralized authority? Bitcoin solved the Byzantine Generals Problem through a new form of consensus called Proof of Work (PoW).

3.2.1 Proof of Work (Nakamoto Consensus)

Public blockchains such as Bitcoin or Ethereum, allow anyone to participate in the consensus process as a miner. Miners compete (or effectively vote) to add new transactions to the blockchain with computing power by expending a certain amount of Central Processing Unit (CPU) cycles to solve a mathematical puzzle. This puzzle is intentionally computationally difficult to solve (Nakamoto, 2009), yet it is very easy to verify the answer.

To add a block of new transactions to the blockchain, a miner must solve the puzzle. The first miner to solve the puzzle sends (proposes) the block to the rest of the network for agreement. If the network agrees on the solution to the puzzle, the miner is rewarded for creating the block and the block is added to the blockchain (the miner wins this round of competition). Through a combination of game theory and economics (effectively betting CPU cycles, which cost money, to win the reward), PoW incentivizes consensus instead of attempting to enforce it. Essentially a miner is rewarded for securing the network.

When nodes synchronize to the network, there is a chance malicious actors may try to send invalid blocks in an attempt to double spend. To prevent this, nodes follow a simple rule of always synchronizing to the longest chain of blocks. This is because the longest chain reflects the majority vote by the network–it is the one miners have done the most work on.

Following the longest chain rule also reduces the number of protocol messages traditionally required to synchronize a large number of distributed nodes, making it possible for a public

blockchain to scale to thousands of nodes. However, the downside of using the longest chain rule is that blocks are never truly final. This is why public blockchains may advise waiting for 12 or more block confirmations to accept a transaction (Block Confirmation, 2016).

An example of why this is necessary is that there are rare instances when the chain will split under normal operation–when multiple miners solve a puzzle at virtually the same time. As noted, nodes choose to build on the longest chain. However, in this case, there are two valid choices. Based on PoW one chain will eventually win, becoming the longest chain (the probability of chains extending in parallel for more than one or two blocks is exceedingly small). The blocks in the shorter chain are often referred to as orphaned; and all transactions in these blocks are effectively invalid until they are packaged into a new block.

Thus, consensus is reached, but transaction settlement times are relatively long and transaction rates are limited.

3.2.2 Byzantine Fault Tolerant (BFT) Consensus

While public blockchains rely on PoW, enterprise (or permissioned) blockchains tend to use more traditional BFT consensus protocols (Byzantine Fault Tolerance, 2017). BFT consensus is based on the idea that a pre-selected, authorized group of validators[2] will create, verify, and attest to new blocks.

These validators take turns creating new blocks and submit a newly created block to other validators for verification and vote. Each validator votes for a block by cryptographically signing it. Once the network receives at least a 2/3 majority vote for a block, it is finalized and added to the blockchain. Since valid blocks will contain the digital signatures of the validators, nodes synchronizing to the blockchain need only check the validator signatures in a block to ensure they are following the correct blockchain.

BFT consensus usually requires a certain minimum number nodes to ensure the network can operate in the face of malicious actors, for example, 3F+1[3], where F is the number of faulty nodes. Compared to PoW, this approach also requires the exchange of more protocol messages to coordinate the consensus process, limiting its scalability. While PoW can support thousands of nodes, BFT is limited to at most hundreds.

Table 1. Consensus Comparison.

	BFT	Nakamoto
Speed (transactions per second)	**Potentially 1000s**	**< 20 on average**
Network Scalability	**100s of nodes**	**1000s of nodes**
Block Finality	**Instant**	**Eventual**

There is not a “one size fits all” consensus algorithm for blockchain technology. Selecting a consensus algorithm for a given use case will require a lot of thought and attention to detail both in the application itself and in the externalities involved.

[2] A validator is the term used for a “miner” in a blockchain using BFT.
[3] This may vary based on the specific protocol.

3.3 Distributed Ledger

A blockchain uses the notion of a ledger to capture and record a history of all transactions. The ledger is distributed and replicated across every machine on the network. New blocks can be appended to the blockchain but prior blocks cannot be modified or deleted.

The term "blockchain" comes from the fact that transactions are batched together in a "block" and blocks are linked together to form a chain. Each block is linked to its parent via a SHA256 cryptographic hash of the contents of the previous block, forming the chain of blocks over time (Figure 1). The cryptographic hash of the contents of a block serve as a fingerprint of the block's contents. Changing anything in the contents of a block results in a completely different hash value, breaking the connection between blocks. The use of a hash to link the blocks coupled with the consensus protocol provides the immutability of the ledger.

Figure 1. Blocks are Linked by the Hash of Their Contents.

Anyone can verify the provenance and authenticity of the blockchain and its contents by following the links and validating the hash in each block as show in Figure 1. For an adversary to alter the blockchain they would need to rebuild the links connecting the blocks from the point of the altered block, which is mathematically impossible based on the hash function described in Section 2.1. The feasibility of such an attack is dependent on the consensus algorithm used by the blockchain because that algorithm decides the conditions for a transaction to be added. The complete blockchain consisting of all blocks and their constituent block data (transactions, metadata, and the hash of the parent's contents) is stored on every node in a key-value database such as Google's open-source LevelDB. In addition to the block data the key-value store also contains the root hash of the Merkle Tree (Merkle Tree, 2017).

The Merkle Tree is a compact representation of the current state of the application for the given set of transactions in a block. A Merkle Tree root hash value efficiently represents the overall state of the data, making it easy to use the root of the tree to determine if a particular data item is a member of the blockchain state. What the actual state represents depends upon the state transition logic of the blockchain application.

Mutating application state is accomplished by sending transactions to the blockchain and processing those transactions for a given block, possibly through smart contracts.

3.4 Smart Contracts

A smart contract is the business logic of a blockchain. It dictates the rules of how state is changed on an object over time. A smart contract provides the ability for developers to extend the functionality of a blockchain by adding application-specific logic.

Although the term smart contract has been around since the late 1990s, Ethereum popularized the term by being the first public blockchain to provide a Turing complete smart contract language.

The goal of Ethereum is to provide a world computer for which anyone can build and deploy blockchain-based applications, often referred to as Decentralized Applications (DAPPS). While other blockchains provide different approaches for adding business logic to the blockchain, Ethereum was specifically designed with a custom virtual machine for running full-featured, deterministic smart contracts.

A critical requirement of any smart contract is that they must execute in a deterministic (Deterministic System, 2017) fashion. Every miner node on the network must agree on the output of their computation. Therefore, it's critical smart contracts always return the same result for a given input. If different machines are getting different results from the execution of the same smart contract, the blockchain will not be able to reach consensus on the correct state of the system.

3.5 Peer to Peer Network

A blockchain application requires both a client and server application, which is often referred to as a blockchain *node.* Nodes connect directly to other nodes to form a fully decentralized network on top of the internet.

While the main responsibility of a peer-to-peer network on the blockchain is to propagate transactions and block information across the network, P2P networks use a specific protocol to provide other functions as well, among those are maintaining connectivity, flow control, node discovery, communicating the consensus protocol, and filtering malformed or non-verified transactions.

Since blockchain P2P networks do not rely on centralized services they are very resistant to failures and denial of service attacks.

3.6 Typical Transaction Flow

By following a transaction through its lifecycle, we can see how all the components of a blockchain work together.

1. A transaction is constructed by an end user, for example Bob wants to transfer 10 widgets to Alice.
2. The transaction is cryptographically signed by Bob and sent to the blockchain network over the P2P network.
3. Each node in the network validates the transaction to ensure it is properly signed and correct. If validation fails, the transaction is dropped and not propagated on to other nodes.
4. At some point in time, a validator or miner node batches a set of transactions together, executes them against one or more smart contracts, creates a block, and proposes the block to the network.
5. Once the majority of the network agrees to the validity of the block, it is added to the blockchain and linked by the cryptographic hash of the parent block.
6. The transaction is complete and permanently recorded on the blockchain.

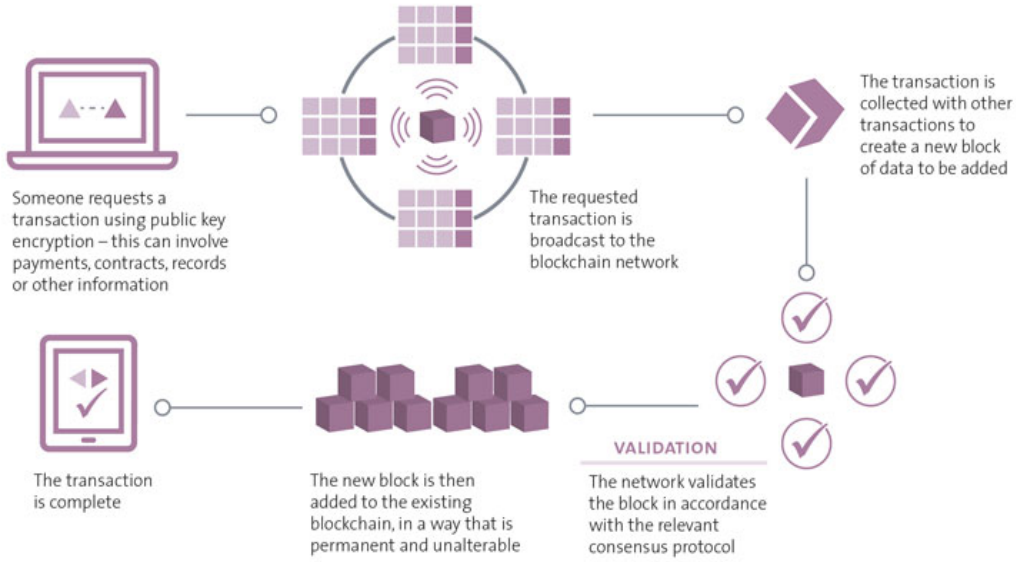

Figure 2. Following a Transaction Through the Blockchain (Blockgeeks, 2017).

4 Public vs. Permissioned Blockchain

There are primarily two[4] different types of blockchains based upon the consensus algorithm they use: public and permissioned.

4.1 Public Blockchain

Public blockchains are the most well-known and battle-tested implementations in production. Bitcoin and Ethereum are two examples. Anyone can join a public blockchain by simply downloading and running the respective open-source implementation. Public blockchains are inherently designed to run in a trust-less environment with the assumption there *will be* malicious actors.

In order to operate in such an environment, most public blockchains use Nakamoto Consensus (Section 3.2.1), which uses a combination of game theory and economic incentives to secure the network via a cryptocurrency with value. Therefore, you must have a cryptocurrency balance in a public blockchain account in order to transact on the blockchain, whether it be a simple financial transaction or a function call to a smart contract. Because of the cryptocurrency requirement, there's no real sense of identity on a public blockchain. Users are associated with a pseudonymous address, which is derived from the public key they use to sign transactions.

4.2 Permission Blockchain

In contrast, a permissioned blockchain is a closed network often using a BFT algorithm. The validators responsible for building the blockchain, as well as the participants in the network, are selected by the group and held accountable for their actions.

[4] There are actually more, but public and permissioned are the dominant models.

There is stronger notion of identity on a permissioned blockchain versus a public one. Because of identity there is no need for a cryptocurrency, or other forms of a token, to secure the network.

Table 2. Public vs. Permissioned Blockchain.

	Who can Participate?	Requires a Cryptocurrency?	Identity of Participants
Public	**Anyone**	**Yes**	**Pseudonymous**
Permissioned	**Closed**	**No**	**Known to the group**

5 Considerations for Using a Blockchain

Bringing trust to a trust-less network is one of the most appealing features for using a blockchain. Many of today's applications have a need to share or move information across organizational boundaries between parties that may or may not trust one another.

Blockchain has the potential to bring trust to such an environment without relying on a 3rd party to mediate transactions. However, careful consideration should be made to determine if blockchain technology is the right fit for the given problem.

1. Current blockchain implementations are not capable of the same volume of transaction throughput that you may find with traditional database technology. For example, public blockchains such as Ethereum can currently process approximately 12-15 transaction per second (TPS), while Bitcoin is capable of about half of that. Permissioned blockchains are much faster ranging from tens to thousands TPS depending on the consensus algorithm. As a comparison, both public and permissioned blockchains are much slower than networks like the Visa processing network, which can process tens of thousands of transactions per second.
2. Blockchain data is not private. Anyone with access to a node, can read all transaction and application state data. However, there are current efforts to provide private transactions using advanced cryptography.
3. The decentralized nature of the technology will in many cases require a redesign of the business processes, as most are based on centralized control and/or trusted 3rd parties. Force-fitting existing processes into a blockchain could result in centralizing a blockchain and limiting its potential.
4. Application development is difficult. Smart contracts operate under an asynchronous programming model as you don't know exactly when or if a transaction is added to the chain. This is in sharp contrast to the dominant request/response model used by most web applications today. Additionally, application logic must be programmed to be deterministic, which can limit the potential logic that can be ported to a smart contract.
5. There are variants of consensus algorithms used by permissioned blockchains. Not all are intended to operate in a trust-less environment. Careful attention should be used in selecting the right consensus algorithm for a given use case.
6. In the case of a permissioned blockchain, consideration needs to be made in terms of the governance process. Since a permissioned blockchain is a closed group, participants must decide who will serve as the validating nodes in the consensus process. Additionally, a process for managing user access must be considered–who can participate, who has the power to add or restrict users, etc. Without careful planning and proper governance, this could lead to a politically centralized blockchain by putting too much power in the hands of a few. Potentially reducing some of the advantages and reasons for considering a blockchain in the first place.

As noted in Section 4, there are predominately two types of blockchains. Each fit a specific need depending on the use case and the needs of the organization. Here are some additional factors you should consider when exploring blockchain technology:

- Do all participating parties trust each other?
- Will more than one user be writing data to the system?
- Is a trusted party **required** as part of the business process?
- Do all participants need to see a synchronized view of the data?
- Does the data need to be private?
- Do you need to control who can deploy smart contracts to the system?

One potential blockchain evaluation framework is shown in Figure 3.

6 Potential Use Cases

Application areas under investigation and of potential value to MITRE sponsors are numerous and include healthcare, identity, supply chain (and logistics), Internet of Things (IoT), additive manufacturing, and secure information sharing among others. Several of these uses cases are presented below.

6.1 Healthcare

Blockchain is being investigated as the foundation for numerous healthcare applications including patient records, drug trials, medical supply chain, and payment processing.

Bruce Broussard, president and CEO of Humana, posits blockchain will become the next big healthcare technology innovation, particularly as it relates to payments and payer contracts. For example, in a situation when a health plan and patient are dealing with a contract, the blockchain can automatically verify and authorize information and the contractual processes. "There is no more back-and-forth haggling with the health plan about what was paid, why it was paid or whether it should have been paid," he wrote. "*With transparency and automation, greater efficiencies will lead to lower administration costs, faster claims and less money wasted.*" (Healthcare Information Tech)

Another potential healthcare application is population health. Instead of relying on health information exchanges or other ways to aggregate data, organizations can eliminate the middleman and access patient databases on a large, population scale. "*Spending time and resources verifying members' trustworthiness (e.g., Health Information Exchange [HIE], all-payer claims database, or local Electronic Medical Records [EMRs]) no longer makes savvy business sense. Blockchain will leap frog population health by providing trust where none exists for continuous access to patient records by directly linking information to clinical and financial outcomes,*" reports CIO. (Blockchain Apps for Healthcare)

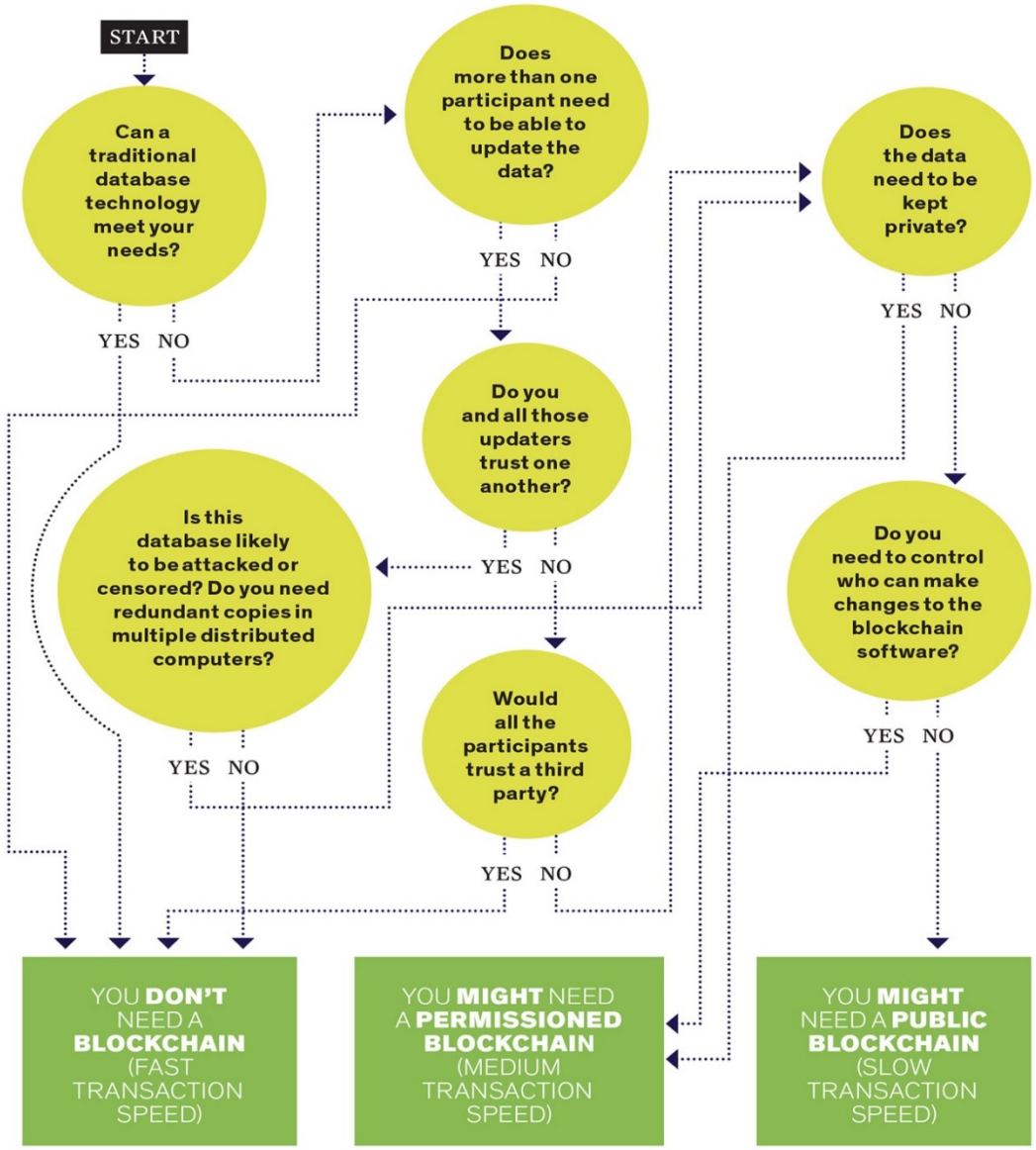

Figure 3. Blockchain Decision Chart (IEEE: Do you need a blockchain).

A registry of all devices connected to the Veterans Affairs (VA) IoT is a potential beneficial use of blockchain. A second use could be digitizing the transactions of Veterans' monetary benefits, like pension and compensation. Health researchers are also exploring using blockchain in healthcare records management. Blockchain could be used to securely exchange large volumes of data while ensuring patient privacy and maintaining data integrity. Further, a blockchain's record-keeping ability could be used in VA hospitals to track a patient's hospital visit, with developments tracked in a ledger as a transaction. (Tech Insight: Blockchain)

6.2 Identity

Online identity management remains a time-consuming and costly process. There is much churn with respect to identity: for example, the Equifax breach has driven Congress and others to rethink two major parts of the current credit system: having credit bureaus store most Americans' identity data and using Social Security numbers as a primary identifier.

The ability to record, track, and manage identity on a blockchain has the potential to vastly improve the efficiency and minimize the cost of identity management; an immutable, trusted source of identity will make it difficult to steal, hack, modify, or otherwise damage reputation, or compromise identity to steal real assets or perpetrate fraud.

A well-constructed blockchain identity solution would help consumers, businesses, and government alike; users can choose how much information to release on an as-needed basis. Trusted information will save businesses and government time and money. "There's no doubt that the blockchain concept, with its power to prevent duplication and divergence from the chain, is highly promising for identity," said Jo Ann Barefoot, a former deputy comptroller at the Office of the Comptroller of the Currency. (Build a Better Credit Bureau, 2016)

Identity is central to many activities of a typical consumer's life. It's also applicable across a wide range of MITRE's sponsors. For example, the Department of Homeland Security (DHS) recently awarded a Blockchain Tech development grant for identity. DHS is developing an identity management solution built on a permission-less blockchain, with a focus on confidentiality (with selective information disclosure), integrity, availability, non-DHS repudiation, provenance, and pseudo-anonymity. (DHS Awards Blockchain Grant)

A good use case related to identity is trusted propagation of cancelled credentials; once your credentials are cancelled on the shared truth blockchain, it is visible to all valid participants and there's no way to game the system.

Advances in biometrics plus digital identity and blockchain present interesting possibilities for MITRE sponsors. HYPR is a startup using blockchain and biometrics for enterprise identity applications. HYPR decentralizes and secures any type of credentials including passwords, Personal Identification Numbers (PINs), and biometric authentication such as fingerprint, face, hand, retina, iris, voice, and behavior. (Decentralized Access Control)

As with any asset management solution, there are issues of data freshness and integrity. Standards for identity management using blockchain are not yet set and best practices are still being developed. Research is needed into the blockchain's ability to protect private information. Once information is recorded on the blockchain, it remains accessible to all parties in the network, so users must be aware to minimize any private information.

6.3 Supply Chain and Logistics

The United States Department of Defense (DoD) is one of, if not the, largest logistics organizations in the world; the challenges of procuring, tracking, deploying, and managing goods is enormous. "*DoD's supply chain depends on an enormous number of globally distributed companies, making it hard to oversee them all. This has traditionally been part of the challenge of developing anti-counterfeit systems. Blockchain may offer a simpler way to comply, broadening the set of companies that can compete for military contracts without watering down oversight.*" (Pentagon has the World Largest Logistic Problem)

Blockchain provides each participant end-to-end visibility based on their level of permission. Each participant in the supply chain ecosystem can view the progress of goods through the supply chain to include origin, transit, and destination. They can also see the status of customs documents, or view bills of lading and other data. Detailed visibility of the good's progress through the supply chain is enhanced with the real-time exchange of original supply chain events and documents. No one party can modify, delete, or even append any record without consensus from others. This level of transparency helps reduce fraud and errors, reduce the time products

spend in the transit process, improve inventory management, and ultimately reduce waste and cost.

Blockchain supply chain solutions could extend to many other government groups as well, e.g., verifying the provenance of medical equipment within the VA, tracking food and drug shipments for the Food and Drug Administration (FDA), and generally managing supply chain risk any government entity.

System issues must be carefully considered when using blockchain in supply chain and coordination applications. Data must be entered securely and accurately. Access to controlled assets must be managed so that system's physical (and software) configurations remain in sync with the blockchain. Smart sensors, various ID tags, and workflow processes are critical items to building a complete blockchain-supported supply chain.

6.4 Internet of Things

The IoT is impacting every segment of industry, consumers, and government. With tremendous growth in smart devices comes many challenges including: security, lifecycle management, device and data integrity, and device authentication.

A potential use case: In June 2016, the DHS' Science and Technology (S&T) Division awarded a $199,000 contract to Factom to study possible blockchain-based advancements for the security of digital identities for the IoT, the upcoming connection and convergence of mobile devices, information technology networks, and connected sensors and devices.

The project, titled "Blockchain Software to Prove Integrity of Captured Data From Border Devices," will create an identity log that captures the identification of a device, who manufactured it, lists of available updates, known security issues, and granted authorities while adding the dimension of time for added security. The goal is to limit would-be hackers' abilities to corrupt the past records for a device, making it more difficult to spoof. It's interesting to note that a North Atlantic Treaty Organization request for proposal also included an IoT section, which underlines the synergy between IoT and blockchain technologies for military applications. (DHS Awards Blockchain Grant of Identity Managment, 2016)

"*IoT devices are embedded within our daily lives-from the vehicle we drive to devices we wear - it's critical to safeguard these devices from adversaries,*" said DHS Under Secretary for Science and Technology, Dr. Reginald Brothers. "*S&T is excited to engage our nation's innovators, helping us to develop novel solutions for the Homeland Security Enterprise.*" (DHS Awards Blockchain Grant of Identity Managment, 2016)

7 Survey of Permissioned Blockchains

7.1 Ethereum

Ethereum (Ethereum Blockchain Application Platform, 2015) is an open-source blockchain platform that enables developers to build and deploy decentralized applications. In addition to the common blockchain components, Ethereum includes a custom virtual machine known as the Ethereum Virtual Machine (EVM) for executing smart contracts.

The EVM is tightly integrated into an Ethereum node, storing the state of smart contracts in a Merkle Tree on the blockchain alongside transactions. Smart contract applications are written in the Solidity language in a syntax that closely resembles JavaScript. Solidity code is compiled to a

binary format and deployed to the blockchain for execution. The flexibility of the EVM allows for the creation of powerful smart contracts.

The default version of Ethereum is a public blockchain using a PoW consensus algorithm. The cryptocurrency of Ethereum is known as Ether and is divisible up to 18 decimal places allowing for very small micro payments.

The Ethereum codebase has a pluggable interface for consensus algorithms, providing the ability to use an alternative consensus model. This is in part due to the growing interest in using Ethereum as a permissioned blockchain in the enterprise space (Enterprise Ethereum Alliance, 2016). Recently, a company contributed a more traditional BFT consensus plug-in that would be a good fit for permissioned blockchains. Initial claims show support of up to 1200 transactions per second.

7.2 Quorum

Quorum (Quorum, 2016) is an open-source version of Ethereum maintained by J.P Morgan. It is specifically designed for permissioned blockchains yet shares many features with Ethereum. Quorum includes three alternatives for consensus: QuorumChain, Raft, and Istanbul BFT.

- **QuorumChain**: The first consensus alternative is an innovative round-robin voting consensus algorithm implemented through an Ethereum smart contract. Parties that can create and/or vote on blocks are pre-configured in the initial block (genesis block) and become part of the state of the QuorumChain contract. Calls to consensus are through the traditional Ethereum transactions model. Since the contract uses a simple round-robin voting approach to consensus (it is not BFT), it is *not* immune to malicious actors and should not be used in a network with untrusted participants.
- **Raft**: The second consensus alternative is an implementation of the well-known Raft algorithm (Ongaro & Ousterhout, 2014). Like QuorumChain, Raft is not BFT and should not be used in a trustless environment.
- **Istanbul BFT**: The third option is based on a BFT algorithm. This is targeted to support limited-trust environments. However, as of this writing the code has not yet been merged into the official release and is still undergoing testing.

One of the key features of Quorum is its support for private transactions. While Quorum supports traditional Ethereum public transactions, it also provides for private transactions between parties. Quorum extends the Ethereum transaction model by running additional nodes called Constellation that are responsible for securing and processing private transactions.

"*Constellation is a general-purpose system for submitting information in a secure way. It is comparable to a network of MTA (Message Transfer Agents), where messages are encrypted with PGP (Pretty Good Privacy). It is not blockchain-specific, and is potentially applicable in many other types of applications where you want individually sealed message exchange within a network of counterparties*." (Quorum Overview)

Quorum stores the state of private and public transactions separately so that private transactions can only be read and processed by the intended parties with access to a specific node. The downside to this approach is that anyone with access to the node can read the private transactions.

7.3 Tendermint

Tendermint (Tendermint Blockchain Consensus, 2017) is an open-source blockchain platform, consisting of three chief technical components: a consensus engine, P2P network layer, and a generic application interface for developing smart contracts.

Tendermint's consensus engine is a BFT consensus protocol ensuring blocks are validated and recorded on every machine in the same order. Tendermint block times are on average 1 second or less, allowing thousands of transactions per second.

For smart contracts, Tendermint provides an extensible application interface, called the Application Blockchain Interface (ABCI). Using the ABCI interface developers can implement smart contracts in several different programming languages to include: Go, Python, JavaScript, Erlang, C++, and more.

Tendermint is designed to be easy-to-use, simple-to-understand, highly performant, and useful for a wide variety of distributed applications. (Introduction to Tendermint, 2016). MITRE has used Tendermint for several prototypes.

7.4 Hyperledger Fabric

Hyperledger Fabric (Hyperledger Fabric, 2016) takes a similar approach to Tendermint as a framework for building blockchain applications. It provides a consensus engine with a BFT consensus algorithm based on Practical Byzantine Fault Tolerance (PBFT), as well as an application program interface (API) for interacting with the blockchain. The original implementation of Fabric was contributed to the open-source Linux Foundation by Digital Asset and IBM.

Smart contracts in Fabric are referred to as "chaincode" and are currently implemented in the Go language and must be run in Docker containers. In addition to chaincode, Fabric also provides additional features:

- Membership service provider: a service to facilitate the enrollment of members that are authorized to participate in the blockchain application
- Channels: the ability for group to create a separate ledger of transactions for privacy

7.5 Guardtime

Guardtime is an Estonian and U.S.-based company that offers a few proprietary blockchain solutions to governments and industry. One of their main solutions is known as Keyless Signature Infrastructure or KSI (Keyless Signatures 2013).

The goal of KSI is not to provide a secure immutable-state machine, but rather to provide a secure methodology to make assertions about the time and integrity of all sorts of digital records while removing as many cryptographic and human assumptions from the security of the system as possible (Guardtime Technology 2017). Guardtime does this through a trusted infrastructure and hash-based signatures instead of asymmetric key-based ones.

Guardtime's KSI is a very different technology than other blockchain platforms, with some benefits and drawbacks as compared to other blockchain systems.

Table 3. Guardtime Pros and Cons

Pros	Cons
1. **Extremely fast signature processing and verification** 2. **Immutable time-stamping of signatures** 3. **Does not require permission to use the blockchain** 4. **Its security comes from minimal assumptions (mostly the cryptographic properties of hash functions)** 5. **Quantum resistant**	1. **Proprietary** 2. **Requires trusted infrastructure of signer nodes to sign data** 3. **Does not store data on the blockchain, only signatures of the data** 4. **While verifying a given signature's presence is easy, searching for a given signature is hard/impossible**

8 Emerging Features

Blockchain technology is advancing at a very fast pace. There are a few features currently in development across several open-source communities that will be of value to future permissioned blockchain applications.

8.1 Private Transactions

Blockchains were originally intended to be publicly auditable. Therefore, all information stored on the ledger (transactions and state) are readable by anyone with access to a node. Recently, a few public blockchains such as Zcash and Monero were developed to protect the privacy of transactions between parties by shielding the contents of a cryptocurrency transaction to everyone but the parties involved.

Privacy of transactions in a permissioned blockchain is one of the most desired features for enterprise users, but production-level support is generally lacking in most permissioned blockchains. The Quorum blockchain outlined in section 7.2 provides support for private transactions. And there is active interest in using sophisticated cryptographic techniques already used by Zcash known as Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARKS) (Non-Interactive Zero-Knowledge Proof, 2017). Support for adding zk-SNARKS to Ethereum is currently under development.

8.2 State Channels

Blockchain transaction throughput is not nearly at the level of speed that most enterprise users are accustomed to with traditional database technology. This is due largely to the overhead of the consensus process. One solution to increase the transaction speed of a blockchain is through the use of state channels.

A state channel is a bi-directional channel between parties. Messages in the channel take the form of transactions. Parties sign messages in the channel and anchor certain transactions to the blockchain at specific checkpoints, making the series of interactions leading to the final transaction impossible to refute later.

The majority of these transactions take place entirely off the blockchain and exclusively between the participants, meaning they are cheap and very fast to execute compared to on-chain payments. (What are State Channels, 2016)

The open-source project Raiden Network (Raiden Network, 2017) has deployed initial support for state channels on Ethereum; however, current efforts are focused primarily on token transfers.

8.3 Bulk Data Storage

Blockchains are not designed to store large data. Information sharing within the enterprise will require the ability to manage large data without directly storing it on the blockchain. One alternative is to store large data in external decentralized storage such as the InterPlanetary File System (IPFS) (IPFS Distributed Web, 2016) and use a hash fingerprint of the document in a transaction on the blockchain as a reference to the externally stored content. MITRE has conducted initial experiments using this approach with moderate success.

Other efforts, such as the BigchainDB (BigchainDB, 2017) project, are working to connect the immutability and decentralization of blockchain technology with traditional database technology to create a best-of-breed approach for the enterprise.

8.4 Connecting Blockchains

As more organizations deploy blockchain solutions there will be a need to allow multiple parallel blockchains to interoperate. This can be difficult depending on transaction models and the various consensus algorithms used by blockchains.

Various open-source projects, such as Cosmos (Cosmos Whitepaper, 2017) and Polkadot (Polkadot, 2017), are working on solutions to provide interoperability between blockchains, but current efforts are predominately focused on public blockchains.

9 Conclusion

Blockchain technology is evolving at a very rapid pace. Understanding the core components of the technology and how they work together is critical to tracking the state of the art. While each component plays a critical role in the technology stack, consensus is at the heart of the system and important to understand. Carefully choosing the right consensus algorithm based on the desired level of trust and security will be critical to a successful blockchain application.

While public blockchains provide the most security as they are designed to operate in a trust-less environment, government users will be most interested in a permissioned blockchain. However, the nature of a permissioned blockchain requires careful planning and governance to establish the parties participating in the consensus process. Without proper governance, there may be a possibility of politically centralizing some of the key functionality of the blockchain, limiting its capabilities, and providing a false sense of security.

As more look to permissioned blockchains to modernize traditional applications, there are several requirements that need to be addressed. For example, privacy and confidentiality on the blockchain, transaction scalability, and blockchain-to-blockchain connectivity. While there's ongoing active research in these areas across several open-source communities, permissioned blockchains need to further evolve to fully meet the needs of the government user.

10 Bibliography

BigchainDB. (2017). Retrieved from BigchainDB: https://www.bigchaindb.com/

Block Confirmation. (2016). Retrieved from Bitcoin Wiki: https://en.bitcoin.it/wiki/Confirmation

blockchain apps for healthcare. (n.d.). Retrieved from CIO: - https://www.cio.com/article/3042603/innovation/blockchain-applications-for-healthcare.html

Blockgeeks. (2017). Retrieved from Blockgeeks: https://blockgeeks.com/

Build a better credit bureau. (2016). Retrieved 2017, from American Banker: https://www.americanbanker.com/news/can-blockchain-be-used-to-build-a-better-credit-bureau

Byzantine Fault Tolerance. (2017). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Byzantine_fault_tolerance

Coinmarket Cap. (2017). Retrieved from Coinmarket Cap: https://coinmarketcap.com/

Cosmos Whitepaper. (2017). Retrieved from Cosmos: https://cosmos.network/whitepaper

Decentralized Access Control. (n.d.). Retrieved from HYPR Corp: https://www.hypr.com/decentralized-authentication/

Deterministic System. (2017). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Deterministic_system

DHS awards blockchain grant. (n.d.). Retrieved from Nasdaq: http://www.nasdaq.com/article/department-of-homeland-security-awards-blockchain-tech-development-grants-for-identity-management-and-privacy-protection-cm667365

DHS Awards Blockchain grant of Identity Managment. (2016). Retrieved from Nasdaq: http://www.nasdaq.com/article/department-of-homeland-security-awards-blockchain-tech-development-grants-for-identity-management-and-privacy-protection-cm667365

Double Spend Problem. (2017). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Double-spending

Enterprise Ethereum Alliance. (2016). Retrieved from Enterprise Ethereum Alliance: https://entethalliance.org/

Ethereum Blockchain Application Platform. (2015). Retrieved from Ethereum : https://www.ethereum.org/

healthcare information tech. (n.d.). Retrieved from Becker Hospital Review: https://www.beckershospitalreview.com/healthcare-information-technology/9-things-to-know-about-blockchain-in-healthcare.html

Hyperledger. (2017). Retrieved from Hyperledger: https://hyperledger.org/

Hyperledger Fabric. (2016). Retrieved from Hyperledger Fabric: https://www.hyperledger.org/projects/fabric

IEEE: Do you need a blockchain. (n.d.). Retrieved from IEEE Spectrum: https://spectrum.ieee.org/computing/networks/do-you-need-a-blockchain

Introduction to Tendermint. (2016). Retrieved from Tendermint: https://tendermint.readthedocs.io/en/master/introduction.html

IPFS Distributed Web. (2016). Retrieved from IPFS: https://ipfs.io/

Members. (n.d.). Retrieved from Enterprise Ethereum Alliance: https://entethalliance.org/

Merkle Tree. (2017). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Merkle_tree

Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). Bitcoin and Cryptocurrency Technologies.

Non-interactive zero-knowledge proof. (2017). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Non-interactive_zero-knowledge_proof
Ongaro, D., & Ousterhout, J. (2014). In Search of an Understandable Consensus Algorithm.
Pentagon has the world largest logistic problem. (n.d.). Retrieved from Defense One: http://www.defenseone.com/ideas/2017/10/pentagon-has-worlds-largest-logistics-problem-blockchain-can-help/141500/
Polkadot. (2017). Retrieved from Polkadot: https://polkadot.io/
Quorum. (2016). Retrieved from JP Morgan: https://www.jpmorgan.com/country/US/en/Quorum
Quorum Overview. (n.d.). Retrieved from Github: https://github.com/jpmorganchase/quorum/wiki/Quorum-Overview
Raiden Network. (2017). Retrieved from Raiden: https://raiden.network/
Reitwiessner, C. (n.d.). *zkSNARKS in a nutshell.* Retrieved from Ethereum Blog: https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/
Tech Insight: Blockchain. (n.d.). Retrieved from VA: Office of Information Technology: https://www.oit.va.gov/library/programs/ts/ti/2017/blockchain.pdf
Tendermint Blockchain Consensus. (2017). Retrieved from Tendermint: https://tendermint.com/
What are State Channels. (2016). Retrieved from State Channels: https://blog.stephantual.com/what-are-state-channels-32a81f7accab

Acronyms

Acronym	Definition
ABCI	Application Blockchain Interface
BFT	Byzantine Fault Tolerant
CPU	Central Processing Unit
DAPPS	Decentralized Applications
DHS	Department of Homeland Security
DoD	United States Department of Defense
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
EMR	Electronic Medical Record
EVM	Ethereum Virtual Machine
FDA	Food and Drug Administration
HIE	Health Information Exchange
IoT	Internet of Things
IPFS	InterPlanetary File System
KSI	Keyless Signature Infrastructure
MTA	Message Transfer Agents
P2P	Peer to Peer
PBFT	Practical Byzantine Fault Tolerance
PIN	Personal Identification Number
PoW	Proof of Work
PGP	Pretty Good Privacy
S&T	Science and Technology
TPS	Transaction Per Second
VA	Veterans Affairs
zk-SNARKS	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge