BLOCKCHAIN

FOR THE ENTERPRISE

WHITE PAPER

KEYHOLE SOFTWARE

Introduction To Blockchain Architecture And Its Value To The Enterprise

WITH CODE WALKTHROUGH

THERE ARE LEGITIMATE **BUSINESS USE CASES** FOR WHICH BLOCKCHAIN MAKES **SIGNIFICANT SENSE** BUT WE RECOMMEND YOU DON'T GET SWEPT AWAY BY THE **ENDORPHIN RUSH**

Blockchain For the Enterprise

A Keyhole Software White Paper

Introduction to Blockchain Architecture and Its Value to the Enterprise

Table Of Contents

Tel 877-521-776 | keyholesoftware.com

Tel 877-521-776 | keyholesoftware.com

White Paper Introduction

While Bitcoin is on the minds of the general public worldwide, blockchain is on the minds of information technology professionals. It is the underpinning technology of the powerful and popular cryptocurrency. What exactly is blockchain and how will it help my business? That is what this white paper seeks to answer.

In the most simple terms, a blockchain is a distributed data system for keeping a ledger of immutable data transactions. We will explore additional complexities through this document, but the simplest way to think of it is a highly distributed transaction log. If you happen to be a developer and you are using Git for source code control, you are already using some of the distributed blockchain elements.

In this white paper, we discuss a number of topics related to blockchain with a particular emphasis on the enterprise. This document will be in three major parts. Part one will include a brief overview and history of blockchain, part two will include a deep technical dive, and the third part is written with the intent to aid managers and executives in their decision making in regards to blockchain. A few topics covered include the following:

- What is a Blockchain?
- Building a Blockchain
- How Are Blockchains Used?
- Blockchain vs. Traditional Data Structures
- Should I Use Blockchain?
- Blockchain Use Cases
- Blockchain Platforms & Providers (In Appendix 1)

This white paper includes many technical elements along with code examples. We believe seeing code in action significantly helps developers to understand concepts. For further hands-on learning, we have created two companion projects that each implement blockchain concepts and algorithms that can be executed. One project is C# and the other is implemented with Java. Please see the Github companion projects by way of the following links:

➢ **Java Blockchain Companion Project**
https://github.com/in-the-keyhole/khs-blockchain-java-example

➢ **C# Blockchain Companion Project**
https://github.com/in-the-keyhole/khs-blockchain-csharp-example

Tel 877-521-776 | keyholesoftware.com

Part One: What Is a Blockchain?

Blockchain technology provides a solution to a challenging distributed computing problem. Before we go into the technical aspects of how blockchain works, let us first discuss the problem it solves. The most popular way to explain the problem is through the following allegory.

A Story of Byzantine Generals

Imagine an epic war being waged in a time before computers or phones. A Byzantine army is in an attempt to seize an enemy town. There are multiple Byzantine army divisions circling the target city, each with one general leading that division. Each separate division must only communicate with each other through messengers.

The generals on the battlefield must agree upon one of two actions: attack or retreat. All divisions must act in unity no matter the decision, assuming that they were to attack with less than their full power, they would surely be defeated. If they do not all retreat, the divisions left would surely be slaughtered. And the armies cannot hold on forever, so a decision must be made.

The only way to communicate between divisions is by word of mouth. Furthermore, they must assume that the messenger may fail. Likewise, one of the generals or messengers is likely sympathetic to the city and would try to purposely sabotage the plans.

This Byzantine Generals' Problem allegory was postulated in the Microsoft research paper by Leslie Lamport, Robert Shostak, and Marshall Pease in 1982.[1]

- ➢ Back to distributed computing: the generals must have an algorithm to guarantee that (a) all loyal generals decide upon the same plan of action, and (b) a small number of traitors cannot cause the loyal generals to adopt a bad plan.

The following image shows how the ideal scenario would appear. All of the generals would coordinate a plan together; their divisions would attack at a single agreed time and day, thus sacking the town.

[1] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem." ACM Transactions on Programming Languages and Systems Journal, July 5, 1982. https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/.

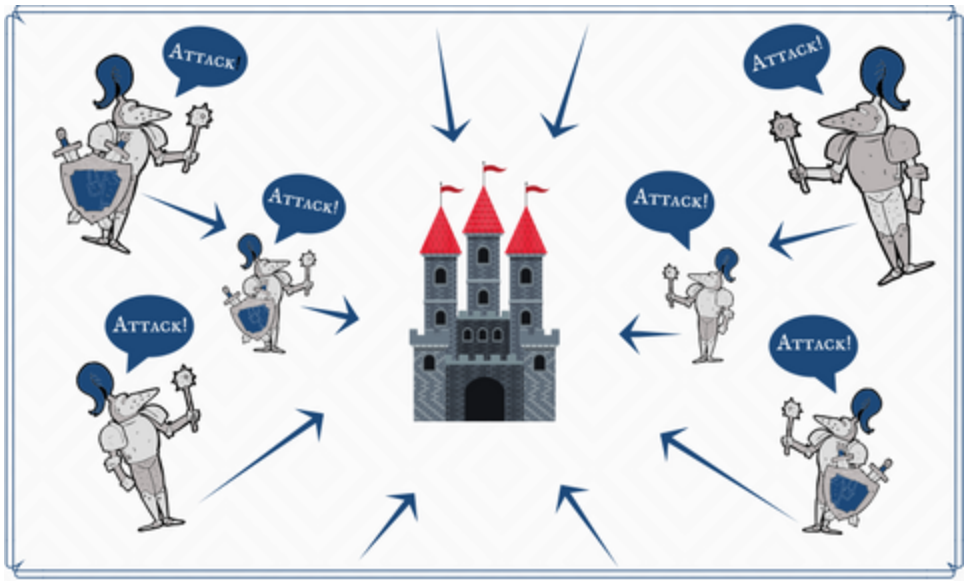

Image 1: Byzantine Generals' Problem Goal, All Division Generals Agreeing On Action In Unison

If all the generals and messengers were trustworthy, it would be a logical and simple solution. Each division speaks to another division until all divisions have been notified of the singular action plan.

But say there are "bad actors" in the chain of command. Perhaps a lieutenant or a messenger is corrupt and changes the message delivered to other divisions. They tell the other division the plan is to retreat instead of attack. Or the messenger is killed, so his message isn't delivered, and a portion of the divisions act on wrong information. This failure in the chain could end like the following:

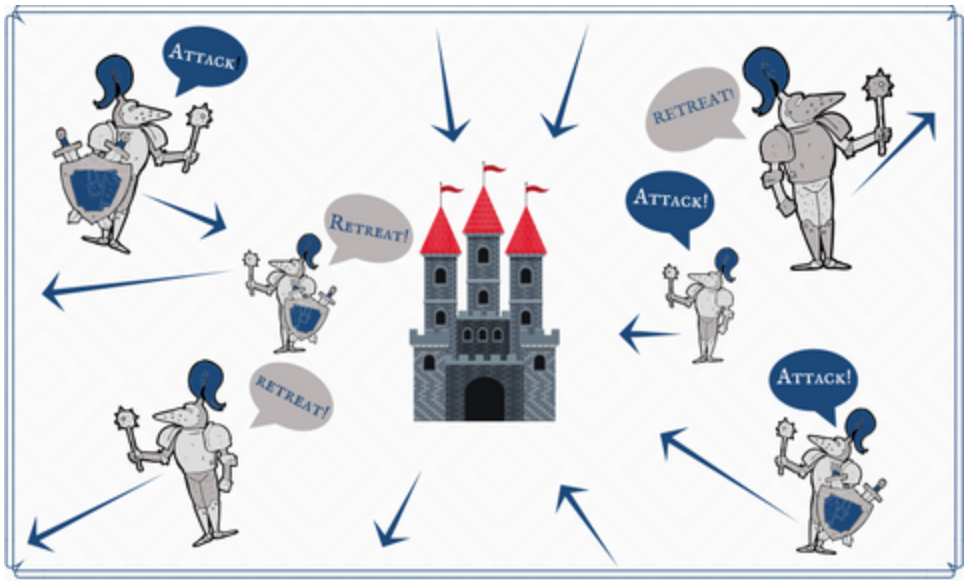

Image 2: Byzantine Generals Problem, Failure To Unify

Tel 877-521-776 | keyholesoftware.com

In the second, more-realistic image, the problem wasn't solved. The generals were unable to reach a consensus. Some led their divisions in an attack and half chose to retreat. Ultimately, the Byzantine army failed to sack the town.

How BGP Applies to Distributed Computing

The Byzantine Generals' Problem is applicable to every distributed computing network. The independent generals act based on the information they are given from other divisions, which could have been tampered with by traitors. There is no central command center ensuring the information given to the general is accurate; there is a chance of failure should the message be bungled by a traitor.

A reliable computer system must be able to function even when its components fail. This is one of the basic problems of distributed computing with open data: how can we trust the data being communicated from independent nodes?

We would argue that the Byzantine Generals' Problem opened the entire field of study on distributed computing. After that, many different approaches were postulated to solve the problem. All were limited to solutions in exponential time. That is, until Satoshi Nakamoto.

Blockchain's Origin

In 2008, an unknown agent or group operating under the pseudonym Satoshi Nakamoto created a white paper entitled *Bitcoin: A Peer-to-Peer Electronic Cash System*.[2] In it, Satoshi offered a solution to the Byzantine Generals' Problem in what was eventually dubbed a "blockchain."

- In the most basic of terms, a blockchain is a distributed, append-only database ledger system.

Think of blockchain as a record of everything that happens—perhaps a transaction, or even an exchange of value, data, goods, and services—exactly as it occurs. The chain puts that data into non-modifiable, encrypted blocks, and distributes those pieces across the network of distributed computers or nodes. Each block is put at the end of a chain of blocks, and every new block includes details of every block in the chain that came before it.

In blockchain, transactions are combined and a hash is computed and linked. Blockchain provides an immutable transaction history that allows consumers to validate consistency of their chain against other members. This allows the system to create consistency across a massive network of independent nodes operating in a peer-to-peer manner. In other words, no middleware or central repository is required.

[2] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." https://bitcoin.org/bitcoin.pdf.

Bitcoin Is A Blockchain, But Blockchains Are Not Bitcoin

Unless you are living under a rock, you've heard about the rise of Bitcoin and other cryptocurrencies. Bitcoin happens to be an application that uses blockchain technology to implement a cryptographic currency mechanism with the specifications defined in Satoshi's white paper.

What makes Bitcoin and similar forms of digital currency unique is that they actually convert real, tangible goods into currency in the form of electricity consumption.

Electricity is a significant factor in the Bitcoin mining process. This is because a transaction block in the Bitcoin ledger is made tamper-proof by calculating a SHA-256 hash of the block data. Computing this hash takes computing cycles. For blockchains that use mining, this is referred to as a **Proof of Work**. The magic of hash functions are one reason that blockchains work. This is referred to as **consensus**, which will be discussed in an upcoming section.

As the blockchain grows, more and more computations are required by computing nodes, therefore more electricity is used. The first one to solve the problem is rewarded with a bit of Bitcoin for themselves, so many people participate in mining. Remember, the Bitcoin blockchain node network is world wide; transactions are being created and blocks mined constantly. As of this writing, there are 511,077 blocks in the chain, and it takes approximately an hour for a transaction to be mined and make its way throughout the network.

There is a screenshot on the following page from Bitcoin's Block Explorer, which is a tool that allows anyone to see the Bitcoin blockchain in action. In the image you can see that there are many transactions occurring.

Mining software must work hard in order to try and get paid. In order for mining to be profitable, speed and scalability are required. Otherwise you'll pay more for the electricity to run your computing nodes than the reward for creating a transaction block. Interestingly, Iceland has become a hotbed for cryptocurrency mining companies due to its cold climate and renewable geothermal energy sources.[3]

[3] Molly Jane Zuckerman, "Culprits Apprehended In Alleged Icelandic Bitcoin Miner Theft." Cointelegraph, 25 Feb 2018. https://cointelegraph.com/news/culprits-apprehended-in-alleged-icelandic-bitcoin-miner-theft.

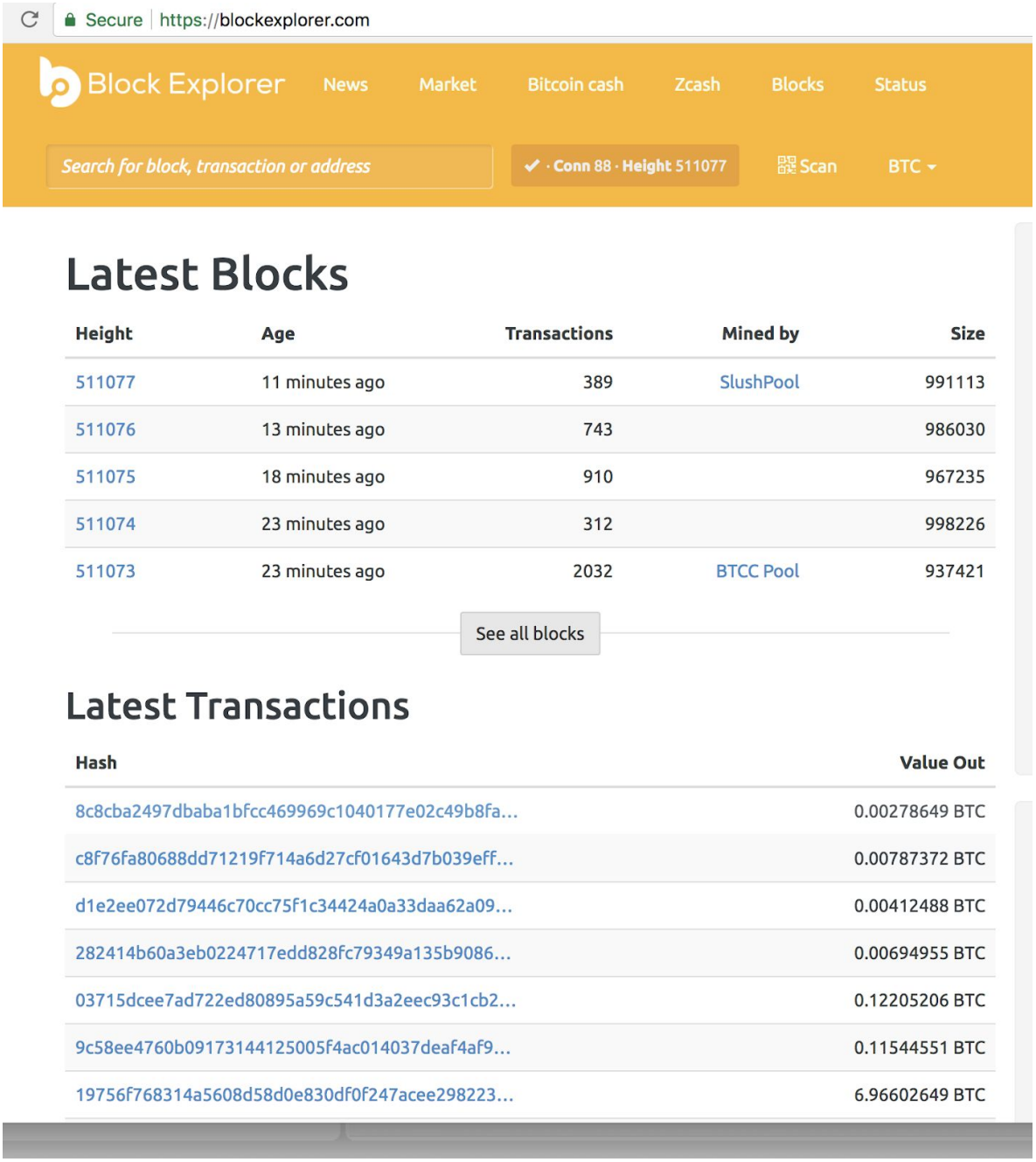

Image 3: Block Explorer

So, the electricity problem is a function of the bitcoin mining consensus mechanism, not blockchain itself. You should understand these mechanisms more after you get through this white paper.

Even though the Satoshi paper describes a system for the cryptocurrency, the underlying blockchain and the cryptographic hash properties can be applied across other domains.

Peer to Peer

A unique feature of blockchain's distributed nature is that it is not a central data store housed on a single server. Rather, the blockchain data is replicated across all the nodes in the distributed system. Nodes in the system communicate with each other using a peer-to-peer network topology. Consider the following image.

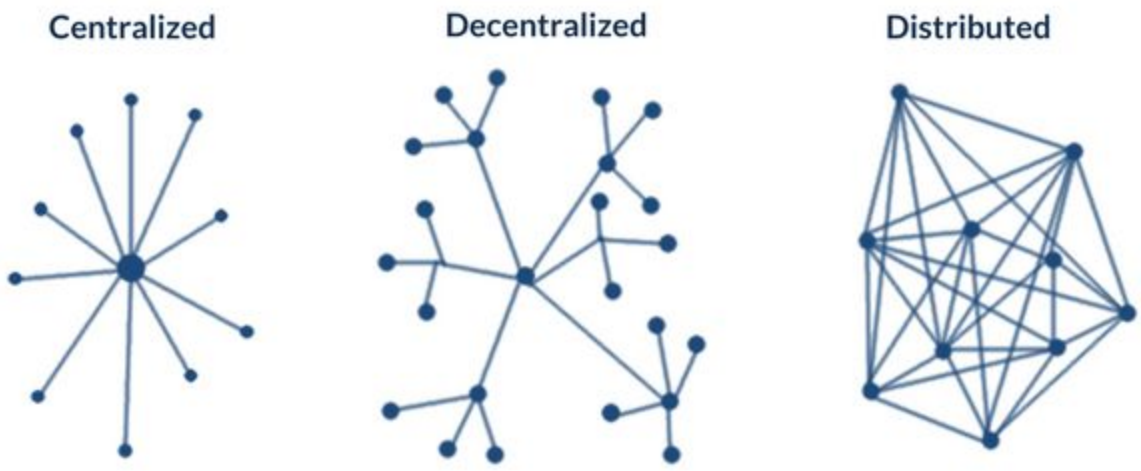

Image 4: Comparison of Centralized, Decentralized, and Distributed Systems

As a comparison, consider a centralized system: one central node does all of the work. It is a challenge to scale. And, like the hypothetical command center in the earlier discussion, it's a single point of failure. In a decentralized system, nodes distribute work to other sub-nodes. It is easier to scale. But in a peer-to-peer or distributed system, nodes are only connected to peers. This distributed architecture provides a stable, fault tolerant, and highly available system. It also eliminates the single point of failure that centralized data stores have.

- ➢ Blast from the past: music and movie sharing services like Napster and Bittorrent are peer-to-peer systems. Thus, we know that peer-to-peer methods can work, even in the consumer space.

Part Two:
Technical Overview of Blockchain

So how does a blockchain keep transactions secure and the system in balance and healthy? The best way to discuss this question is to dig into the technical specifics of building an actual blockchain with code. This section is technical in nature. For hands-on learning, we suggest you refer to the companion projects found on the official Keyhole Software GitHub via the following links:

- **Java Blockchain Companion Project**
  https://github.com/in-the-keyhole/khs-blockchain-java-example

- **C# Blockchain Companion Project**
  https://github.com/in-the-keyhole/khs-blockchain-csharp-example

Hashing

To begin, hashing is an important concept to understand. It is used extensively in blockchain. Currently in Bitcoin alone, there are over 10 *Septillion* hashes.[4] They are used both for integrity verification as well as identification for blocks, transactions, and addresses.

A **hash function** is mathematical, taking an input (any data, any length) and transforming it into a fixed-length output. The process of taking some data and applying a hash function to it is called **hashing**. The result is a **hash**.

In blockchain, transactions are combined and a **hash** is computed. It is a part of what forms a block.

A Secure Hashing Algorithm (SHA) is used to generate the hash. This is a one-way and non-reversible process; it is highly improbable to use the output to determine what the input was.[5] There are a variety of hashing algorithms, but what you should know in regards to blockchain is SHA-2 and its SHA-256 hashing function. Bitcoin uses SHA-256, as it is one of the strongest hash functions currently available.[6]

- For reference, see the Java companion project to this white paper. It has a JUnit hash test that shows how a hash is computed: https://github.com/in-the-keyhole/khs-blockchain-java-example/tree/master/src/test/java/helpers.

[4] "Hash Rate." Bitcoin. http://bitcoin.sipa.be/.
[5] ConsenSys, "Blockchain Underpinnings: Hashing." Medium, 13 Jan 20.
https://medium.com/@ConsenSys/blockchain-underpinnings-hashing-7f4746cbd66b.
[6] "SHA-2." Wikipedia. https://en.wikipedia.org/wiki/SHA-2 (Accessed 21 Feb 2018).

You can also give hashing a try with this online utility. Try out the SHA-256 hash.[7] You will notice that even a small change in the initial data will output a significantly changed hash.

Before blockchain, a common use of hashing was as a secure way to store passwords in databases.

Merkle Trees

Merkle trees are an important foundational component of a blockchain.

A Merkle tree connects data using SHA-256 hashes in a tree structure. An initial root hash is generated; this is the Merkle Root. As nodes or child leaf nodes are added, the main root nodes will contain a list of their child node hashes.[8] The diagram below depicts this.

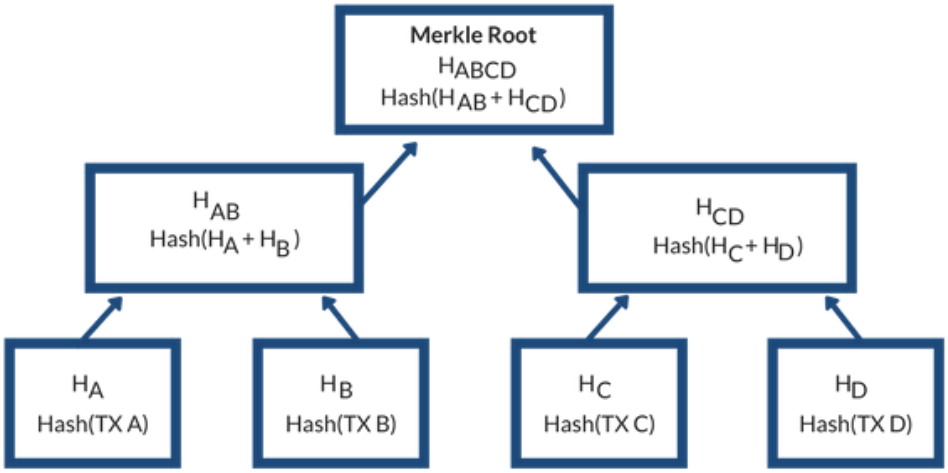

Image 5: Merkle Tree Diagram

The tree structure maintains the order that data is added to the tree and allows it to quickly validate changes to child elements. You must simply walk up the tree and compare hashes for changes. The advantage of this tree structure is that the validity of the data in a tree branch can be validated quicker by comparing with parent hashes up the tree.

- ➢ If you use a Git source code repository, a Merkle Tree hash data structure is used by Git to determine changes to files stored in the repository.

This results in a `O(log(N))` time where `N` is the total amount of data elements. This is as opposed to a list structure which would be `O(N)` time. Blockchains will create a Merkle tree of hashes derived from block data and transactions. This tree is typically a binary tree. How Merkle trees are applied to the blockchain transactions will be discussed after this paper

[7] "SHA256 Hash." Blockchain Demo. https://anders.com/blockchain/hash.html.
[8] Marc Clifton, "Understanding Merkle Trees - Why use them, who uses them, and how to use them." Code Project, 13 Mar 2017. https://www.codeproject.com/Articles/1176140/Understanding-Merkle-Trees-Why-use-them-who-uses-t.

introduces the block and transaction elements.

- **Question:** Isn't there a finite chance of a hash collision? I know the chance in SHA-256 is tiny, but the amount of data is huge.
- **Answer:** It will take a *very* long time to find one. "Let's say you were trying to perform a collision attack and would "only" need to calculate 2^{128} hashes. At the rate Bitcoin is going, it would take them $2^{128}/(300\times10^{15}\bullet86400\bullet365.25) \approx 3.6\times10^{13}$ years. In comparison, our universe is only about 13.7×10^{9} years old. Brute-force guessing is not a practical option." [9]

Transactions

In a blockchain, blocks contain a **Merkle tree** of transaction objects. Transactions contain any kind data object to suit a use case, along with a hash, a parent hash, timestamp, and other audit information. When applied to Bitcoin cryptocurrency, data relevant to this will contain input account, amount, output account, and a public key to securely identify accounts. Below is a general diagram of a Bitcoin-based blockchain transaction data structure.

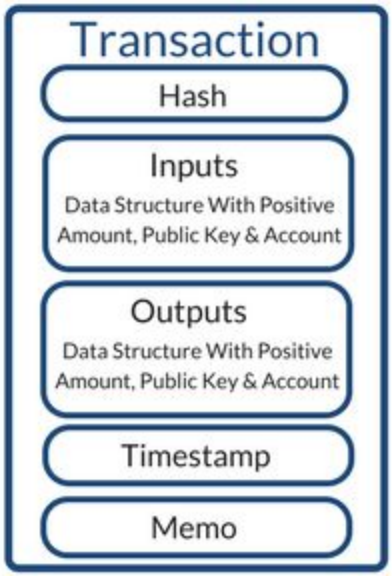

Image 6: Bitcoin Transaction Elements Example

At a minimum, blockchain transactions require a hash, some kind of identity property, and timestamp information. However other data properties can be included which are specific to the unique use case.

Now that we have defined the ability to easily create a Merkle tree of transaction data, we can move on to what makes a block data structure.

[9] Galvatron. Cryptography Stack Exchange, 29 May 2017.
https://crypto.stackexchange.com/questions/47809/why-havent-any-sha-256-collisions-been-found-yet.

A Block

A **block** is the elemental unit of a blockchain. It consists of a block header and transactions in a block. Multiple blocks make up a blockchain. As blocks are confirmed, they are added to the end of the chain. Blocks are linked so they can be traversed.

In the case of Bitcoin, a block will also contain an answer to a mathematical puzzle. The puzzle is difficult-to-solve and the answer will be unique to each block. Only with the correct answer will new blocks be submitted to the network. This is the process of "**mining**," competing to be the next to find the answer that "solves" the current block.[10]

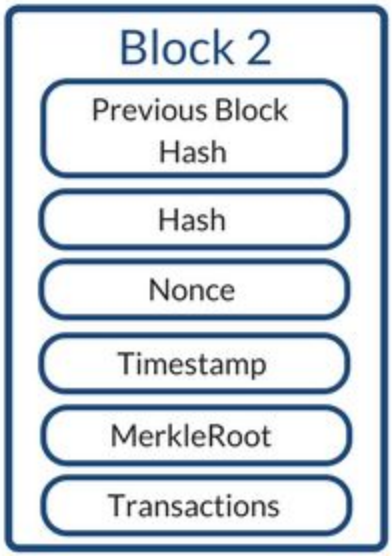

Image 7: Simplified Block Example

Shown below is a block data structure implemented in Java. Notice that the `transactions` property is a generic type definition. This allows transactions in a block to be of any type of data.

- ➢ If this was a block for a Bitcoin blockchain, transactions could be a `java.util.List` of type "Transaction," which would represent a debit or credit amount.

```
...
public class Block<T extends Tx> {
	public long timeStamp;
	private int index;
	private List<T> transactions = new ArrayList<T>();
	private String previousHash;
	private String merkleRoot;
	private String nonce = "12345";
```

[10] "Block." Bitcoin Wiki. https://en.bitcoin.it/wiki/Block (accessed 20 Feb 2018).

```
    // caches Transaction SHA256 hashes
  public Map<String,T> map = new HashMap<String,T>();
    public Block<T> add(T tx) {
           transactions.add(tx);
           map.put(tx.hash(), tx);
           computeMerkleRoot();
           return this;
    }

    public void computeMerkleRoot() {
           List<String> treeList = merkleTree();
           // Last element is the merkle root hash if transactions
           setMerkleRoot(treeList.get(treeList.size()-1) );
    }

...
```

Our `block` object is pretty straightforward, but let's go over each of its properties:

- The **Timestamp**, which is used as a verification of when the block originated.
- The **Index,** which is the designated location in the chain.
- The **MerkleRoot** is a the top of the tree hash and is created by hashing transactions.
- **Transactions** are a transfer of value. The value can be any type of object, from a single entity to a list of individual transactions or messages. In Bitcoin specifically, an individual block may contain multiple individual transactions.
- The **`PreviousHash`** is the SHA-256 hash of the previous hash in the chain.
- **`GetHash`** combines the values of the timestamp, index, transactions, and the previous hash. It then gets the hash for that value.

Blocks will create a Merkle tree of the transaction hashes and use this to validate blocks and provide easier lookup searching of transactions within the block. Blocks are organized into a linear sequence over time.

➢ The companion project has a Merkle tree JUnit test that tests how a block creates a tree and uses it to validate transactions.

```
@Test
public void merkleTreeTest() {

  /// create blockchain, add transactions

 SimpleBlockchain<Transaction> chain1= new SimpleBlockchain<Transaction>();

chain1.add(new           Transaction("A")).add(new           Transaction("B")).add(new
Transaction("C")).add(new Transaction("D"));

 // get a block in chain
 Block<Transaction> block = chain1.getHead();

 System.out.println("Merkle Hash tree :"+block.merkleTree());

  // get a transaction from block
  Transaction tx = block.getTransactions().get(0);
```

```
// see if hash is valid... using merkle Tree...
block.isTransactionValid(tx);
assertTrue( block.isTransactionValid(tx) );

 // mutate the transaction data
  tx.setValue("Z");

  assertFalse(block.isTransactionValid(tx) );

  }
```

With our block now ready, we can begin to assemble our chain.

A Chain of Blocks

With a block defined, they are chained together and managed by a `blockchain` object. This is where the Merkle tree implementation is applied.

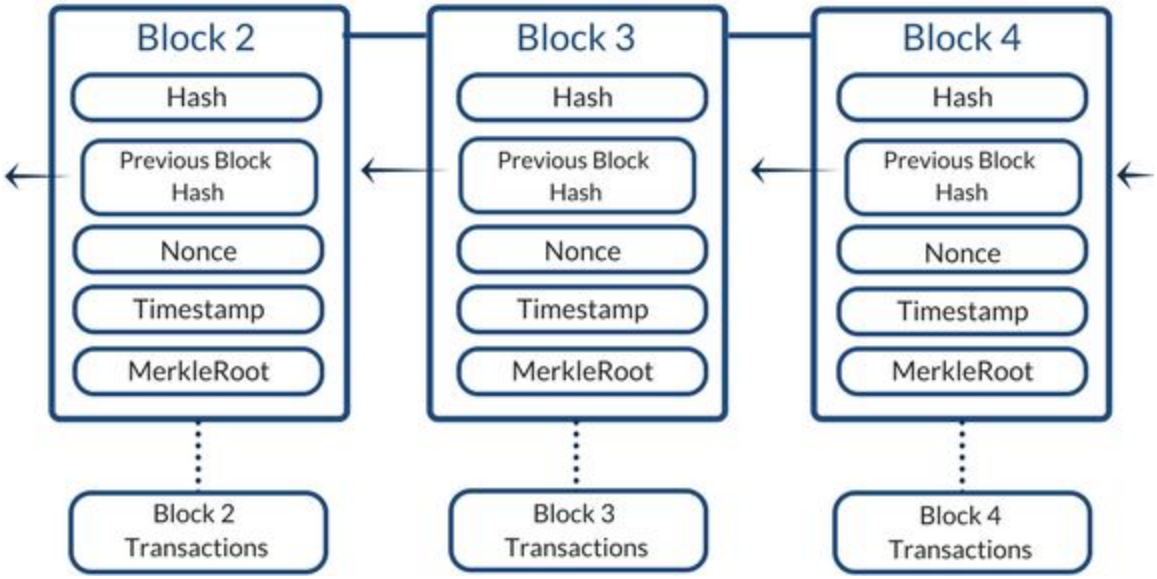

Image 8: Simplified Blockchain

- The companion Java project has a JUnit test implemented that will exercise the blockchain. Here is a link to the unit test in the Java Github project: https://github.com/in-the-keyhole/khs-blockchain-java-example/blob/master/src/test/java/simple/chain/SimpleChainTest.java

An example `SimpleBlockChain` class definition is shown below. You will notice it is fairly straightforward. It essentially houses a list of blocks and maintains the head block of the chain.

```
public class SimpleBlockchain<T extends Tx> {
	public static final int BLOCK_SIZE = 10;
	public List<Block<T>> chain = new ArrayList<Block<T>>();

	public SimpleBlockchain() {
		// create genesis block
```


Tel 877-521-776 | keyholesoftware.com

```
		chain.add(newBlock());
	}

	public SimpleBlockchain(List<Block<T>> blocks) {
		this();
		chain = blocks;
	}

	public Block<T> getHead() {

		Block<T> result = null;
		if (this.chain.size() > 0) {
			result = this.chain.get(this.chain.size() - 1);
		} else {

			throw new RuntimeException("No Blocks have been added to
chain...");
		}

		return result;
	}

	public void addAndValidateBlock(Block<T> block) {

		// compare previous block hash back to genesis hash
		Block<T> current = block;
		for (int i = chain.size() - 1; i >= 0; i--) {
			Block<T> b = chain.get(i);
			if (b.getHash().equals(current.getPreviousHash())) {
				current = b;
			} else {

				throw new RuntimeException("Block Invalid");
			}

		}

		this.chain.add(block);

	}

	public Block<T> newBlock() {
		int count = chain.size();
		String previousHash = "root";

		if (count > 0)
			previousHash = blockChainHash();

		Block<T> block = new Block<T>();

		block.setTimeStamp(System.currentTimeMillis());
		block.setIndex(count);
		block.setPreviousHash(previousHash);
		// chain.add(block);
		return block;
	}

	public SimpleBlockchain<T> add(T item) {

		if (chain.size() == 0) {
			// genesis block
			newBlock();
```

```
			}

			// See if head block is full
			if (getHead().getTransactions().size() >= BLOCK_SIZE) {
				newBlock();
			}

			getHead().add(item);

			return this;
		}

		/* Deletes the index of the after. */
		public void DeleteAfterIndex(int index) {
			if (index >= 0) {
				Predicate<Block<T>> predicate = b -> chain.indexOf(b) >= index;
				chain.removeIf(predicate);
			}
		}

		public SimpleBlockchain<T> Clone() {
			List<Block<T>> clonedChain = new ArrayList<Block<T>>();
			Consumer<Block> consumer = (b) -> clonedChain.add(b.Clone());
			chain.forEach(consumer);
			return new SimpleBlockchain<T>(clonedChain);
		}

		public List<Block<T>> getChain() {
			return chain;
		}

		public void setChain(List<Block<T>> chain) {
			this.chain = chain;
		}

		/* Gets the root hash. */
		public String blockChainHash() {
			return getHead().getHash();
		}

}
```

This is a simple implementation of a blockchain. As you can see, it is generic; it supports generic transaction data.

How Are Blocks Created?

Nodes create transactions in the the peer network, transmitting them to other nodes. A consensus mechanism puts transactions into a block, sending the block to nodes in the network to confirm and validate. Transactions are immutable, which is due to the SHA-256 hash of the transactions and that it also contains the previous block's hash in the blockchain. Nodes can confirm and validate blocks from the consensus easily.

In the Bitcoin blockchain, the consensus mechanism is Proof of Work (POW). It is performed by miners that listen for new transactions and compete to create and add blocks back into network. You might think that "bad actor" miners would have the opportunity to fudge block

data. This would be computationally difficult due to the fact that nodes validate blocks being added to the chain, ensuring computed hashes are linked up the chain.

Block Size and Validation

Block size is typically determined by number of bytes. There is no magic size to a block. The number of transactions in the block can vary depending upon the type of data stored in the blocks.

Bitcoin currently limits blocks to a size limit of approximately 1MB. Satoshi defined the block size for Bitcoin at 1MB to prevent mining spammers from loading the network down with large spam blocks, though increasing the limit has been debated.[11]

Example Implementation: Pulling It All Together

Each of the companion projects implement a simple blockchain; both provide a good way to see some of the blockchain concepts in action. Using the Java companion project, let's start with a simple blockchain implementation. Below is a partial Java class definition.

```
...
public class SimpleBlockchain<T extends Tx> {
	public static final int BLOCK_SIZE = 10;
	public List<Block<T>> chain = new ArrayList<Block<T>>();

	public SimpleBlockchain() {
		// create genesis block
		chain.add(newBlock());
	}
...
```

Notice the parameterized type. This allows the implementation to handle different transaction types. The constructor creates an initial block by calling the `newBlock()` method shown below. The initial block is referred to as the genesis block.

```
...
public Block<T> newBlock() {
		int count = chain.size();
		String previousHash = "root";

		if (count > 0)
			previousHash = blockChainHash();

		Block<T> block = new Block<T>();

		block.setTimeStamp(System.currentTimeMillis());
		block.setIndex(count);
		block.setPreviousHash(previousHash);
		// chain.add(block);
		return block;
	}
...
```

[11] "Block Size Limit Controversy." Bitcoin Wiki. https://en.bitcoin.it/wiki/Block_size_limit_controversy (accessed 26 Feb 2018).

Notice that the blocks are chained together using a `java.util.List` with the latest added block being the head. This is just for demonstration. A real blockchain implementation cannot just assume the last record added is the current head block, but it works for this simple example. When the implementation creates a new block, it sets the `previousHash` with the latest block's hash in the chain.

```
...
public Block<T> getHead() {

              Block<T> result = null;
              if (this.chain.size() > 0) {
                     result = this.chain.get(this.chain.size() - 1);
              } else {

                     throw new RuntimeException("No Blocks have been added to
chain...");
              }

              return result;
       }
...
```

With the blockchain implemented, we can now define the blocks contained in the blockchain.

Block Type

Below is a partial definition of blockchain class properties.

```
...
public class Block<T extends Tx> {
       public long timeStamp;
       private List<T> transactions = new ArrayList<T>();
       private String hash;
       private String previousHash;
       private String merkleRoot;
       private String nonce = "0000";
...
```

The `Block` class is defined with a generic transaction type, in addition to the block identifying properties for the block header and a list of `transaction` objects. The `block` method that adds `transaction` objects to the block follows.

```
...
public Block<T> add(T tx) {
          transactions.add(tx);
          map.put(tx.hash(), tx);
          computeMerkleRoot();
          computeHash();
        return this;
       }
...
```

When added, it computes a new hash for the block as well as computing the Merkle tree root hash with the new transaction. With the block implementation defined, the final data object

is the `Transaction`.

Transaction Type

For this example, the `Transaction` object is defined as a type with two properties and implements a `Tx` interface to support polymorphism.

```
public class Transaction implements Tx {

	private String hash;
	private String value;

	public String hash() { return hash; }

	public Transaction(String value) {
		this.hash = SHA256.generateHash(value);
		this.setValue(value);
	}
```

Again, this is a simplistic implementation. Bitcoin transactions have much more to them: input transaction properties, output transaction properties, address hash, and a PKP-based public key to support a digital signature. Public key has been omitted for this example block transaction type.

Creating a BlockChain Unit Test

The companion projects have JUnit tests that test the simple blockchain implementation. We break down the Java companion project source below.

Two blockchain objects are created. A `clone` method is used to make a copy of the `chain` object. Imagine the clone emulating the peer-to-peer, transmitting blocks between nodes. The chains have three transactions: "`A`", "`B`", and "`C`".

```
...
@Test
public void testBlockchain() {

SimpleBlockchain<Transaction> chain1 = new SimpleBlockchain<Transaction>();
chain1.add(new Transaction("A")).add(new Transaction("B")).add(new Transaction("C"));
SimpleBlockchain<Transaction> chain2 = chain1.Clone();
...
```

An additional transaction "`D`" is added to `chain1` and the two chain hashes are compared. They will be false, since `chain1` has an additional transaction.

```
...
chain1.add(new Transaction("D"));

System.out.println(String.format("Chain 1 Hash: %s", chain1.getHead().getHash()));
System.out.println(String.format("Chain 2 Hash: %s", chain2.getHead().getHash()));
System.out.println(
String.format("Chains Are In Sync: %s",
chain1.getHead().getHash().equals(chain2.getHead().getHash())));
...
```

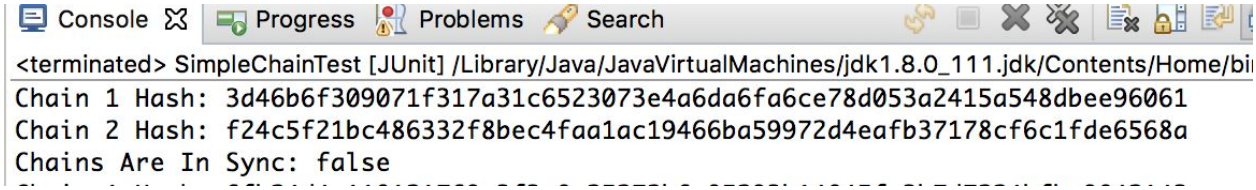

Image 9: Creating a Blockchain Unit Test

The same transaction "`D`" is added to the the second chain. The hashes are compared which result in an assertion of `True`.

```
...
chain2.add(new Transaction("D"));

System.out.println(String.format("Chain 1 Hash: %s", chain1.getHead().getHash()));
System.out.println(String.format("Chain 2 Hash: %s", chain2.getHead().getHash()));
System.out.println(
String.format("Chains Are In Sync: %s",
chain1.getHead().getHash().equals(chain2.getHead().getHash())));
assertTrue(chain1.blockChainHash().equals(chain2.blockChainHash()));
...
```

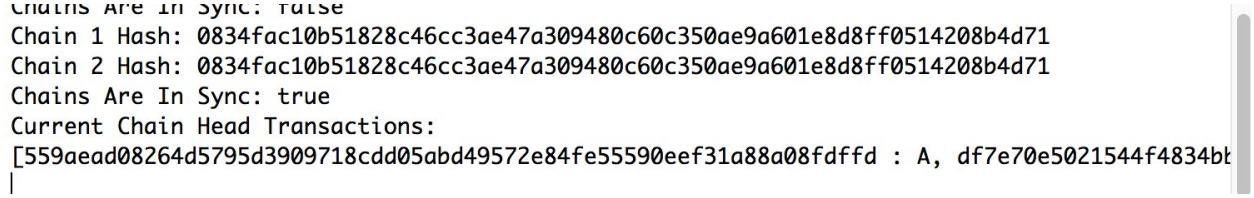

Image 10: Creating a Blockchain Unit Test

The next assertion checks the head block's transaction `merkleTreeRoot` hash with the block's `merkleTreeRoot` hash.

```
...
// Block Merkle root should equal root hash in Merkle Tree computed from block
transactions
Block headBlock = chain1.getHead();
List<Transaction> merkleTree = headBlock.merkleTree();
assertTrue(headBlock.getMerkleRoot().equals(merkleTree.get( merkleTree.size() - 1)));
...
```

Mining a Block JUnit Test

Bitcoin's consensus mechanism utilizes a Proof of Work (POW) algorithm that creates a hash with leading zeros for the blockchain hash, referred to as **mining**. The following sections will discuss the various other types of consensus protocols that can be applied.

As the reader, you will need to use your imagination a bit for this unit test as mining will need to be simulated. In an actual blockchain implementation, transactions are created and communicated across the node via peer-to peer-network. A consensus mechanism will collect transactions, create a block, and submit it back to the peer network for confirmation and

validation. If the hashes and format of blocks don't jibe, nodes will reject the block and not add it to the blockchain.

A `chain` object and a `Miner` object, which has a reference to a chain object, are created here. Thirty transactions are created and passed to the miner object. An assertion should validate that the 30 transactions should result in three blocks in the chain, as we set the transactions per block to 10.

```
...
@Test
public void blockMinerTest() {

 // create 30 transactions, that should result in 3 blocks in the chain.
SimpleBlockchain<Transaction> chain= new SimpleBlockchain<Transaction>();

// Represents a proof of work miner
  Miner miner = new Miner(chain);

 // This represents transactions being created by a network
         for (int i = 0; i < 30; i++) {
            miner.mine(new Transaction(""+i));
          }

     System.out.println("Number of Blocks Mined = "+chain.getChain().size());
     assertTrue( chain.getChain().size() == 3);

  }
...
```

The `mine(Transaction)` method of the `Miner` object will perform Proof of Work for each block, then create a block, and then add it to the blockchain. Miner methods involved in this are shown below.

```
...
public void mine(T tx) {
       transactionPool.add(tx);
       if (transactionPool.size() > SimpleBlockchain.BLOCK_SIZE) {
              createBlockAndApplyToChain();
       }
}

private void createBlockAndApplyToChain() {

    Block block = chain.newBlock();
    // set previous hash with current hash
    block.setPreviousHash(chain.getHead().getHash());
    // set block hashes from POW
    // block
     block.setHash(proofOfWork(block));
     chain.addAndValidateBlock(block);
       // empty pool
      transactionPool = new ArrayList<T>();
 }

       private String proofOfWork(Block block) {
```

```
            String nonceKey = block.getNonce();
            long nonce = 0;
            boolean nonceFound = false;
            String nonceHash = "";

            Gson parser = new Gson();
            String serializedData = parser.toJson(transactionPool);
String message = block.getTimeStamp() + block.getIndex() + block.getMerkleRoot() +
serializedData+ block.getPreviousHash();

      while (!nonceFound) {

                  nonceHash = SHA256.generateHash(message + nonce);
                  nonceFound = nonceHash.substring(0,
nonceKey.length()).equals(nonceKey);
                  nonce++;

            }

            return nonceHash;

      }
```

Introducing Nonce

So now we understand the basic implementation of a blockchain. However, you are probably wondering just how it is *so secure* and magical?

It makes sense from a consistency standpoint. If you are astute, however, you might think "I could just go as far back as I wanted and re-write the chain to be malicious." That is where the **nonce** comes in.

- A **nonce**, ("number only used once"), is a number added to a hashed block that, when rehashed, meets the difficulty level restrictions. The nonce is the number that blockchain miners are solving for.[12]

The nonce value is based on the hash of the string data. The goal is to append a numerical value to the end of the message so that, when combined, you get a certain set of zeros at the beginning of the hash.

Strictly speaking, the matching criteria could be any set of values 0-F at any location(s) in the hash. You are not limited to zeroes at the beginning. The probability of any predetermined arrangement is equally as likely. What changes the complexity is the length of these values in your predetermined set.

This computational complexity gives us an enforceable time buffer. Essentially, to infiltrate the system, one would have to have enough computational power to re-evaluate the nonce

[12] "Nonce." Investopedia. https://www.investopedia.com/terms/n/nonce.asp.

for every block in the chain up to the existing block. Going back to any block before the current one makes this process highly improbable due to the nature of transaction syncing. In a sufficiently large enough set of agents, you will always be behind the rest of the computers in the network that are "good actors."

Nonce Example

So armed with this, we will create a nonce value. Calculating a nonce is what miners are doing in order to achieve a consensus: validate a block with transactions which are requesting to be added to the blockchain. As mentioned, this is referred to as Proof of Work.

First, we will hash our message using our previous hashing method.

```
String hash = helpers.SHA256.generateHash("TEST String");
              System.out.println(hash);
              assertTrue(hash.length() == 64);
```

Next, we will select a nonce pattern. In our case we are going to look for leading zeros in the hash.

```
String nonceKey = "12345";
// E.g. "00000" :
String zeroGoal = new String(new char[nonceKey.length()]).replace("\0", "0");
```

Finally, we will start adding our nonce value to the end of the message so as to try to "mine" a new hash that matches our nonce criterion.

```
while (!isNonceFound) {
  nonceHash = SHA256.generateHash(message + nonce);
  isNonceFound = nonceHash.substring(0, nonceKey.length()).equals(zeroGoal);
  if (!isNonceFound) {
     nonce++;
  }
}
```

In this example, we use the nonce key of `12345`.

```
Message: Keyhole Software
Hash: bbc6cfc49e4051b169805e3343b62f0847e1426f8c87639d9d20f34fe222b920
Nonce: 2178184
Nonce Hash: 00000262769a95de33eac1c3d48f74a274eff0bdfab6ba922738744a51745dc0
Nonce Search Time: 8208 ms
```

Image 10: Finding a Nonce

As we increase the complexity, the amount of time to find an appropriate key goes up exponentially. We tested this and the following values were taken during some testing with various nonce lengths.

Count	Speed Test	Speed	Iterations
1	< ms	-	9
2	2 ms	2	352
3	36 ms	36	6,110
4	60 ms	60	10,264
5	1,380 ms	1,380	256,267
6	76,541 ms	76,541	13,989,388

Table 1: Time To Nonce Values

➢ If you'd like to explore this further, you can test it in the companion source code which has a nonce JUnit test that can be executed: https://github.com/in-the-keyhole/khs-blockchain-java-example/blob/master/src/test/java/nonce/NonceTest.java.

Consensus

The following concepts are true in a blockchain implementation:

- Blockchain data is replicated across many nodes that participate in the blockchain.
- Transaction data can only be appended.
- Transactions in the chain of data are immutable and incorruptible due the previously-mentioned cryptographic magic.
- Each node cannot be implicitly trusted to add only valid data transactions to the blockchain, as the Byzantine general story describes.

Consensus is the method in which the blockchain network can validate and trust transactions before they are added to the chain. Blockchains are deployed in consensus-based protocol which is either public or permissioned.

➢ **Public consensus** means anyone can attempt to add transactions and participate in consensus. Users participate in an anonymous manner. (Think Bitcoin.)

➢ **Permissioned-based** protocols require nodes to be authorized and identified to participate in the consensus or to add transactions to the chain. Users are not anonymous. (Think some enterprise applications.)

Protocols

Four types of consensus algorithms are commonly used to implement a protocol. Each of them would require a detailed multi-page white paper to describe how they work, so below we will just provide a description of what they do.

Practical Byzantine Fault Tolerant (PBFT) Algorithm

This method of consensus is used by Hyperledger, Stellar, and Ripple. Think back to the allegory presented at the beginning of this white paper. There are generals and traitors. How does each general determine the best path of action?

With the PBFT approach, each general maintains an internal state of ongoing information and status. When the general receives a new message, she uses her internal state to run a computation or operation that tells her what to think about the message. Then, after reaching her individual decision about the new message, that general shares that decision with all the other generals in the system. A consensus decision is determined by the total decisions submitted by all generals.[13]

Nodes in the network would be identified as having the permission to confirm adding transactions. All consensus nodes would have a seed value that is appended to the transaction data. All permissioned consensus node's data and seed value will be a part of a calculation to determine if the transaction data is valid.

This approach scales faster, but at the cost of anonymity.

Proof of Work (POW)

A Proof of Work system requires its users to actually perform some type of work to participate. This is utilized by Bitcoin, Ethereum, and other crypto-currencies in the form of "miners" working to solve a block and have that block accepted into the blockchain.[14] The Proof of Work algorithm can be used as a public consensus protocol.

Transactions are blocked together. These blocks are not confirmed while nodes (i.e. miners) attempt to solve the computing problem of the pending transactions in the block. The first node to solve the problem is able to add the transaction block to the chain and is broadcast to the network.

The POW algorithm calculates the Nonce with a given difficulty (i.e number of leading zeros) for the hash of the blockchain transaction. Within each block the linked or previous hash is a part of the new hash.

It's nearly impossible to generate bogus POW nonce hashes. So if the hash is invalid, it will not be added to the chain. Or, if it is added, it will not be reachable or in the chain of blocks connected by hashes.

---

[13] Chris Hammerschmidt, "Consensus in Blockchain Systems. In Short." Medium, 27 Jan 2017. https://medium.com/@chrshmmmr/consensus-in-blockchain-systems-in-short-691fc7d1fefe

[14] Andrew Tar, "Proof of Work, Explained." CoinTelegraph, 17 Jan 2018. https://cointelegraph.com/explained/proof-of-work-explained.

Sidebar: Mining Bits for Fun and Profit

One of the key components of the blockchain is the amount of effort required to commit a block to the chain. With Bitcoin, the system has been tuned so that it takes about 10 minutes of computation time to "mine" one block to commit to the chain.

We go into the technical specifics on how this mining is done elsewhere in the paper, but for now, think of it like a looking for a key in a pile of millions of keys -- you know one of them is the right one, but no way to know which it is without trying each one. This is a time-consuming act. The complexity for finding these keys is set to scale with the rise in power of computers so that each block will take approximately ten minutes globally to find.

So why would someone do this? Why would someone commit their hardware resources and electricity toward finding this needle in a haystack? Simple: they do it for the money.

At the beginning of each set of transactions that go with a block, there is a special type of transaction that rewards the miner for their work. These rewards will halve every 210,000 blocks that are committed, approximately every four years. The original reward was set at 50 coins. That reward is currently at 12.5 bitcoins as of the writing.[15] It will halve again to 6.25 approximately June 2020.[16]

There is an additional type of reward for committing blocks in the form of transaction fees. Miners take a small portion of each transaction that is committed as a part of the block as a fee for committing the transaction. At some point the reward for transaction fees will be higher than the mining reward, but we have not reached that threshold.

After a miner finds the key for their block of data, they send it to the rest of the network for review. If a majority of the network agrees that it should be committed, achieving consensus, then it is. Any other workers that were working on the same block grab the newly-committed block and start mining the next block.

Proof of Stake (POS)

Proof of Stake is another systematic approach to achieve distributed consensus. It was first used by Peercoin in 2012.[17] Also, Ethereum and its creator, Vitalik Buterin, are making a transition from Proof of Work to Proof of Stake.[18][19]

[15] "Mining." Bitcoin Wiki. https://en.bitcoin.it/wiki/Mining#Reward (Accessed 22 Feb 2018).
[16] "Controlled Supply." Bitcoin Wiki. https://en.bitcoin.it/wiki/Controlled_supply (Accessed 20 Feb 2018).
[17] Ameer Rosic, "Proof of Work Versus Proof of Stake." Blockgeeks. https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/.
[18] "Ethereum's Switch to Proof of Stake – Better Than Proof of Work?" UTB; Use The Bitcoin, 30 Jan 2018. https://usethebitcoin.com/ethereums-switch-proof-work-proof-stake/.
[19] Alyssa Hertig, "Ethereum's Big Switch: The New Roadmap to Proof-of-Stake." Coindesk, 5 May 2017. https://www.coindesk.com/ethereums-big-switch-the-new-roadmap-to-proof-of-stake/.

It is still an algorithm with the same goal of consensus, but the process is quite different. Unlike Proof of Work which rewards miners for completing new blocks, the creator of a new block is chosen deterministically depending on "wealth," also known as **stake**. In this type of system, there is no block reward; the "miners" (in this system, called "forgers") take the transaction fees. This can be vastly more cost effective.

This type of consensus is similar to the POW. The difference is that instead of allowing any node to participate in the POW computation, with Proof of Stake, only nodes that have an actual stake or participation in the blockchain would be able to approve new blocks. It can take place as a lottery, with each potential participant possessing the chance to win that new block based on their statistical "stake" in the network.

A benefit of Proof of Stake is that it doesn't have the energy consumption needs of a distributed, consensus-based Proof of Work. Miners need a lot of energy. One Bitcoin transaction required the same amount of electricity as powering 1.57 American households for one day.[20] Proof of Stake allows for a greener and cheaper form of consensus.

Delegated Proof of Stake (DPOS)

A Delegated Proof of Stake system is a highly efficient and flexible consensus model. Current cryptocurrency projects that use this model include BitShares and Steem.

DPOS uses stakeholder approval to resolve issues in a democratic way.[21] Block producers are selected deterministically. Vote strength is determined by the wealth of the voter. If a producer is a bad actor, meaning they're spammer, scammer, or someone who doesn't do a quality job securing the network, ongoing voting will flush out the bad actor.[22]

This approach is similar to the Proof of Stake system. A number of dedicated entity nodes are established and certified to apply their POS signature.

This approach allows fast consensus confirmation, but it introduces a more centralized approach. This isn't necessarily bad if there are enough dedicated entity nodes. But, for the sake of speed, centralization means if these nodes aren't available, Proof of Stake won't happen.

Other Consensus Protocols

The previous protocols are the most well known and have been implemented in the wild, but they are by no means the only available protocols. Innovation continues in the space, and some blockchain platforms even allow custom consensus mechanisms to be applied.

[20] John Lilic, "Bitcoin's Energy Consumption." Blockgeeks. http://blockgeeks.com/bitcoins-energy-consumption/.
[21] "Delegated Proof of Stake Consensus." Bitshares. https://bitshares.org/technology/delegated-proof-of-stake-consensus/.
[22] Leah Stella, "Explain Delegated Proof of Stake Like I'm 5." Hackernoon, 28 Sept 2017. https://hackernoon.com/explain-delegated-proof-of-stake-like-im-5-888b2a74897d.

Smart Contracts

Smart Contracts are an important feature of a decentralized ledger like blockchain. They are also called self-executing contracts, blockchain contracts, or digital contracts. This is because in addition to just storing blocks of data within blockchain transactions, executable code snippets can also be stored.

So, say a smart contract is written as executable code in the blockchain. The individuals involved are anonymous, but the contact is the public ledger. They can be defined to be executed-based upon some kind of criteria, such as a date/time, or a threshold that is reached in other accounts. When a triggering event like an expiration date is hit, the contract will automatically execute itself according to the agreed-upon terms.

One of the leading open source blockchain platforms, Hyperledger- Fabric, has a mechanism for this, referring to this as "chaincode." Currently chaincode is written in the Go language, but they have said they will support other languages such as Java.[23] Ethereum also has a smart contract high level language that runs on a virtual machine they call Ethereum Virtual Machine (EVM).[24] [25] Chaincode can be digitally signed in the same fashion as Bitcoin to ensure only authorized parties can execute.

These scripts are engineered to accept and operate upon the block transaction. Plus, regulators can use the blockchain to understand the activity in the market while the privacy of who participated in the smart contract is respected.[26]

Putting It All Together

So now we have a combination of using Merkle trees, a consensus-based nonce value to validate work, and a peer network of nodes to share. This gives us a transaction log with the following:

- A Distributed, Immutable Data Store
- Shared Transaction History
- Cryptographically Verifiable
- No Centralized Control or Single Point of Failure

With these underlying blockchain architecture fundamentals, there are endless opportunities to introduce it to other business domains other than cryptocurrency.

[23] "Chaincode for Developers." Hyperledger Documentation. http://hyperledger-fabric.readthedocs.io/en/release/chaincode4ade.html.
[24] Dr. Gavin Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger." http://gavwood.com/Paper.pdf.
[25] Manuel Araoz, "The Hitchhiker's Guide to Smart Contracts in Ethereum." Zeppelin Solutions, Last Updated 6 Oct 2017. https://blog.zeppelin.solutions/the-hitchhikers-guide-to-smart-contracts-in-ethereum-848f08001f05.
[26] Ameer Rosic, "Smart Contracts: The Blockchain Technology That Will Replace Lawyers." Blockgeeks. https://blockgeeks.com/guides/smart-contracts/.

Part Three:
Blockchain For Business

In this section, we answer some logical questions about blockchain that can help serve as a guide for those who might be thinking about putting it into their business pipeline.

Real-World Use Case

One way to see the benefits and help understand this technology is to see it applied to a common business problem.

Here at Keyhole, we have a timesheet system where employees enter their time spent on projects worked. This is an activity that occurs in most enterprises. This time is typically entered into a timesheet system data repository of some sort.

This data is supplied to a payroll processing system and into an accounting system. Most enterprises will outsource payroll processing. Depending upon size, they will also outsource accounting functions. This typically requires some kind of ETL integration with the outsourced systems, as shown by the diagram below.

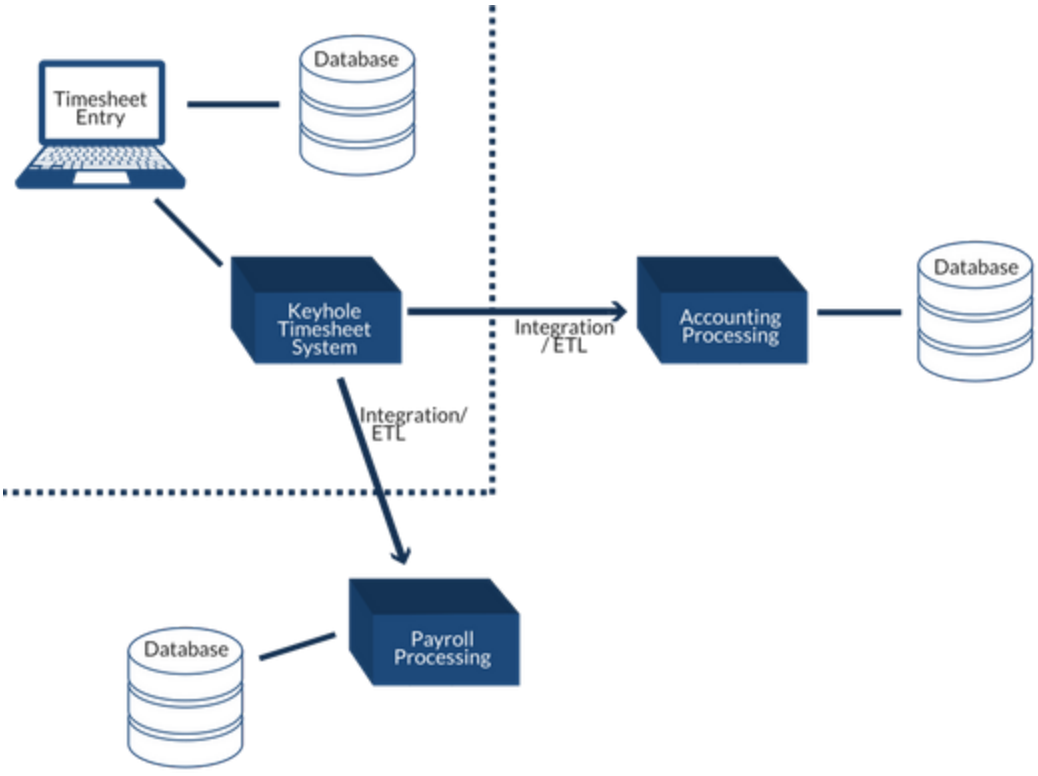

Image 11: Typical Timesheet System Integrating with Payroll Processing & Account Processing

Tel 877-521-776 | keyholesoftware.com

Implementing this integration means employee timesheet data is managed and duplicated across three different systems. IT personnel in all these organizations must build and support the integration mechanisms.

Here is where a blockchain could shine. If all payroll processing companies supported and participated in a timesheet information blockchain, then they could eliminate integration and ETL costs by simply accessing the blockchain for employee timesheet data. Here is a diagram showing the introduction of the blockchain:

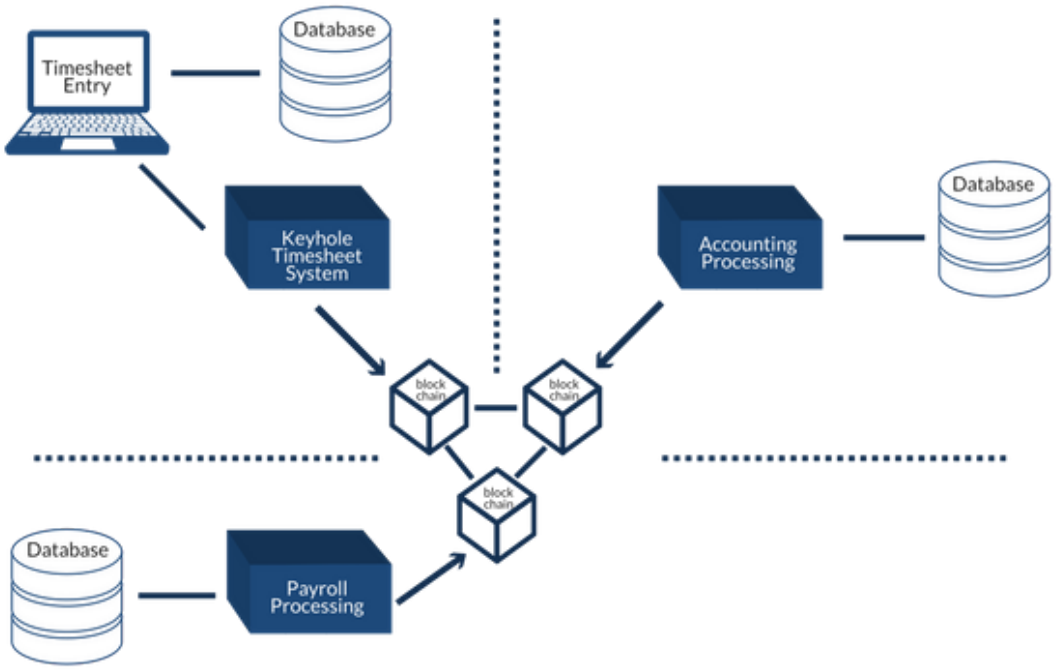

Image 12: Timesheet System Integrating with Payroll Processing & Account Processing via Blockchain

The blockchain example diagram looks subtly different, but is profoundly different in a real-world implementation. A single blockchain will store all timesheet transaction information. The timesheet system data entered into the blockchain is immediately available to the payroll processing system and the accounting system.

The blockchain data is shared and safe; payroll and accounting will not be able to change the timesheet information. They can can indicate payroll has been processed by adding a status indicator to the blockchain for the employee time. The general ledger can likewise access—but not mutate—the other blocks, but can indicate posting to the general ledger.

These transactions are immutable and timestamped. This is opposed to the traditional example in which multiple data stores have to span and record-keep the timesheet data across multiple systems and data stores. In this implementation, there is no centralized timesheet data store and server software that must be maintained and supported. Each organization will have nodes sharing timesheet data.

Caveats

Compute Resources/Attack Surface

What makes blockchain so powerful in currency is its wide distribution and overall incentivization.

To maintain stability of the system, “good actors” must control 51% of all computational resources on the given chain. The problem comes in if you are trying to privately control a public blockchain within your own pool of resources. Most existing botnets can easily out-compute your resources available and transfer ownership of any goods in the chain to themselves.

With currency, this is less likely because there is enough benefit in being a good actor that good actors outweigh bad actors. Organizations in a public blockchain outside of the currency use case will have to consider and attempt to keep “bad actors” from attempting to control computing resources (i.e nodes). Deploying a permissioned block can solve/reduce the problem of someone attacking or taking over the blockchain.

Authority and Ownership

What makes blockchain technologies powerful is the distributed nature. There are no middlemen. This tends to be a problem for businesses that want to monetize solutions, as in many cases, what makes your solution valuable (and monetizable) is that you own it and are thereby selling access to it. Most businesses make money because they are the middleman between the consumer and resource.

Consider a distributed blockchain for a use case within an industry. The industry participants should collectively define and own it, or not own it but use and support it. Therefore, in our opinion, some kind of association or consortium could be established for long-term sustainability. Without widespread adoption and usage, there is not much value in a distributed blockchain, when compared to traditional centralized technologies are easier to deploy, maintain, and control.

Blockchain Use Cases

Blockchain may be a good solution wherever transparent and immutable records are useful. New innovators across the industry spectrum will continue to find new ways to leverage the benefits of blockchain technology. As such, any list of use cases will continually grow as new ideas are brought to the table and the technology continues to mature.

Smart Contracts

In terms of smart contracts, blockchain technology goes a long way to, essentially, automate

trust. It uses permanently-retained and immutable historical data to authenticate that every party of the contract is who they say they are. It is transparent. The contract specifically defines the rules and penalties around an agreement (like in a traditional contract), but goes a step farther to automatically enforce those obligations.

Blockchain smart contracts ensure that agreements are verifiable, permanent, and coded to meet the terms of its parties - without the expense of a middleman. The smart contract executes the instructions consistently and without fail, perhaps enacted based upon a time-based or condition-based trigger. Smart contract protocols potentially strengthen up the legal standing of contracts, which makes them extremely applicable to enterprise needs.

By being able to treat different physical assets as digital assets to be traded, it becomes very easy to confirm ownership and settle claims via the historical log of ownership transfer.

Because of the transparency and immutability features offered by Smart contracts, blockchain could take over functions usually done by intermediaries like banks, escrow services, and even legal services.

Decentralized Autonomous Organizations

A Decentralized Autonomous Organization (DAO), sometimes known as a Decentralized Autonomous Corporation (DAC), is an organization that is run explicitly through rules that are encoded as smart contracts.

Cryptocurrencies are a great example of decentralized autonomous organization. There is no central authority as to the rules of the system or the organizational structure. But yet the system continues to function. These types of systems are still relatively new compared to all things and so whole bodies of research are needed to fully understand long-term impact.

Business Process Improvement

Audit Trails

Remember that at its core, blockchain is just an encrypted, distributed ledger. With that in mind, audit trails are a good candidate for use with blockchain as its persistence mechanism.

Blockchain provides a strong and fully documented audit trail. Using it to track data from its inception to present will likely reduce disputes and discourage fraudulent activity. It could also reduce costs associated with auditing as records can be instantly and independently verified.

Information Security

Banks and other large institutions have a responsibility to secure the account and personal information of their users from hackers and unauthorized access. Blockchain offers a means to

significantly decrease the possibility of errors and ensure the integrity of records. It can create a record of who has accessed information or records automatically and set permissioned controls for viewing that information. This has important implications for health records.

Supply Chain Management

Being a ledger, blockchain provides an ability to track products from origin to completion. This makes it an excellent potential fit for supply chain management due to end-to-end visibility in real-time.

Every time a piece of the supply chain would change hands, that transaction would be documented. Many users may access, inspect, or add to the data. They however cannot change or delete it. It creates a permanent history which could dramatically reduce time delays, human error, and doubt. Additionally, it can track processes against predefined rules to ensure it is kept compliant with any industry regulation.

Various large retailers have begun working with blockchain to determine how it can help their supply chain in regards to fraud and safety, which will be discussed in following sections of this paper.

Internet of Things

At the risk of being too broad, as blockchain technology continues to mature, a likely pairing could occur: blockchain and the Internet of Things. Using IoT devices with data sent as an immutable ledger to the blockchain could have substantial impact on real-time data transparency.

This could have significant benefits in a vast majority of industries. For example, using IoT for for real-time control of commercial property and real estate, like monitoring access, credential management, and legal standings. For commercial vehicles and transportation, the tracking of trip data and journey stops. Other examples include the tracking and categorization of product movement and delivery, ensuring those products are kept within shipping regulations via IoT sensors like temperature; smart home utility metering and control via IoT devices; facility optimization, leveraging energy coming from local solar or wind microgrids in an effort to be more energy efficient; the sale of excess renewable energy autonomously; enhanced security for management of personal data taken in by IoT device, and more.

Order Fulfillment And Payment Of Digital Assets

The existing financial system is very complex and that complexity creates risk. While financial trades happen virtually instantly, actually going through the process to settle those trades remains quite complex. Synchronising internal ledgers can be a significant task, particularly when financial institutions are using different systems to settle accounts. It may take several

days to finally exchange assets which ties up capital and, as such, increases risk. Blockchain technology and cryptocurrency is significantly faster and takes away many of the intermediaries found in traditional banking.

Currently, payment processors and merchant services are quite expensive for e-commerce companies. Blockchain could be a good fit for e-commerce companies, using cryptocurrency and smart contracts to automate fulfillment of orders - especially for the delivery of digital goods.

Cryptocurrency also may have an impact on global transactions. No longer are participants restricted to national currency borders; cryptocurrency can speed up and simplify cross-border payments.

Identity Management

Currently, identity management is often handled by identity cards, badges and documents. No matter how secure you think it is, designs can be counterfeited. How do you verify them? Blockchain can provide a identity infrastructure solution to store identity details. Anyone who needs to verify identity would simply need to query the open blockchain.

With the strong encryption native to block chain and it being a ledger, it could potentially be used as an identification system of record for authenticating ownership of identity and being able to transact that ownership to a new holder in an auditable fashion.

When To Use Blockchain

Let’s think about it in terms of when to use blockchain as a go-to solution. Your idea might be a really great candidate for using blockchain if it meets the following qualifications:

- Provides value to your organization’s consumers, vendors, or industry;
- they are incentivised well enough to be “good actors”; and
- the control of the system can be pushed out to the users of the system and not a central authority.

Examples of potential use cases include:

- Global Property Exchange
- Global Trade Tracking
- National Voting Platforms
- Medical Records
- Fintech (Financial Institution Technology)
- Government Identity And Registration
- Digital Currency

The vast majority of the solutions that are going to be most effective will be at a national,

global, or industry level. Unfortunately, that's not just saying that your company is global.

Shipping is an excellent example use case. A single shipping company like UPS or FedEx would not be a great candidate for blockchain. Alternatively, say a nation wanted to start auditing all shipments from all carriers. Or, all of the major industry carriers formed an open coalition for open shipping data because it behoved all of them to watch trends more broadly. These could both be great use cases.

Another good example would be property. Let's say you wanted to bring all of the states together and unmuddle the property transfer process. This might happen as a coalition, non-profit, or government entity, applying value for doing the computational work associated with each property transfer. This might be a great use case for a custom blockchain.

Doing a single state or county would not, however, because the computational resources from bad actors globally could far outweigh that smaller set of local resources.

Current Blockchain Use Case Spotlights

As an example of real-world, current implementations, we'll provide three spotlights: Walmart, Fortune 500's JD.com, and Alibaba.

International e-commerce conglomerate Alibaba has been working toward using blockchain in supply chain management efforts to combat fake goods. It has formed an Asia-Pacific Blockchain Consortium with its retailers, including AusPost, Fonterra, Blackmores, and PwC. Its goal is to develop a blockchain "Food Trust Framework" for its supply chain, helping to improve integrity and traceability of goods from "paddock to plate."[27] [28]

Alibaba has used RFID devices and unique QR codes on each product. Every time that product changes hands, it is scanned and recorded. That data is uploaded to the blockchain where it is accessible by various parties, like the brand, manufacturers, logistics firms, and customers. Once the customer receives the package, they can scan to see the product's history and verify its origin. In this approach, counterfeiters not only have to contend with copying the product and its packaging, but the entire digital infrastructure behind it on the blockchain.[29]

Walmart and JD.com have both worked on efforts to use blockchain to provide real-time supply chain trackability with a focus on food safety. It was reported in December of 2017 that Walmart and JD.com were among several companies partnering under the Blockchain

[27] Pete Rizzo, "An Asia-Pacific Blockchain Consortium is Forming Around Food Supply Chain." Coindesk, 22 May 2017. https://www.coindesk.com/pwc-teams-up-with-alibaba-for-food-supply-blockchain-test/.
[28] Tas Bindi, "Alibaba and AusPost team up to tackle food fraud with blockchain." ZDNet, 24 March 2017. http://www.zdnet.com/article/alibaba-and-auspost-team-up-to-tackle-food-fraud-with-blockchain/.
[29] Eva Xiao, "Alibaba, JD tackle China's fake goods problem with blockchain." Tech In Asia, 17 Oct 2017. https://www.techinasia.com/alibaba-jd-ecommerce-giants-fight-fake-goods-blockchain.

Food Safety Alliance.[30]

On one pilot effort, Walmart uses blockchain for its supply chain management of pork products in China. Its goal is to accurately and transparently display farm origination details, batch numbers, factory and processing data, storage temperatures, expiration dates, and shipping details. Pilot programs like this will be key to the organization's adoption of blockchain. It's reported that Walmart could save up to an estimated $1tn if it were to move its global supply chain to blockchain.[31] Walmart is reportedly using an IBM Blockchain solution, which is based on Hyperledger.[32] [33]

In addition to food safety efforts, JD.com has approached blockchain for logistics. The JD.com supply chain arm, JD Logistics, joined a transportation blockchain alliance known as Blockchain in Transport Alliance (BiTA).[34] BiTA was established with the aim of developing and promoting blockchain standards for global freight and logistics companies[35] and now includes FedEx, UPS, BNSF Railway, Union Pacific, YRC Worldwide, Penske, GE Transportation, and more.

Why Is Blockchain So Hot Right Now?

At the time of writing this paper, Bitcoin has been hovering between $10,000 and 11,000 USD. Everyone is watching this, eagerly wanting in on a piece of that pie. If you had mined 300 Bitcoin when it first started, you'd be sitting on approximately $3,000,000 worth of assets. There are not many of endeavours in life that can yield that kind or return in such a short amount of time.

The question starts coming in of "What exactly makes it so powerful and how can I monetize it?" Bitcoin's mass appeal may, arguably, be more about autonomy, freedom, a little greed, and cutting out the long-standing banking institutions' control of government-backed currencies. This benefit is debatable, but the potential underlying blockchain technology is much more appealing in the IT landscape.

Blockchain has the potential to change and disrupt how traditional line of businesses share

[30] "Stan Higgins, "Walmart, JD.com Back Blockchain Food Tracking Effort in China." 14 Dec 2017. https://www.coindesk.com/walmart-jd-com-back-blockchain-food-tracking-effort-china/
[31] Michael del Castillo, "Walmart Blockchain Pilot Aims to Make China's Pork Market Safer." Coindesk, 19 Oct 2016. https://www.coindesk.com/walmart-blockchain-pilot-china-pork-market/.
[32] Olga Kharif, "Wal-Mart Tackles Food Safety With Trial of Blockchain." Bloomberg Technology, 18 Nov 2016. https://www.bloomberg.com/news/articles/2016-11-18/wal-mart-tackles-food-safety-with-test-of-blockchain-technology.
[33] Roger Aitken, "IBM & Walmart Launching Blockchain Food Safety Alliance In China With Fortune 500's JD.com." Forbes, 14 Dec 2017. https://www.forbes.com/sites/rogeraitken/2017/12/14/ibm-walmart-launching-blockchain-food-safety-alliance-in-china-with-fortune-500s-jd-com/.
[34] Wolfie Zhao, "E-Commerce Giant JD.com Launches Blockchain Startup Accelerator." Coindesk, 27 Feb 2018. https://www.coindesk.com/e-commerce-giant-jd-com-launches-blockchain-startup-accelerator/.
[35] Daniel Palmer, "Chinese Retail Giant JD.com Joins Blockchain in Transport Alliance." Coindesk, 2 Feb 2018. https://www.coindesk.com/chinese-retail-giant-jd-com-joins-blockchain-in-transport-alliance/.

data. This is being validated by big companies like IBM and Microsoft investing in, and providing leadership in, blockchain technologies. If these organizations can disrupt or breathe new life in the typical line of enterprise automation, they can sell more software and services.

Keyhole Recommendation

The buzz is extraordinarily high. While blockchain does have its benefits, it makes your annual budget ripe for picking if you aren't educating yourself on the underlying technology. Companies will tell you everything you want to hear with how blockchain will solve every problem you've ever had and then some. After all, while blockchain is secure, it is also incredibly computationally-heavy.

Our recommendation is to not get swept away in the endorphin rush. There are some legitimate use cases where blockchain makes a ton of sense. Most of them are not for profit, but for cost-saving efficiencies.

White Paper Conclusion

Traditionally, the awareness of new and disruptive technologies usually stay within IT circles. Particularly when that exciting technology isn't a tangible device that users can touch and use, it stays in tech-centric companies. Blockchain, however, has transcended IT circles, making mass media news mostly due to Bitcoin.

As we have shown in this white paper, blockchain is an innovative technology that can provide efficiencies in the way data is shared in an autonomous manner. If we had to pick a single buzzword for this impact of this technology, "autonomy" would be it. Given the benefits, it's unlikely for blockchain to be a thing of the past. The question is, how will the technology be leveraged as the technology matures?

The best way to begin to understand a technology is to actually do something with it; apply blockchain technology to some use case or an example application. We suggest you play with the source code, which in the distributed blockchain world, is the law of blockchain. Here at Keyhole Software, we initially did this by studying the BitcoinJ open source implementation. Even though it is a Bitcoin implementation, it is also a blockchain implementation.

Keep your eyes and ears open for how blockchain is being applied in the real world. Take note of what platforms others use, then try applying. It's important to understand intimately how blockchain works before you make a decision to use a blockchain platform. We recommend you use a platform, rather than building your own blockchain framework platform. Once you understand how blockchain works, then ideas for applying it should start flowing.

From a business perspective, we strongly encourage due diligence in having your technical teams understand the inner workings, algorithms, and protocols of blockchain prior to moving forward just because of the hype.

Appendix 1:
Blockchain Frameworks and Platforms

Organizations have a plethora of choices available in implementing and deploying a blockchain solution. They could build their own based upon the Satoshi whitepaper, or use one the many open source frameworks that can be customized to various needs.

Blockchain platforms provide a custom, opinion-based implementation that are ready to use for cryptocurrency and other smart-contract blockchain implementations.

We will discuss some arguably-popular open source blockchain frameworks. There are a variety of open source projects out there. We've chosen to highlight these because of potential applicability to our enterprise clients, particularly in regards to smart contracts. This is certainly not an exhaustive list, as we'll only be briefly summarizing six projects: Ethereum, Hyperledger Fabric, QTUM, Cordano, NEO, and Multi Chain.

Ethereum

Ethereum, arguably the most popular, is a decentralized platform that runs smart contracts on a custom built blockchain "that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference."[36] It is a generic platform for all kinds of transactions and applications, including the creation of crypto-currencies and democratic autonomous organizations (DAOs). As mentioned previously in this paper, Ethereum is working on a move from a Proof of Work protocol to a Proof of Stake protocol.

Smart contracts are treated as autonomous scripts or stateful decentralized applications that are stored in the Ethereum blockchain for later execution. All smart contracts are executed on the Ethereum Virtual Machine (EVM), a distributed global computer, and simultaneously executed by every node in the network. An Ethereum transaction contract code can trigger actions (data reads and writes, for example) which have a cost measured in **gas.** Each gas unit consumed by a transaction is paid for in Ethereum currency, Ether. This price is deducted from the Ethereum account sending the transaction.[37] Having a built-in cryptocurrency can make it a good match for applications that need it.

Ethereum allows anyone (via pseudonym) to build and use decentralized applications that run on blockchain technology. It focuses on total transparency and has a matured eco-system. Everyone can download the blockchain ledger and view all transactions. Like Bitcoin, no one controls or owns Ethereum.

- Website - https://ethereum.org/

[36] Ethereum. https://ethereum.org/.
[37] Vivek M George, "White Paper: A Next-Generation Smart Contract and Decentralized Application Platform." Ethereum Github. https://github.com/ethereum/wiki/wiki/White-Paper.

- White Paper - https://github.com/ethereum/wiki/wiki/White-Paper
- Github - https://github.com/ethereum

Hyperledger Fabric

HyperLedger Fabric is an open source, permissioned blockchain foundation protocol for enterprise use. It is a distributed ledger solution, underpinned by a modular architecture designed to support pluggable implementations of different components, including configurable consensus and membership services. [38]

One note: out-of-the-box, Hyperledger Fabric allows you to choose the consensus mechanism modularly. Practical Byzantine Fault Tolerance (PBFT) is available, where a new block is added if more than 2/3 of all validating peers submit the same response. It does not have its own cryptocurrency, though it is possible to develop a native currency or digital token with chaincode.[39]

Fabric also supports and executes Smart Contracts (called "chaincode" in Fabric). Transaction processing is performed by predefined users. Its modular architecture allows Fabric to be customized to a multitude of applications. It essentially provides an ala carte option with lots of flexibility in regards to options you want to use versus the ones you do not.

It was created by Hyperledger Consortium under the backing of the Linux foundation and led by a number of industry players including IBM and Digital Asset Holdings. IBM and others have also used it to implement enterprise blockchains and Blockchain-as-a-Service offerings.[40] It is data and application agnostic, with industry leaders representing finance, banking, Internet of Things, supply chain, manufacturing, and technology.[41]

- Website - https://www.hyperledger.org/projects/fabric
- White Paper - https://docs.google.com/document/d/1Z4M_qwILLRehPbVRUsJ3OF8Iir-gqS-ZYe7W-LE9gnE/
- Github - https://www.hyperledger.org/projects/fabric

HyperLedger Cello: This project is a Hyperledger project in incubation. Hyperledger Cello is a blockchain provision and operation system, which helps manage blockchain networks in an efficient way[42] Other Hyperledger tools exist, including Composer, Explorer, and Quilt.

IBM Blockchain: IBM Blockchain, built on top of the HyperLedger project, is available as part

[38] Hyperledger Fabric Github Repository. https://github.com/hyperledger/fabric.
[39] Martin Valenta, Philipp Sandner, "Comparison of Ethereum, Hyperledger Fabric and Corda." Medium, 25 Jun 2017. https://medium.com/@philippsandner/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6.
[40] Ron Miller, "IBM Unveils Blockchain As A Service Based on Hyperledger Fabric." TechCrunch, 19 March 2017. https://techcrunch.com/2017/03/19/ibm-unveils-blockchain-as-a-service-based-on-open-source-hyperledger-fabric-technology/.
[41] Hyperledger White Paper. https://docs.google.com/document/d/1Z4M_qwILLRehPbVRUsJ3OF8Iir-gqS-ZYe7W-LE9gnE/.
[42] Hyperledger Cello Github Repository. https://github.com/hyperledger/cello.

of its Bluemix service catalog. It offers additional security and infrastructure facilities for enterprises.

QTUM

QTUM, pronounced Quantum, is a hybrid blockchain application platform. It provides a Turing-complete blockchain stack with the ability to execute smart contracts and decentralized applications.

Qtum's core technology combines a fork of Bitcoin Core, an Account Abstraction Layer allowing for multiple Virtual Machines including the Ethereum Virtual Machine (EVM), and Proof-of-Stake consensus. It's a huge benefit that it is backwards compatible, both with Bitcoin gateways and existing Ethereum contracts.

An interesting part of Qtum is its focus on mobile. Qtum offers a way for blockchain-based applications to be accessed by mobile devices, yet still be decentralized and secure. In particular, its Simple Payment Verification protocol allows smart contracts to be executed in lite wallets via mobile applications. QTUM is seen as a promising contender in the competition with Ethereum.

- Website - https://qtum.org/en/
- White Paper - https://qtum.org/en/white-papers
- Github - https://github.com/qtumproject

NEO

NEO, formerly known as Antshares, is often known as the "Ethereum of China." The two are very similar, and while both are open source, NEO has the full backing of China's government compared to Ethereum's support by a democratic foundation of developers.[43] Neo was developed by Shanghai-based blockchain R&D company OnChain.[44]

According to its website, NEO is a non-profit community-based blockchain project that utilizes blockchain technology and digital identity to digitize assets, to automate the management of digital assets using smart contracts, and to realize a "smart economy" with a distributed network.

Instead of Proof of Work, NEO uses a dBFT (decentralized Byzantine Fault Tolerant) consensus mechanism which is more energy-efficient.[37]

- Website - https://neo.org/
- White Paper - http://docs.neo.org/en-us/index.html
- Github - https://github.com/neo-project/neo

[43] Joe Liebkind, "4 Blockchain Contenders in Competition with Ethereum." Investopedia, 14 Dec 2017. https://www.investopedia.com/news/4-blockchain-contenders-competition-ethereum/.
[44] Ameer Rosic, "What is Neo Blockchain? Beginners Guide." Blockgeeks. https://blockgeeks.com/guides/neo-blockchain/.

Cardano

Cardano is an open source smart contract platform, similar to Ethereum, with a focus on security through a layered architecture. The platform uniquely uses Haskell, a programming language with a high degree of fault tolerance. It was released by IOHK (Input Output HongKong), whose CEO Charles Hoskinson was the former CEO of Ethereum.[45] The national research and education network of Greece plans the first Cardano pilot project by verifying student diplomas via Cardano's blockchain.

Cardano uses a proof of stake algorithm to achieve decentralized consensus in its network.[46] In this protocol, slot leaders generate new blocks in the blockchain and verify the transactions. Anyone holding a Cardano ADA coin can become a slot leader.[47]

- Website - https://www.cardanohub.org
- Documents - https://whycardano.com/, https://www.cardanohub.org/en/academic-papers/
- Github - https://github.com/input-output-hk/cardano-sl/

MultiChain

MultiChain is a open source platform for the creation of private blockchains (permissioned blockchains) either within or between organizations. Its main focus is the financial sector and as such, privacy and control is a significant focus. Its core aim is threefold: (a) to ensure that the blockchain's activity is only visible to chosen participants, (b) to introduce controls over which transactions are permitted, and (c) to enable mining to take place securely without proof of work and its associated costs. It also provides the ability to work with multiple blockchains at one time, even allowing the creation of connections between the activity in different chains.[48]

It is backwards compatible with Bitcoin and Bitcoin Core, including the peer-to-peer protocol, transaction/block formats and Bitcoin Core APIs/runtime parameters.[49] It provides a simple API and a command-line interface to preserve and set up the chain.

- Website - https://www.multichain.com
- White Paper - https://www.multichain.com/download/MultiChain-White-Paper.pdf
- Github - https://github.com/MultiChain

[45] Ramsteen, "Cardano (ADA) - 3rd generation of Cryptocurrency." SteemKR. https://steemkr.com/cryptocurrency/@ramsteem/cardano-ada-3rd-generation-of-cryptocurrency.
[46] Shawn Gordon, "Cardano Lists ADA Futures on BitMEX." Bitcoin Magazine, 8 Jan 2018. https://bitcoinmagazine.com/articles/cardano-lists-ada-futures-bitmex/
[47] Steven Buchko, "What is Cardano? | Beginner's Guide." Coincentral, 5 Dec 2017. https://coincentral.com/cardano-beginner-guide/.
[48] Dr Gideon Greenspan, "MultiChain Private Blockchain - White Paper." https://www.multichain.com/download/MultiChain-White-Paper.pdf.
[49] Multichain Github Repository. https://github.com/MultiChain.

Context

As mentioned, this is not an exhaustive list. Other open source blockchain platform alternatives include Corda, Eos, HydraChain, NXT, OpenChain, Quorium, and so many more.

As with any emerging technology that shows promise, many players attempt to rush into the market. Cryptocurrencies are the first commonly-known application of blockchain and they are still experiencing growing pains. But as the blockchain begins to make significant inroads to other business domains, we will see an emergence of frameworks, patterns, and best practices.

About Keyhole Software

Keyhole Software is a software development and consulting firm with a team that loves technology. Our expert employee consultants excel as "change agents," helping our clients to be successful with software technologies and approaches that bring competitive advantage.

We consult nationally with clients across the United States. The Keyhole Software corporate office is located in Leawood, Kansas, just south of Kansas City. Additional teams are located in St. Louis, Chicago, Lincoln, and Omaha. We frequently assist our nationwide clients with custom application design, development, and modernization initiatives with Java, JavaScript, and .NET technologies. See recent Keyhole Software projects here.

Keyhole was founded on the principle of delivering quality solutions through a talented technical team. As such, knowledge transfer is important to us. To our clients, we offer various techniques to provide the most value: one-on-one or group mentoring, lab/lecture educational courses, and access to our knowledge transfer engine GrokOla.

Related Services Snapshot

- Technology Consulting – Expert Keyhole Consultants work with organizations to analyze the current state of applications, recommend technical directions, and develop strategic plans for modernization. We can help your organization to assess, prioritize, and implement blockchain technology.

- Application Development – Individual members or entire teams of Keyhole Software Consultants perform expert development services: application analysis, design,

development, testing, and enhancement of custom applications.

- Education – Instruction of a wide range of custom courses and technical mentoring programs for knowledge transfer to groups or individuals.

Contact Keyhole Software

Company Website: https://keyholesoftware.com
Products Website: https://keyholelabs.com

Phone: 877-521-7769
Email: asktheteam@keyholesoftware.com
Headquarters: 8900 State Line Road Suite 455 Leawood, KS 66206

White Paper References

In addition to the footnotes, please see the following excellent resources for further information and learning:

- **Keyhole Software - Java Blockchain Companion Project**
  https://github.com/in-the-keyhole/khs-blockchain-java-example
- **Keyhole Software - C# Blockchain Companion Project**
  https://github.com/in-the-keyhole/khs-blockchain-csharp-example
- Chris Hammerschmidt, "Consensus in Blockchain Systems. In Short." Medium, 27 Jan 2017. https://medium.com/@chrshmmmr/consensus-in-blockchain-systems-in-short-691fc7d1fefe.
- Christian Cachin, "Resilient Consensus Protocols for Blockchains." IBM Research Blog, 18 October 2017. https://www.ibm.com/blogs/research/2017/10/resilient-consensus-protocols-blockchains/.
- "Confirmation." Bitcoin Wiki. https://en.bitcoin.it/wiki/Confirmation.
- Joe Liebkind, "4 Blockchain Contenders in Competition with Ethereum." Investopedia, 14 Dec 2017. https://www.investopedia.com/news/4-blockchain-contenders-competition-ethereum/
- Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." https://bitcoin.org/bitcoin.pdf.
- Block Explorer. https://blockexplorer.com.
- "Merkle Trees." Bitcoin Wiki. https://en.bitcoin.it/wiki/Protocol_documentation#Merkle_Trees

Footnotes

[1] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem." ACM Transactions on Programming Languages and Systems Journal, July 5, 1982. https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/.

[2] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." https://bitcoin.org/bitcoin.pdf.

[3] Molly Jane Zuckerman, "Culprits Apprehended In Alleged Icelandic Bitcoin Miner Theft." Cointelegraph, 25 Feb 2018. https://cointelegraph.com/news/culprits-apprehended-in-alleged-icelandic-bitcoin-miner-theft.

[4] "Hash Rate." Bitcoin. http://bitcoin.sipa.be/.

[5] ConsenSys, "Blockchain Underpinnings: Hashing." Medium, 13 Jan 20. https://medium.com/@ConsenSys/blockchain-underpinnings-hashing-7f4746cbd66b.

[6] "SHA-2." Wikipedia. https://en.wikipedia.org/wiki/SHA-2 (Accessed 21 Feb. 2018).

[7] "SHA256 Hash." Blockchain Demo. https://anders.com/blockchain/hash.html.

[8] Marc Clifton, "Understanding Merkle Trees - Why use them, who uses them, and how to use them." CodeProject, 13 Mar 2017 https://www.codeproject.com/Articles/1176140/Understanding-Merkle-Trees-Why-use-them-who-uses-t.

[9] Galvatron. Cryptography Stack Exchange, 29 May 2017. https://crypto.stackexchange.com/questions/47809/why-havent-any-sha-256-collisions-been-found-yet.

[10] "Block." Bitcoin Wiki. https://en.bitcoin.it/wiki/Block (accessed 20 Feb. 2018).

[11] "Block Size Limit Controversy." Bitcoin Wiki. https://en.bitcoin.it/wiki/Block_size_limit_controversy (accessed 26 Feb 2018).

[12] "Nonce." Investopedia. https://www.investopedia.com/terms/n/nonce.asp.

[13] Chris Hammerschmidt, "Consensus in Blockchain Systems. In Short." Medium, 27 Jan. 2017. https://medium.com/@chrshmmmr/consensus-in-blockchain-systems-in-short-691fc7d1fefe

[14] Andrew Tar, "Proof of Work, Explained." CoinTelegraph, 17 Jan 2018. https://cointelegraph.com/explained/proof-of-work-explained.

[15] "Mining." Bitcoin Wiki. https://en.bitcoin.it/wiki/Mining#Reward (Accessed 22 Feb. 2018).
[16] "Controlled Supply." Bitcoin Wiki. https://en.bitcoin.it/wiki/Controlled_supply (Accessed 20 Feb. 2018).
[17] Ameer Rosic, "Proof of Work Versus Proof of Stake." Blockgeeks. https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/.
[18] "Ethereum's Switch to Proof of Stake – Better Than Proof of Work?" UTB; Use The Bitcoin, 30 Jan. 2018. https://usethebitcoin.com/ethereums-switch-proof-work-proof-stake/.
[19] Alyssa Hertig, "Ethereum's Big Switch: The New Roadmap to Proof-of-Stake." Coindesk, 5 May 2017. https://www.coindesk.com/ethereums-big-switch-the-new-roadmap-to-proof-of-stake/.
[20] John Lilic, "Bitcoin's Energy Consumption." Blockgeeks. http://blockgeeks.com/bitcoins-energy-consumption/.
[21] "Delegated Proof of Stake Consensus." Bitshares. https://bitshares.org/technology/delegated-proof-of-stake-consensus/.
[22] Leah Stella, "Explain Delegated Proof of Stake Like I'm 5." Hackernoon, 28 Sept 2017. https://hackernoon.com/explain-delegated-proof-of-stake-like-im-5-888b2a74897d.
[23] "Chaincode for Developers." Hyperledger Documentation. http://hyperledger-fabric.readthedocs.io/en/release/chaincode4ade.html.
[24] Dr. Gavin Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger." http://gavwood.com/Paper.pdf.
[25] Manuel Araoz, "The Hitchhiker's Guide to Smart Contracts in Ethereum." Zeppelin Solutions, Last Updated 6 Oct 2017. ttps://blog.zeppelin.solutions/the-hitchhikers-guide-to-smart-contracts-in-ethereum-848f08001f05.
[26]Ameer Rosic, "Smart Contracts: The Blockchain Technology That Will Replace Lawyers." Blockgeeks. https://blockgeeks.com/guides/smart-contracts/.
[27] Pete Rizzo, "An Asia-Pacific Blockchain Consortium is Forming Around Food Supply Chain." Coindesk, 22 May 2017. https://www.coindesk.com/pwc-teams-up-with-alibaba-for-food-supply-blockchain-test/.
[28] Tas Bindi, "Alibaba and AusPost team up to tackle food fraud with blockchain." ZDNet, 24 March 2017. http://www.zdnet.com/article/alibaba-and-auspost-team-up-to-tackle-food-fraud-with-blockchain/.
[29] Eva Xiao, "Alibaba, JD tackle China's fake goods problem with blockchain." Tech In Asia, 17 Oct 2017. https://www.techinasia.com/alibaba-jd-ecommerce-giants-fight-fake-goods-blockchain.
[30] Stan Higgins, "Walmart, JD.com Back Blockchain Food Tracking Effort in China." 14 Dec 2017. https://www.coindesk.com/walmart-jd-com-back-blockchain-food-tracking-effort-china/
[31] Michael del Castillo, "Walmart Blockchain Pilot Aims to Make China's Pork Market Safer." Coindesk, 19 Oct 2016. https://www.coindesk.com/walmart-blockchain-pilot-china-pork-market/.
[32] Olga Kharif, "Wal-Mart Tackles Food Safety With Trial of Blockchain." Bloomberg Technology, 18 Nov 2016. https://www.bloomberg.com/news/articles/2016-11-18/wal-mart-tackles-food-safety-with-test-of-blockchain-technology.
[33] Roger Aitken, "IBM & Walmart Launching Blockchain Food Safety Alliance In China With Fortune 500's JD.com." Forbes, 14 Dec 2017. https://www.forbes.com/sites/rogeraitken/2017/12/14/ibm-walmart-launching-blockchain-food-safety-alliance-in-china-with-fortune-500s-jd-com/.
[34] Wolfie Zhao, "E-Commerce Giant JD.com Launches Blockchain Startup Accelerator." Coindesk, 27 Feb 2018. https://www.coindesk.com/e-commerce-giant-jd-com-launches-blockchain-startup-accelerator/.
[35] Daniel Palmer, "Chinese Retail Giant JD.com Joins Blockchain in Transport Alliance." Coindesk, 2 Feb 2018. https://www.coindesk.com/chinese-retail-giant-jd-com-joins-blockchain-in-transport-alliance/.
[36] Ethereum. https://ethereum.org/.
[37] Vivek M George, "White Paper: A Next-Generation Smart Contract and Decentralized Application Platform." Ethereum Github. https://github.com/ethereum/wiki/wiki/White-Paper.
[38] Hyperledger Fabric Github Repository. https://github.com/hyperledger/fabric.
[39] Martin Valenta, Philipp Sandner, "Comparison of Ethereum, Hyperledger Fabric and Corda." Medium, 25 Jun 2017. https://medium.com/@philippsandner/comparison-of-ethereum-hyperledger-fabric-and-corda-21c1bb9442f6.
[40] Ron Miller, "IBM Unveils Blockchain As A Service Based on Hyperledger Fabric." TechCrunch, 19 March 2017.

https://techcrunch.com/2017/03/19/ibm-unveils-blockchain-as-a-service-based-on-open-source-hyperledger-fabric-technology/.

[41] Hyperledger White Paper. https://docs.google.com/document/d/1Z4M_qwILLRehPbVRUsJ3OF8Iir-gqS-ZYe7W-LE9gnE/

[42] Hyperledger Cello Github Repository. https://github.com/hyperledger/cello.

[43] Joe Liebkind, "4 Blockchain Contenders in Competition with Ethereum." Investopedia, 14 Dec 2017. https://www.investopedia.com/news/4-blockchain-contenders-competition-ethereum/.

[44] Ameer Rosic, "What is Neo Blockchain? Beginners Guide." Blockgeeks. https://blockgeeks.com/guides/neo-blockchain/.

[45] Ramsteen, "Cardano (ADA) - 3rd generation of Cryptocurrency." SteemKR. https://steemkr.com/cryptocurrency/@ramsteem/cardano-ada-3rd-generation-of-cryptocurrency.

[46] Shawn Gordon, "Cardano Lists ADA Futures on BitMEX." Bitcoin Magazine, 8 Jan 2018. https://bitcoinmagazine.com/articles/cardano-lists-ada-futures-bitmex/

[47] Steven Buchko, "What is Cardano? | Beginner's Guide." Coincentral, 5 Dec 2017. https://coincentral.com/cardano-beginner-guide/.

[48] Dr Gideon Greenspan, "MultiChain Private Blockchain - White Paper." https://www.multichain.com/download/MultiChain-White-Paper.pdf.

[49] Multichain Github Repository. https://github.com/MultiChain.