# Blockchain 101 Guide

for

# Blockchains Demonstration

Project 9

Version 1.0 approved

Prepared by Vanessa Soares, David Mistretta, Daniel Champagne, and Ian Eichorn

University of Massachusetts Dartmouth

2 December 2017

# Table of Contents

# Chapter 1: What is a Blockchain?

A blockchain is a shared distributed ledger that allows for the recording of transactions and asset tracking on a network. An asset can be anything from  a house, a car, land, or even patents, copyrights, or branding. Anything of value can be tracked and traded on a blockchain network. This reduces the risk and cuts costs for anyone involved in the tracking process. In terms of Bitcoin and the blockchain, think of the blockchain as the operating system and Bitcoin as the application that is running on the operating system. Bitcoin is only the first use case for the Blockchain [5].

Blockchains are a new and developing technology that may further protect against data breaches. They are defined as incorruptible digital ledgers of economic transactions that have the ability to be programmed to record everything of value. Blockchains have created the backbone of a new internet system by allowing the distribution of digital information but not allowing that digital information to be copied. [1].

They were originally devised for Bitcoin, the digital currency, but the technology community is finding other potential uses for them. Blockchains allow digital information to be distributed but not copied. Information shared on a blockchain exists as a shared and continually updated database. Information is public, easily verifiable, and is not stored in one single location. Therefore, it makes it extremely difficult for hackers to corrupt the data since no centralized version of this information exists. Since blockchains store blocks of information that are identical across the network, the blockchain can't be controlled by one single entity and it has no single point of failure. Therefore, data breaches may be prevented by potentially storing user information or any other confidential information on a blockchain. A blockchain can be viewed as a spreadsheet that is duplicated across a network of computers that's designed to regularly update this spreadsheet [1].

A verified piece of data forms a block that has to be added to the chain in order for a blockchain to be created. Blockchain users have to use their respective keys and computing systems to run algorithms that solve complex math problems. Once a problem is solved, the block is added to the chain and the data it contains exists on the network forever (can't be altered or removed) [11]. In order to update data, the owner has to add a new block on top of the previous block which creates a specific chain of code. The entire chain across the network changes accordingly. Every single alteration is tracked and no data is lost/deleted because users can always look at previous version of the block to identify what's different in the latest version [11].

A blockchain is a linked-list-like structure that acts as an immutable ledger. It is distributed among all members of a decentralized network. This ledger tracks and records any arbitrary piece of data in such a way that it cannot be changed. The distributed nature of the blockchain implies that multiple copies exist. The distributed nature is also what lends itself to being immutable. If there are multiple versions of the blockchain on the network, the official ledger is the version that is held by the majority of the peers on the network [18].

Hashins and Merkle Trees

A hash function is a tool that converts data of variable length to a unique fixed size digest. Hash functions exhibit pre-image resistance, second pre-image resistance, and collision resistance. Blockchain implementations typically use the SHA-256 hash [18].

A merkle tree is also known as a hash tree and it is a balanced binary search tree. Any parent node is the hash of the concatenation of its children. The tree root is the tallest node and is known as the Merkle Root. A condensed version of all the data stored in the block is the merkle tree. Hashing and merkle trees are the foundation of the immutable ledger [18].

5 Key Responsibilities of the Network Node

- Discover fellow peers and compute the genesis block
- Wait for data from a user node to signal the start of the race to compute/mine a block
- Repeatedly randomize or increment the nonce until the block header's hash meets the blockchain's criteria (e.g. a certain number of leading 0's)
- Once a solution is found, share it with the other peers on the network.
- Upon solving their own block, or receiving a block solved by a peer, each node must verify that the block fits onto the blockchain. This is important since nodes do not trust their peers on the network.

The above network structure gives Blockchain its distributed and decentralized properties [18].

# Chapter 2: Basic Concepts and Benefits of a Blockchain

## I. Decentralization

Decentralized technology lets us store data on a network that can be accessed over the Internet [4]. Through the decentralized technology, the owner of the data has direct control through their private key which is directly linked to the data [4].

Anything that happens on the blockchain network is a function of the network as a whole [1]. Therefore, the blockchain is managed by all nodes on its network instead of a single entity. Decentralization is defined as the networking operating on a user-to-user/peer-to-peer basis [1].

Blockchains are defined as public ledgers of all transactions and communications that have ever been executed on the network [3]. They are comprised of a network of nodes that get a copy of the blockchain that is automatically downloaded once they join the blockchain network [1]. A block is the "current" part of the blockchain that records some or all of the recent communications [3]. Once the communication or transaction is completed, the block goes into the blockchain as a permanent database [3]. Once a block is completed, a new block is generated [3]. These blocks are linked together in the form of a chain in linear chronological order with every block containing a hash of its previous block [3].

Since blockchains store blocks of data that are identical across the network, blockchains cannot be controlled by a single entity and they have no single point of failure [1]. The blockchain network operates on a consensus mechanism that automatically checks in with itself every 10 minutes and reconciles every transaction that happens in these 10 minute intervals [1].

Reasons for Decentralization [4].

**Empowered Users:** allows users to keep control of their information and transactions.
**Fault Tolerance:** less likely to fail accidentally since they rely on many separate components that are not likely to fail.
**Durability and Attack Resistance:** Since the blockchain doesn't have a central point of control and can better survive malicious attacks, decentralized systems are more "expensive" to attack and destroy or manipulate.
**Free from Scams:** it is much more difficult for users to cause harm to other users by scamming.
**Removing Third-Party Risks:** enables users to make an exchange without a third party as an intermediary which eliminates risks.
**Higher Transaction Rate:** transactions can reduce times to minutes and can be processed anytime compared to how transactions are done through banks now.
**Lower Transaction Costs:** done through eliminating 3rd party intermediaries and overhead costs for exchanging assets.
**Transparency:** changes to public blockchains are viewable by all parties and all transactions are immutable which means they cannot be altered or deleted.
**Authenticity:** the blockchain is complete, consistent, timely, accurate, and widely available.

### II. Consensus

For a transaction to be valid, all participants must agree on the validity of the Transaction [5]

### III. Provenance

Participants know where the asset came from and how its ownership changed over time [5]

### IV. Immutability

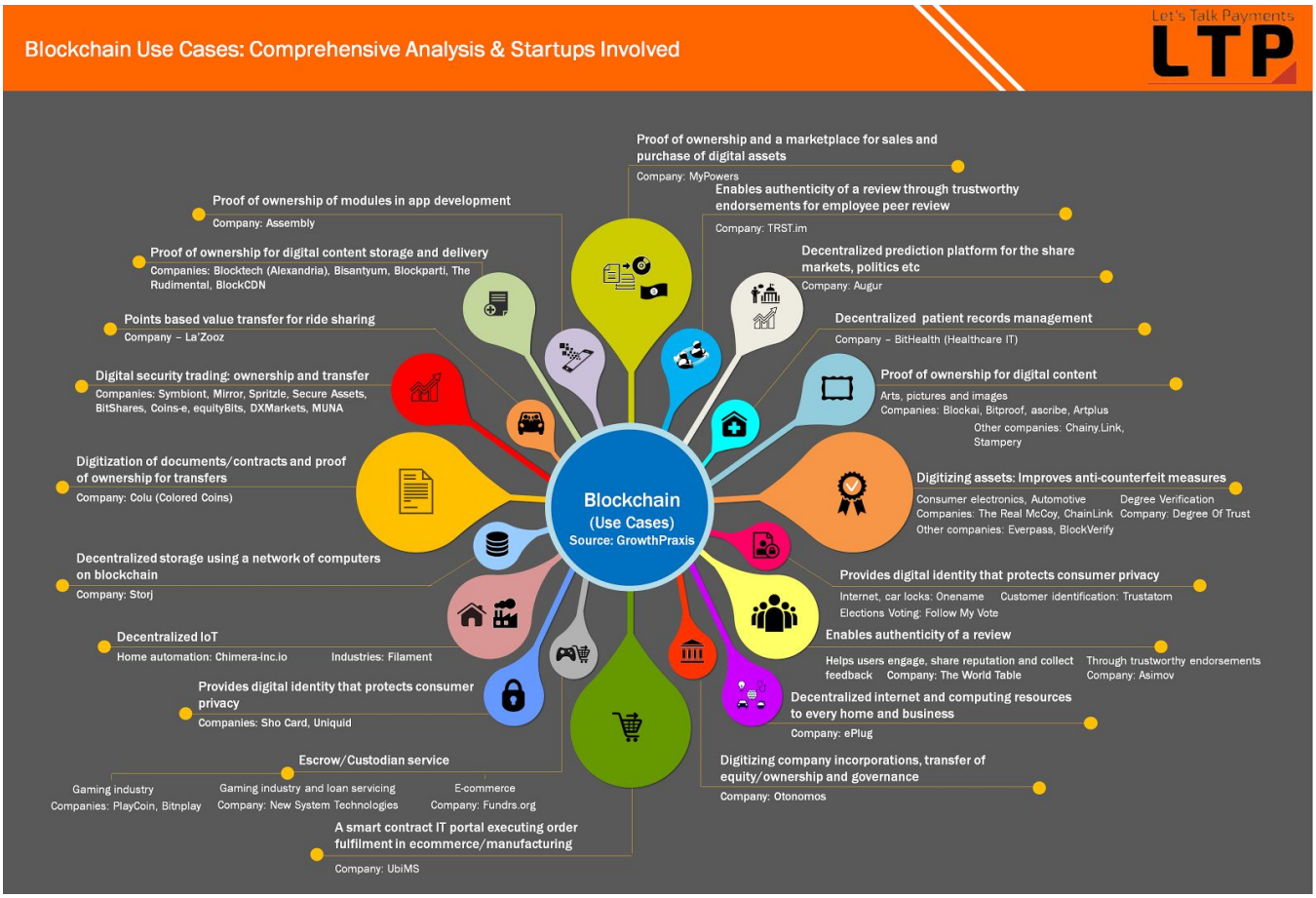
No participant can tamper with a transaction once it has been recorded to the blockchain. If a transaction was made by mistake and a change needs to be made, a new transaction must be used to reverse the error. Both transactions remain on the blockchain [5]

### V. Finality

A single, shared ledger provides one place to go to determine who owns the asset or if a transaction has been completed [5]

## Chapter 3: Blockchain Applications and Use Cases

The concepts of parallel blockchains and sidechains allow for tradeoffs and improved scalability using independent blockchains which allow for further innovation [3]. Various benefits of blockchains include decentralization, recording and validating every transaction which provides security and reliability, authorization of transactions by miners which make the transactions immutable and prevent hacking threats, and they discard the need for third-party or central authority for P2P transactions [3]. Various companies and institutions are studying blockchains to apply them to various areas such as money transfers, risk management, smart bonds, and cryptocurrencies.




## Supply Chain Management

The immutability of the blockchain makes it suitable for tracking goods as they move and change locations in the supply chain [12]. Entries on a blockchain can be used to queue events with a supply chain [12]. Blockchain provides a new way to organize tracking data and putting it to use [12].

## Healthcare

Age, gender, and basic medical history data would all be suitable information to be stored on a blockchain in the healthcare industry. None of this information should be able to identify a patient which allows it to be stored on a shared blockchain accessible by numerous individuals without any concerns for privacy [12]. Devices will be able to store data on a healthcare blockchain that can append data to a person's medical record [12].

## Real Estate

The average person sells their house every 5-7 years, and the average person will move almost 12 times in their lifetime [12]. This information could be very useful to store on a blockchain for the real estate industry as it could expedite home sales by quickly verifying finances, reduce fraud as a result of blockchain's encryption, and offer transparency through the selling and purchasing process [12].

Media

Writers and content creators can spread their works on a blockchain and receive immediate payment [12]. Comcast's advanced advertising group developed a new technology to let companies may ad buys on broadcast and over-the-top TV through the blockchain [12].

Energy

Blockchains could be used to execute energy supply transactions and provide the basis for metering, billing, and clearing [12]. Other potential applications include ownership documentation, management of assets, guarantees of origin, emission allowances, and renewable energy certificates [12].

Record Management

National, state, and local governments must maintain records of all individuals such as birth and death dates, marital status, and property transfers [12]. Some of these records only exist in paper form. At times, citizens have to physically go to their local office to make changes which is time-consuming. Blockchain technology can simplify the record keeping process and make it far more secure [12].

Identity Management

Blockchains offer enhanced methods for managing identities as well as digitizing personal documents. Developing digital identity standards is a highly complex process. A universal online personal identity solution requires the cooperation of private entities and government.

Currently, Illinois is experimenting with blockchain technology to replace birth certificates as a form of identification [8]. This would give citizens more control over their data as well as reassurance that their information is more secure than before [8]. The Illinois Blockchain Initiative partnered with identity solutions firm Evernym to create an online ledger that is only accessible to the ID owner and additional granted individuals [8]. This is similar to how the technology could be used to share information between hospitals [8].

Voting

Blockchains are fault tolerant, do not allow someone to change the past records, hack the present records, or alter access to the system [12]. Every node on the blockchain with access can see the same results and every vote on the blockchain can be irrefutably traced to its source without sacrificing voter anonymity [12]. End-to-end verifiable voting systems will give the voter the ability to verify that their vote was correctly recorded and counted [12]. Blockchains would especially be useful in the event that a ballot was missing,

in transit, or modified and it can be detected by the voter and caught before the election is over [12].

Cybersecurity

Cisco believes that the blockchain can play a role in managing networks built on switches, firewalls, and other appliances [7]. Cisco has joined the Hyperledger Project, an open source initiative that is trying to develop cross-industry blockchain technologies. Blockchains can be used to manage switches, routers, firewalls, and internet of things gateways [7]. The blockchain technology is promising where network management is done through a centralized controller [7]. Blockchains could make management tasks possible across multiple vendors by keeping record of an appliance's current state and configuration history which ensures that changes made to a device don't cause a network outage [7].

Smart Contracts

Distributed ledgers enable the coding of simple contracts that'll only execute when specific requirements are met (e.g. status change of equipment).

Land Title Registration

Blockchains make record keeping far more efficient and can be applied to recording property titles as well as registered cars.

Connected Cars

Toyota is partnering with MIT to utilize blockchain technology for connected cars. This would allow an unspecified number of people to access data as well as manage payment between cars and charging spots [9]. Blockchains would also allow for accurately preserving vehicle mileage and maintenance records. It can link headquarters to vehicles to manage times and fees for car sharing [9]. This information is key to measuring the value of a vehicle and calculating insurance rates [9]. Blockchains would also allow car-sharing companies to locate any vehicle at any time and determine who to charge for any services [9].

Center for Disease Control

Blockchain technology could allow public health workers respond better to a crisis [10]. The CDC, state and local departments, and other organizations need to routinely share public health data so they can control the spread of various infectious diseases [10]. Blockchains can especially assist with public health surveillance by efficiently managing data during a crisis or allow for better tracking of opioid abuse [10]. Moving such important data between peers in a secure, compliant, and transparent manner as quickly as possible is key to the business model [10].

For example, let's evaluate the scenario of a pandemic. The CDC has an existing mobile app used by local health workers to log information about patients and determine medications that should be given to various patients. However, personally identifiable information cannot be stored on the cloud. Storing it in an approved manner takes more time. By using the blockchain, storing and sharing data can be done much faster while complying with security and privacy laws [10].

Preventing DDoS Attacks

Hackers use several techniques to instigate an attack such as sending junk requests to a website, increasing traffic until the site can't keep up with requests [11]. The attack goes on until the site gets overwhelmed with requests and crashes [11]. The main difficulty in preventing DDoS is with the existing DNS (Domain Name System) [11]. DNS is a partially decentralized one-to-one mapping of IP addresses to domain names and works like an internet phone book [11].

Blockchains would allow for DNS to be fully decentralized and distribute the contents to a large number of nodes which makes it nearly impossible for hackers to attack a network [11]. Domain editing rights would only be granted to domain owners and no one else could make changes [11]. This reduces the risk of data being accessed/changed by unauthorized parties [11]. By using blockchains to protect data, a system can ensure that it is invulnerable to hackers unless every node on the network is wiped clean simultaneously [11]. If current DNS would operate on the blockchain, users would still be able to register domain names but only authorized users could make changes to their domains [11]. Data would be stored on various nodes and every user on the network would have a copy. It would be virtually impossible to hack/destroy it completely [11].

Safeguarding Data

Blockchains allow for the distribution of every piece of data to nodes throughout the system [11]. If someone tries to alter the data, the system analyzes the whole chain and compares them to the meta-packet and excludes any that don't match up [11]. The only way to wipe out the entire blockchain is to destroy every node on the network. Even if just one node remains on the system, the whole system can be restored despite all the other nodes being compromised [11].

Preventing Fraud and Data Theft

Blockchains prevent potential fraud and decrease the chance of data being stolen/compromised [11]. In order to destroy/corrupt a blockchain, a hacker would have to destroy the data stored on every user's computer in the blockchain network [11]. This can sometimes be millions of computers. It is nearly impossible for a hacker to simultaneously bring down an entire blockchain network [11]. Even if the Blockchain were impacted by an attack, nodes not affected by the attack would continue to run as normal on the network

[11].
Bigger blockchain networks with more users have an infinitely lower risk of getting attacked by hackers since the complexity is higher to penetrate such a network [11].

Decentralized Storage, Recordkeeping, and Peer-to-Peer Sharing
Users are able to store all of the data in their network on their computer if they choose to do so. This results in earning money for renting “extra” storage space and they can ensure that the chain won’t collapse. If a hacker tries to tamper with a block, the whole system analyzes every block of data to find the one that differs from the rest. If the system finds this kind of block, it simply excludes it from the chain and identifies it as false. There is no central authority or storage location. Every user on the network stores some/all of the blockchain and everyone is responsible for verifying data to make sure false data can’t be changed and existing data can’t be removed [11].

# Chapter 4: Blockchains and Security

Blockchains eliminate the risk with data being held in a central location since they store data across its network [1]. Computer hackers will no longer be able to exploit central points of vulnerability when networks implement the blockchain for data [1]. Blockchain uses encryption for security with public and private keys as a basis. A public key is a long and randomly-generated string of numbers and is used as the user's’ address on the blockchain [1]. Any actions done by the user on that network get recorded as belonging to that specific public key [1]. The private key is used as a “password” that gives its owner access to assets on the network [1].

Attacks on the Blockchain Network

Consensus Attack
A consensus attack occurs when a majority of the nodes on the network work together to commit a Shallow Block Attack to commit invalid data to the Blockchain. This attack takes advantage of the fact that the official ledger is the one present at a majority of the network’s peers [18].

Consensus Attack Countermeasures
A high number of peers are the best method of countering a possible consensus attack since it is more costly to attack a network with a large number of peers. It would also be beneficial to keep track of who is allowed to become a miner on the network [18].

Shallow Block Attack
A shallow block attack occurs when a single malicious node computes blocks containing malicious data that fit into specific locations on the chain, and then swaps them with actual

blocks. This attack is called “shallow” because it typically targets blocks at the top of the chain since too much hashing power would be needed to compute a block that is deep in the Blockchain [18].

Shallow Block Attack Counter

There isn’t much you can do to counter shallow block attacks. However, in order for these to be effective, they must be coupled with a consensus attack and the counters for that attack are listed above [18].

# Chapter 5: Blockchains and Web 3.0

[In Development]

# Chapter 6: Blockchain and the CIA Triad

There are two main types of blockchains: public and private. Public Blockchains are permissionless, meaning that data is publically available to anyone who wishes to participate on the network [2]. Private Blockchains are permission based platforms established by groups or individual firms, or divisions within an organization. Data can only be accessed by those users who are part of the group and are properly authenticated [2]. Blockchains can help with digital identities and maintaining data integrity. They have the ability to improve cyber defense since they can secure and prevent fraudulent activities and detect data tampering based on their underlying characteristics of immutability, transparency, auditability, data encryption, and operational resilience [2].

## Confidentiality

The CIA triad represents confidentiality, integrity, and availability. Confidentiality is defined as the property that sensitive information is not disclosed to unauthorized individuals, entities, or processes [2]. The main concern of confidentiality is ensuring only interested and authorized parties can access the correct and appropriate data to them [2]. If an attacker can gain access to the blockchain network, then they are more likely to gain access to the data. Therefore, authentication and authorization controls need to be implemented [2]. Various blockchain implementations are starting to address data confidentiality and access control challenges by providing full block data encryption and AAA capabilities. [2]. Full encryption of blockchain data ensures that it won’t be accessed by unauthorized parties while this data is in transit.

With public blockchains, there’s no requirement to control network access since the chain protocols allow anyone to access and participate in the network [2]. Private blockchains

require that appropriate security controls are in place to protect network access. Blockchains can provide advanced security controls: leveraging the public key infrastructure (PKI) to authenticate and authorize parties, and encrypt their communications [2].

Today, if an attacker gains access to the blockchain network and its data, this doesn't always mean that the attacker can read or retrieve the information stored on the blockchain. Full encryption of data blocks can be applied to guarantee its confidentiality. End-to-End Encryption ensures that only those who have authorization to access the encrypted data i.e. through their private key can decrypt and see the data [2]. Using encryption keys along with PKI on a blockchain can provide higher levels of security. Implementing secure communication protocols on a blockchain guarantees that even when an attacker tries to do a man-in-the-middle attack, the attacker won't be able to either forge the interlocutor's identity or disclose any data while in transit [2]. Accessing the blockchain from multiple devices using the same key can put the organization at a higher risk of losing control of their private keys.

Users are protected by their unique Blockchain ID [18].

## Integrity

Integrity is defined as guarding against improper information modification or destruction. It includes ensuring information nonrepudiation and authenticity [2]. This may consist of data encryption, hash comparison, or digital signatures. Blockchains have built in immutability and traceability that provide means for data integrity.

Blockchains are secure from the perspective that they enable users to trust that the transactions stored on them are valid [2]. Blockchains use sequential hashing and cryptography. They also have a decentralized structure. These 3 characteristics make it very challenging for any party to tamper with it compared to a standard database [2]. 51% of users need to agree a transaction is valid before it is added to the blockchain (consensus model protocol).

Blockchains guarantee that nothing will be erased. The solution to the "right to be forgotten" problem of integrity is to encrypt the personal information written in the system to ensure that forgetting the keys will ensure that the information is no longer accessible. Therefore, we should focus on blockchain's value to provide unaltered evidence of facts by writing the hash of transactions to it, while the transactions themselves are stored outside of the system. This ensures integrity of transactions and enables the ability to erase transactions, leaving only small traces of forgotten information on the chain._Every

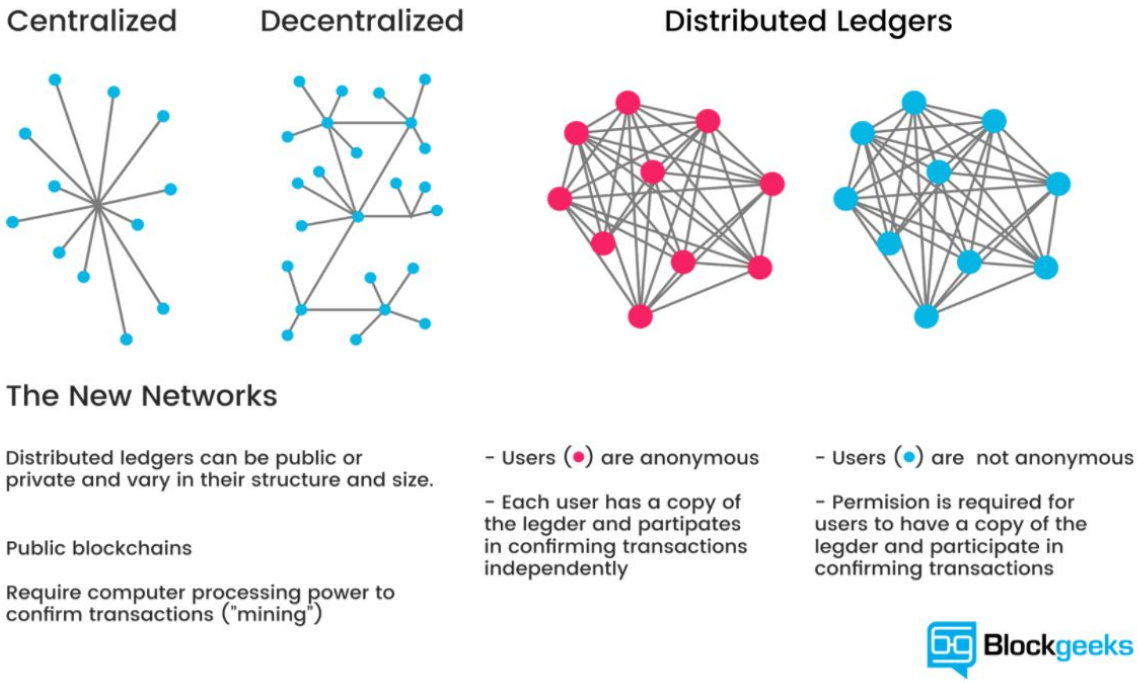
transaction added to a blockchain (private or public) is digitally signed and timestamped. The organization can trace back to a specific time period for each transaction and identify the party (through their public address) on the blockchain [2]. Nonrepudiation is the assurance that someone can't duplicate the authenticity of their signature on a file or the authorship on transaction that they originated [2]. It increases the reliability of the system (detection of tamper events or fraudulent transactions) since every transaction is cryptographically associated to a user. Any new transaction added to the blockchain will result in a change of the global state of the ledger. The implication is that with every new iteration of the system, the previous state of the system will be stored which results in a fully traceable history log [2]. This provides entities with an extra level of reassurance that data is authentic and hasn't been tampered with.
Blockchains can guarantee data integrity since each set of data is guarded by a merkle tree and hashing [18].

## Availability

Availability means ensuring timely and reliable access to and use of information. The Distributed Denial of Service (DDoS) is the most common attack and can cause the most disruption to internet services as websites are disrupted and apps become unresponsive [2]. DDoS attacks on blockchain networks are costly since they attempt to overpower the network with large volumes of small transactions. The decentralization and peer-to-peer characteristics of the blockchain make it harder to disrupt than client-server architectures. Blockchains are also subject to DDoS attacks but they are able to withstand them a lot better than regular network architectures. Since Blockchains have no single point of failure, this decreases the chances of an IP-based DDoS attack disrupting normal operation. If a node is taken down, data is still accessible via other nodes on the network since they all have a full copy of the blockchain at all times [2]. Bitcoin has withstood cyber-attacks for more than 7 years.  Organizations can still face risks from external events outside of their control despite using blockchains (i.e. global internet outage) [2]. Blockchains have operational resistance due to their combination of peer-to-peer nature and number of nodes on the network, operating in a distributed and 24/7 manner. Organizations can make a node under attack redundant while the rest operate as usual.  Even if a major part of the blockchain is under attack, it'll operate as normal due to the distributed nature of the technology. Obviously, blockchains aren't "bullet-proof."

The blockchain's existence at numerous peers on the network protects it from ever being lost. Thus, guaranteeing its availability [18].

# Chapter 7: Centralization vs. Decentralization




# Chapter 8: Security Properties of the Blockchain

As shown by the CIA Triad, block chains consist of many security properties, and some natural flaws. However even these flaws can be addressed to sustain the securability of the block chain. By nature of block chains, anyone connected to the network has access to all the nodes. However to counter this, many blockchains encrypt all their information, so only people with the key can access it.

# Chapter 9: Smart Contracts and the Blockchain

Smart Contracts are accounts holding objects on the Ethereum blockchain [6].

# Chapter 10: Blockchain Performance and Maintenance

[In Development]

# Chapter 11: Blockchain Capabilities and Limitations

The nature of blockchains means that a large network of users is required to ensure that it is robust. [16] A blockchain with only a few users puts it at greater risk of failure and security risks such as a 51% attack. For example, a blockchain with 100 users versus one with 1000 is more likely to experience failures if users begin to disappear, or change the blockchains data . The large user base ensures the robustness of the blockchain, as authenticity is guaranteed if a very large amount of users can confirm it.

# Chapter 12: Blockchain Tutorials

## Creating a Private Ethereum Blockchain in Windows

1. Install Geth and Mist
   GETH: https://geth.ethereum.org/downloads/ - install appropriate version
   MIST: https://github.com/ethereum/mist/releases --install appropriate version
   a. Unzip Geth and Mist to a safe location. Geth is the executable that syncs the blockchain to your computer. Mist is used to track the sync progress and is also the platform to use for connecting to the public Ethereum blockchain or a testnet.

2. Run Ethereum-Wallet.exe within the Mist/Ethereum Wallet folder. This will visually show the progress of the Ethereum Sync

3. Run Geth.exe within the Geth folder previously downloaded. This will initialize the syncing of the blockchain.

## Installing Bash for Windows 10

Bash is a Ubuntu based shell that allows windows to run linux based software and commands. Bash is recommended as many of the commands will be linux based. In order to use it be sure to have windows 10 64 bit and the windows 10 anniversary update.

1. Open settings, Go to “Update and Security” then “For developers.
   a. Enable” Developer Mode”
2. Open Control Panel, click on “Programs” then “Turn windows features on or off”
   a. Under programs and features enable, “Windows Subsystem for Linux(Beta)” and click ok.
3. Reboot your computer.

4. Go to start and type “Bash” and run the Bash command
    a. This will install Bash and you will be prompted to continue. Type y to continue and complete the installation.

## Development using Windows Terminal with Cygwin

Cygwin is a terminal for windows 7 used to recreate a linux OS terminal. If you use windows 7, or windows 8.1, Cygwin is the terminal that you want. If you have windows 10, you can use Bash for Windows. If you have an UNIX OS, you can use the default terminal.
Download Cygwin: https://www.cygwin.com/

Cygwin Instructions:

Follow the download prompts for Cygwin. It will have you choose a mirror to download from. Don’t be afraid to choose whichever one and then follow through keeping all the defaults. After you’ve downloaded, you can create a shortcut on the desktop.
Download Bash for Windows: //link needed

In this tutorial I will walk you through the solidity development process. This tutorial is directed towards those who are new to new to blockchain development. I will show you how to get all the applications you need to begin work as a blockchain developer and also walk you through creating your first solidity contract and the mechanics behind it.

## Installing Node.js, npm, git, web3.js, truffle, and connecting to the ethereum testnet.

Node.js, npm, git, web3.js, and truffle are essential ingredients for Blockchain development. //add short explanation of why

1. Install nodejs : https://nodejs.org/en/ 8.9.0
    a. Install the appropriate node.js version, then confirm it’s installed by typing $node -v in your terminal.
    b. Note: Npm should be installed with nodejs. Check with npm -v in the console. Skip step 2 if version was installed.
2. Install npm via terminal if needed.
    a. $npm install --global. https://docs.npmjs.com/getting-started/what-is-npm
    b. You can check your version of npm by typing
        $npm -v
3. Install git - https://git-for-windows.github.io/
    a. Follow install instructions and keep default selections. Once you finish, restart your terminal. Type

$git version

4. Install web3.js via terminal
    a. Web3.js is the main ethereum javascript library that comes packed with useful API. Web3.js can be very tricky to install. The first command to try is
        $npm install web3
        A lot of times you will run into errors, especially if you are using a Windows OS (although sometimes you won't). Try
        $npm install web3@^0.20.0
5. Install truffle via terminal
    a. $npm install -g truffle
    b. To check if you have truffle installed, type
        $truffle
        This will give you a list of the commands you can use in truffle
6. Install ethereum testnet via terminal
    a. $npm install -g ethereumjs-testnet

These are the main tools you need to begin life as a blockchain developer. For editing solidity and javascript files, we recommend Visual Studio Code; but you can use whichever IDE you prefer.

## Using Truffle to create smart contracts

Once you have everything installed and downloaded (node.js, npm, git, web3.js, truffle, and ethereum testnet), you are now ready to create your first blockchain project! The first thing you want to do is open up your terminal and navigate to a workspace you enjoy. If you use Cygwin, the terminal will open up in C:\cygwin64\home\UserName . This is the workspace I use. To create a folder within that workspace, type

$mkdir SD_Blockchain

The name of my project in this instance is SD_Blockchain.To navigate into this workspace type

$cd SD_Blockchain

If you are familiar with Linux you will recognize a lot of these commands. That's because Cygwin is a terminal meant to emulate linux. Once you are in your workspace, you can check which files are in your terminal with the command

$ls

Once you are in your workspace, for the first time, you should have no files in it. To begin your first contract you will use a truffle command

$truffle init

This will initialize truffle within that workspace and create three folders and two Javascript files. The folders are [contracts] [migrations] [test] and the files are truffle.js and truffle_test.js.

[contracts] will be where you make all of your contracts. To create a .sol file within the contracts folder first navigate into contracts

```
$cd contracts
```

You will notice there is already a contract in their, called migrations. Keep that their as it's used when you migrate any new contracts after compilation. Now you want to create a new .sol file. To do so,

```
$touch FileName.sol
```

This will create a new file with filetype .sol. You can edit this file with Visual Studio Code or whichever IDE you prefer.

Next you will have to add to the [migrations] folder. You will see their is already a javascript file in it called 1_initial_migration. You will want to create a new .js file

```
$touch 2_filename_migration.js
```

It doesn't really matter what you call it as long as it indicates which .sol file it's for.

# Development using Ubuntu 16.04 (Bash/Cygwin for Windows)

## Installing Ethereum

Use the following commands:
sudo apt-get install -y software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install -y ethereum

## Creating the Genesis Block

Use the following commands:
mkdir genesisblock - make the directory
cd genesisblock - get into the directory
nano genesis.json - make the file

Copy and paste the following genesis block into your file:
[insert our genesis block here]
Save this new file.

## Command List for Libraries

List of linux commands - https://ss64.com/bash/

# Coding with Solidity and Javascript

## Truffle Development

Commands:
Truffle console - this command initializes the Truffle console that lets you enter commands.
Compile - compiles your solidity code.
Migrate - migrates your contract to a file that can be deployed.
Deploy - deploys your contract onto the blockchain.
addBlock.new(“Constructor Initialization”)
addBlock.at(“contract_Address”).addData(“hello”,1,1234)

## Getting Your Application to Communicate with a Solidity Contract

First, you’ll need to declare your web3 provider. Use the following line below to declare your provider as your local host.
var web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

You’ll need to set your default account. For the purpose of development and testing, we’ll set the default account to the first account in the testRPC.

```
web3.eth.defaultAccount = web3.eth.accounts[0];
```

This is the tricky part. You will need to pass the JSON representation of the contract ABI into the contract instance. Go to remix.ethereum.org and paste your solidity contract into a new text file. In the compile window, click the “Details” button and a popup window will appear. Click the clipboard icon for the Interface:ABI section. That is the JSON representation of your contract. Structure your statement like ours:

var addBlockContract =
web3.eth.contract([{"constant":true,"inputs":[{"name":"key","type":"uint256"}],"name":"getdata","outputs":[{"name":"","type":"bytes32"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":true,"inputs":[{"name":"key","type":"uint256"}],"name":"getID","outputs":[{"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"inputs":[{"name":"data","type":"bytes32"},{"name":"ublock_no","type":"uint8"},{"name":"uid","type":"uint256"}],"name":"addData","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"},{"constant":true,"inputs":[{"name":"key","type":"uint256"}],"name":"getblock_no","outputs":[{"name":"","type":"uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":true,"inputs":[{"name":"","type":"uint256"}],"name":"blockchain","outputs":[{"name":"block_no","type":"uint8"},{"name":"id","type":"uint256"},{"name":"data","type":"bytes32"}],"payable":false,"stateMutability":"view","type":"function"}]);

Now, we have a working contract. But, we can’t facilitate communication yet. To do that, we’ll need an instance of the contract to work with. Declare the following statement and replace the address with your current contract address. You’ll find it as the second to last parameter after running addBlock.new(‘Constructor Initialization’):

```
var addingBlock = addBlockContract.at('0xcd09404926caf1ae13dbae49f4d853930b95a192');
```

# Resources for API/Language Documentation

**Solidity Documentation**
https://solidity.readthedocs.io/en/develop/

**Web3js API Documentation**
https://github.com/ethereum/wiki/wiki/JavaScript-API

**Nodejs API Documentation**
https://nodejs.org/api/index.html

**Javascript Documentation**
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

**HTML Documentation**
https://developer.mozilla.org/en-US/docs/Web/HTML

**CSS Documentation**
https://developer.mozilla.org/en-US/docs/Web/CSS

# Glossary of Terms

**51% Attack**
When more than half of the computing power of a network is controlled by a single entity/group. This entity or group may issue conflicting requests to harm the network.

**Blockchains**
Blockchains are shared, trusted, and public ledgers of transactions. Everyone is able to see the transactions but no single user controls them. Blockchains are cryptographed, secure, and tamper resistant. They are distributed databases that are perfect for storing information such as values, identities, agreements, property rights, and credentials. Any information placed into a blockchain will remain there forever. Blockchains are decentralized, disintermediated, affordable, and censorship resistant. Various applications include Bitcoin, Namecoin, Sia, and Ethereum.

**Block**
A block is a file that is used to permanently record data. It is a record of some or all of the most recent transactions that haven't been recorded in any previous blocks. New blocks are added at the end of the blockchain and can't be changed or removed once they are

written. Each block memorializes what happened in the minutes before it was created and keeps a record of some or all recent transactions as well as a reference to its previous block.

**Block Explorer**
A block explorer is an online tool used for exploring the blockchain. It allows for watching and following all transactions in real time on the blockchain. They can serve as blockchain analysis and provide information such as total network hash rate and transaction growth.

**Block Height**
The number of blocks connected in the blockchain.

**Chain Linking**
Chain linking is the process of connecting two blockchains. This allows for transactions between the two chains and also allows blockchains to communicate with other sidechains.

**Client**
A software program executed by a user on a desktop, laptop, or mobile device to launch an application.

**Consensus**
A consensus requires that an agreement is made among a number of processes for a single data value.

**Consortium Blockchains**
A consortium blockchain is a blockchain where the consensus process is controlled by a preselected set of nodes. Each user operates a node and they must sign every block for the block to be valid. The right to read the blockchain may be public or restricted to its participants. Consortium blockchains are considered to be “partially decentralized.”

**Cryptographic Hash Function**
A cryptographic hash function is a math algorithm that takes an input that can be any kind of digital data (ex: password file) and produces a single fixed length output. The main properties of cryptographic hash functions are as follows: easy to compute a hash value for any message; infeasible to generate a message from its hash function except through brute force; infeasible to change a message’s contents without changing the hash; infeasible to find two messages with the same hash; deterministic so that the same message always has the same hash output. Cryptographic hash functions may have security applications, indexing data in hash tables, fingerprinting, detecting duplicate data or uniquely identify files, and as checksums to detect accidental data corruption.

**dApp (Decentralized Application)**
An application is decentralized if it meets the following criteria: completely open-source, operate autonomously, and with no entity controlling the majority of its tokens and all changes being decided on by a consensus of its users; data and operation records must be cryptographically stored in a public and decentralized blockchain to avoid central points of failure; must use a cryptographic token necessary for access to the application; must generate tokens according to a standard cryptographic algorithm acting as a proof of valid contributing nodes.

**Distributed Network**
A network where processing power and data are spread over the network nodes instead of having a centralized data center.

**Digital Signature**
A digital code generated by a public key encryption that is attached to an electronically transmitted document to verify its contents and the sender's identity.

**Ethereum**
Ethereum is an open software platform that uses blockchain technology. It enables developers to write smart contracts and build and deploy decentralized applications.

**Fork**
Forks are created when two blocks are created at the same time. This essentially creates two parallel blockchains with one of the two being the winning blockchain. The winning blockchain gets determined by its users by the majority choosing which blockchain the clients should listen to.

**Genesis Block**
The very first block in a blockchain.

**Hash**
Performing a hash function on the output data. This is used for confirming transactions.

**Hardfork**
A hardfork is a change to the protocol of the blockchain that makes previously invalid blocks or transactions valid. It requires all users to upgrade their clients.

**Hashcash**
Hashcash is a proof-of-work system that is used to limit email spam and DoS attacks.

**Light Node**

A light node is a computer on the blockchain network that only verifies a limited number of transactions.

**Lightning Network**
A lightning network is a decentralized network that uses smart contract functionality on the blockchain to allow for instant payments across a network of participants. It allows transactions to happen instantly without worrying about block confirmation times. Lightning networks also allow two participants on the network to create an entry, conduct transactions between each other, and record the state of the transactions on the blockchain.

**Merkle Tree**
A Merkle tree's main concept is to have some piece of data linking to another. This can be accomplished by linking things together via a cryptographic hash. The content itself can be used to determine the hash and this allows us to address the content. The content becomes immutable because if you change anything in the data, the hash changes and the link is different. Every block points to the previous block and a block becomes invalid if it is modified.

**Node**
A node is any computer that is connected to the blockchain network. Nodes are considered to be full nodes if they fully enforce all rules of the blockchain. Most nodes are lightweight nodes but full nodes form the network backbone.

**Oracles**
Smart contracts on the blockchain that cannot access the outside network on their own. Oracles sit between a smart contract and the external world. They provide data needed by the smart contract and send its commands to external systems.

**Peer-to-Peer Network**
The decentralized interactions between two parties or more in a highly-interconnected network. Participants of a P2P network deal directly with each other through a single mediation point.

**Public Address**
The cryptographic hash of a public key. These can act as email addresses that can be published anywhere unlike private keys.

**Private Keys**
String of data that allows you to access the tokens. These act as passwords that are kept

hidden from anyone but the owner of the address.

**Private Blockchains**
Private blockchains are blockchains where write permissions are centralized to a single organization. Read permissions can be public or restricted.

**Proof of Authority (PoA)**
A proof of authority is a method of consensus in a private blockchain. It gives one or more clients with one particular private key the right to make all blocks on a blockchain.

**Proof of Work (PoW)**
A proof of work system is a measure used to deter denial of service attacks and other service abuses such as spam. PoW systems require some work from the service requester. Typical work may include processing time by a computer.

**Public Blockchains**
A public blockchain is a blockchain that allows for anyone to read it, send transactions, and participate in the consensus process. Public blockchains are secured by crypto economics which is the combination of economic incentives and cryptographic verification by using proof of work or proof of stake. Public blockchains are considered to be fully decentralized.

**Ring Signature**
Ring signatures are cryptographic and provide a decent level of anonymisation on a blockchain. They ensure that individual transaction outputs cannot be traced. A message signed with a ring signature is endorsed by someone. It should be computationally infeasible to determine which group member key was used to produce the ring signature.

**SHA (Secure Hash Algorithm)**
A SHA is a group of cryptographic hash functions published by the National Institute of Standards and Technology. It takes an input of any size and form, mixes it up, and creates a fixed size output known as a hash. A hash can be viewed as a fingerprint of the data. They are one-way functions and cannot be decrypted back to their original form unless brute force is used.

**Smart Contracts**
Smart Contracts are computer protocols that facilitate, verify, or enforce the negotiation or performance of a contract. They usually have a user interface and emulate the logic of contractual clauses. They aim to provide security superior to traditional contract law.

**Softfork**
A soft-fork is a change to the protocol where only previously valid blocks are made invalid.

It is backward-compatible since old nodes will recognize new blocks as valid. This type of fork requires only a majority of miners to upgrade and enforce the new rules.

**Solidity**
A programming language used for developing Smart Contracts.

**SVP (Simplified Payment Verification) Client**
SVP Clients are lightweight clients that don't download and locally store the entire blockchain. They provide a way to verify transactions without having to download the whole blockchain. They only download the block headers by connecting to a full node.

**State Channel**
A state channel is an interaction that is made off of the blockchain without significantly increasing participant risk. This is done by moving interactions off of the chain which can lead to improvements in cost and speed. They function by locking part of the blockchain state so a set of participants can completely agree with each other to update it.

**Swarm**
A Swarm is a distributed storage platform and content distribution service. Its primary objective is to provide a decentralized and redundant store of Ethereum's public record. Essentially, it stores and distributes dApp code and data as well as blockchain data.

**Token**
A token is a digital identity for something that can be owned and is created as a sophisticated smart contract system with permission systems and interaction paths attached to them.

**Testnet**
A testnet is a second blockchain used by developers to test new versions of client software. This is done to prevent putting data at risk.

**Transaction Block**
Collection of transactions gathered into a block that can then be hashed and added into the blockchain.

**User Keys (Public and Private)**
**Public Key:** A User's address on the blockchain. (A long string of randomly generated numbers)
**Private Key:** Gives access to a User's digital assets.
Having a public and private key is what allows users to share digital assets through the blockchain safe and securely. You must safeguard and protect your private key however;

and if you lose your private key you will be unable to retrieve your assets.

**Wallet**
A file that houses private keys and usually contains a software client that allows access to view and create transactions on a specific blockchain.

**Whisper**
A whisper is a part of the Ethereum protocol that allows for messaging between users on the same network that runs the blockchain. Its main task is the provision of a communication protocol between decentralized applications.

# References [IEEE]


[1]"What is Blockchain Technology? A Step-by-Step Guide For Beginners", Blockgeeks, 2017.

[2] E. Piscini, D. Dalton and L. Kehoe, "Blockchains and Cyber Security", Deloitte, 2017.

[3] "Know more about Blockchain: Overview, Technology, Application Areas and Use Cases - Lets Talk Payments", Lets Talk Payments, 2017.

[4]"4 Key Features of Blockchain", Techracers, 2017.

[5]M. Gupta, Blockchain for Dummies. for Dummies: A Wiley Brand, 2017.

[6] "Ethereum Project", Ethereum.org, 2017

[7] A. Gonsalves, "Cisco says blockchain ledger technology has networking role", SearchSDN, 2017.

[8] K. Leary, "Illinois is experimenting with blockchains to replace physical birth certificates", Futurism, 2017.

[9] N. Shimizu, "Blockchain's new client: Connected cars- Nikkei Asian Review", Nikkei Asian Review, 2017.

[10]M. Orcutt, "Why the CDC thinks blockchain can save lives", MIT Technology Review, 2017.

[11]"Using Blockchain Technology to Boost Cyber Security", Hacker Noon, 2017.

[12]A. Meola, "The growing list of applications and use cases of blockchain technology in business & life", Business Insider, 2017.

[13]"Blockchain Glossary." Blockchainhub, 16 October 2017.

[14]"Comprehensive Blockchain Glossary: From A-Z - Blockgeeks." Blockgeeks. 16 October 2017.

[15]"What Is Blockchain Technology? A Step-By-Step Guide For Beginners." Blockgeeks. 16 October 2017.

[16]"What are Blockchain's Issues and Limitations? - CoinDesk", CoinDesk, 2017.

[17] Antonopoulos, Andreas M. (2017-06-12). Mastering Bitcoin: Programming the Open Blockchain. O'Reilly Media. Kindle Edition.

[18] Church, Zach. "Newsroom." Blockchain, Explained, 25 May 2017, mitsloan.mit.edu/newsroom/articles/blockchain-explained/.

[19] Laurence, Paul. Blockchain: Step-By- Step Guide to Understanding and Implementing Blockchain Technology. Kindle Edition.