2018

Demonstrating the Functionality and Efficacy of Blockchain-based System in Healthcare Using Simulation Tools

Jad S. Mubaslat
Wright State University

DEMONSTRATING THE FUNCTIONALITY AND EFFICACY OF BLOCKCHAIN-BASED SYSTEMS IN HEALTHCARE USING SIMULATION TOOLS

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Industrial and Human Factors Engineering

by

JAD S. MUBASLAT
B.S.B.M.E., Ohio State University, 2015

2018
Wright State University

Wright State University
GRADUATE SCHOOL

March 21, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Jad S. Mubaslat ENTITLED Demonstrating the Functionality and Efficacy of Blockchain-based Systems in Healthcare Using Simulation Tools BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Industrial and Human Factors Engineering.

Frank W. Ciarallo, Ph.D.
Thesis Director

Jaime E. Ramirez-Vick, Ph.D., Chair
Department of Biomedical, Industrial
and Human Factors Engineering

Committee on
Final Examination

Frank W. Ciarallo, Ph.D.

Jaime E. Ramirez-Vick, Ph.D.

Subhashini Ganapathy, Ph.D.

Barry Milligan, Ph.D.
Interim Dean of the Graduate School

ABSTRACT

Mubaslat, Jad S. M.S.I.H.E., Department of Biomedical, Industrial and Human Factors Engineering, Wright State University, 2018. *DEMONSTRATING THE FUNCTIONALITY AND EFFICACY OF BLOCKCHAIN-BASED SYSTEMS IN HEALTHCARE USING SIMULATION TOOLS.*

Blockchain and cryptocurrencies have been a rapidly growing industry and area of academic research since Bitcoin's launch in early 2009. Blockchains have already been applied in spaces outside of finance, such as healthcare. This work explores the Bitcoin blockchain and a blockchain-based care coordination system through the use of modeling and simulation tools. Two agent based models are constructed: one to represent the Bitcoin blockchain, and a second to represent a blockchain-based care coordination system, MDChain. Insight is provided that is relevant to current issues within the Bitcoin community and predictions are made as to its future. The feasibility of constructing a blockchain-based care coordination system is also demonstrated while discussing the requirements that a healthcare solution should have. The work provides a foundation for advancing current understanding of blockchain systems, and to further the development of simulation models of blockchains.

Contents

List of Figures

List of Tables

Acknowledgment

Thank you Dr. Frank W. Ciarallo for providing structured, thorough, constructive, continuous feedback throughout this process. I sincerely appreciate you putting the effort forth to understand Bitcoin/blockchain technology. Working with you has been a pleasure, and one of the brightest aspects of my academic experience at Wright State University.

Thank you to my committee members, Dr. Jaime E. Ramirez-Vick and Dr. Subhashini Ganapathy, for giving me the opportunity to perform my work in this new realm that is blockchain technology.

Thank you to my friends for being an incredible group of individuals who always support me in whatever I do; Chad Davis has been with me on this Bitcoin/cryptocurrency/blockchain journey since the start of BitQuick.co in 2013 and I'm glad to continue working with you and being great friends.

And finally, a sincere extension of my gratitude to my mother, Lamees Mubaslat, father, Saed Mubaslat, and brother, Rahmey Mubaslat. I could not ask for a more solid group of family members. To my mother for encouraging a healthy work-life balance; to my father for providing me with advice on both technical and non-technical issues; and to my brother for acting as an excellent counter-balance and showing me different perspectives.

This list is of course incomplete, as I could not possibly list all the individuals and organizations who have impacted my life in varying ways throughout the years, and ultimately shaping the thesis I have written. I will always be grateful for the help I receive, no matter how big or small.

Introduction

This thesis explores the Bitcoin blockchain and a blockchain-based care coordination system through the use of modeling and simulation tools. A literature review is first presented which explains the relevance of modeling and simulation to the blockchain and healthcare fields, discusses healthcare and care coordination related works generally, blockchains and cryptocurrencies generally, and security and scalability issues related to blockchains. This is followed by a study of the Bitcoin blockchain by analyzing a constructed agent based model. Finally a study of a blockchain-based care coordination system is presented by analyzing a second agent based model that was constructed. Insight is provided that is relevant to current issues within the Bitcoin community, the feasibility of a blockchain-based care coordination system is demonstrated, and a foundation for further blockchain simulations to be built in the future is provided. The model developed is the first known application of simulation tools to explore blockchain-based healthcare systems. The two models were constructed using AnyLogic modeling software.

Blockchains and healthcare systems are naturally complex and involve numerous stakeholders who abide by varying rules depending upon their goals and incentives. Since modeling and simulation tools, and more specifically agent based modeling, are valuable tools for exploring complex systems, blockchains and healthcare are a suitable target for application. Additionally, agent based modeling is useful in contexts where the system being studied displays nonlinear behavior among independent and autonomous entities that lead to an emergent whole [1].

In this thesis, models are considered to play three potential roles: generators, mediators, or predictors, as shown below in Figure 1.1. Generator models are used to develop new theories about how systems behave, predictors provide specific outputs to study already well understood systems, while mediators have properties of both generator and predictor models [2].

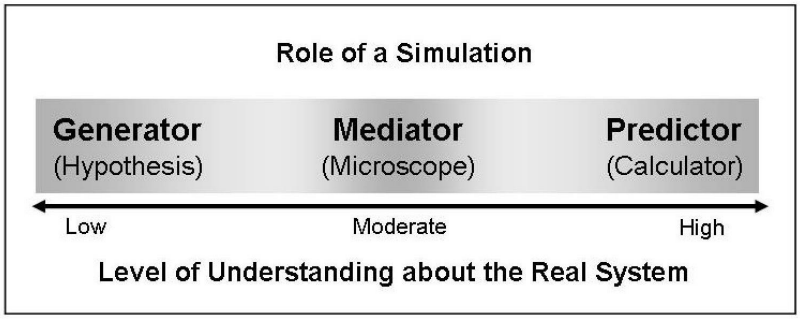


Figure 1.1: Role of a Simulation [2]

To understand blockchains more generally, it can be helpful to first understand Bitcoin's blockchain, the first public implementation of a blockchain. Bitcoin is a peer-to-peer digital currency, and program, released in 2009 that was first described in a white paper issued by pseudonym Satoshi Nakamoto in late 2008 [3]. "Bitcoin" refers to the network while "bitcoin" refers to the native token built into the network, known as a "cryptocurrency". Anyone who possesses bitcoins can send any amount of money, nearly instantly, with low fees, anywhere in the globe, and without dependence upon a third party, such as a government or private bank. This is accomplished by utilizing a cryptographically connected distributed ledger that is secured with a native underlying token to incentivize honest behavior of participants. More information as to how Bitcoin functions is provided in subsection 3.1.1. Since Bitcoin's launch, thousands more cryptocurrencies and tokens have been launched, having a combined market capitalization of over $750 billion in November, 2017 [4].

This thesis develops a model that incorporates Bitcoin users and miners that make

transactions among each other and engage in mining activities. This model is used to study varying configurations and implementations of blockchains by simulating varying scenarios which are outlined below and discussed in further detail in section 3.2.6:

- Default - Replicates the current Bitcoin network
- Bitcoin Cash - Replicates the current Bitcoin Cash network, a separate cryptocurrency
- Litecoin - Replicates the current Litecoin network, a separate cryptocurrency
- Segwit2x - Replicates a proposed upgrade to the current Bitcoin network that failed to instantiate

The metrics and time periods used to perform model validation and explore scalability are outlined in Table 1.1. To validate model behavior, metrics such as the total size of the blockchain, total supply of bitcoins in circulation, average bandwidth usage, and total number of transactions were recorded and compared to data from the real Bitcoin network from January 1, 2009 until January 8, 2016. To explore issues related to scalability, metrics such as average bandwidth usage, transaction fees, and memory pool size were recorded from July 31, 2017 until June 30, 2019. The results of these simulations support a discussion of scalability issues related to the real Bitcoin network. In addition, based on these simulations, and the trends in activity in 2017, predictions can be made on the expected rise in fees in early 2018.

Table 1.1: Time periods simulated and metrics collected during Bitcoin simulations

Topic	Model Validation	Scalability
Metrics Used	• Total Size of the Blockchain • Total Supply of Bitcoin in Circulation • Total Number of Transactions • Average Bandwidth Usage	• Transaction Fees • Transactions in Memory Pool • Average Bandwidth Usage
Time Period	January 1, 2009 until January 8, 2016	July 31, 2017 until June 30, 2019

The model developed is considered to be a mediator model, as it provides insight into the behavior of the Bitcoin network while also providing some specific computational results. The main interface of the simulation can be seen on the following page in Figure 1.2. The left panel, in red border, provides a visual representation of how the Bitcoin network behaves; the right panel, in blue border, displays charts and statistics related to the simulation and contains 9 items labeled 1R-9R. These items are further described in subsection 3.2.3.

The underlying blockchain technology can be applied to more industries than finance alone. Blockchains contain critical technical features that enable their operational capabilities, and are outlined below in Table 1.2. These technical features and operational capabilities should be used as underlying concepts when considering applying a blockchain to varying industries, and are further discussed in subsection 4.1.1.

Table 1.2: Operational Capabilities vs Technical Features of Blockchains

		Technical Features							
		Immutable Ledger	Consensus	Smart Contracts	Multi-Sig	Cryptography	Asset Digitization	P2P	Capability Total
Operational Capabilities	Transfer of Value	1	1	1	1		1		5
	Security	1	1	1	1	1			5
	Auditability	1	1			1	1		4
	Decentralization of Trust	1	1	1				1	4
	Feature Total	4	4	3	2	2	2	1	

Healthcare is one industry that may benefit from the application of blockchain technologies. Several blockchain projects have already targeted the healthcare industry, and the work previously done is further discussed in subsection 2.3.3 and section 4.1. Blockchain related healthcare projects that have publically traded tokens were valued at a combined value of $407M as of February, 2018 [4].

Healthcare systems involve complex interactions between dynamic participants with varying demands and behaviors. Care coordination is the act of recording a patient's health activity as they interact with various stakeholders. Care coordination solutions should fulfill

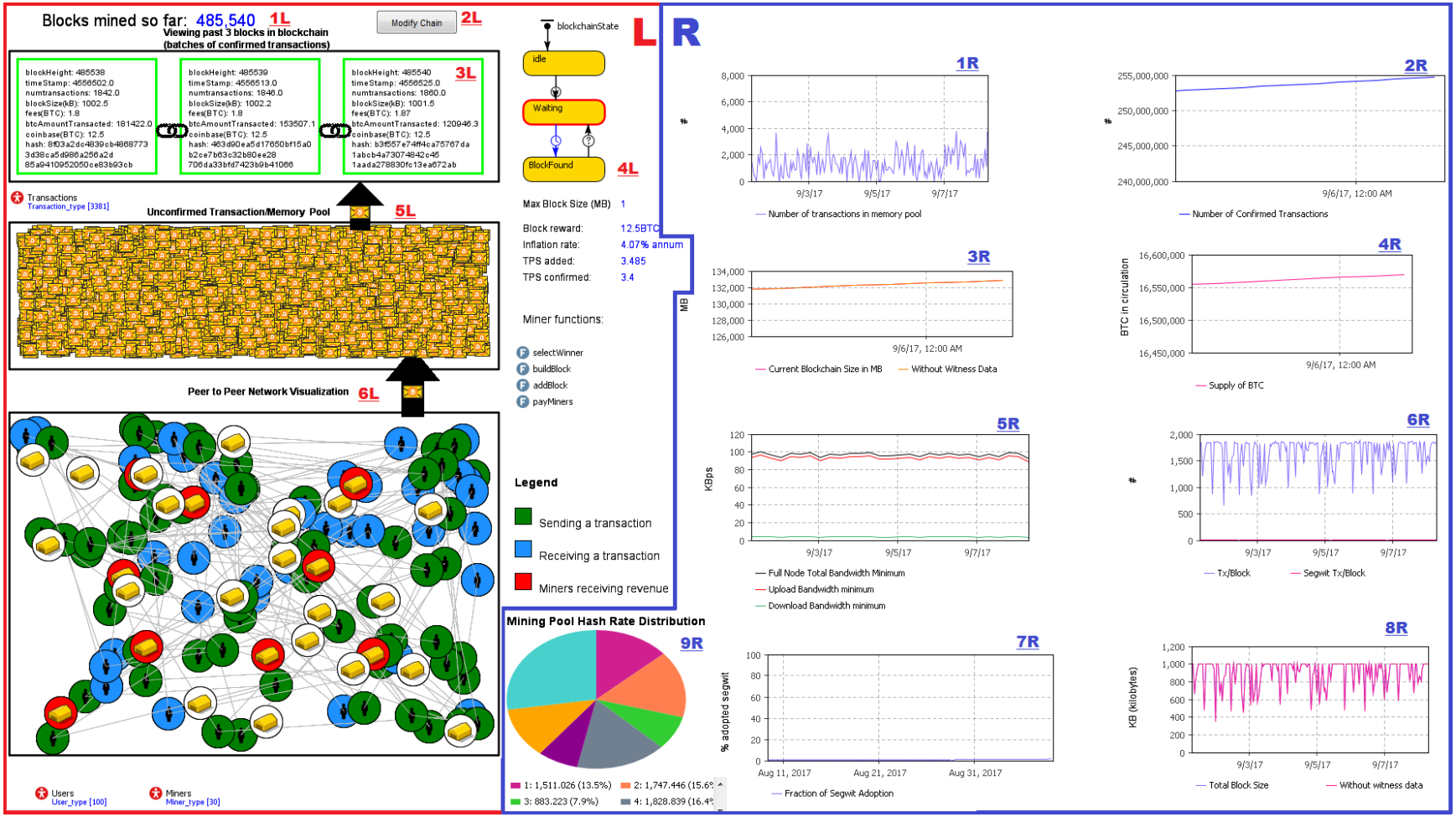


Figure 1.2: Bitcoin Blockchain Simulation in Progress GUI

varying healthcare system requirements. SimplyVital Health is already using blockchain technology to perform care coordination of patients [5]. In this thesis, using the AnyLogic model, care coordination is studied and modeled as a specific activity within the healthcare ecosystem. The operational capabilities that a blockchain can provide are outlined and displayed in relation to which healthcare system requirements they contribute to fulfilling in Table 1.3 and are further discussed in section 4.1.2.

Table 1.3: Requirements of Healthcare vs Operational Capabilities of Blockchains

		Operational Capabilities				
		Auditability	Transfer of Value	Security	Decentralization of trust	Requirement Total
Healthcare Requirements	Cost Reduction	1	1	1	1	4
	Fraud Prevention	1		1	1	3
	Identity Management			1	1	2
	Record Availability	1	1			2
	HIPAA Compliance	1		1		2
	Universality of Record	1			1	2
	Auditability	1	1			2
	Reconciliation of Records	1				1
	Interoperability	1				1
	Encourage Patient Engagement		1			1
	Capability Total	8	4	4	4	

Care coordination is studied by developing an agent based model of a hypothetical blockchain-based care coordination system, named MDChain and is outlined in Figure 1.3. MDChain is used to track the activity of opioid addicted patients as they interact with varying stakeholders in the healthcare ecosystem, such as hospitals, rehabilitation treatment centers, and general practitioners. The model is considered to be a generator model, as it does not predict specific results but does provide support for the understanding of blockchain-based healthcare systems.

The modeled network architecture can be seen in the main interface that is shown below in Figure 1.4, and further described in subsection 4.2.3. The ability to record patient information in a distributed, immutable manner is demonstrated by providing sample record outputs from simulation runs. These results are presented in section 4.3 and used to

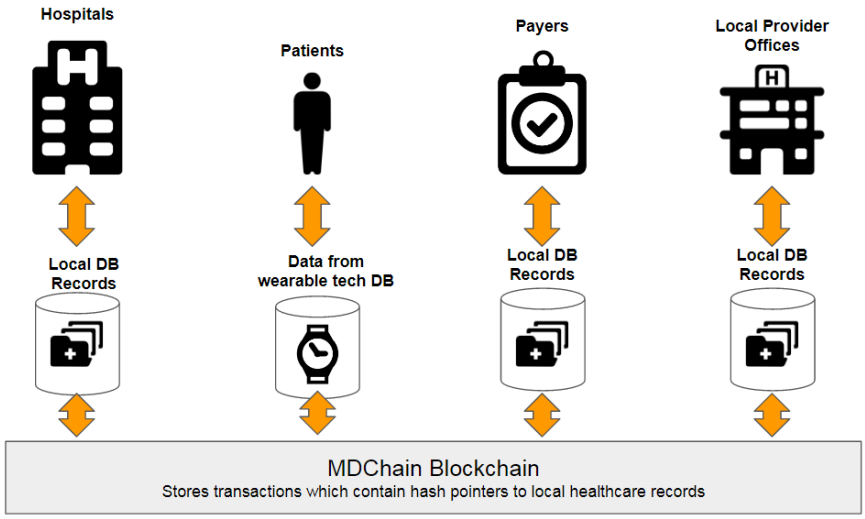


Figure 1.3: Diagram of Healthcare Stakeholders and Local Records in MDChain

further demonstrate the feasibility and efficacy of blockchains as a technological solution to contribute to the advancement of the healthcare industry.

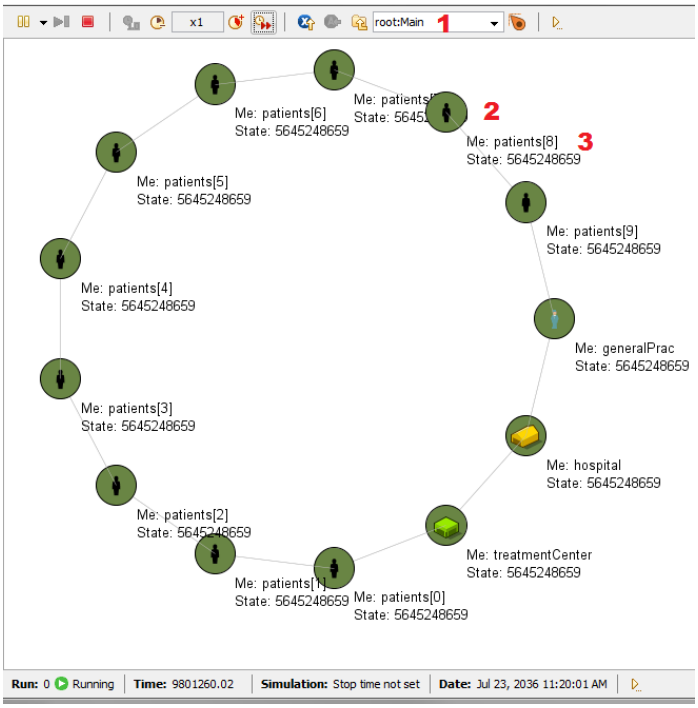


Figure 1.4: Main Interface for MDChain Simulation Run

Literature Review

This literature review is broken into sections that describe modeling and simulation related works, healthcare related works, and blockchain, cryptocurrency, or Bitcoin related works. Some resources are multidisciplinary in nature, such as the application of modeling and simulation to healthcare networks, application of blockchain towards healthcare, or application of modeling and simulation to blockchain networks.

2.1 Simulation

Heath [2] reveals insight as to the purpose of developing models and simulations. Simulations have the ability to give valuable information without needing to completely replicate a true system. The simulation is used as a representation of the true system. Heath identifies three roles that a simulation may serve depending on the level of understanding about the real system, as aforementioned above in Figure 1.1.

If a system is well understood, a predictor model can be built such that a simulation will behave in a precise manner, providing realistic outputs when given realistic inputs. As less information is known regarding the behavior of a system, the role of simulation moves toward the generator end of the spectrum. Generator models can be used to hypothesize about the underlying structure and resulting behavior in a complex, poorly understood system. Agent based modeling is particularly useful as generator models; by observing, modeling, and then simulating individual behaviors for agents, one can try to

replicate emergent properties from the real world from the interactions that occur between the simulated agents.

2.1.1 AnyLogic Modeling Software

AnyLogic modeling software is used to build models using discrete event, agent based, and system dynamics tools [6]. The models in this thesis utilize discrete event and agent based modeling tools. These models contain parameters, variables, functions, events, state charts, and distributions, and are further described below:

- Parameters - Parameters are used to store static attributes related to an agent. Examples include a person's gender, or the name of an entity.
- Variables - Variables are used to store dynamic attributes related to an agent. Examples include a person's age, or the revenue received by a corporation.
- Functions - Functions are used to define custom complex behaviors that an agent can perform. Examples would include the construction of a block by a Bitcoin miner, or sending a message to another agent.
- Events - Events are used to schedule actions at scheduled moments of time. Events can occur once, or occur multiple times with a custom defined distribution for inter-event times.
- State Charts - State charts exist within an agent and represent the various states an agent can take. Agents can have more than one state chart, and state charts can have nested state charts. Examples would include the health status of a patient, or state of an item as it flows through a supply chain.
- Distributions - Probability distributions can be used in functions or to define the reoccurrence rate of an event. AnyLogic contains a variety of built-in probability

distributions, including exponential, binomial, uniform, Poisson, logistic, and more. AnyLogic can also process input data sets and create custom probability distributions that replicate the data.

2.2 Healthcare

Patil and Seshadri [7] discusses the variety of security-related and privacy-related risks that electronic health records pose to both corporations and consumers. Increasing amounts of data (referred to as "big data") are being harvested as medical records become digitized. This abundance of information can ideally be used to improve care process, delivery, and management while also lowering costs. The data may go beyond clinical in nature, and may also include social, financial, physical, genomic, and psychological information. Much of this data is housed in data centers with varying levels of security. Even if a data center is compliant with the Health Insurance Portability and Accountability Act ("HIPAA"), this does not guarantee patient record safety. The authors state concerns surrounding the governance of healthcare data, ability to perform real-time security analytics, and maintaining adequate levels of privacy for patients. It is suggested that technological breakthroughs will be needed to properly address these issues.

van Panhuis et al [8] covers similar topics, but with more emphasis on sharing of data for the benefit of public health. The health data being stored could be used to further inform health policy decisions made at the local, national, and global level. The authors develop a systematic framework to explain the challenges involved in sharing this data. Challenges were either technical, motivational, economic, political, legal or ethical in nature. Some specific challenges include the lack of financial incentive for data providers to share information between themselves, a technological inability to preserve data, ownership and copyright issues, lack of reciprocity when sharing information, among other key challenges.

Care coordination is one activity that may enable improved treatment for patients and reduced costs, but depends upon the resolution of the key issues mentioned above that data sharing initiatives face. Schultz and McDonald [9] aims to develop a consensus as to the definition of care coordination by reviewing 57 unique definitions. The authors identify 5 major themes when viewing care coordination definitions:

- Numerous healthcare participants are involved in care coordination
- A degree of interdependence exists between the participants and their activities
- Participants require knowledge of others' roles and resources
- Requires the sharing and exchange of relevant data
- Ultimately aims to facilitate effective delivery of healthcare

Care coordination is still an evolving field, and the authors believe that the development of conceptual models can contribute to generating evidence as to what practices work best in care coordination, and what models contribute to the improvement of quality of care.

2.2.1 Healthcare Modeling and Simulation

Kanagarajah et al [1] discusses in further detail why agent based modeling is suitable for healthcare modeling. The authors describe healthcare as a generally complex adaptive system, and demonstrate the nonlinear behaviors and complexities of such systems through the use of simulation. A hypothetical simulation of an emergency department is also made.

Zeigler [10] focuses on the efficacy of using modeling and simulation to support care coordinating and fee-for-performance health payer models. US healthcare currently involves various uncoordinated systems that promote independent pricing, with little regard for quality. The authors also develop a new generic model using discrete event systems

specifications for the distributed tracking of individual patients that experience varying intervention pathways. This model stems from the philosophy that healthcare reform and care coordination can be modeled as a system-of-systems.

Barajas and Akella [11] also aims to leverage modeling and simulation to gain information surrounding a patient's health based on electronic health records. The model dynamically estimates a patient's mortality rate over time by incorporating varying types of patient related data, such as lab results, vital readings, provider notes and more. 15,000 electronic health records were used to test the model, and revealed that the model was able to detect an increase in the probability of mortality before a mortality event occurs.

2.3 Blockchain, Cryptocurrency and Bitcoin

"Bitcoin: A peer-to-peer electronic cash system" [3] was a white paper released by a pseudonym "Satoshi Nakamoto" in 2008 that described a peer-to-peer electronic version of cash, known as "Bitcoin". Bitcoin enables payments to be made between parties without reliance upon a third party, such as a financial institution, for clearing. The network utilizes a distributed cryptographic ledger, now widely referred to as a "blockchain", to keep track of all transactions. The ledger is secured by rewarding the distributed participants with bitcoins for behaving honestly. The combination of blockchain technology and a digital currency has given rise to a new category of assets referred to as "cryptocurrencies". More information as to how Bitcoin functions is provided in subsection 3.1.1.

Decker and Watternhofer [12] describes how the distributed members of the Bitcoin network are connected in a peer-to-peer fashion. Transactions and blocks that are mined are propagated through multiple hops to eventually be stored on all nodes on the network; all nodes broadcast their current state to their neighbors on a best-effort basis. When a block is added to the blockchain by a miner, it typically takes roughly 40 seconds for that block to be broadcast to at least 90% of the network. The authors state that this reliance

upon block propagation can cause delays in transaction clearing, and also poses a threat to the network itself by creating an advantage for attackers who create smaller blocks.

Since Bitcoin's creation, thousands of additional cryptocurrencies have also been created. Ethereum is the second largest cryptocurrency by market capitalization as of February, 2018. "Ethereum: A next-generation smart contract and decentralized application platform" [13] describes the idea for Ethereum. Essentially Ethereum acts as a decentralized computational network. Ethereum allows for code to be stored inside of Ethereum addresses, known as "smart contracts", that enable automated execution of instructions. While this functionality expands the use cases that Ethereum can address, it also creates for an increased attack surface to the network. Some of these attacks are further explored by Atzei et al [14].

Blockchains that do not contain an underlying cryptocurrency have also begun to emerge. One of these is "Hyperledger Fabric" [15] and is intended to be used in enterprise environments. In Hyperledger Fabric, all participants' identities are known, which removes the necessity for implementing a cryptocurrency to incentivize mining.

Rosenfeld [16] describes a protocol whereby certain bitcoins have additional meta data attached to them to allow them to represent other assets in a digital manner. This can be used to digitize real world assets on the Bitcoin blockchain. Asset digitization is further discussed in section 4.1.1.

Pilkington [17] writes an excerpt from a research book that studies the broad societal impact of digital technologies. The chapter surrounding blockchain provides a holistic overview of blockchain technology and suggests potential applications such as blockchain-based voting systems, disruptions to financial technology, logistical improvements for supply chains, digital identity providers and more. The work also notes that "a blockchain does not need to be a shared ledger, nor does it need to have a distributed consensus. It can be completely centralized as long as its data/state is externally verifiable and all data is immutable" and further describes the differences between public, private, and hybrid

blockchains.

2.3.1 Security and Scalability

Anyone who runs Bitcoin compatible software must abide by the community established consensus protocol. Bitcoin Core is one of the most well known software implementations of the Bitcoin protocol, and is open source [18]. Consensus as to the state of Bitcoin's blockchain is accomplished by a system known as *proof-of-work*. *Proof-of-work* involves scanning for a specific value that when hashed using SHA256, the hash begins with a certain number of zero bits. A hash function is a function such that the input data cannot be reverse engineered given a specific output of that data; every output for a SHA256 input is unique and indistinguishable from any related inputs. SHA256 is a specific implementation of a hash function designed by the National Security Agency [19].

Bitcoin utilizes a variety of underlying advanced cryptographic tools including block ciphers, Diffie-Helman key exchange, asymmetric cryptography, elliptical curve cryptography, digital signing, and more. Rosic [20] discusses these terms in greater detail.

Gervais et al [21] introduces a quantitative framework used to objectively compare *proof-of-work* blockchains in terms of security. The framework incorporates network layer parameters and evaluates their impact on the security of a blockchain-based system. The paper focuses on analyzing *double spend* attacks and *selfish mining* attacks. The *double spend* attack refers to the scenario when a user initiates two conflicting transactions such that the same bitcoin is spent twice, possibly undoing an original payment. The *selfish mining* attack refers to miners withholding block solutions to increase their relative rewards.

Baqer et al [22] discusses the impact that denial of service attacks can have on the Bitcoin network by empirically analyzing a spam transaction campaign that occurred in July, 2015. The authors utilized clustering methodology to classify transactions as spam. They conservatively estimated that at the peak of the spam campaign, up to 23% of transactions were classified as spam throughout a 10 day period.

Bonneau et al [23] provides rationale to the importance of research surrounding Bitcoin, as well as a systematic exposition on Bitcoin and other alternative cryptocurrencies. The paper discusses numerous proposed designs for cryptocurrencies. Some of the concepts discussed include alternative consensus models, varying currency distribution models, and key management tools.

Goldfeder and Bonneau [24] develop insights into multi-signature schema that can be leveraged in Bitcoin. Specifically, they propose a method whereby shared control of a wallet, secure bookkeeping, secure delegation of authority and two-factor security policies can be enacted.

Tschorsch and Scheuermann [25] discusses the Bitcoin protocol, provides an overview of *proof-of-work* and blockchains generally, discusses security issues and scalability issues related to digital currencies. The authors provide a comprehensive review of the Bitcoin field, as well as its characteristics and related works, while also suggesting future research directions. The authors believe it is uncertain if Bitcoin can retain its current state of robustness as the network scales, and as bitcoin mining rewards reside. If Bitcoin is to scale to higher transaction rates, system participants must be able to process the increased computational load, which may contribute towards network centralization. They conclude that Bitcoin and other cryptocurrencies are not yet fully understood, and provide for a highly interesting field of research.

Herrera-Joancomartí and Perez-Sola [26] provides a comprehensive review of Bitcoin's features and discusses the scalability issues that surround Bitcoin. The authors discuss payment channels as a method of off chain scaling. The authors also discuss new privacy issues that arise when using off chain scaling solutions.

Croman et al [27] explores fundamental and circumstantial choke points present in the Bitcoin protocol that may prevent Bitcoin from scaling to increased throughput while still retaining low latency. They identify that Bitcoin can only scale to 7 transactions per second given its current architecture, as opposed to Visa's peak rate of 56,000 transactions per

second. The authors suggest that modifications to the time between creation of blocks and the size of the data contained within blocks are first steps towards increasing throughput, but are not sufficient alone. Various scaling proposals are discussed; however, none of the alternative solutions have yet been proven to scale blockchains in a secure manner.

Lopp [28] analyzes the feasibility of Bitcoin scaling to billions of users by increasing the size of Bitcoin blocks and relying on simplified payment verification (SPV). SPV nodes in the Bitcoin network do not retain the full blockchain, and instead only store limited information from the full blockchain while trusting that this limited information is being provided in an honest manner from a full node. If the Bitcoin network were capable of performing 300 transactions per second, the author claims that operating a full node on the Bitcoin network would cost in excess of $2,500 for initial setup. The article states that roughly 98% of current full node operators would not be willing to pay more than $100 per month to maintain their node.

BitFury Group [29] was a paper written in 2015 that studies the pros and cons of a Bitcoin block size increase. The authors conclude that they believe the maximum block size should be increased in order to allow Bitcoin to continue scaling in the near term; however, they state that if the block size were immediately increased to 8MB from 1MB, then over 80% of node operators would be unlikely to be able to allocate sufficient resources towards node operation. Their analysis can be seen below in Table 2.1. The authors also explore alternative scaling proposals.

Poon and Dryja [30] is a white paper that describes a scaling solution for Bitcoin transactions that does not require all transactions be settled on the blockchain. This type of solution is referred to as an "off-chain" or "2nd layer" solution. The idea behind the Lightning Network is to utilize a technology called "payment channels" to allow users to connect in a network fashion through multiple payment channels. Transactions made through the Lightning Network would be instant, facilitate micropayments, payments across blockchains and more. They state that the Lightning Network, in combination with 133MB blocks, would

Table 2.1: Resource consumption by full nodes as the block size increases [29]

Characteristic	Scale factor	Block size, MB ($= N/2$)						
		0.5	1	2	4	8	16	32
Transaction throughput, tps	N	1.75	3.50	7.00	14.0	28.0	56.0	112
Number of txs in a block	N	1050	2100	4200	8400	16800	33600	67200
Blockchain storage per day, MB	N	72	144	288	576	1152	2304	4608
Blockchain storage per year, GB	N	26	51	103	205	411	821	1643
Transaction processing time, ms	1	0.33	0.33	0.33	0.33	0.33	0.33	0.33
Block verification time, s	$N + 0.09N \log_2 N$	0.07	0.15	0.33	0.71	1.51	3.23	6.86
Average bandwidth, kB/s	N	74	148	296	592	1184	2368	4736
Daily traffic, GB	N	6.2	12.4	24.8	49.6	99.2	198	397
Yearly traffic, TB	N	2.2	4.4	8.8	17.7	35.4	70.7	141
RAM usage, GB[1]	N	2	4	8	16	32	64	128
Immediately excluded nodes, %	n/a	0	20	40	75	90	95	95
Excluded nodes in 6 months, %	n/a	5	25	50	80	95	95	95

provide the ability for up to 7 billion people to open two channels per year and perform unlimited transactions within the channels.

Back et al [31] proposes another 2nd layer scaling solution dubbed *pegged sidechains*. The basic idea involves creating an alternative blockchain that has its own token that is pegged to Bitcoin in a 1:1 ratio. The peg is enforced by requiring a Bitcoin user to lock up their BTC on the Bitcoin blockchain while they convert it to the pegged sidechain currency. For example, perhaps a sidechain is built that is capable of 1,000 transactions per second. Users could transfer their BTC into the sidechain, make thousands of transactions, then transfer those tokens back into the Bitcoin blockchain.

Wuille [32] is a presentation by Bitcoin Core developer, Pieter Wuille, where he describes the benefits that *Segregated Witness* has in terms of blockchain scalability. *Segregated Witness* is a Bitcoin upgrade that allows transactions to be structured so that some of their data does not count towards the 1MB block size limit. This could allow Bitcoin blocks to contain as much as 4MB of total data depending on the amount of users utilizing the technology and the types of transactions being created. He also states that *Segregated*

Witness fixes technological issues that prevented Lightning Network and sidechains to be properly implemented. *Segregated Witness* was activated on the Bitcoin network in August, 2017.

"Bitcoin Cash - Peer-to-Peer Electronic Cash" [33] is the white paper that describes an alternative cryptocurrency called *Bitcoin Cash* that is created as the result of a Bitcoin hard fork. A Bitcoin hard fork occurs when the network splits into two versions due to incompatible rule changes. In the case of *Bitcoin Cash*, some users had a fundamental disagreement about implementing *Segregated Witness* and instead wanted to increase the Bitcoin block size to 8MB.

An upgrade proposal to the Bitcoin network, named *Segwit2x*, appeared in 2017 that would double the block size in November, 2017; however, the upgrade did not have consensus among the Bitcoin community. Implementation of *Segwit2x* would likely have split Bitcoin into two versions due to the hard fork: one with 2 MB blocks and one with 1 MB blocks with identical histories up until the moment of the hark fork. The upgrade was later cancelled by a select group which was developing the code for the upgrade [34]. Song [35] explores in greater depth the technological flaws that were present in *Segwit2x* that would have prevented it from executing properly.

2.3.2 Blockchain Modeling and Simulation

Previous work with simulations studied Bitcoin, blockchains, and cryptocurrencies. Many cryptocurrencies, including Bitcoin, feature a testnet; the testnet is a separate blockchain with valueless coins that can be used for testing [36].

Bornholdt and Sneppen [37] appeared in 2014 and models the demand of Bitcoin and other cryptocurrencies using a model derived from sociology, the Moran process. The general idea is that the value of a particular cryptocurrency can be estimated based off of its popularity. Their model specifically implements an environment of agents that can engage in investing, mining, and trading various cryptocurrencies.

Carlsten et al [38] appeared in 2016 and explores the necessity of the block reward for Bitcoin to operate correctly. The authors developed a model of the Bitcoin blockchain system with the goal of studying the incentive issues that surround Bitcoin mining, without necessarily being an accurate predictor of mining behavior in practice. They claim the results of the study are only made stronger by the presence of simple assumptions in their model, as it shows that undesirable behaviors occur despite the simple set of assumptions provided. The simulator was built using the C++ programming language.

Cocco et al [39] appeared in 2017 and utilizes agent based modeling to explore cryptocurrency markets and their future iterations. In their model Random Traders and Chartists transact with one another through the exchange of bitcoins. Their model was able to demonstrate some characteristics shared with true cryptocurrency markets, such as price absolute returns, and the number of bitcoins increasing over time. The model was written using the Smalltalk language.

Chen et al [40] appeared in 2017 and models smart contract execution through a decentralized network of nodes using an agent based model. The model is based on principles of game theory and agent based model analysis. The possibility for users to be prevented from manipulating smart contracts for their own malicious benefit is explored.

Terna [41] appeared in 2017 and is a thesis that aims to better understand the conditions that affect diffusion of a cryptocurrency through a network. The authors develop an agent based model and use clustering analysis to analyze the results. The model is derived from a disease spread model. Two versions of their model, a single layer and a multilayer model, were developed. The language used to code their program was called Netlogo.

Laskowski [42] appeared in 2017 and proposes the use of a model that mimics disease spread within a population (SIR model) to explore the development of practical participatory decision support systems. The model incorporates elements from agent based modeling, blockchains, smart contracts, and virtual reality. The blockchain enables the provenance and transparency of decision making within the model.

Yasaweerasinghelage et al [43] appeared in 2017 and uses architectural performance modeling and simulation tools to explore latency issues within blockchain-based systems. Latency is a fundamental issue in the design of blockchain-based systems and these tools support this fundamental analysis. Their results can be used to further discuss the trade offs involved in using blockchain-based systems that are related to security, performance, and cost.

Norgaard et al [44] appeared in 2018 and utilizes agent based modeling to explore the difference in structure between virtual black markets that leverage cryptocurrencies as payment methods versus traditional black markets. The motivation of the study was to better understand the type of network architecture that emerges from the individual agent behavior.

2.3.3 Blockchain and Healthcare

Healthcare is one industry that may pose as an opportunity for blockchain technologies. In August, 2016 the Office of the National Coordinator for Health Information Technology operated an ideation challenge, the Health IT challenge, where 15 white papers were awarded cash prizes for exploring the use of blockchain in health IT and health-related research [45].

One of the participants in the Health IT challenge was IBM's Global Business Services Public Sector Team, who wrote "Blockchain: The Chain of Trust and its Potential to Transform Healthcare – Our Point of View" [46]. They believe that blockchains could impact pain points within healthcare that include interoperability, accessibility, data integrity, privacy, security, healthcare delivery models, cost, fraud, process complexity, consumer engagement, contracting, and compliance. They present a list of 16 potential use cases, and explain in depth 3 of the use cases: healthcare pre-authorization payment infrastructure, counterfeit drug prevention and detection, and distribution of clinical trial results.

Azaria et al [47] is another entrant in the Health IT challenge and describes a proto-

type healthcare blockchain-based system named "MedRec". MedRec utilizes blockchain to handle electronic health records. MedRec provides authentication management, confidentiality, accountability and ability to share data with researchers. They incentivize stakeholders in the healthcare ecosystem, such as public researchers and public health authorities, to secure the network by performing *proof-of-work* in exchange for having access to aggregated, anonymized healthcare data.

Rabah [48] is a review paper that outlines the various challenges and opportunities that blockchain applications face in the healthcare industry. The authors believe that blockchain technology will advance efforts to improve patient care, treatment efficacy, security, and reducing costs. They suggest that electronic medical record management will be made more efficient, disintermediated, and secured through blockchain technology.

Kuo et al [49] provides a comprehensive overview of biomedical and healthcare applications that could be developed using blockchain technology. They view decentralized management of records, immutability of audit trails, data provenance, availability of data, security, and privacy as benefits to be gained from the implementation of blockchain technology over traditional distributed database management systems. They describe 4 potential use cases:

- Improved Medical Record Management
- Enhanced Insurance Claim Process
- Accelerated Clinical/Biomedical Research
- Advanced biomedical/health care data ledger

The authors also discuss potential challenges that blockchain applications may face in a healthcare environment relating to transparency, confidentiality, speed, scalability, and resistance to malicious actors.

Dufel [50] discusses blockchain, alongside other peer-to-peer technologies, and their application towards healthcare. BitTorrent, a peer-to-peer file sharing system, is discussed,

as well as distributed hash tables. The authors suggest that only a combination of blockchain, distributed hash tables, and BitTorrent would able to effectively create a peer-to-peer health information exchange system. A breakdown of the role that each technology provides can be seen below in Table 2.2.

Table 2.2: Role of peer-to-peer technologies in providing Health Information Exchange services [50]

	Implementing Technologies		
Services	**Blockchain**	**DHT**	**BitTorrent**
Master Patient Index	NO	YES	NO
Provider Registry	NO	YES	NO
Data Search	NO	YES	NO
Data Storage & Retrieval	NO	NO	YES
Direct Messaging	NO	YES	NO
Event Bus	NO	YES	NO
Data Marketplace	YES	NO	NO
Consent & Compliance	YES	NO	NO

Zhang et al [51] provides metrics that can be used to evaluate the feasibility, intended capability, and compliance of a blockchain-based application in the healthcare space. A summary of their suggested metrics is provided below in Table 2.3.

Zhang [52] aims to fill the gap of information regarding software architectural styles and recommendations for constructing blockchain-based healthcare applications. In order to do this, they discuss challenges in addressing healthcare interoperability, develop a case study of a blockchain-based healthcare application they are constructing, and suggest how using familiar software patterns can directly address some challenges. The application developed by the authors uses the Ethereum test blockchain to provide a web portal that

Table 2.3: Summary of metrics for evaluating blockchain-based applications in healthcare [51]

	Assessment Metric
1	Entire workflow is HIPAA compliant
2	Framework employed needs to support Turing-complete operations
3	Support for user identification and authentication
4	Support for structural interoperability at minimum
5	Scalability across large populations of healthcare participants
6	Cost-effectiveness
7	Support of patient-centered care model

patients can use to access and update medical records and fulfill prescription requests.

"Mercantis" [53] was a winning project proposed at the Distributed Health Hackathon during 2017 that leveraged Ethereum smart contracts to create a decentralized marketplace for healthcare data. The proposed system enables patients to control their medical information, and monetize it if they chose to do so by selling it to researchers; researchers can have improved access to data by leveraging the decentralized data market.

SimplyVital Health [5] is a startup company founded in 2016 that aims to leverage blockchain technology in the context of healthcare. Their publicly available web materials indicate that they are working on two products: "ConnectingCare" and "Health Nexus". ConnectingCare is a care coordination tool used by providers that leverages blockchain technology to create a secure audit trail. Health Nexus plans to be its own healthcare focused blockchain with an underlying cryptocurrency token called "HLTH".

[54] is an online GitHub repository of existing healthcare related blockchain projects. Figure 2.1 provides an overview of healthcare related blockchain projects that are referenced in the repository.

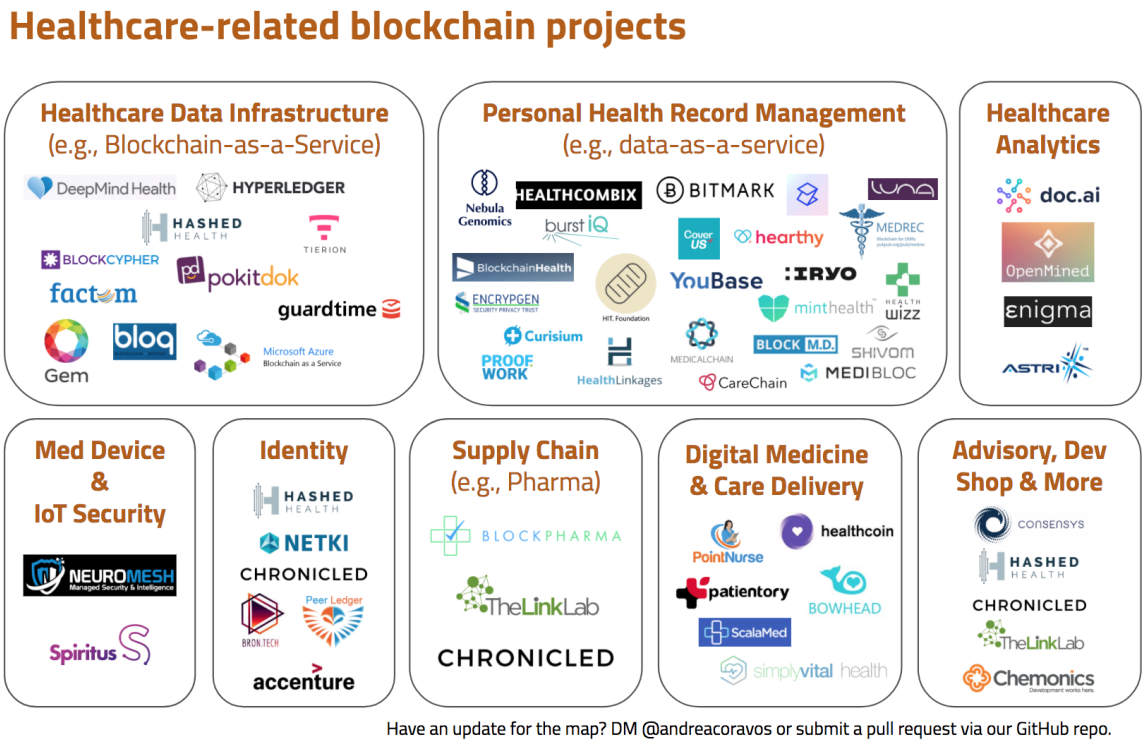


Figure 2.1: Healthcare-related Blockchain Projects [54]

Using agent-based modeling to understand Bitcoin and blockchain

In 2009, a first-of-its-kind decentralized digital currency program, called “Bitcoin”, was released. Bitcoin utilizes an ongoing immutable cryptographic chain of transactions that acts as a decentralized peer-to-peer ledger. This underlying distributed database has been referred to as “blockchain” technology, and holds the potential to revolutionize the way that value or information can be moved between parties. Blockchains typically possess technical features that enable traceability and auditability of transactions, built-in cryptocurrencies, and public key infrastructure for identity management. Due to their distributed nature, blockchains and blockchain-based cryptocurrencies can be maintained without a central authority, which minimizes the risk associated with a single point of failure and enables censorship resistance; however, decentralization introduces scaling and performance draw backs. We model the Bitcoin blockchain using an agent-based model in AnyLogic. The model mimics important aspects of the Bitcoin network and includes miners, users, transactions, and a cryptographically connected chain of blocks to act as a blockchain. We use this model to demonstrate blockchain mechanics, and to explore and forecast aspects of blockchains including resource requirements, and the effects from adopting varying protocol implementations. The model predicts that Bitcoin transaction fees will rise exponentially by early 2018 if scaling solutions are not implemented.

3.1 Introduction

A white paper describing a peer-to-peer electronic cash system, dubbed "Bitcoin", was published by a pseudonym, "Satoshi Nakamoto", in 2008 [3]. The Bitcoin network was subsequently started on January 3, 2009 [55]. Bitcoin functions as a decentralized digital currency program, also known as a "cryptocurrency", and is the first of its kind. Bitcoin enables its users to send payments to one another ("transactions"), without a third party, in an irreversible manner, by leveraging a series of digital signatures commonly referred to as a "blockchain" or "blockchain technology". Since its launch, Bitcoin has reached prices of up to $20,000 and has motivated the release of hundreds of alternative cryptocurrencies. The combined market capitalization of cryptocurrencies exceeded over $750 billion in November, 2017 [4]. Since then, cryptocurrencies have experienced volatile price action, dropping to as low as $330 billion in early February, 2018, and returning back to a combined market capitalization of $500 billion in late February, 2018.

Modeling and simulation can be a valuable tool in better understanding how systems function, and can be applied to better understand the mechanics of a blockchain. The model constructed in this study is an abstract representation of how the Bitcoin protocol functions. This chapter aims to further explore the Bitcoin blockchain by developing an agent based model of the Bitcoin network. It is hypothesized that one can replicate select behaviors of the real Bitcoin network through the development of agent based models. Using the model to simulate realistic scenarios, we are able to explore current issues of importance to the Bitcoin community, and develop insight into future work to be done.

3.1.1 How Bitcoin Works

First, to define some initial terminology, "Bitcoin" refers to the network protocol while "bitcoin" ("BTC") refers to the electronic coin currency itself that is used as a unit of account in the Bitcoin network. The specific mechanics of Bitcoin are described in the

original white paper [3]. In this section, Bitcoin is briefly explained in an abstract way. The mechanics of Bitcoin are also demonstrated in the methods section.

Bitcoin "users" are anyone who uses the Bitcoin blockchain to initiate bitcoin transactions. Users can include miners, individuals sending money to friends or family, investors, and more. To use the Bitcoin network, users must either download the entire blockchain and run Bitcoin software locally, or they may use third party services that run their own Bitcoin software. These third parties may include Bitcoin wallet providers, Bitcoin exchanges, merchants, and more. Users are connected in a peer-to-peer manner, such that any user in the Bitcoin network can communicate with any other user through a series of hops.

Users create transactions whenever they want to send bitcoins to another user. Bitcoin is a "push-only" system, meaning transactions can only be pushed by a sender, and not pulled by a receiver. This is contrary to something like a credit card that authorizes a merchant to pull funds from the credit card. Whenever a user creates a new transaction or learns about a new transaction from a neighbor, they broadcast it to their connected peers. These new transactions remain in an unconfirmed state inside varying users' "memory pool" until the transaction is confirmed by being included in a block by a Bitcoin miner. The local memory pool maintained by each user can differ between users, and is an asynchronous store of transaction information.

Blocks of transactions are added to the blockchain by specialized nodes on the network, dubbed "miners", who compete to create ("mine") the next block in what is called a *proof-of-work* system. Miners can also be thought of as transaction validators or processors. The *proof-of-work* system essentially functions as a lottery to award the next block where the number of tickets you have is directly correlated to the amount of computing power you have. A new block is awarded every 10 minutes on average, following an exponential distribution. *Proof-of-work* is also used to reach consensus as to the state of the Bitcoin network. If there are ever two conflicting blockchains, the blockchain with the highest amount of *proof-of-work* will be treated as the real version.

The creator of each block is rewarded a small amount of bitcoin, in order to incentivize honest participation in the mining process. This results in the total supply of BTC expanding over time. The expansion rate is predictable and decreases over time, eventually leading to a hard limit of 21 million BTC being created by the year 2140.

To reduce the variability of income in this lottery-esque system, miners often join together in mining pools, whereby they proportionally split the revenues from block mining rewards. For example, if a mining pool consists of 3 miners with 30, 20 and 10 units of computing power, and any single miner in that pool discovers the next block, the rewards are typically distributed proportionally as 50%, 33% and 17% respectively to each member based on units of computing power; however, this does lead to a degree of centralization and if one single mining pool possesses a majority of network computing power, this creates a vulnerability to a 51% attack [25].

The Bitcoin blockchain is an immutable, ever-growing record of every transaction that has occurred on the network since it began. The blockchain is composed of a cryptographically connected chain of blocks of information, as seen in Figure 3.1 below; each block contains a set of transactions made by users on the network, the cryptographic hash of the previous block, and other related information.

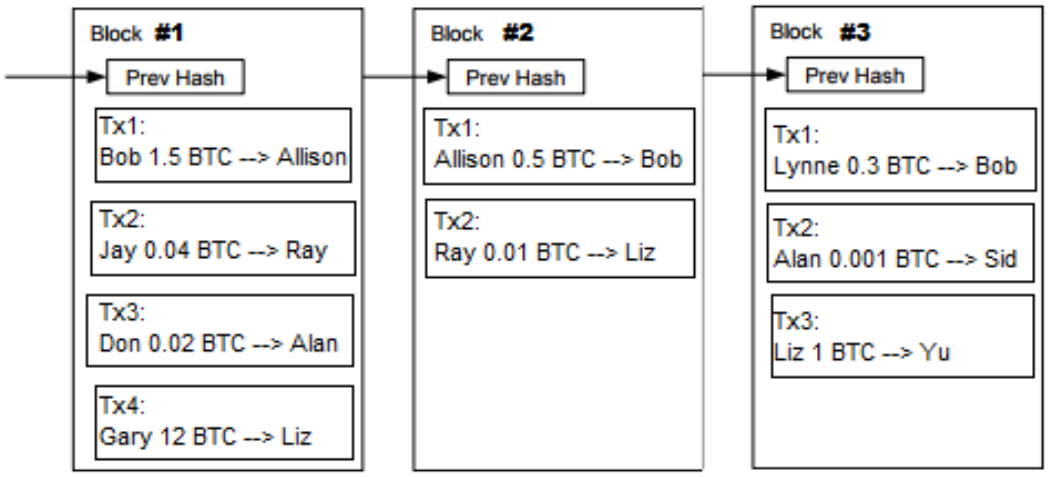


Figure 3.1: Representation of the Bitcoin Blockchain

The blockchain maintained by every user on the Bitcoin network should be the same, and acts as a synchronous data store. A duplicate of the blockchain is held by every node on the network, so if a transaction or block is detected as invalid, the rest of the network will reject that copy of the blockchain.

3.1.2 Scalability

Since every transaction must be recorded on the blockchain, and the blockchain itself must be replicated across every participating node, scalability becomes a concern for any blockchain-based system. The fewer users that are able to participate in a blockchain-based system, the more centralized it becomes. When the blockchain is more centralized, and is thus stored on fewer nodes, then the blockchain becomes more vulnerable to attack.

A long debated topic amongst the Bitcoin community surrounds how to properly scale Bitcoin to handle higher transaction throughput while still retaining key properties of network decentralization. Currently, Bitcoin blocks feature a maximum size limit of 1 MB. Research on the scaling debate can be found dating back to as early as 2015, such as the white paper written by the BitFury Group in September, 2015 [29], and also coincides with the time period when 1 MB blocks were beginning to be mined. Since then, a number of proposals have emerged to address the scalability issue that include modifications to the maximum block size, block time interval, micropayment channels [30], and sidechains [31]; however, only one major change has been added that addresses the block size capacity issue, named *Segregated Witness*.

When an upgrade is suggested in the Bitcoin community that results in software incompatability, this is dubbed a "hard fork", and if implemented, results in a break of network consensus. Contrarily, an upgrade that does not break compatibility is dubbed a "soft fork". In August, 2017, *Segregated Witness* was added to the Bitcoin network via a soft fork, which essentially provides a discount on transaction data of 75% for *Segregated Witness* style transactions. *Segregated Witness* also enables micropayment channel

and sidechain technologies to be developed more easily [32].

An upgrade proposal named *Segwit2x* also appeared in 2017 that would double the block size in November, 2017; however, the upgrade did not have consensus among the Bitcoin community. Implementation of *Segwit2x* would likely have split Bitcoin into two versions due to the hard fork: one with 2 MB blocks and one with 1 MB blocks with identical histories up until the moment of the hark fork. The upgrade was later cancelled by a select group who was developing the code for the upgrade [34]. Also around this time a hard fork that featured an 8 MB block increase without *Segregated Witness* activation was also executed, resulting in a separate cryptocurrency called *Bcash* or *Bitcoin Cash* [33].

3.1.3 Simulation Efforts for Bitcoin and blockchain

Simulations are valuable tools for studying how a system behaves without interfering with real world activity. Models are used in varying roles, ranging from being a precise predictor of real world behavior, to being a generator of new hypotheses to support understanding of system behavior [2]. Agent based models specifically can be beneficial when exploring systems that involve complex behavior among multiple entities [2]. The simulation presented in this work serves as more of a mediator model, as it provides further understanding of how Bitcoin functions while also making some specific predictions. Since blockchain networks require multiple nodes and resources to participate, testing and studying a live blockchain network can be difficult. This creates an opportunity for the application of simulation.

Previous simulation work has been performed to study Bitcoin, blockchains, and cryptocurrencies. Many cryptocurrencies, including Bitcoin, feature a testnet; the testnet is a separate blockchain with valueless coins that can be used for testing [36]. Some models have been developed to analyze the security of the Bitcoin blockchain [21, 38]. Agent based modeling has also been used to study the economics of cryptocurrency markets [37, 39, 44], decentralized execution of smart contracts [40], and other blockchain related topics [41, 42].

3.2 Methodology and Model Design

An agent-based model of the Bitcoin blockchain was developed to analyze how resource consumption scales over time, how participants interact with each other, and to be able to visualize these interactions. The simulation was developed in the AnyLogic modeling software, which is written in Java and was further described in subsection 2.1.1. The source code for the model can also be found at https://github.com/champbronc2/BitcoinAnyLogic.

The model attempts to replicate behaviors of the real Bitcoin network without the complete implementation of the Bitcoin protocol itself. By reducing the granularity of the model, simulations can be carried out for long periods of time with relatively low resource consumption compared to creating and running a Bitcoin testnet network [36]. The Bitcoin blockchain was chosen as a specific implementation to model, since it was the first public implementation of a blockchain, and also has the highest availability of data for model validation. The model includes a GUI for manipulation of select initial parameters, visualization of agent interactions, and environment statistics.

In addition to model validation, two exploratory experiments were performed to further explore the effects of varying block sizes, varying block times, and *Segregated Witness* activation. Validation includes a comparison of the total number of transactions, total blockchain size, total bitcoin supply and bandwidth consumption over time relative to the real Bitcoin network. An additional exploratory experiment was performed involving four unique configurations of *Segregated Witness* activation, maximum block size, and average time between blocks.

3.2.1 Agents, Parameters and State Variables

The model consists of 5 agent types: main, users, miners, transactions, and blocks. The main agent type contains a population of users, miners, transactions, and blocks. Any

parameters, variables, state charts, functions, events, or custom distributions that belong to each agent are described below in Table 3.1.

Table 3.1: Breakdown of model components by agent type

	Main	Users	Miners	Transactions	Blocks
Parameters	• initialSupply • initBlockSize • initTxNum • initMemPool • initialBlockHeight • blockTime • maxBlockSize • startingBlockReward • halvingRate • useSegwit • hardforkDelay • newMaxBlockSize	• segwitDelay	None	• amount • size • segwit • feeSizeRatio • toWhom • fromWhom	• btcAmount • transactions • fees • blockSize • blockHeight • timeStamp • hash • coinbase • segwitTransactions • segwitSize
Variables	• txRate • blockHeight • blockReward • currentMaxBlockSize	• balanceBTC	• miningPool • balanceBTC • hashRate	• fee	None
Functions	• selectWinner • buildBlock • addBlock • payMiners • attack • validateChain	None	None	None	None
Events	• updateTxRate • hardfork • recordData	None	None	None	None
State Charts	• blockchainState	• userState • segwitState	• minerState	• txState	None
Distributions	None	• sizeDistribution	none	None	None

Main

Parameters are created for the initial supply of BTC (initialSupply), initial blockchain size in kilobytes (initBlockSize), initial number of confirmed transactions in the blockchain (initTxNum), initial number of unconfirmed transactions in the memory pool (initMemPool), initial number of blocks mined (initialBlockHeight), average time between mined

blocks in minutes (blockTime), maximum block size in bytes (maxBlockSize), starting block reward in BTC (startingBlockReward), the block reward halving rate in number of blocks (halvingRate), indicator for activation of *Segregated Witness* in August 2017 as a boolean (useSegwit), and the delay in days (hardforkDelay) and new block size in bytes (newMaxBlockSize) if a *Segwit2x* hard fork occurs.

Variables are created for the average transaction rate per day per user (txRate), current number of blocks mined in the blockchain (blockHeight), current block reward in BTC (blockReward), and the maximum size of the next block in bytes (currentMaxBlockSize).

Functions are created to select the next mining pool to mine a block (selectWinner), to build a block from transactions (buildBlock), to add a block to the blockchain (addBlock), to pay the block reward and transaction fees to the miners (payMiners), to simulate a modification of the blockchain (attack), and to simulate the repair and validation of the blockchain (validateChain).

An event is created to update the transaction rate variable once each day (updateTxRate), to initiate a hard fork to change the current maximum block size (hardfork), and an event to record historical simulation data including, but not limited to, the current time, number of confirmed transactions, floating bitcoin supply, blockchain size and bandwidth usage (recordData).

A state chart is created that alternates between states of waiting for a block to be mined, and a block being mined (blockchainState).

Users

Users have a variable for their balance of bitcoins (balanceBTC), and a parameter determining their delay until adopting *Segregated Witness* (segwitDelay).

Users have a state chart (userState) whereby they can transition between states of waiting, receiving a transaction from a miner or user, and sending a transaction to another user. Users also have a second state chart (segwitState) that describes whether they have

adopted *Segregated Witness*.

A distribution of historical Bitcoin transaction sizes also exists within the user (sizeDistribution), and is used to generate random transaction sizes in bytes.

Miners

Miners have a variable to represent the pool that the miner belongs to (miningPool), the balance of bitcoins held by the miner (balanceBTC), and the computing hash rate of the miner (hashRate).

The miner's state chart (minerState) allows the miner's state to alternate between waiting to mine a block, receiving revenue from a recently mined block, and upgrading their hash rate by spending bitcoins with a transfer of BTC to a random user.

Transactions

Transactions have parameters for the amount of bitcoins transacted in BTC (amount), the size of the transaction in bytes (size), the usage of *Segregated Witness* as a boolean (segwit), the fee rate paid in terms of satoshis per byte (1 satoshi is one hundred millionth of a single bitcoin) of data (feeSizeRatio), to whom (toWhom) and from whom (fromWhom) the transaction is being made.

Transactions have a variable for the the total fee paid for the transaction in BTC (fee).

The state chart (txState) only has a state of unconfirmed. Once an individual transaction is confirmed, or removed from the memory pool after a timeout, the individual transaction agent is destroyed.

Blocks

Each block contains parameters to describe the number of bitcoins transacted in the block (btcAmount), the number of transactions (transactions), total fees included in all transac-

tions in the block in BTC (fees), total size of the block in kilobytes (blockSize), the height of the block (blockHeight), the time the block was found (timeStamp), a SHA256 hash of the previous block's data (hash), the mining reward for that block in BTC (coinbase), the number of *Segregated Witness* transactions included in the block (segwitTransactions) and the size of the data for those *Segregated Witness* transactions in the block in kilobytes (segwitSize).

3.2.2 Initialization

When the simulation is started, a window is displayed that contains basic information about the simulation, and the ability to toggle certain parameters. A screen shot of this window can be seen below in Figure 3.2. The user may toggle the start date between January 1, 2009 and July 31, 2017, the maximum block size, the average block time, the activation of *Segregated Witness*, and the ability to specify a hard fork to increase the maximum block size to some new specified size on a specified date. The "Run" button can be pressed to begin the simulation. 30 miner agents and 350 user agents are generated at initialization and remain constant throughout the life of the simulation.

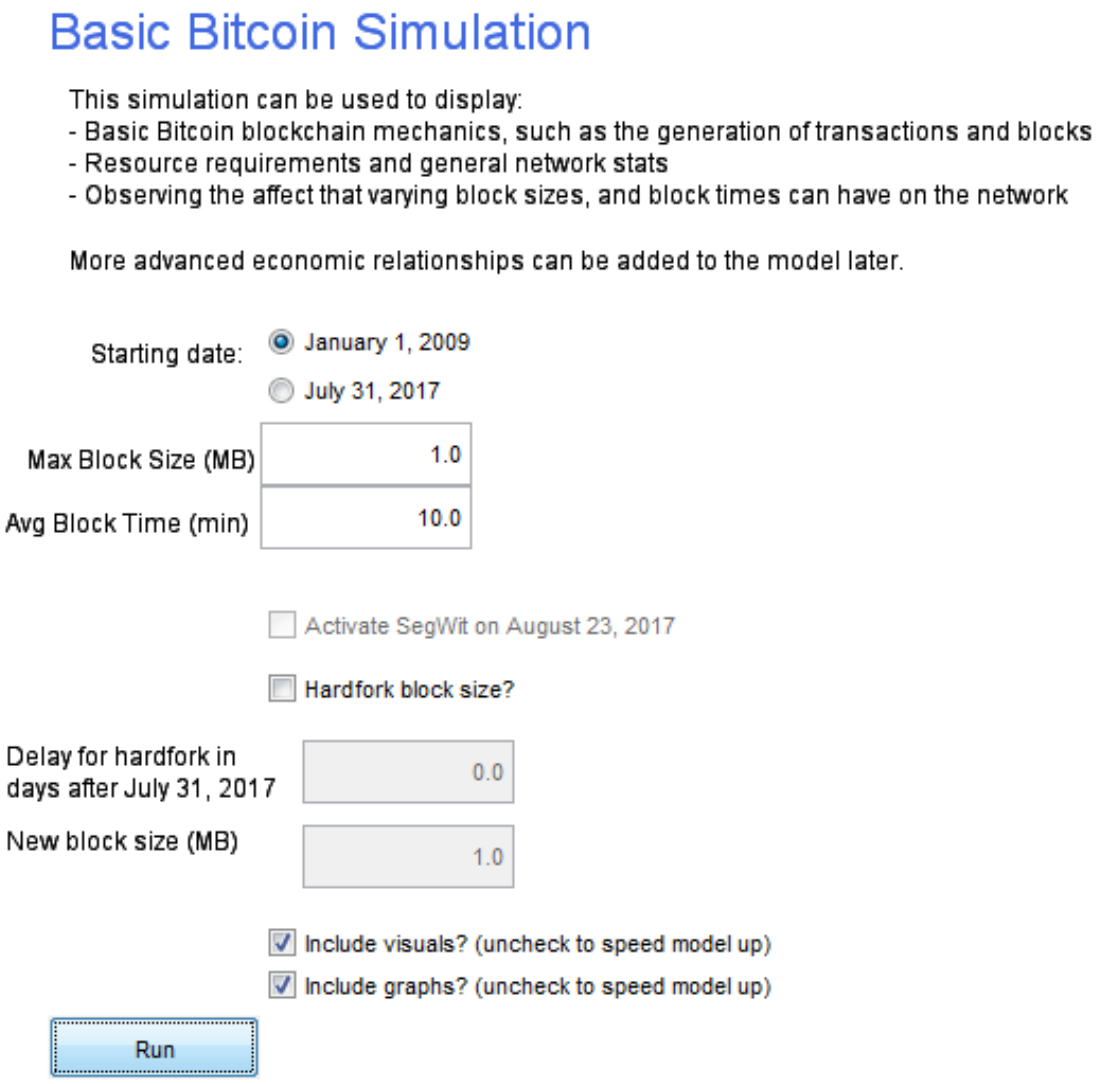


Figure 3.2: Simulation Initialization GUI

Other parameters cannot be toggled, and are automatically determined at model start up. The default parameters for each start date are displayed in Table 3.2 below. The initial parameters for the start date of July 31, 2017 were approximated using true values in the Bitcoin network on that date [55].

	January 1, 2009	July 31, 2017
blockTime (minutes)	10	10
maxBlockSize (bytes)	1000000	1000000
halvingRate (blocks)	210000	210000
startingBlockReward (BTC)	50	50
initialBlockHeight	0	478495
initialSupply (BTC)	50	16482000
initBlockSize (kilobytes)	0.2	126721000
initTxNum	1	243480000
initMemPool	1	5000
startDelay (days)	0	3125
useSegwit	FALSE	TRUE
hardforkDelay (days)	0	0
mewMaxBlockSize (bytes)	1000000	1000000

Table 3.2: Parameter values on initialization for both start dates

3.2.3 Processes and Interface Overview

Once a simulation run begins, a new window will be presented, and is shown in Appendix A in Figure A.1. The left panel, in red border, provides a visual representation of how the Bitcoin network behaves and contains 6 items labeled 1L-6L (described below); the right panel, in blue border, displays charts and statistics related to the simulation and contains 9 items labeled 1R-9R (described below).

Item 1L displays the number of blocks mined on the network so far, which is equivalent to the height of the blockchain in number of blocks.

Item 2L is a button “Modify Chain” that triggers the “attack” function which simulates an attempt to modify a previously mined block; specifically, the button attempts to increase the coinbase mining reward to 100 BTC in the third most recently mined block. When the next block is mined, the chain will fail validation, as the SHA256 hash contained within the block following the modified block will not match the SHA256 hash of the modified block itself. The user is then shown details regarding the error, and asked if they want to repair the blockchain in the dialog box shown below in Figure 3.3. This simulates the retrieval of the correct block from a peer by replacing the invalid 100 BTC coinbase reward with the valid reward.

Item 3L displays the raw data contained in the past 3 consecutive mined blocks. This

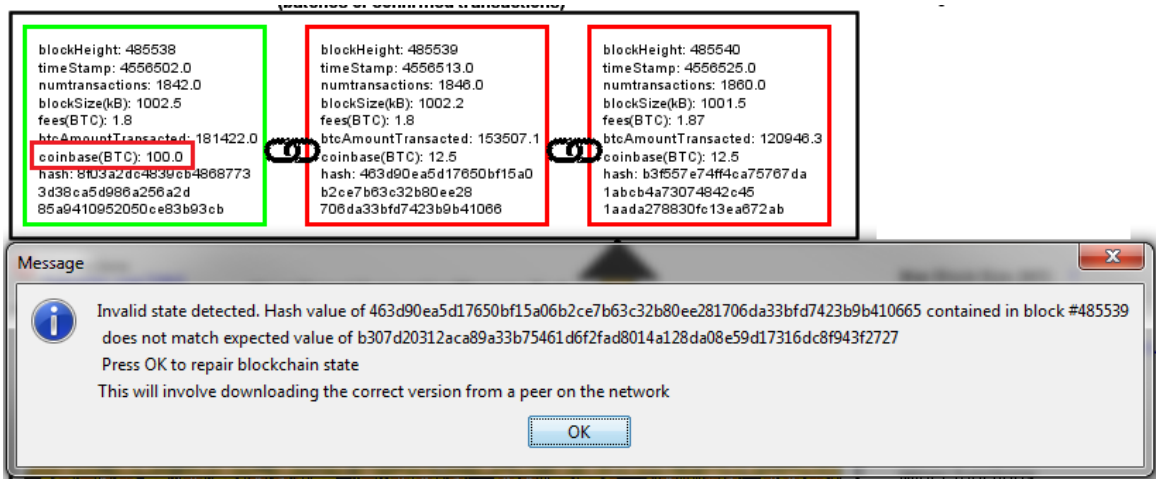


Figure 3.3: Dialog Box for Blockchain Validation and Repair

is done by converting the 3 most recently created block agents into strings. If the border of a block is red, the hash value contained within that block does not match the SHA256 hash of the previous block, thus deeming the block invalid. If the border is green, the block is valid.

Item 4L displays the current state of the Bitcoin network via simple statistics and a state chart, which alternates between waiting for a block to be found, and mining a block. Below the state chart, the current maximum block size, mining block reward, supply inflation rate, transactions per second being added to the memory pool and the transactions per second being confirmed and added to the blockchain are shown.

Item 5L is a visual representation of unconfirmed transactions in the local memory pool, which are representative of transaction agents in an unconfirmed state. As transactions are confirmed and included into blocks, or generated by users, they are removed from or added to the unconfirmed transaction memory pool respectively.

Item 6L is a visual representation of the Bitcoin peer-to-peer network. User and miner agents are displayed, as well as their direct connections to peers. Everyone is connected by a single network; any user or miner can communicate with any other user or miner through a series of hops. The legend shows that green is associated with a user sending a transaction, blue is associated with a user receiving a transaction, and red is associated with

a miner receiving revenue from their mining pool.

Items 1R through 8R display various sets of time series data. Item 7R is only visible if *Segregated Witness* is activated during the initialization of the model. Item 9R is a pie chart displaying the cumulative hash rates for miners belonging to each mining pool.

The main, miner and user agents each contain internal processes that control their behavior and interactions with other agents. These are described below. Block agents do not contain any internal processes. A graphical representation of the processes performed by each agent is provided as part of each description.

Main Processes

The main agent houses all other agents, and can be thought of as a local node. Every X minutes the main agent transitions from an idle state to one of mining the next block. X has an exponential distribution with an average time equal to the block time selected at initialization. First, the main agent runs a function to validate the current blockchain state by computing the SHA256 hash of each block and checking for consistency with the hash present in the preceding block. If any errors are found, the user is prompted to repair the chain state. This is referred to as the "Validation" state in Figure 3.4 below.

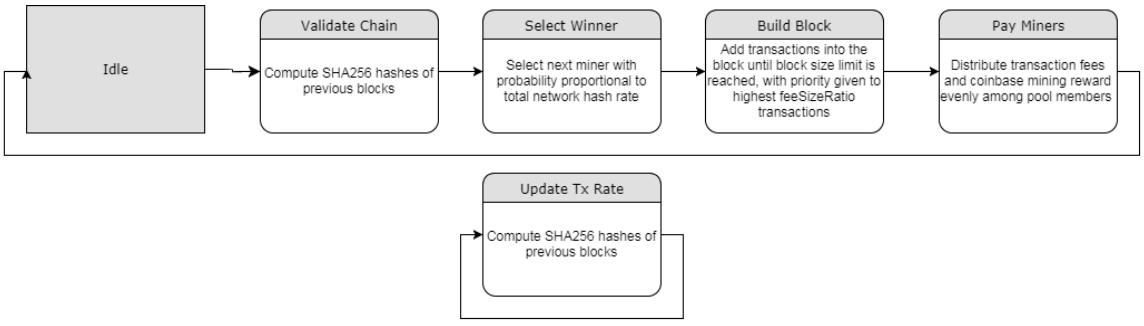


Figure 3.4: Main Agent Processes

After validation, a random miner is selected to mine the next block, with the probability of any single miner winning being proportional to their individual hash rate. This is

referred to as the “Select Miner” state.

Next, the main agent builds a block of transactions that would be representative of what a miner would build in reality. All unconfirmed transactions are ordered by descending feeSizeRatio, and then added to the next block until the size of the block meets the maximum allowable block size. If the transaction utilizes *Segregated Witness*, the transaction size is reduced by 75%. As each transaction agent is accounted for in the block, it is sent a message that it has been mined, so that the transaction agent can destroy itself. This is referred to as the “Build Block” state and results in a new block agent being created.

Finally, in the “Pay Miner” state, the sum of transaction fees and the coinbase mining reward is distributed proportionally, based on hash rate, between all members of the pool to which the winning miner belongs to.

The main agent also has one separate process, noted as “Update Tx Rate” that runs once a day to update the rate at which users generate transactions.

Miner Processes

Miner agents alternate between two functional states and one idle state, as shown below in Figure 3.5. A miner moves to the “Receive” state and changes its color to red for 1 minute if it is part of a mining pool that the main agent is paying revenue to.

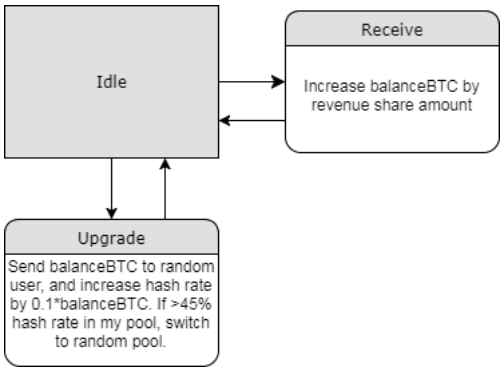


Figure 3.5: Miner Agent Processes

Miners also transition to the "Upgrade" state at an average rate of one time per week based on an exponential distribution. In this state miners change their color to green for 1 minute, and create a transaction agent with a destination to a random user for the amount of the miner's entire balance, and increase their hash rate by 0.1 units for each BTC spent. This simulates the act of purchasing more computing power from a vendor. Also during the "Upgrade" state, the miner will check to see if their pool's hash rate is greater than 45% of the total network; if so, the miner will attempt to switch to a random pool resulting in less than 45% of hash rate being controlled by that particular pool. If no suitable pool can be found after 30 random switches, a message will be displayed to the user that "No suitable mining pools found that don't result in 45% hash power in single pool".

User Processes

User agents, similar to miners, alternate between two functional states and one idle state as seen below in Figure 3.6. A user will move to the "Receive" state and change its color to blue for 1 minute if it receives a message from a transaction that was confirmed, indicating some amount of BTC has been transferred to the user.

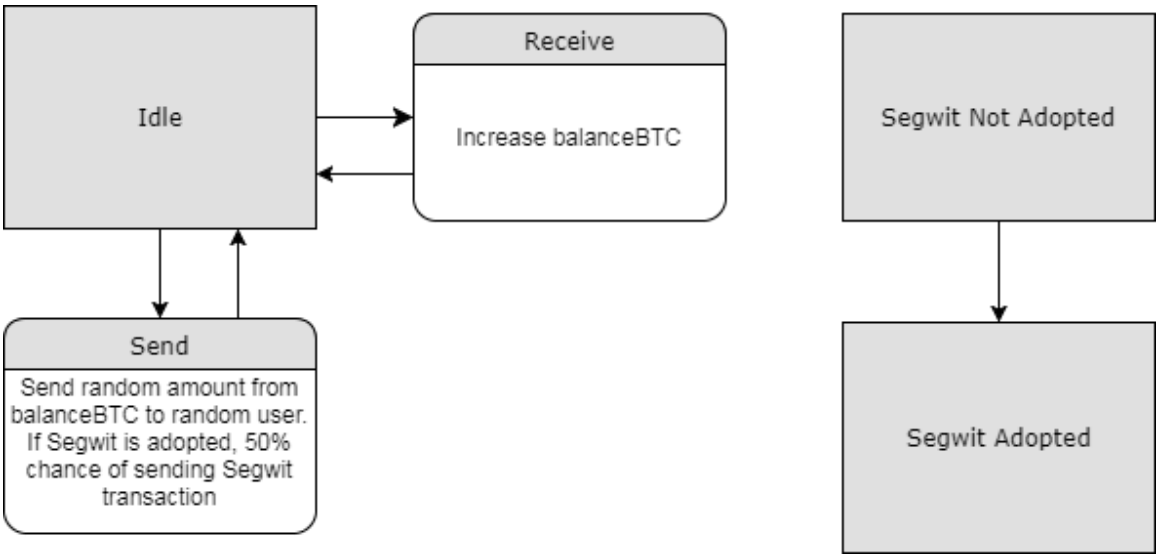


Figure 3.6: User Agent Processes

Users also transition to the "Send" state at an average rate that is specified in the main agent based on an exponential distribution. In this state, users change their color to green for 1 minute, and create a transaction agent to transfer BTC to another randomly selected user if they can afford to pay the fee for their transaction. The size in bytes of the transaction is selected from a size distribution derived from real Bitcoin network data [56] on 9/4/2017. If the user has adopted *Segregated Witness*, the transaction has a 50% chance of being a *Segregated Witness* transaction. The transaction amount is also subtracted from the user's balance at this time.

Users also have a separate state for *Segregated Witness* adoption if the feature is toggled at model initialization. After a pre-determined amount of time, the user will adopt *Segregated Witness* and be able to generate *Segregated Witness* transaction types.

Transaction Processes

Transaction agents contain a simple internal process whereby if they receive a message from the main agent that they have been mined, they move to a "Confirmed" state, as seen below in Figure 3.7. In this state, the transaction sends a message to the receiving user to increase its balance, and the transaction agent is then destroyed. If a transaction remains unconfirmed for longer than 336 hours, a transition is made to automatically remove the transaction from the memory pool in accordance with the default behavior of the Bitcoin Core client [18].

3.2.4 Concepts and Assumptions

Many aspects of the Bitcoin network, such as the supply expansion rate and block mining time, are hard coded into the Bitcoin protocol itself. Some of these concepts are represented precisely in the model, while others are implemented using a variety of assumptions described below.

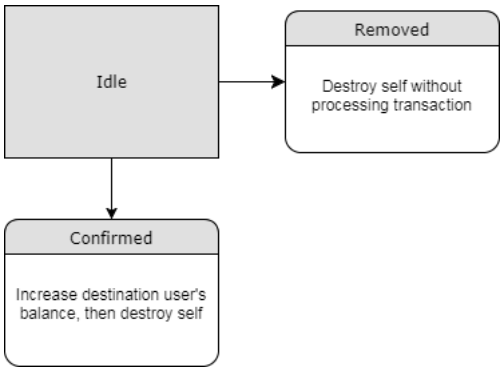


Figure 3.7: Transaction Agent Processes

The following assumptions were made in the main agent:

- The number of miners (30), mining pools (8) and users (350) remains constant over time.
- No price element is included in the model.
- Bandwidth consumed is estimated for a full node connected to 25 peers and is approximated using Formula 3.1 below.
- The block reward is hard coded to halve every 210,000 blocks, as described in the Bitcoin protocol.
- Blocks are mined with an exponential distribution with mean blockTime minutes between blocks.

Bandwidth consumed is defined as the average number of kilobytes consumed per second over a 6 hour time period and was approximated using Formula 3.1, described below. The sum of kilobytes of data from all blocks from *t* to *t-6* is taken, where *t* is the number of hours elapsed in the simulation so far. The size of each block is doubled, since the transactions in each block must have also been relayed alone at one point. 26 is used since each block or transaction must be downloaded once and uploaded 25 times to reach

each peer. We then divide by 21600 seconds, which is the number of seconds in 6 hours.

$$\frac{\sum_{t-6}^{t} Blocks.blockSize[\text{KB}] * 2 * 26[\text{peers}]}{21600[\text{seconds}]} \tag{3.1}$$

Miner assumptions:

- Hash rate can be directly purchased at an exchange rate of 1 BTC per 0.1 arbitrary hash rate units.
- Miners will try to leave their current pool for a random pool if they notice their current pool is approaching 51%.
- Miners will fill a block with transactions until the maximum block size is reached.
- Miners prioritize including transactions in blocks based on the transaction fee per byte.

User assumptions:

- If *Segregated Witness* is activated, the adoption rate is created by forecasting the current adoption trend as a sigmoid curve, as defined in Formula 3.2 below.
- The rate that users initiate transactions is approximated by regressions and forecasts from data from the actual Bitcoin network, and is described below.

The *Segregated Witness* adoption rate in the model was created using Formula 3.2 below. This formula was derived by forecasting current *Segregated Witness* adoption trends in the real Bitcoin network [57] and is further described in Appendix B.

$$\%adoption = \frac{1}{1 + e^{-0.0264025 * (x[\text{days}] - 138[\text{days}]))}} \tag{3.2}$$

The transaction rate for users is divided into four distinct eras:

1. From 1/1/2009 until 5/16/2010, remain constant at 100 tx/day

2. From 5/17/2010 until 6/12/2011, increase exponentially to 3,756 tx/day (R-square 0.75)

3. From 6/13/2011 until 5/2/2012, decrease linearly from 6,900 tx/day to 5,052 tx/day (R-square 0.177)

4. From 5/3/2012 and onward, increase exponentially (R-square 0.94 for data until 9/17/2017)

The rules for each of these eras were determined by performing regressions on real Bitcoin network data [55].

Transactions:

- The size of each transaction is drawn from a distribution fitted to historical Bitcoin transaction sizes from August 28, 2017 until September 4, 2017 [56] and further discussed in Appendix C

- The ratio between transaction fee and size is drawn from a distribution fitted to historical Bitcoin fees from September 1, 2017 until September 14, 2017 [58] and further discussed in Appendix D

3.2.5 Model Validation

Four metrics were assessed from January 1, 2009 until January 1, 2016 in order to validate the behavior of the model with the real Bitcoin network [55, 59]. The total number of confirmed transactions, supply of Bitcoin, blockchain size and bandwidth consumption are analyzed using the default simulation settings. Eight unique runs were performed to assess model accuracy and precision. Each metric is recorded in 3 month intervals. The data was then plotted in multiple time series plots, and a 90% confidence interval was developed for each metric.

Table 3.3: Setup for varying segregated witness activation and block size

Config Name	Start Date	Hardfork	Hardfork Delay (days)	New Max Block Size (MB)	Block Time (min.)	Segwit
Default + Segwit	7/31/17	FALSE	N/A	N/A	10	TRUE
Bitcoin Cash	7/31/17	TRUE	0	8	10	FALSE
Segwit2x	7/31/17	TRUE	109	2	10	TRUE
Litecoin	7/31/17	FALSE	N/A	N/A	2.5	TRUE

3.2.6 Experiments

After model validation was performed, experiments were performed to explore possible future impacts from varying Bitcoin parameters and features, such as average block time, maximum block size and the activation of *Segregated Witness*. These experiments are intended to provide insights into current issues surrounding Bitcoin and other cryptocurrencies, without requiring literal implementation.

Segregated Witness and Varying Block Size

Several scenarios were created to investigate the resource and network performance of alternate blockchain protocols similar to the Bitcoin protocol. These scenarios are also inspired by currently live alternative implementations in the cryptocurrency space. The effect of *Segregated Witness* activation, varying block time and varying block size on resulting resource requirements and network throughput are analyzed. Starting the simulation on July 31, 2017, 4 different scenarios with 4 replications are run for a period of 2 years, as shown in Table 3.3. The first scenario is the default configuration but with *Segregated Witness* active. The second scenario is based on *Bitcoin Cash*, which features an 8 MB block size and no *Segregated Witness*. The third scenario is based on the deprecated Bitcoin scaling proposal, *Segwit2x*, which would have had 2 MB blocks by November 17, 2017 and *Segregated Witness* activated. The fourth scenario is based on Litecoin, which features a 2.5 minute block time and *Segregated Witness* activated. In addition, transactions were set to remove themselves from the memory pool after only 24 hours instead of 336 hours, and the number of transactions that underwent this process were recorded.

Data on the resulting blockchain size, bandwidth usage, transaction fees, and memory pool growth are recorded. Three time series charts are developed that plot all 4 configurations over time. One plot is developed to characterize the transaction fee in terms of satoshis per byte for transactions that are included into blocks, a second plot to show the size of the memory pool over time, and a third plot to compare bandwidth consumption. A table was also developed to show the total blockchain size at specified times.

3.3 Results

In this section, results from model validation and experimentation are presented. Full sized charts for Figures 3.8 and 3.9 are available in Appendix E.

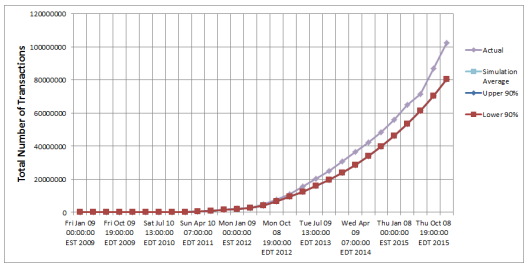
3.3.1 Model Validation

Data regarding the total number of transactions, supply of BTC, size of the blockchain in megabytes and average bandwidth consumed in kilobytes per second were collected in simulations from January 1, 2009 until January 8, 2016 in 3 month intervals. A total of 8 unique runs were performed, resulting in 232 data points collected for each of the four metrics.

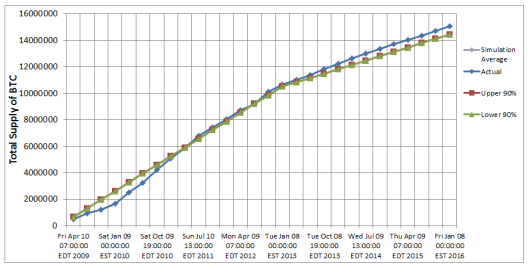
Figures 3.8a through 3.8d below display time series plots for each metric including the simulated data and the real Bitcoin blockchain data. The average, and 90% confidence intervals were calculated for the simulation data.

3.3.2 Scaling Experiments

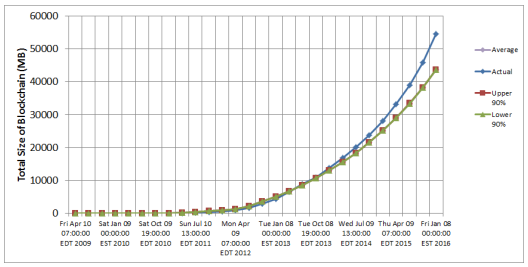
Data regarding the final blockchain size in megabytes, bandwidth usage in kilobytes per second, transaction fees in terms of satoshis per byte and number of transactions in the



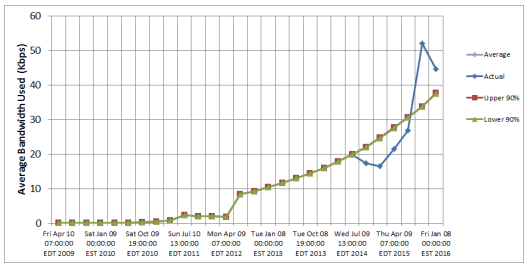
(a) Total Number of Transactions



(b) Total Supply of BTC



(c) Total Size of Blockchain



(d) Average Bandwidth Usage

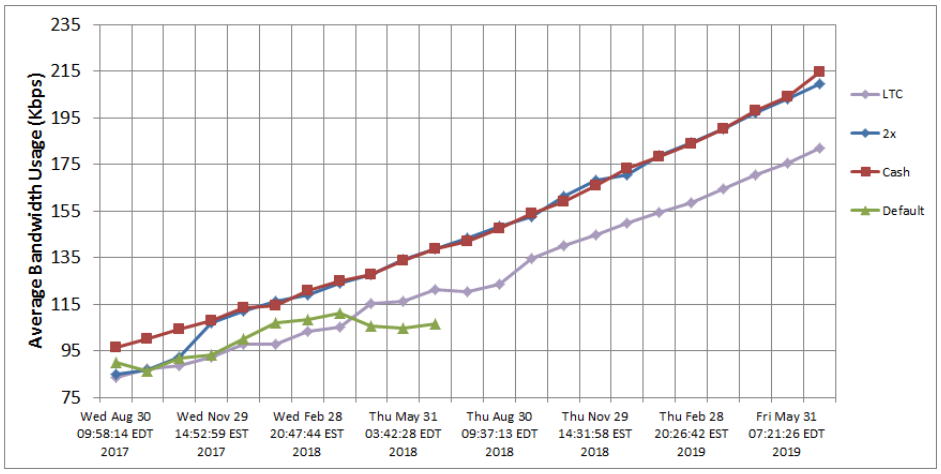
Figure 3.8: Time series plots of validation simulation results and real data

memory pool were collected from July 31, 2017 until June 30, 2019 in 1 month intervals. The default configuration was only able to run until June 30, 2018 due to simulation and computation limitations. Each configuration was replicated 4 times, resulting in 16 unique runs.

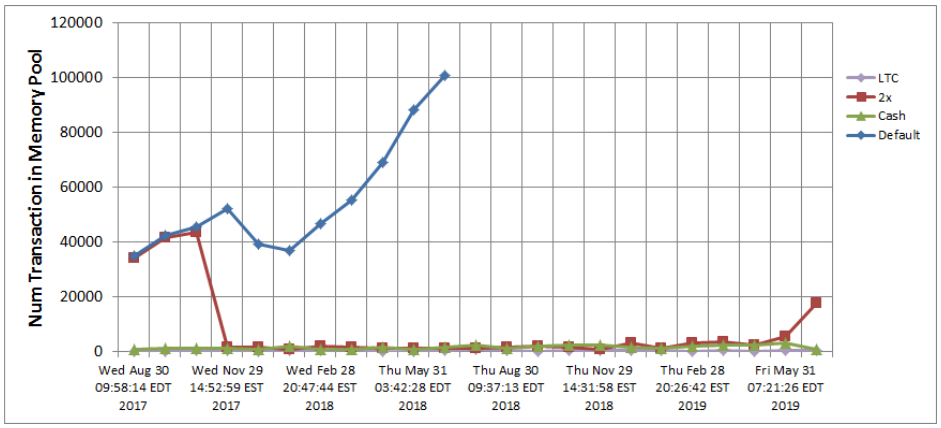
Figures 3.9a through 3.9c below display time series plots for each metric across all 4 configurations. Each data point is representative of the average of the 4 replications for each configuration.

Table 3.4 below shows the resultant blockchain size for each configuration in 4 month intervals.

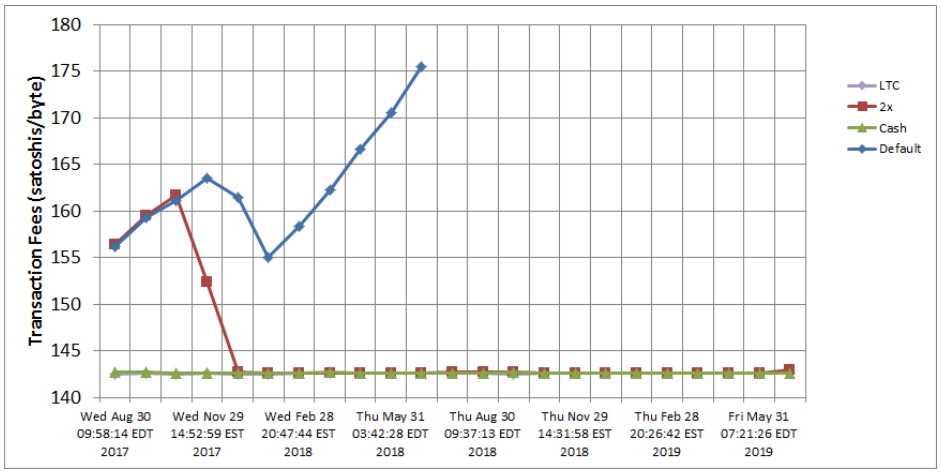
90% confidence intervals were calculated for each metric collected at regular intervals for both model validation and scaling experimentation. Table 3.5 below displays the widest confidence interval found for each metric used for validation in terms of percentage of the average value.



(a) Average Bandwidth Usage



(b) Transaction Fees



(c) Memory Pool Size

Figure 3.9: Time series plots comparing varying configurations

Table 3.4: Blockchain Size (GB) for Varying Configurations and Dates

–	**Default**	**Bitcoin Cash**	**Segwit2x**	**Litecoin**
7/31/17	126.7	126.7	126.7	126.7
11/29/17	144.6	147.0	144.8	144.2
3/31/18	165.5	170.4	166.5	164.4
6/30/18	181.8	190.3	184.7	181.6
11/29/18	209.1*	228.6	219.6	214.6
3/31/19	230.9*	264.5	252.4	245.7
6/30/19	247.3*	295.2	280.2	272.2

* - extrapolated value

Table 3.5: Widest 90% confidence interval width for each metric represented as a percentage of the average value

Metric	**Total Transactions**	**Mempool**	**Chain Size**	**Avg. Bandwidth**
90% CI Width	0.06%	0.01%	0.08%	1.12%

Table 3.6 below displays the widest confidence interval found for each metric during scalability experimentation in each scenario in terms of percentage of the average value.

Table 3.6: Widest 90% confidence interval width for each metric and scenario represented as a percentage of the average value

Scenario	**Fees**	**Mempool**	**Avg. Bandwidth**	**Chain Size**	**Total Transactions**
Default	0.08%	1.99%	1.05%	0.01%	0.01%
Bitcoin Cash	0.07%	10.50%	0.27%	<0.01%	<0.01%
Segwit2x	0.06%	14.65%	0.46%	0.58%	0.01%
Litecoin	0.01%	14.70%	0.62%	<0.01%	<0.01%

3.4 Discussion

3.4.1 Model Validation

The total number of transactions, total supply of BTC, total size of the blockchain, and average bandwidth consumption were plotted over time, as seen in Figure 3.8, in order to validate the behavior of the model relative to the real Bitcoin network historically. Across

all 4 metrics, the 90% confidence interval widths were less than one percent of the average value, indicating precise outcomes for replicated runs. The general direction of all 4 metrics also coincided with the real Bitcoin network.

When viewing the total number of transactions seen in Figure 3.8a, the final number of transactions in the simulation is 80,345,721, while the real value is 102,612,806, a difference of 21.7%. At some point between July 10, 2011 and October 9, 2011 the total transactions in the simulation and the real Bitcoin network were equal. This may be due to under estimation of regression values used in the simulation; however, the difference was never larger than 21.7% and the real Bitcoin network experiences regular variance in the rate of transactions being made each day [55]. For instance, the rate of transactions per day on the real Bitcoin network went from 241,000 on September 16, 2015 to 118,000 by September 18, 2015. This under performance may also be explained due to the prevalence of abnormal increases in transaction rates in the real Bitcoin network due to spam transaction attacks, in which transactions are made with the sole intention of congesting the Bitcoin network. Previous research has indicated that spam transactions may comprise a non-neglible amount of transaction activity [22].

The resulting size of a blockchain is directly related to the amount of transaction data contained within it. This characteristic is confirmed when looking at the final blockchain sizes in Figure 3.8c, as the final difference between each blockchain size is roughly 20%, similar to the difference observed in total number of transactions.

When viewing the supply of bitcoin in circulation as seen in Figure 3.8b, the final supply in the simulation was 14,449,078 BTC while the real Bitcoin network had a supply of 15,062,925, a difference of roughly 4%. Until April 2011, the number of BTC in the simulation was higher than the real Bitcoin network. The difference may be attributed to the variance of Bitcoin mining over time. During periods in 2009, Bitcoin experienced block times of up to 426 minutes, which would result in the supply being released more slowly; however, since 2013 the real Bitcoin network has consistently experienced block

times slightly below 10 minutes, resulting in a slightly higher rate of supply inflation [60].

Collection of data for the bandwidth consumption rate of the real Bitcoin network was limited to July 9, 2014 until November 29, 2017. The source of the data is from 1 particular Bitcoin node. The data is highly variable, as the number of connected peers may vary at any given time [59]; however, in general, the average bandwidth consumption in the simulation coincided with the direction of the real Bitcoin network.

Based on the performance of the aforementioned metrics, it seems to be a reasonable conclusion that one can use agent based modeling to replicate specific behaviors of blockchain-based networks.

3.4.2 Scaling Proposals

The simulation results in Figure 3.9 provide insight as to the impact of varying scaling proposals.

When viewing the resultant bandwidth usage in Figure 3.9a, it can be seen that the default configuration resulted in a peak bandwidth usage of 111 Kbps, whereas *Bitcoin Cash* and *Segwit2x* continued to rise throughout the period, experiencing a peak bandwidth consumption of 215 Kbps in June, 2019. Since bandwidth usage is directly correlated to the amount of transaction data being put on the network, it makes sense that as long as throughput capacity is not reached (the rate of transactions added to the memory pool exceeds that of transactions added to the blockchain), bandwidth consumption will continue to increase, as is seen by the *Segwit2x* and *Bitcoin Cash* proposals. For comparison, the average worldwide internet connection speed was 7.2 Mbps as of Q1 2017 [61]. If the bandwidth usage for the *Bitcoin Cash* configuration in Figure 3.9a is extrapolated using a polynomial regression, it would still take roughly 20 years until bandwidth consumption exceeded 7.2 Mbps.

The size of the memory pool and the average transaction fee per byte are shown in Figures 3.9b and 3.9c respectively. This information can be used to assess when through-

put capacity is exceeded for a particular configuration; if the fee rises, it is indicative that throughput capacity has been exceeded. The *Bitcoin Cash* and Litecoin configuration simulations indicate sufficient transaction capacity throughout the simulation period. When viewing the *Segwit2x* configuration, the hard fork occurring in November is sufficient in clearing the memory pool backlog and dropping fees back to 140 bytes per satoshi; however, the memory pool begins to grow again in June, 2019 indicating that the *Segwit2x* proposal may have only provided sufficient capacity for roughly 2 years. When viewing the default configuration, which features *Segregated Witness* activation, the memory pool briefly decreases from November, 2017 until January, 2018, after which the memory pool and transaction fees grow rapidly.

In reality, transaction fees and memory pool size have dropped considerably during January and February, 2018, as seen in Figure 3.10. This is due to a significant decrease of transaction rate behavior compared to what was forecast in the model based on information available in September, 2017. The Bitcoin network experienced roughly 200,000 transactions per day in September, 2017 but has fallen to just 154,000 by February, 2018, a drop of 23% [55]. This may be due to decreased user activity and/or transaction batching, whereby multiple transactions are combined into one larger, more efficient transaction.

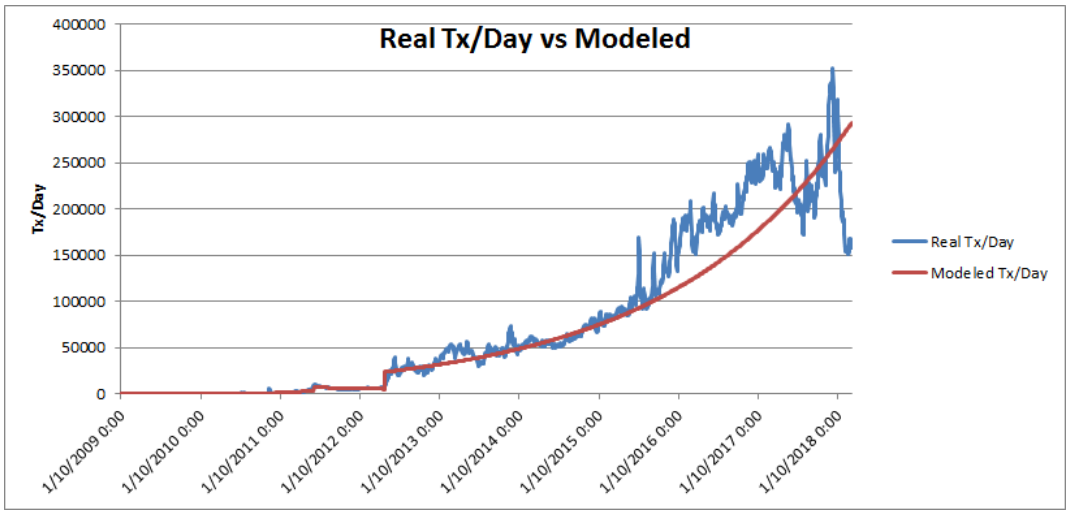


Figure 3.10: Real Bitcoin Network Transactions Per Day Versus Modeled [55]

An increasing difference between chain configurations can be observed when viewing the resulting blockchain sizes for each configuration in Table 3.4. After 1 year, the 4 configurations differ by 8.5 GB. After 2 years, a larger difference can be observed between the default configuration, and the *Bitcoin Cash*, *Segwit2x* and Litecoin configurations of 47.9 GB, 32.9 GB, and 24.9 GB respectively. Since 2011, average hard drive space for general consumer grade hardware has increased roughly 173 GB per year, resulting in an average hard drive capacity of 1.4 TB in 2016 [62]. Figure 3.11 below shows the resulting blockchain sizes plotted against the average hard drive capacity over time, assuming that each configuration has full blocks for the entire year. With 1 MB or 2 MB blocks, the average hard drive capacity would always exceed that of the blockchain, and for 4 MB blocks (which can be compared to 4x the rate of blocks in the case of Litecoin) or 8 MB blocks, it would take nearly 35 years or 5 years respectively until the blockchain size exceeded that of the average hard drive.

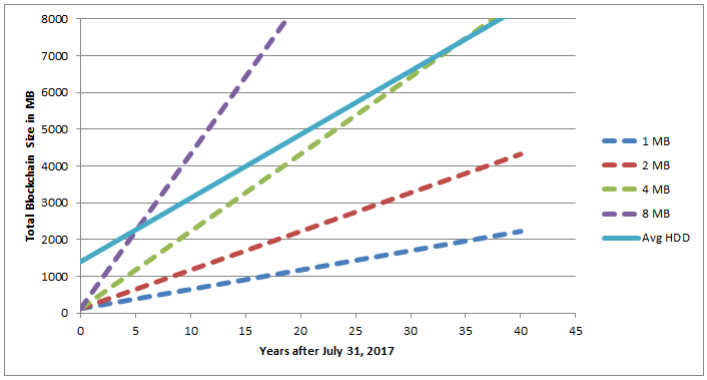


Figure 3.11: Blockchain Size vs Average HDD Capacity

This study on the scalability issue differs from the BitFury Group's study in that this study utilizes forecasting for determining transaction rates instead of assuming all blocks will always be full, regardless of network activity. Assuming the average Bitcoin transaction is roughly 400 bytes, and 1 MB is mined every 10 minutes, this results in a maximum theoretical throughput of 4.17 transactions per second; for comparison, in 2016 Visa re-

ported nearly 2,000 transactions per second and PayPal at 136 transactions per second [26]. This would require a block size increase to 600 MB or 40.8 MB respectively if all transactions were to continue to settle on the blockchain.

Ultimately the results show that the default configuration, the *Segwit2x* proposal, *Bitcoin Cash*, and Litecoin all would provide suboptimal results for scaling if the goal is to match the likes of Visa or PayPal. While the *Segwit2x* proposal and *Bitcoin Cash* configurations provide temporary relief to memory pool size problems and exploding transaction fees, they also introduce the native risks associated with hard forks; for example, the *Segwit2x* hard fork was dead on arrival due to a bug in the code [35]. Also the results tend to support the notion that none of the proposals would significantly negatively impact decentralization when considering the impact on bandwidth consumption, and hard drive space. An optimal blockchain scaling solution should involve more than just *Segregated Witness* activation, block size variation, and block time variation. Other studies have also suggested that changes to maximum block size and block time intervals are only incremental steps towards increased throughput [27].

3.4.3 Future Work

Agent-based modeling can be used to explore various topics in the Bitcoin ecosystem, such as selfish mining strategies, reputation based systems, game theory economics, and more. Various modifications could be made to the current model that would allow different aspects of blockchains to be studied and are listed below.

- *Mining* - The inclusion of varying mining strategies could be employed to study what strategies lead to optimal revenues. Mining could also be used to study varying consensus models, and to test the resilience of certain blockchain configurations to malicious actors. For instance, what incentives are required to prevent a mining pool from overtaking more than 51% of hashing power? Or an alternative mining

algorithm, such as *proof-of-stake* [13], could be simulated.

- *Economic Components* - The inclusion of price dynamics, varying user populations and/or adoption rates of Bitcoin globally could be modeled to shed insight as to how Bitcoin's price, or user base, grows over time. Even further, the complete removal of an underlying cryptocurrency could be analyzed, as is the case in certain blockchain implementations such as Hyperledger [15].

- *Scaling Proposals* - Alternative scaling solutions, such as the Lightning Network [30], could be added to the simulation. Also the estimation of additional factors, such as I/O disk speed requirements and RAM consumption could be added to the simulation [28]. Also the incorporation of multiple blockchain copies, as is observed in real blockchain-based systems, would add to the granularity of the simulation, and allow consensus failures to be analyzed.

- *Incorporate New Data* - The model was developed using data available through September, 2017. This data was used to make forecasts to describe transaction activity, transaction size, fees, and other assumptions. Reconstructing the same model using updated information through March, 2018 can provide further insight to current Bitcoin community issues, especially those related to scalability as discussed in subsection 3.4.2.

The model is not limited to the modifications suggested above however, as agent based models can be used to simulate a wide variety of complex behaviors arising from individual interactions. As the field of blockchain technology continues to grow, more tools from varying domains will be used to study and understand how blockchains behave, with greater levels of detail.

Demonstrating the feasibility of a blockchain-based care coordination system to improve treatment of patients

Current Electronic Health Record (EHR) management is dominated by centralized companies, and isolated data stores that lack interoperability. There is a lack of medical record universality, auditability of records and profit incentive for centralized organizations to share data with competitors. In addition, cyber security has become an increasing concern; nearly 90% of healthcare organizations have experienced a data breach in the past two years, and 45% have experienced more than five [7]. Data sharing initiatives in the EHR space have already shown the potential to save hundreds of thousands of lives, and provide savings on the scale of $10B+, but suffer from a lack of implementation [8]. These breaches and data sharing initiatives also bring up concerns of health data ownership and privacy. A model simulation is used to demonstrate the feasibility of a blockchain-based care coordination system among opioid addicted patients that would foster improved care, control for patients and transparency for providers, among other improved outcomes.

4.1 Introduction

A decentralized digital currency, called "Bitcoin", was launched in 2009. Bitcoin utilizes a series of cryptographically interconnected blocks of transactions, commonly dubbed as a "blockchain" [3]. The blockchain acts as a distributed ledger, retaining the history of every Bitcoin transaction that has occurred since the start of the network; similar to the internet, anyone is free to join the Bitcoin network without censorship. This enables value to be transferred without third party intermediaries, such as governments or banks. Value can be defined as more than just currency, and in the context of EHR, the value is the healthcare information itself. Since Bitcoin's inception, hundreds of other digital currencies and varying implementations of blockchain technology have been created. The total market cap of these varying digital currencies traded at an excess of $800B in January, 2018 [4].

This chapter uses simulation tools to study the feasibility of implementing a blockchain-based system in a healthcare environment in order to improve coordination of care. A model is developed that simulates the actions of opioid addicted patients and healthcare providers recording patient information on a shared blockchain combined with local record stores. The model can be toggled between a public permissionless blockchain or a private permissioned blockchain. In the literature, simulations have been used for evaluating and demonstrating the efficacy and feasibility of developing varying care coordination architectures, [10] and for dynamically modeling a patient's health state [11]. Blockchain technology may enable solutions where data can be shared in a distributed manner that increases interoperability, security, immutability, and consumer privacy. In addition, blockchain-based systems may also foster improved care for patients, allow patients to control their records and provide increased transparency for all system participants [46]. While this chapter focuses on care coordination specifically, blockchains have the potential to affect a multitude of activities within the biomedical and healthcare ecosystem; these may include improved exchange of clinical health data, claims adjudication, pharmaceutical supply chain management, improved clinical trial data for population health research, and more [48, 49]. In

August, 2016 the Office of the National Coordinator for Health Information Technology operated an ideation challenge where 15 white papers were awarded cash prizes for exploring the use of blockchain in health IT and health-related research [45]. Several blockchain projects have already targeted the healthcare industry, as can be seen below in Figure 4.1 [54]. Patientory, Medibloc, Medishares and Medicalchain have publically traded tokens that are valued at a combined value of $407M as of February, 2018 [4]. Work has already been done to explore metrics for assessing blockchain-based healthcare applications [51].

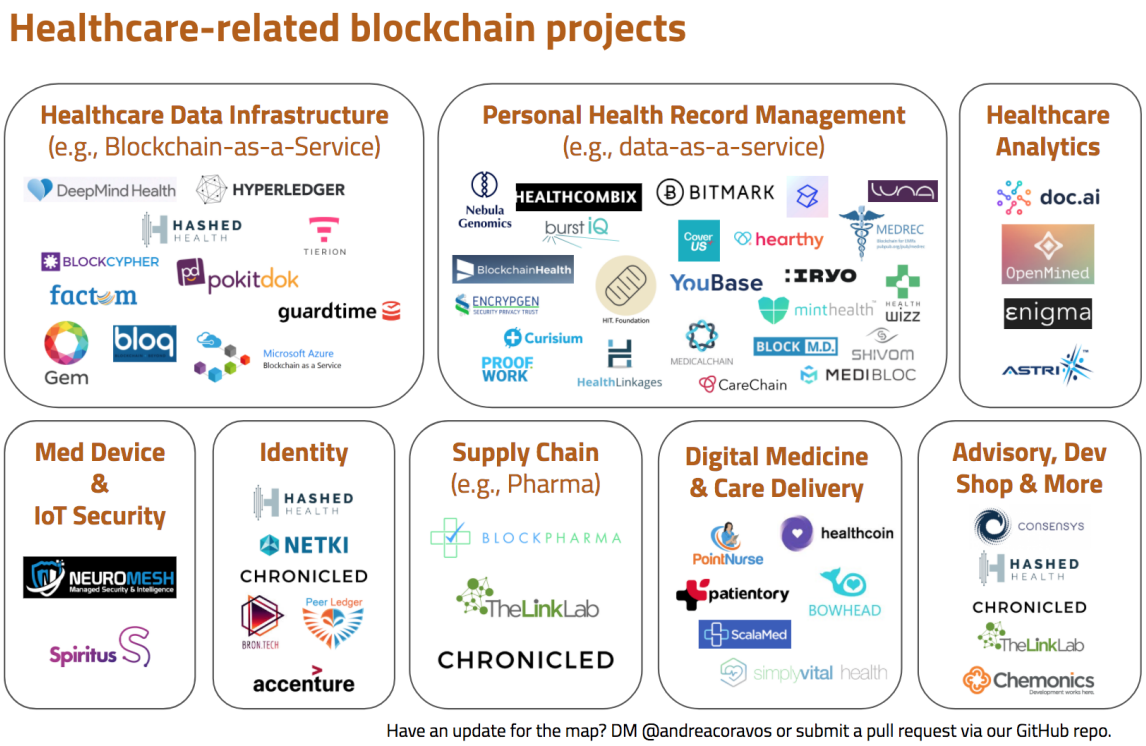


Figure 4.1: Healthcare-related Blockchain Projects [54]

4.1.1 Blockchain

Blockchains possess a multitude of technical features than enable varying operational capabilities. In this section these technical features and operational capabilities are first described, and then organized into a matrix to display which technical features contribute to

enabling specific operational capabilities.

Technical Features of Blockchain Technology

Immutable Ledger Since each block of data in a blockchain is cryptographically connected to the previous, modifications in the ledger will result in an invalid state. A blockchain is an append-only data structure, meaning data can only be added to it but never removed. Even if a blockchain were run without distributed consensus, as long as external parties can still view the data, they can verify the data is immutable. The Bitcoin blockchain, the longest running blockchain to date, has remained immutable since inception. This immutability enables users to trust that if a transaction is included in the blockchain it will not be removed, without requiring reassurances from a central third party [17].

Consensus of State A blockchain must have some mechanism by which all participants can arrive to a consensus as to the state of the network. The consensus protocol is the set of rules which the participants of a blockchain agree upon for determining the state of the blockchain.

In the case of Bitcoin, the consensus protocol is accomplished by *proof-of-work*. *Proof-of-work* allows anyone to attempt to solve a complicated math problem (typically done by specialized computers) in order to compete for the right to form the next block of data in the blockchain (referred to as "mining"). The user who successfully performs this *proof-of-work* is rewarded with bitcoins [23]. This reward creates the incentive for miners to participate in the network.

Another common alternative to *proof-of-work* is the *proof-of-stake* system; whereas in *proof-of-work*, the odds of a miner forming the next block are proportional to the computational resources dedicated to solving a specified math problem, in *proof-of-stake* your odds of forming the next block are proportional to the total amount of cryptocurrency being held by the miner [13].

Both *proof-of-work* and *proof-of-stake* require an underlying cryptocurrency to achieve consensus in an environment where anyone is allowed to participate (known as a "public permissionless" environment) in the consensus process. Blockchain protocols without an underlying cryptocurrency can achieve consensus; however they require that the participants in the consensus protocol are limited and identified (known as a "private permissioned" environment) [15].

Automated Smart Contracts Smart contracts are programs that exist on a blockchain. Since they exist on the blockchain, any participant in that blockchain can view the contract code and determine if it executed correctly. Smart contracts, unlike contracts written by humans, can execute without the action of a third party. The execution of smart contracts proceeds by meeting a set of pre-defined conditions [14]. MedRec, a prototype for an electronic health records system, proposed that patient-provider relationships, relevant permissions, and data retrieval instructions could be implemented through the use of Ethereum smart contracts [47].

Multi-signature Schema Multi-signature transactions require that at least m of n users sign off on a particular transaction, instead of just 1 user authorizing a particular transaction [24]. This can enable basic escrow functionality; for example, Bob and Alice could utilize a 2 of 3 multi-signature address as an escrow with Carol acting as a third party arbiter to settle any sort of dispute that may occur if both Bob and Alice do not agree to sign off on a transaction.

Cryptography Cryptography refers to the transfer of information in such a way that preserves the privacy of the data being transmitted through the use of mathematical principles. Cryptography is critical to the foundation and operation of any blockchain system, and is used in various aspects of a blockchain, including the creation of blocks, transactions, addresses, privacy preserving mechanisms and more [20]. A function commonly used in

cryptocurrencies is that of a “one-way hash function”. A one-way hash function produces a unique output for any given input of data; the output of a oneway hash function cannot be used to determine the original input in any way. Bitcoin utilizes the SHA256 hash function [19].

Asset Digitization Asset digitization refers to the idea of converting either tangible or non-tangible assets into assets on a blockchain, such that their ownership can be verified without a third party. One of the earliest methods of asset digitization created was “colored coins”, which is a protocol on top of the Bitcoin blockchain that allows certain bitcoins to have additional meta data attached to them. This meta data could represent the deed to property, ownership of a stock, ownership of a web domain, ownership of certain medical records or devices, among other applications [16].

Peer-to-peer Networking Blockchain systems leverage peer-to-peer networking, instead of a client-server model. In a client-server model, all users trust a single server to obtain information and system updates. In a peer-to-peer system, information is propagated to all users who participate. Decentralization of information does come at a cost however, as every user must be aware of every other users entire history of activity at all times [12].

Operational Capabilities of Blockchain Technology

The technical features mentioned above lead to more abstract operational capabilities that were identified in blockchain based systems. Transfer of value, security, auditability and decentralization of trust were identified as 4 of blockchain’s operational capabilities after surveying the literature.

- Transfer of Value - Blockchains enable the ability to determine ownership of data through public-key cryptography. This data is defined to be the “value” being transferred in a blockchain based system. It could be the rights to access a certain record,

or simply ownership of a native token or cryptocurrency.

- Security - Blockchains utilize cryptography to ensure immutability of data stored in a blockchain. Blockchains eliminate a central point of failure in a system.

- Auditability - Transactions in a blockchain cannot be removed, since blockchains are append-only data structures. This makes them valuable tools for tracking the chronological history of actions by participants in a system over time.

- Decentralization of Trust - Blockchains are replicated across the various participants in a blockchain based system, increasing the overall resiliency of such a system. Compromising a blockchain based system will typically require that a majority of the participants are compromised, as opposed to just a single entity in typical centralized database management systems.

In Table 4.1 below, a 1 can be interpreted as a technical feature contributing to the enabling of a certain operational capability. The table can be used as a reference to interpret which technical features are most integral to enabling blockchain operational capabilities.

Table 4.1: Operational Capabilities vs Technical Features of Blockchains

		Technical Features							
		Immutable Ledger	Consensus	Smart Contracts	Multi-Sig	Cryptography	Asset Digitization	P2P	Capability Total
Operational Capabilities	Transfer of Value	1	1	1	1		1		5
	Security	1	1	1	1	1			5
	Auditability	1	1			1	1		4
	Decentralization of Trust	1	1	1				1	4
	Feature Total	4	4	3	2	2	2	1	

Based on the table above, the most critical technical features in a blockchain are its immutable ledger and consensus of state. Both of these technical features contribute in some manner to enabling all the underlying operational capabilities listed. These operational capabilities will ultimately determine how well a blockchain can be applied to a particular set of requirements for an application.

4.1.2 Care Coordination

Care coordination does not have a singular definition, but in general refers to the idea of multiple care providers coordinating together in some manner to track and manage the care that a specific patient receives. Increased coordination by healthcare providers and an increased awareness of the roles and resources available by each provider should result in overall improved care for patients, and reduction in costs. The activity of care coordination has been identified as a critical component to achieving improved quality of care [9].

Healthcare System Requirements

Varying requirements were identified as components to an improved care coordination system, which are outlined below. These requirements are then organized in a matrix against the operational capabilities that blockchains have, as outlined earlier, to focus on which operational capabilities have the potential to fulfill specific requirements in a blockchain-based care coordination system.

(a) Cost Reduction - Healthcare costs grew 4.3% in 2016, and represented 17.9% of GDP for the USA [63].

(b) Fraud Prevention - Security breaches in the healthcare space are becoming an increasing concern. Some studies have reported that over 90% of hospitals have incurred a data breach every two years. These breaches pose both security and privacy risks for patients [7].

(c) Identity Management - Healthcare providers need to be able to verify the identity of the person or provider that is requesting treatment. The US Office of the National Coordinator for Health Information Technology mentions “verifiable identity and authentication of all participants” as one of the critical components for healthcare systems nationwide [48].

(d) Record Availability - Having real-time access to healthcare data improves both clinical care coordination and clinical care in emergency situations. In addition, public health efforts can also be improved by enabling rapid detection of public health related threats [49].

(e) HIPAA Compliance - The Health Insurance Portability and Accountability Act is a certification that ensures security policies and procedures have been developed by relevant healthcare providers and is a mandatory requirement for those who store healthcare patient data [64].

(f) Universality of Record - The construction of a universal health record can reduce the amount of manual paperwork, overhead, and overall costs by providing a transparent record set that is accessible from anywhere around the globe [48].

(g) Auditability - The presence of an audit trail can assist in improved claim auditing and fraud detection. An audit trail also allows a patient's medical history to be reconstructed and verified to ensure that optimal treatment decisions are made for the patient by their provider [49].

(h) Reconciliation of Records - Reconciliation of differences between records can result in increased costs when trying to access historic patient data [48].

(i) Interoperability - Different healthcare systems run into issues when sharing data due to a lack of interoperability [47]. The US Office of the National Coordinator for Health Information Technology has already issued a nationwide roadmap for healthcare systems to achieve interoperability [48].

(j) Encourage Patient Engagement - Clinical outcomes for patients can be improved by increasing consumer engagement. Blockchain based systems may enable emerging approaches for disease treatment and precision medicine if patients are encouraged to take ownership over their information [46].

In Table 4.2 below, a 1 can be interpreted as a capability contributing to the fulfillment of a healthcare requirement. The table can be used as a reference to interpret which blockchain-based operational capabilities are most integral to fulfilling healthcare system requirements, and which requirements may need to be addressed by supplemental solutions.

Table 4.2: Requirements of Healthcare vs Operational Capabilities of Blockchains

		Operational Capabilities				
		Auditability	Transfer of Value	Security	Decentralization of trust	Requirement Total
Healthcare Requirements	Cost Reduction	1	1	1	1	4
	Fraud Prevention	1		1	1	3
	Identity Management			1	1	2
	Record Availability	1	1			2
	HIPAA Compliance	1		1		2
	Universality of Record	1			1	2
	Auditability	1	1			2
	Reconciliation of Records	1				1
	Interoperability	1				1
	Encourage Patient Engagement		1			1
	Capability Total	8	4	4	4	

Based on the table above, blockchain's capability to enable auditing of records is a critical component to fulfilling various healthcare system requirements. Auditability supports eight of the ten identified requirements of healthcare systems. Transfer of value, security, and decentralization of trust are also important operational capabilities of blockchains, supporting either four or five out of ten healthcare system requirements.

4.2 Methodology and Model Design

A proof of concept model was developed to demonstrate how a healthcare focused blockchain could be used in a hypothetical system involving opioid dependent patients. The model includes a general practitioner, a treatment facility, a hospital, and patients. All participants in the system are able to access relevant healthcare data in a distributed manner by uti-

lizing a hypothetical blockchain, MDChain, with a native cryptocurrency, MDChain coin (MDC). The model was built using AnyLogic modeling software, which uses the Java programming language and was further described in subsection 2.1.1. The model includes a cryptographic peer-to-peer ledger with a very basic mining algorithm put into place to maintain consensus.

Each participant has its own public identifier (known as a public key), a copy of the MDChain blockchain, and a set of local records. Patients can migrate between states of being healthy, prescribed opioids, addicted to opioids, attending therapy, and being a recovered addict. All patient interactions with healthcare providers, as well as interactions from their own wearable devices, are stored asynchronously as local records across the various nodes; information regarding where these local records are stored (known as "hash pointers") are stored synchronously across all participants in the MDChain blockchain as transaction data inside of blocks. This design reduces the amount of data stored in the public blockchain, since it only includes the location of records, rather than the full record. This design still maintains the distribution of data and the ability to implement offline permissions in the future to ensure compliance with HIPAA and other relevant regulations. A diagram of the proposed MDChain system is shown below in Figure 4.2.

The model also implements functionality allowing any participant to compile a verifiable complete history of any other participant of interest's records by only using the participant of interest's public key. The model concretely demonstrates the auditability, availability, universality and reconciliation of these patient records distributed throughout the MDChain network. Such a system may also reduce fraud, reduce costs, incentivize patient engagement and improve the overall treatment of patients throughout the system. The proposed architecture is simulated and empirically scored based on the technical features, operational capabilities, and requirements discussed earlier. These scores, in combination with the matrices developed earlier, can be used to facilitate future research as to the optimal architecture of a blockchain-based healthcare system. The source code for the model

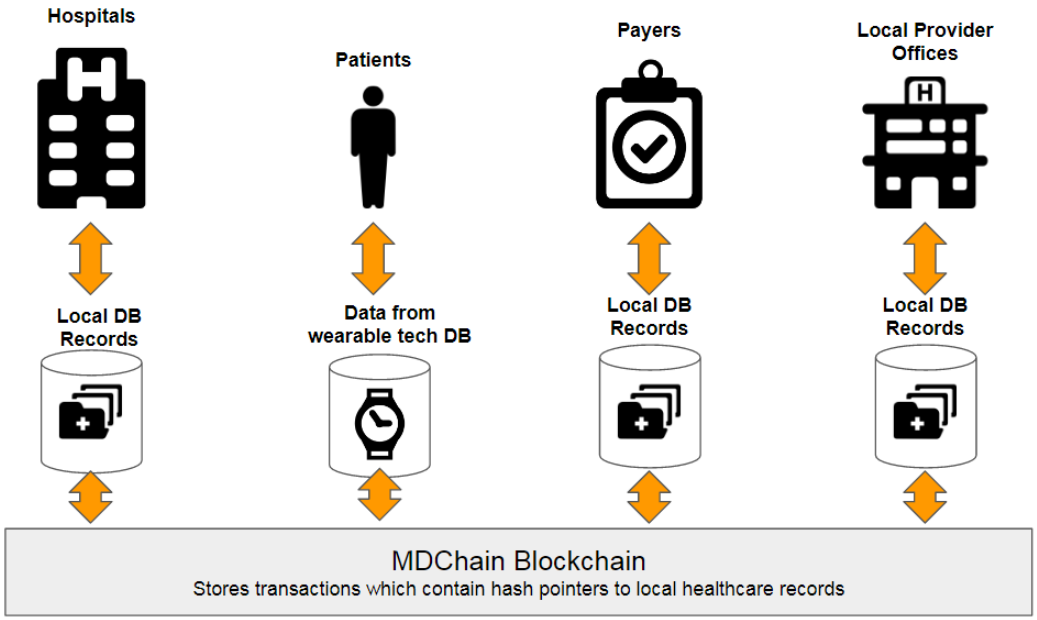


Figure 4.2: Diagram of Healthcare Stakeholders and Local Records in MDChain

can also be found at https://github.com/champbronc2/BitcoinAnyLogic.

4.2.1 Agents, Parameters and State Variables

The model includes 5 agent main types, and 4 agent sub-types. An agent sub-type inherits all the characteristics from its main type. The 5 main agents include main, participant, blocks, transactions, and local records. The 4 sub-types are all inherited from the participant main type and include patients, general practitioner, hospital, and treatment center. Any parameters, variables, state charts, functions, events, or custom distributions that belong to each agent are described below in Table 4.3.

Main

A function in the main agent fetches a local record when given a specified hash pointer (getRecord).

An event in main selects a random participant to mine the next block of transactions

Table 4.3: Breakdown of simulation components by agent type.
.*- indicates unused variable

	Main	Participant	Patients (sub-type of participant)	General Practioner (sub type of participant)	Hospital (sub type of participant)	Treatment Center (sub type of participant)	Blocks	Transactions	Local Records
Parameters	None	• pubKey • privKey*	None	None	None	None	• prevBlockHash • timestamp • txNum • MDCAmount • fees • size • blockHeight • coinbase • transactionData	• timeStamp • txID • from • to • amount • fee • hashPointer	• timeStamp • txID • data • hashPointer
Variables	None	• topBlockHash • topBlockString • blockHeight • con0 • con1 • balance*	None	None	None	None	None	None	None
Functions	• getRecord	• mineBlock • sendBlocks • toJSON • buildTx • buildRecord • fetchRecord	• constructHistory	None	• constructHistory	• constructHistory	• toJSON	• toJSON	• toJSON
Events	• selectMiner	• setConnections	None	• sync	None	None	None	None	None
State Charts	None	None	• patientState	None	None	None	None	None	None
Distributions	None	None	None	None	None	None	None	None	None

on the network (selectMiner).

Participant

All participants have parameters for a public key (pubKey) and a private key (privKey). In practice, the private key would be used to sign messages and prove the identity of any participant, however this functionality was left out of the model. The public key is treated as a unique identifier for each participant in the MDChain network.

Variables are initialized to store the number of blocks in the blockchain (blockHeight), a cryptographic hash of the most recent block in that participant's blockchain (topBlockHash), and a JSON [1] formatted representation of that block (topBlockString). Each participant is only connected to two other participants in a ring-type fashion in order to reduce the total number of connections in the model. Variables are used to store the name of the two connected agents (con0 and con1). A variable also exists to store the current balance of MDC, but is not used within this version of the model as the cryptoeconomics of the

[1]JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. - www.json.org

system are not explored in this simulation.

Participants have functions to create a new block (mineBlock), broadcast a new block to its peers (sendBlocks), create a new transaction (buildTx) or local record (buildRecord), and find any record on the MDChain network given a specific hash pointer value (fetchRecord). A general function was implemented to serialize the information of a participant (toJSON). The serialized information includes that participant's current variable and parameter states.

An event for each participant is triggered at start up (setConnections) that defines the variables con0 and con1 mentioned above. These events create the static configuration used in the simulations.

Each participant contains a population of blocks, transactions and local records, referred to as blockchain, memPool, and localRecords respectively within the model.

Patients, General Practitioner, Hospital and Treatment Center Patients, the general practitioner, the hospital and the treatment center inherit all parameters, variables, functions, and events from the participant agent type. Additionally, participants have a function enabling them to construct the complete history of any participant in the MDChain network given a specific public key (constructHistory).

Patients also have a state chart (patientState) that is meant to emulate the opioid addiction process. The state chart alternates between states of being healthy, requiring a prescription, having an addiction, being under treatment for addiction, and having recovered from addiction. More details regarding the processes involved by patients is explained in subsection 4.2.3.

The general practitioner has an event run at initialization to ensure the correct variables for con0 and con1 are set (sync).

Blocks

Blocks contain parameters to define the hash string of the preceding block (prevBlock-Hash), a timestamp for when the block was created (timestamp), the number of transactions contained within the block (txNum), the total amount of MDC transacted by all transactions within the block (MDCAmount), total amount of MDC transaction fees paid to the miner of the block (fees), the size of the block in kilobytes (size), the height of the block in the blockchain (blockHeight), the amount of MDC paid to the miner as a reward (coinbase), a random color value (blockColor) such that the state of consensus can be observed in the model, and a JSON representation of the transactions included in the block (transaction-Data).

A general function was implemented to serialize the information of a block (toJSON). The serialized information includes that block's current variable and parameter states.

Transactions

Transactions contain parameters to define when the transaction was created (timeStamp), a unique string to identify the transaction (txID), a string array of who the transaction was sent from (from), a value for which agent the transaction is being sent to (to), the amount of MDC transacted (amount), the transaction fee (fee), and a string hash pointer that corresponds to a local record (hashPointer).

A general function was implemented to serialize the information of a transaction (toJ-SON). The serialized information includes that transaction's current variable and parameter states.

Local Records

Local records contain similar information parameters as transactions, such as the time the local record was created (timeStamp), the corresponding transaction ID on the blockchain

(txID), details regarding the patient interaction (data), and a corresponding hash pointer string (hashPointer).

A general function was implemented to serialize the information of a local record (toJSON). The serialized information includes that specific local record's current variable and parameter states.

4.2.2 Initialization of a Run by a User

Upon the launch of a simulation, the user is presented with an interface to select the type of blockchain architecture to model, as seen in Figure 4.3 below. The public permissionless architecture allows all agents to participate in the blockchain, including the ability to find blocks, broadcast transactions and look up data using the blockchain; in the private permissioned architecture, patients are not connected in the network, and are unable to view the blockchain, broadcast transactions, or find blocks.

Figure 4.3: Initial Interface with Blockchain Architecture Selection

The simulation run can then begin by pressing the "Run" button. This begins a simulation using December 4th, 2017 as an arbitrary start date. A population of 10 patients, 1 general practitioner, 1 hospital, and 1 treatment center are created and linked together in a ring-type network, such that each participant has exactly 2 connections. This enables any single participant to find a continuous path to any other participant.

All participants have a randomly assigned public key and private key. Since MDC balances are not functionally required for the demonstration model, all agents begin with a balance of 0 MDC. All participants also start with a single block in their blockchain, known as a genesis block, and no transactions or local records. All patients start in a healthy state in their state chart.

4.2.3 Processes and Interface Overview

The main interface is presented once the simulation begins, and can be seen below in Figure 4.4. Item 1 is a drop down bar that can be used to switch between different interfaces. The main interface is shown by default. Item 2 is the image representation of each participant. The color of the participant corresponds to the color of their most recent block. When all participants have the same color, this is an indication that they are in consensus as to the state of the blockchain. Item 3 is textual information regarding the participant, such as their name, and state. The state of a participant is the first 10 characters of the topBlockHash, and is another indicator that all participants are in consensus.

If the private permissioned option is chosen during initialization, the network architecture will differ as seen below in Figure 4.5. In this setup, patients are disconnected from the providers and do not have permission to read or write to the blockchain.

By using the drop down bar (item 1), one can switch to the interface for each participant. The interface for the patient is shown below in Figure 4.6. The interface for patients, the general practitioner, the treatment center and the hospital all contain the same information, except for item 1 in Figure 4.6. Item 1 is exclusive to the patient interface, and shows the current state of the patient in its statechart. Item 2 is a module used to demonstrate the functionality of the MDChain blockchain; by entering any patient's public key and pressing go, the edit box will populate that participant's history in JSON format. Relevant information regarding parameter values, variable values, and population sizes are shown in the top

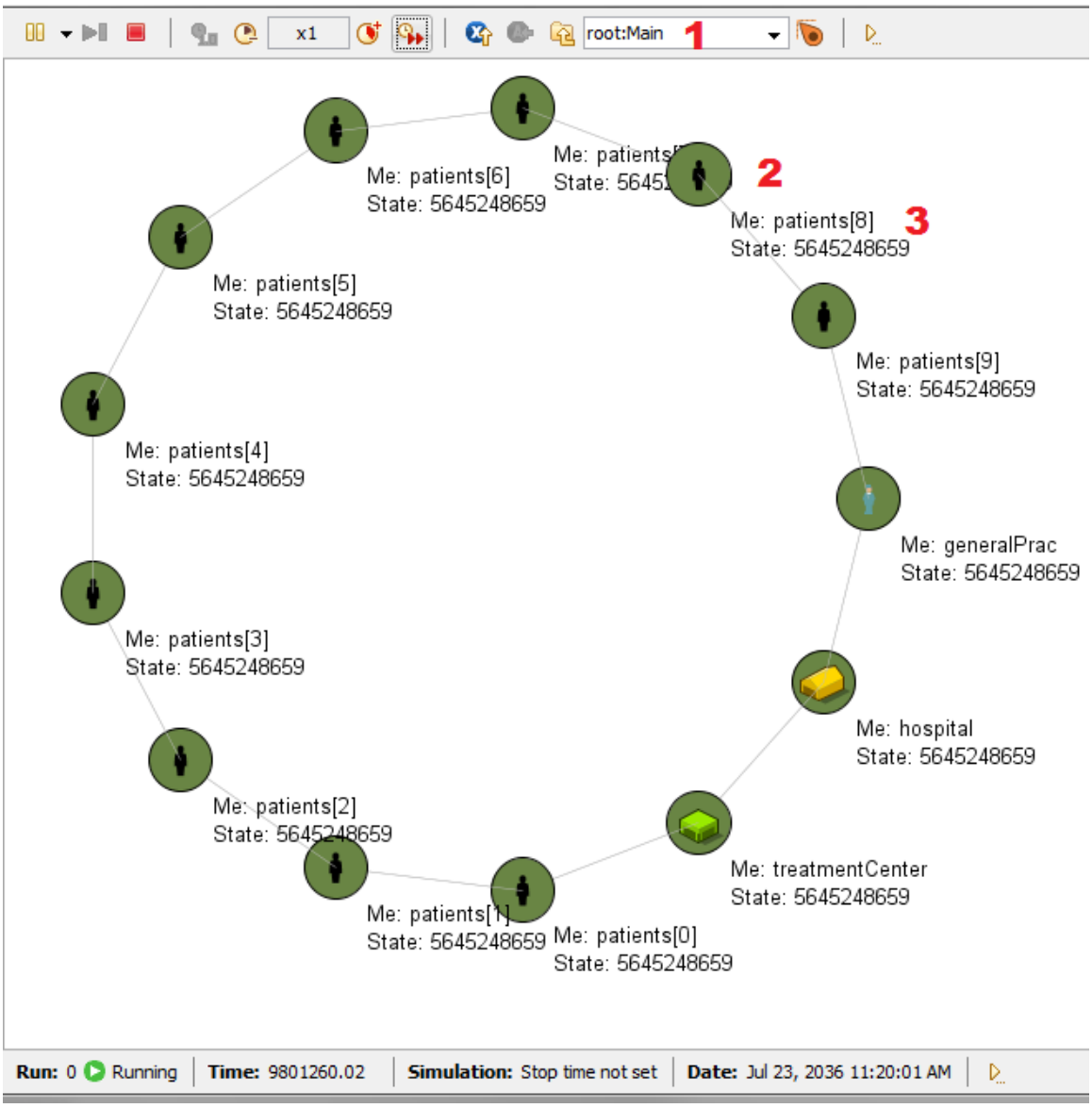


Figure 4.4: Main Interface for Public Permissionless Option

right.

All events are driven by the patient sending a message to the Hospital, GP, or TreatmentCenter. The associated provider will provide a response to the patient based on the type of message sent, and the provider will create a transaction where both the provider and patient destroy some tokens, while both act as signers of the transaction.

The main agent, and each participant agent have internal processes related to the min-

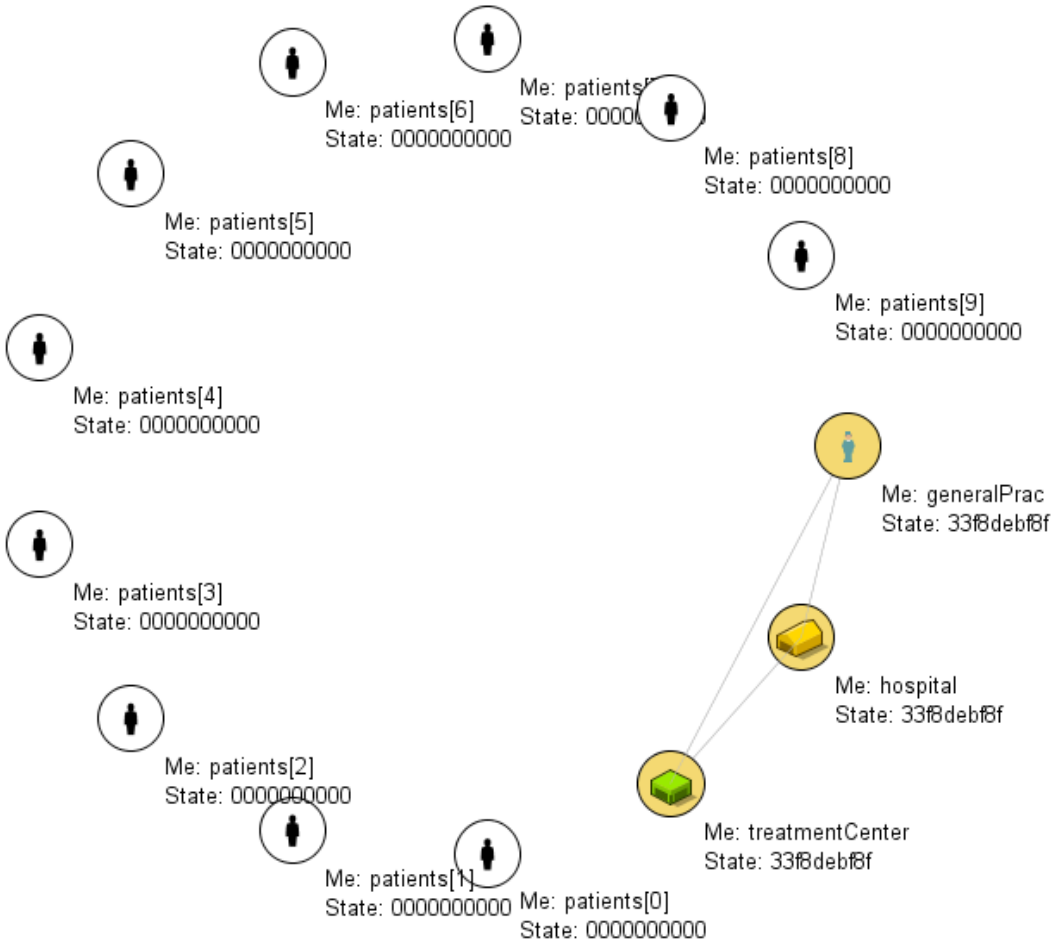


Figure 4.5: Main Interface for Private Permissioned Option

ing of blocks, relaying of new blocks, and the creation of new transactions and local records based on patient interactions with healthcare providers. Transactions, blocks and local records themselves do not contain any internal processes in the prototype model.

Main Processes

The main agent contains a single process that serves the purpose of determining which agent in the MDChain network will be responsible for mining the next block. The event "mineBlock" is triggered at an average rate of 12 times per year; when "mineBlock" is triggered, a message containing the text "winning miner" is sent to the agent, as seen in

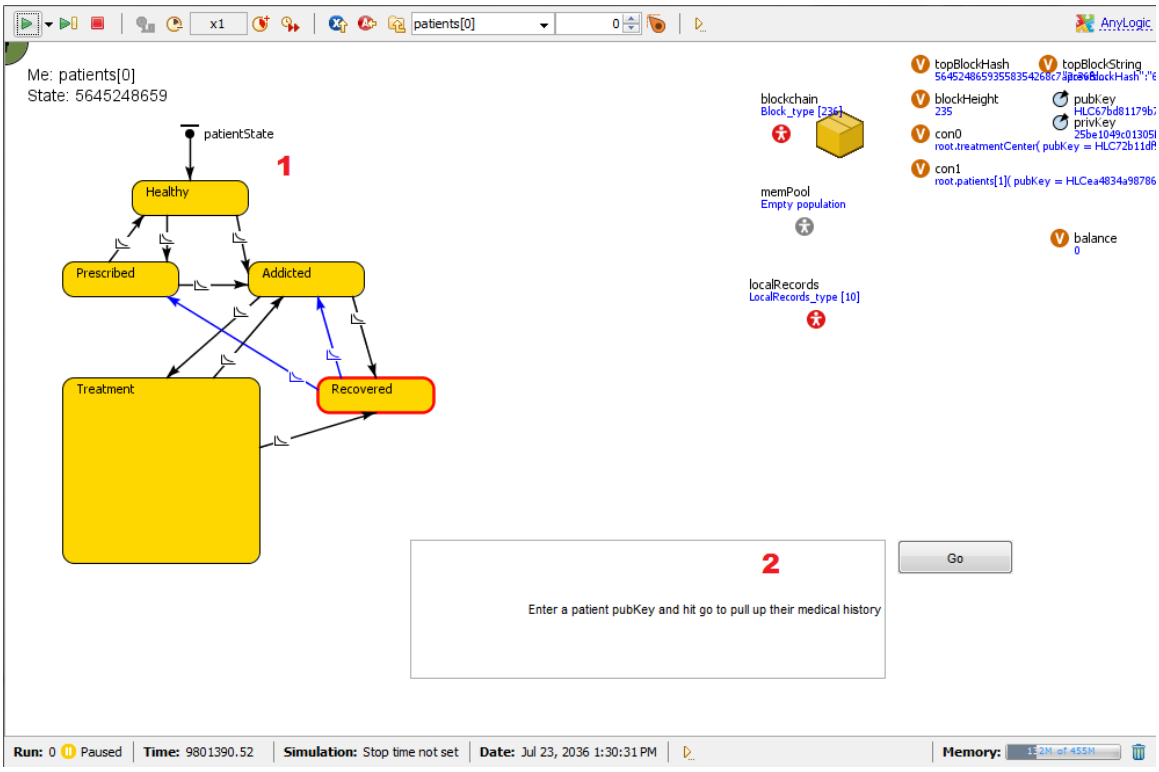


Figure 4.6: Participant Interface

Figure 4.7 below.

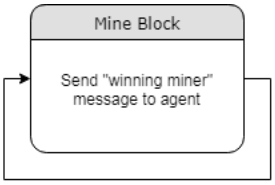


Figure 4.7: Main Mining Process

When using the public permissionless architecture, all participants have an equal probability of mining the next block; when using the private permissioned architecture, patients are not permitted to mine the next block.

Participant Processes

Every transaction that is created corresponds to either a block being mined, or a health-related state change in a patient agent. A participant will begin creating the next block after receiving a message of "winning miner" from the main agent, as seen in Figure 4.8 below; the agent will first add a transaction to their memPool to reward themselves 50 MDC using the buildTx function. No local record is created for this transaction since there is no patient interaction occurring. Next, the agent will utilize the mineBlock() function to iterate through each transaction in their memPool and include the details related to each transaction into the block. The memPool is then cleared back to 0. Finally, the agent will broadcast its newly constructed block to its neighbor, con0, with the message "BLOCK UPDATE MSG" and a JSON representation of the newly formed block via the sendBlocks() function.

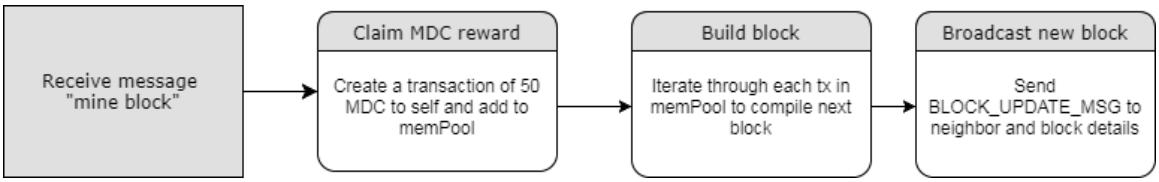


Figure 4.8: Participant Mining Block Process

When agents receive a message containing the text "BLOCK UPDATE MSG", they undergo the process shown below in Figure 4.9. First, the agent will check if their block height is lower than that of the block being sent to them. If the block height is equal or greater, no action is taken; however, if the block height of the agent is lower than that of the block being received, the agent will parse the message to create the new block. Finally, the agent will also broadcast the new block to its neighbor, con0, with the message "BLOCK UPDATE MSG" and a JSON representation of the newly formed block via the sendBlocks() function.

Each patient state change corresponds to a transaction and local health record. There

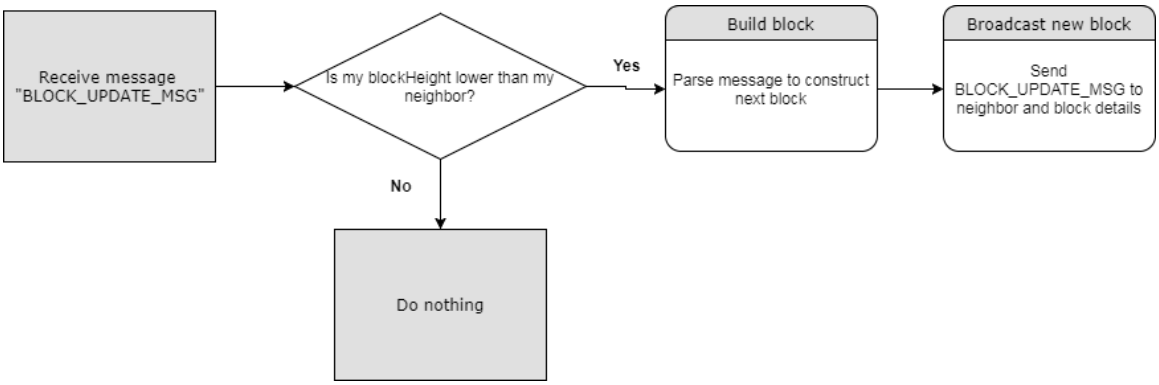


Figure 4.9: Participant Receiving Block Process

are a total of 10 possible unique state changes that correspond to 6 different processes being triggered, as seen in Figures 4.10-4.15. Each transition in the state chart occurs at a random time with an average rate of once per year, with equal probability of each event occurring. Diagrams are created to visualize each of these 6 unique processes. It may also be observed that all interactions are initialized by the patient, ideally giving the patient a greater degree of agency.

Prescription of pain killer A patient can be prescribed pain killers from their general practitioner when they move from a state of "healthy" or "recovered" to that of "prescribed", as seen below in Figure 4.10.

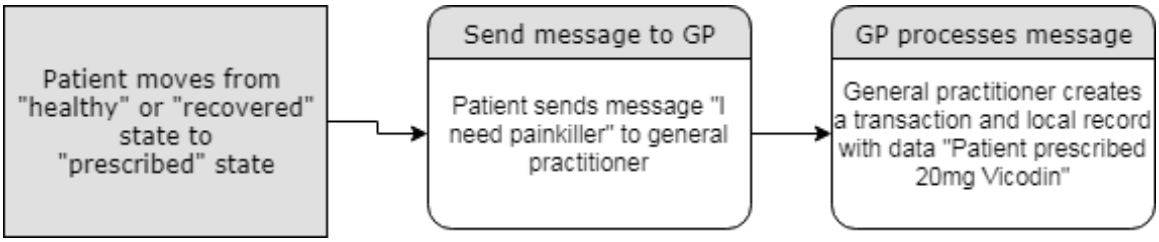


Figure 4.10: GP Prescribing Patient Painkiller Process

First the patient initiates a message to their general practitioner, "I need painkiller." Once the general practitioner receives this message, they will build a transaction and local record with the details of the painkiller prescription, when it happened, and a unique hash pointer. The transaction is seen in the blockchain as coming from the general practitioner

and the patient, and being sent to a burning (a specialized address with no associated private key) address, such that the MDC spent are destroyed. The transaction and local record exist in the general practitioner's environment.

Return to healthy state from prescribed state A patient can self-report to the blockchain that they have returned to a "healthy" state from a "prescribed" state, as seen below in Figure 4.11. This is meant to emulate a wearable device directly interacting with the MDC network. In the private permissioned architecture, this process is disabled since patients are unable to write to the blockchain.

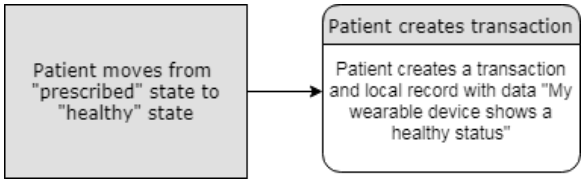


Figure 4.11: Patient Returning to Healthy State from Prescribed State Process

The patient will construct a transaction and local record that contains the details from the wearable device, when it happened, and a unique hash pointer. The transaction is seen in the blockchain as coming from the patient, to a burning address. The transaction and local record exist in the patient's environment.

Overdose from opioids A patient can move to a state of "addicted" from either the "healthy", "prescribed", "treatment" or "recovered" states, as seen below in Figure 4.12. When this occurs, the patient interacts with the hospital agent.

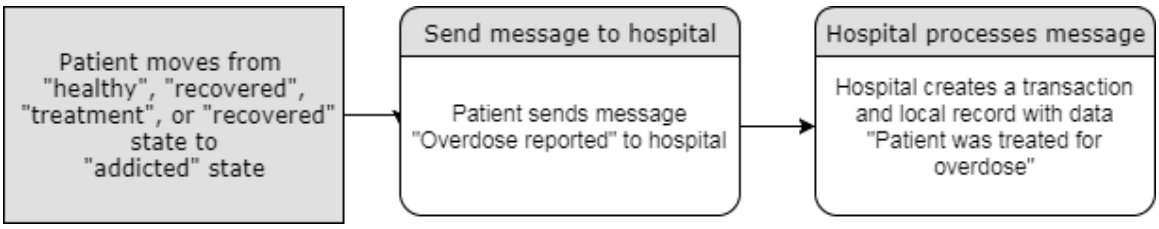


Figure 4.12: Hospital Diagnosing Patient with an Overdose Process

Initially the patient sends a message to the hospital, “Overdose reported.” Once the hospital receives this message, they will build a transaction and local record with the details of the overdose event, when it happened, and a unique hash pointer. The transaction is seen in the blockchain as coming from the hospital and the patient, and being sent to a burning address. The transaction and local record exist in the hospital’s environment.

Admission into therapy treatment center A patient can be admitted to a therapy treatment center and move into the state of “treatment” from a state of “addicted”, as seen below in Figure 4.13.

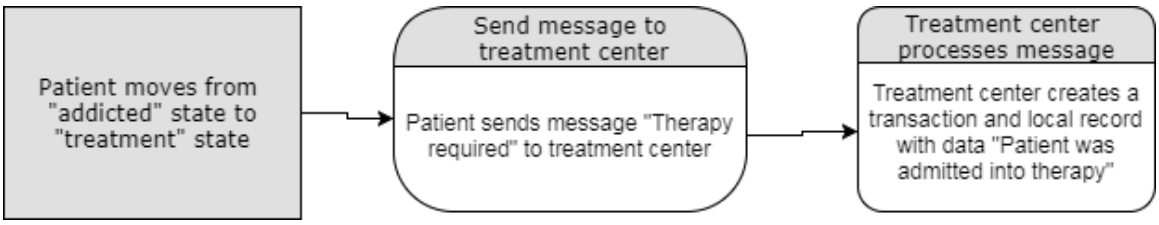


Figure 4.13: Treatment Center Admitting Patient to Therapy Process

First the patient initiates a message to the treatment center, “Therapy required.” Once the treatment center receives this message, they will build a transaction and local record with the details of the therapy needed, when the request was made, and a unique hash pointer. The transaction is seen in the blockchain as coming from the treatment center and the patient, and being sent to a burning address. The transaction and local record exist in the treatment center’s environment.

Recovery state from addicted state A patient can self-report to the blockchain that they have overcome their addiction, and move from the “addicted” state to the “recovered” state, as seen below in Figure 4.14. This emulates a wearable device interacting with the MDC network directly.

The patient constructs a transaction and local record that contains the details from the wearable device, when recovery occurred, and a unique hash pointer. The transaction is

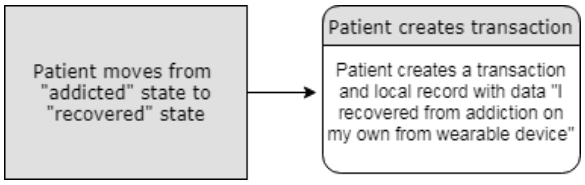


Figure 4.14: Patient Returning to Recovered State from Addicted State Process

seen in the blockchain as coming from the patient, to a burning address. The transaction and local record exist in the patient's environment. In the private permissioned architecture, this process is disabled since patients are unable to write to the blockchain.

Recovery state from therapy treatment center Patients can also move to a state of "recovered" from a state of "treatment", as seen below in Figure 4.15; this requires the patient to interact with the treatment center.

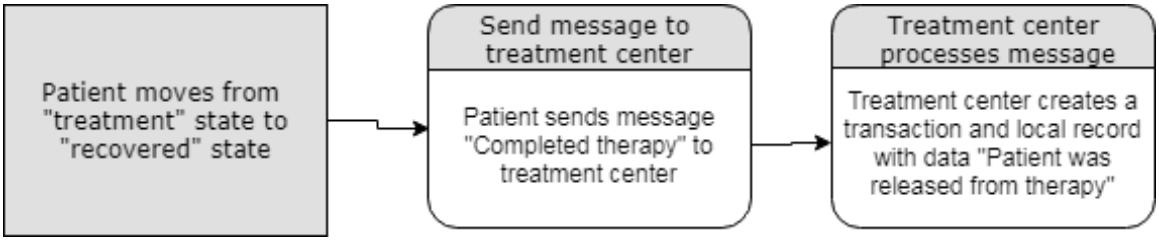


Figure 4.15: Treatment Center Discharging Patient from Therapy Process

First the patient initiates a message to the treatment center, "Completed therapy." Once the treatment center receives this message, they will build a transaction and local record with the details of the completed therapy, when it happened, and a unique hash pointer. The transaction is seen in the blockchain as coming from the treatment center and the patient, and being sent to a burning address. The transaction and local record exist in the treatment center's environment.

4.2.4 Concepts and Assumptions

Assumptions were made in the construction of the model, both pertaining to the architecture of the blockchain protocol, and the opioid addiction cycle. The assumptions made in regards to the blockchain protocol are presented below:

- Participants do not broadcast their memPool, and are thus asynchronous
- All participants have an equal chance at mining the next block
- All transactions are 1 kilobyte in size
- All transactions are for 1 MDC with a fee of 0.001 MDC
- Blocks have no size limit
- Transactions do not affect participant balances
- MDC sent to a burn address are removed from circulation
- The creator of the transaction will always store the corresponding local record of that transaction

As stated earlier, the processes correspond to an assumed opioid process functionality. Patients switch states at a random time at an average rate of 1 time per year using an exponential distribution for the time between state changes.

4.2.5 Model Exploration

A single run of the simulation is performed for a 50 year period in simulated time. After 50 years have passed, the simulation will be paused and a demonstration of the ability to compile patient records will be performed with the public permissionless architecture. This will be done by showing that from the hospital interface the medical history of a

random patient, in this case patient 4, can be compiled using only that agent's public key. Screenshots of the private architecture will also be presented for comparison.

The simulations will then be empirically evaluated for their ability to demonstrate which criteria described in the earlier matrices are fulfilled by the constructed model, and present a comparison of each architecture. The scores can then be used to facilitate further discussion as to which requirements the proposed blockchain architecture fulfills naturally, which depend on the way a blockchain system is designed, and which may be solved with complementary solutions.

4.3 Results

A single run of the simulation using a random number seed of 2 and a public permissionless blockchain was performed for a 50 year period of simulated time, beginning on December 4th, 2017. The complete resulting blockchain and each local record were exported to database tables.Record compilation was performed for patient 4, having a public key value of HLC182f4589c[...]5249dac4a6, as seen below in Figure 4.16.

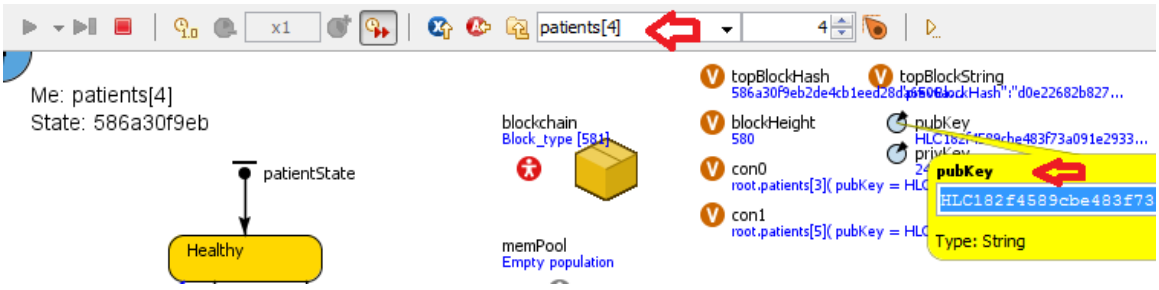


Figure 4.16: Obtaining Public Key Value for Patient 4

The interface was then switched to the hospital's, and the public key was entered into the edit box, as seen below in Figure 4.17.

After pressing go, the full record is populated in the edit box, as seen below in Figure

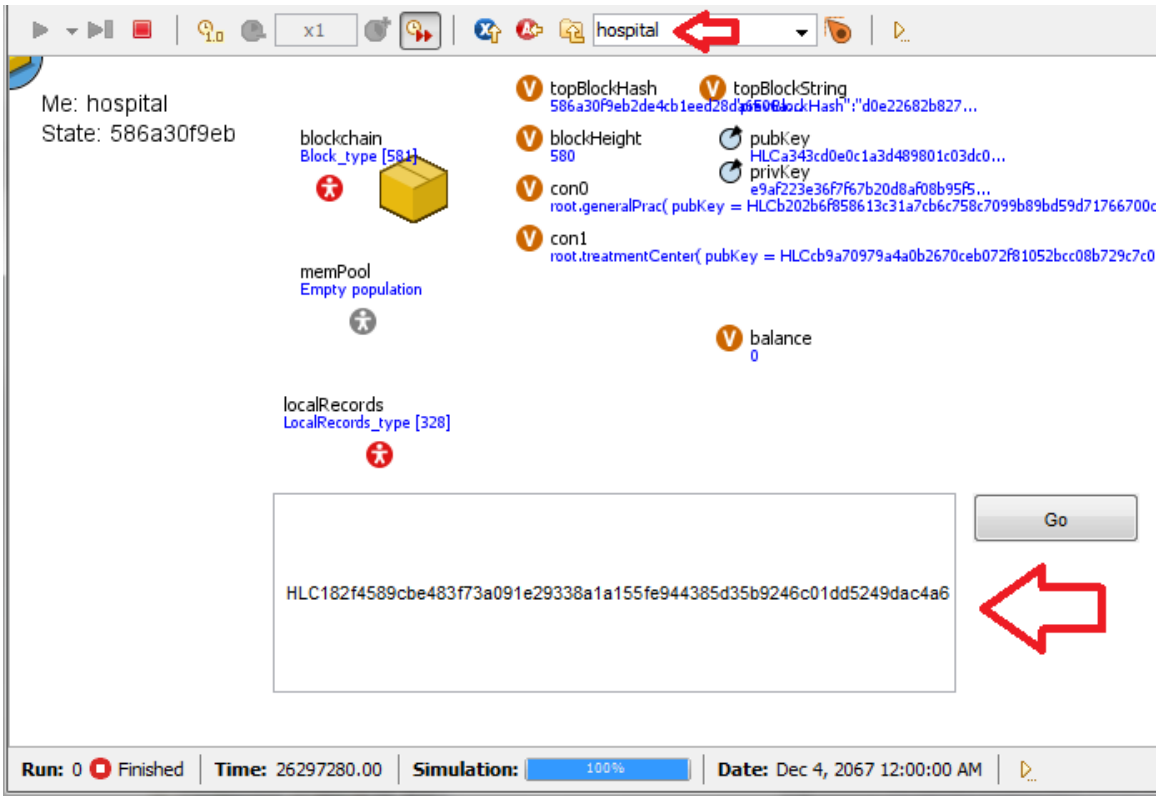


Figure 4.17: Filling Edit Box in Hospital Interface with Public Key Value for Patient 4

4.18.

An excerpt of the first 2 records is also provided below.


```
{"timeStamp":"86236.89275113701","txID":"c7b19c1aa6f37dc322483263b-
dc3ddc8758b4290b9af16cb06fcfc970e217c7f","from":"HLC182f4589cbe483-
f73a091e29338a1a155fe944385d35b9246c01dd524dac4a6,null","to":"HLCb-
urnAddress0x","amount":"1.0","fee":"0.001","hashPointer":"7bcea088-
5e4166d7a9ea808daa214116bf4226b63759a94852504832fe529760"}
My wearable device shows a healthy status85121.81857470065
{"timeStamp":"299060.28129702253","txID":"716f226c56fb4d1c165c81b-
75f4ec0e99b96edbaa1e6a3558a4daeb6592fb676","from":"newlymintedcoi-
ns","to":"HLC182f4589cbe483f73a091e29338a1a155fe944385d35b9246c01-
```

dd5249dac4a6","amount":"50.0","fee":"0.001","hashPointer":"7774a6-
7dde1b67eb8ded3e0e1f69f1bce1446556b2fd4b1ba951adb6693f6b3c"}

local record not found

The patients compiled record was then compared against the exported databases to verify proper execution.

Three tables, Tables 4.4 through 4.6, are constructed that score the presence of technical features, operational capabilities, and healthcare system requirements respectively. Each table presents whether the constructed model directly supports these technical features, operational capabilities, or requirements for healthcare. Each item is provided a code of “Low”, “Medium”, or “High”:

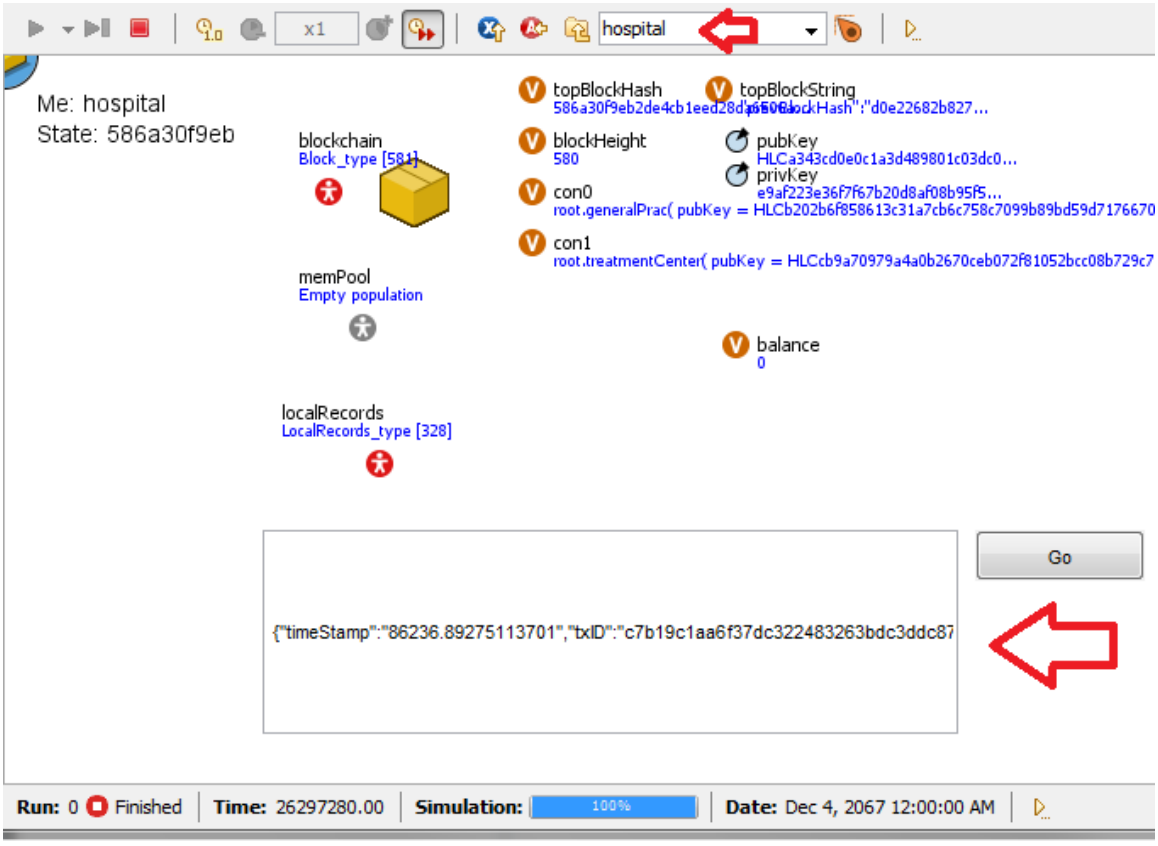


Figure 4.18: Populated Records for Patient 4 from Within the Hospital’s Interface

- Low - the constructed model did not have a component or function to support the corresponding item.
- Medium - the constructed model had a component or function to support the corresponding item, but was not explicitly utilized.
- High - the constructed model directly utilizes a component or function to support the corresponding item.

The justification for the coding is explained in the discussion section.

Table 4.4: Scoring Ability of Model to Demonstrate Blockchain Technical Features

Feature	Immutable Ledger	Consensus	Smart Contracts	Multi-Sig	Cryptography	Asset Digitization	P2P
Public	High	High	Low	Medium	High	Low	High
Private	Medium	High	Low	Medium	High	Low	Medium

Table 4.5: Scoring Ability of Model to Demonstrate Blockchain Operational Capabilities

Capability	Transfer of Value	Security	Auditability	Decentralization of Trust
Public	High	Medium	High	High
Private	High	Medium	High	Medium

Table 4.6: Scoring Ability of Model to Demonstrate Healthcare Requirements

Requirement	**Public**	**Private**
Costs Reduction	Medium	Medium
Fraud Prevention	Medium	Medium
Identity Management	High	Medium
Record Availability	High	Medium
HIPAA Compliance	Low	Medium
Universality of Record	High	High
Auditability	High	High
Reconciliation	High	High
Interoperability	Medium	High
Encourage Patient Engagement	Medium	Low

4.4 Discussion

As seen in Figure 4.18, the data relevant to a patient interaction can be found by combining on-chain data (synchronous data stored in the blockchain) with off-chain data (asynchronous local records stored in each node). A key design decision when building blockchain systems is how much data is stored on the blockchain itself, as this ultimately has critical implications related to scalability, privacy and interoperability [50]. For this system, it was decided that data regarding the actual patient visit would be held off-chain. In the model, this only included short messages describing the interaction that occurred; however, in reality this would likely include far more information, including information protected by HIPAA regulations. Storing data on-chain would allow any participant in this blockchain system to read the data; in a permissionless implementation of a healthcare blockchain, this may introduce difficulty in satisfying current regulations. The data could be encrypted while being stored on-chain, but this would introduce performance drawbacks and still increase the overall attack surface on the medical records, since a replica of the blockchain data is stored on each participant's node.

By allowing records to be stored offline, legacy healthcare record systems can still be used. This also means that HIPAA compliance can still be performed off-line. Utilizing the blockchain-based system in this manner simply enables interoperability, and sharing of data in a secure manner; however, simply using a blockchain does not make an entity HIPAA compliant or secure on its own. The proposed model would act as a second layer to the current healthcare infrastructure, as opposed to outright replacing it.

Model Demonstration of Technical Features

Table 4.4 outlines how well the model demonstrates the implementation of an immutable ledger, consensus among nodes, smart contracts, multi-signature wallets, cryptography, asset digitization and a peer-to-peer network.

Immutable ledger was given “High”, as the model presents a ledger that is cryptographically immutable. Since each subsequent block in the model contains a SHA256 digest hash of the block preceding it, any modification of a previous block could be detected since the hash values would not match.

Consensus was also given “High”, since all participant nodes remain in the same state throughout the running of the model. The current model did not include any malicious actors but in reality malicious actors may be able to participate in permissionless systems. This makes the task of achieving consensus more difficult than in a permissioned blockchain system. General purpose algorithms utilized in current mainstream cryptocurrencies, such as *proof-of-work* [3] and *proof-of-stake* [13], may be suitable for a healthcare based ecosystem.

Smart contract functionality was given “Low”, as the model did not directly demonstrate any smart contract use; however, the use of smart contracts may still be immensely valuable in the space of healthcare. The ability for insurance companies to automatically perform payments by leveraging a blockchain system may significantly reduce operating costs for payers, and ultimately contribute to lowered treatment costs. Smart contracts may also help to enable interoperability [52].

Multi-signature usage was given “Medium”, as the model utilized multi-signature schemes in an abstract manner, albeit no cryptographic signing was taking place. For example, when a transaction in the model is made between a patient and a healthcare provider, the transaction includes the public key of both participants; this is meant to be an abstract representation of the ability for a system to be constructed in such a manner that both participants are required to sign a transaction for it to be valid, implying that consent was given by each participant.

Cryptography was given “High”, since each block was cryptographically linked using the SHA256 algorithm. In addition, each hash pointer was generated by conducting a SHA256 digest of the patient interaction data itself.

Asset digitization was given “Low”, as the proposed healthcare system did not incorporate any type of physical assets being represented digitally on the blockchain; however, this does not rule out the use of digital assets in a healthcare blockchain-based system. One use case involves developing a secure marketplace for the exchange of verified medical research and medical record data [53].

Peer-to-peer networking was given “High”, since the model leveraged a peer-to-peer network architecture. Each participant node was connected to only 2 other neighboring nodes. In a real blockchain-based system, each node can have more than 2 connections, but is unlikely to require being connected to every single node. Peer-to-peer networks have the ability to support large numbers of users while maintaining interconnectivity with limited hops.

Model Demonstration of Operational Capabilities

Table 4.5 outlines the codified results for blockchain operational capabilities, which are enabled by the existence of the aforementioned technical features.

The ability to transfer value is demonstrated in the model by the existence of MDC token, and is thus given a score of “High”.

Security is given “Medium”, since some security properties are implied, such as the immutability of the ledger; however, because no malicious actors were demonstrated and a rudimentary consensus algorithm was implemented, security was only given “Medium”.

Auditability was given “High” since the ability to recover patient records based on blockchain data was directly demonstrated. Any node could compare the local records given by any firm to the hashed data stored on the blockchain to verify the integrity of any particular record; if the hash of the data stored on the chain does not match the hash of the local record returned, then the data has been tampered with.

Decentralization of trust was also given “High” since no single node in the system was entrusted with storing the entire set of local records, which is contrary to the way current

healthcare systems operate.

Model Demonstration of Requirements

Coding of healthcare system requirements is shown in Table 4.6.

Cost reduction was given "Medium", as the model did not directly demonstrate any economic elements; however, higher availability of data in healthcare can lead to reduced costs in various ways: improved treatment efficacy leading to reduced expenditures on treatment, reduction in administrative costs due to record reconciliation and auditing, reduction in fraud due to data integrity, and more.

Fraud prevention was also given "Medium", since the model did not include any direct demonstrations of malicious actors or fraudulent activity. The model, however, does offer elements that would contribute to a reduction in fraud through record auditing. The inclusion of identity management may also contribute to reduction in fraud, as the private and public key pair can be used to authenticate the user.

The model assigns public and private keys to each system participant, and acts as a built-in identity management component, giving identity management a score of "High". Blockchain-based systems utilize public/private key pairs to authenticate the control of addresses, and in a healthcare system can also be used as authentication mechanisms. These keys can also be used in combination with legacy identity management systems. For example, a link could be created in a legacy identity management system and a blockchain-based system by simply noting in the legacy system the public key belonging to the user in the blockchain system. While the proposed model does not actually use private keys for any functions, in reality the private key would be used to offer the patient ownership over their records.

As demonstrated in the results section, the data from local records is available to all participants in the system, giving record availability a score of "High". In the proposed system, no redundant local record stores were utilized; however, in reality, multiple copies

of local records could be maintained, offering improved record availability. For example, if a patient goes to their doctor and has a checkup performed, the doctor will naturally store the record as they already do in legacy systems, but in the blockchain-based system, perhaps the patient themselves can also have a mechanism by which to store the data themselves. New businesses could be created that offer to rent storage space of encrypted healthcare records for users in exchange for MDC tokens. This allows the data to be highly available to all users in the system, without burdening all users to store every transaction that occurs within the system.

HIPAA compliance is not directly addressed through the proposed model, and is thus given a score of "Low". The proposed system still requires that the off-chain local records store data in a HIPAA compliant manner. The proposed system does not positively or negatively affect HIPAA compliance directly.

Universality of records is given a score of "High" since regardless of which node is compiling a patient's record, the patient's record is complete. In current legacy systems, healthcare providers that are not part of a data sharing initiative will be unable to collect the complete history of that patient. Current data sharing initiatives are operated by central entities, which do not properly align the incentives of all participants to share their data openly. By creating a permissionless blockchain that is open to all participants, and incentivized by an underlying currency, data sharing could ideally occur on a more open playing field. Certain architectures could provide MDC token incentives for users when contributing data, and require the payment of MDC tokens to access data on the network.

Blockchain-based systems provide inherent tools that enable auditing of records. The current model developed displays the ability to compile a record, and by combining local records and on-chain data, one can perform an audit of the records by checking the hash values provided. Auditability was given a score of "High". This also demonstrates the ability to reconcile records stored in varying locations, leading to a score of "High" for reconciliation.

Local records could be stored in any format. As long as the hashed record of the data provided matches that of the on-chain data, the data can be trusted to be an original version. If the data were all stored on-chain, the same format would likely need to be used to enable interoperability; however, by storing the records off-chain, the blockchain layer acts as an intermediary layer that enables interoperability of incompatible systems. Since all records were stored in the same format, we believe this feature was not directly demonstrated, but was supported, and thus interoperability was given "Medium".

The MDC token was not directly used in any manner to encourage patient engagement; however, the inclusion of a native cryptocurrency enables new models to encourage patient engagement. Some examples would include rewarding patients for contributing data to the MDChain blockchain itself, or rewarding patients for sharing data with third party researchers. Since this functionality was not directly demonstrated, a score of "Medium" was given.

4.4.1 Future Work

While this model demonstrates the efficacy and feasibility of a blockchain-based healthcare system, future work should be done to explore optimal design architecture. These topics may include:

- Simulation and modeling of varying token incentive structures. The economics of a blockchain-based ecosystem with a native cryptocurrency should be explored in depth, as they may introduce an opportunity to create new business models within healthcare. This may include modifications to the mining and initial distribution of the token, fees associated with using the blockchain network, and/or models to increase system participation.

- Further exploration as to what data should be stored on chain versus off chain. Data stored on chain must be replicated across every participating node, which affects the

scalability of such a system.

- Internal processes could be added to transactions to model the execution of smart contracts. Smart contracts could be used to manage permissions to certain data, facilitate automated reporting, expedite payment requests, and other potential use cases.

- Simulating and estimating resource requirements for system participants. These requirements will limit the amount of data and number of participants that can use such a system. Simulation tools have already been used to predict latency of blockchain-based systems [43].

- Privacy and security concerns for data stored, shared or secured in a blockchain. This would also include research as to the possibility of HIPAA compliant blockchain architectures.

Demonstrating the requirements of a blockchain-based healthcare system with a basic simulation must be fine tuned specifically to the healthcare industry, which introduces new challenges in addition to demonstrating the underlying operational capabilities and technical features that a blockchain can offer. Varying design decisions can be made when constructing a blockchain-based healthcare system that affect the ability of certain requirements to be met. Blockchains may be used as a valuable tool for the transfer and auditing of information within a healthcare ecosystem, but further research remains to be done that explores the performance trade offs involved when moving from legacy systems to the integration of more decentralized blockchain-based systems. Blockchains and cryptocurrency technologies are still in their infancy, and it may take several years until they can be used in commercial production environments, but their potential remains promising.

Conclusion

Blockchains hold the potential to disrupt various industries in addition to finance alone. In this thesis, agent based modeling tools were used to model and study the Bitcoin blockchain, as well as develop a proposed hypothetical blockchain-based healthcare model, MDChain. Blockchain-based systems and healthcare are both complex, dynamic systems making them prime targets for study through model development and simulation. Both models were developed using AnyLogic modeling software, which is written using the Java programming language. The work also provides a comprehensive review of previous modeling and simulation work that has been done in the blockchain space, and the healthcare space. It is the first known attempt to apply modeling and simulation tools to the blockchain healthcare space, and is a novel approach in studying blockchain-based healthcare applications.

The model that was developed to simulate the Bitcoin blockchain was used to demonstrate the mechanics of a blockchain, and also used to make specific predictions about the real Bitcoin network. The model predicted that fees would continue to rise throughout early 2018; however, due to changes in the behavior of the users in the real Bitcoin network, this prediction was ultimately false. We also suggest that an effective scaling solution for Bitcoin would involve more than simply tweaking parameters related to block time and block size. Future modifications could be made to the current model to facilitate the exploration of topics surrounding cryptocurrency mining, consensus algorithms, economic elements such as market prices, scaling proposals, and more. These modifications were outlined in further detail in subsection 3.4.3.

The model built for blockchain-based care coordination within the healthcare ecosystem demonstrates how such a system may be constructed, and is used to further facilitate discussion surrounding effective design of such a system. Various design decisions, such as the presence of a native cryptocurrency token, permitted participants in the network, what data to store on the blockchain and others are considered. The model is also used to educate individuals as to the technical features and operational capabilities of blockchains and how those operational capabilities contribute to addressing requirements within the healthcare ecosystem. We find that a blockchain-based care coordination system strongly contributes to enabling record availability, record universality, auditing of records, reconciliation of records, and interoperability of records.

Future work should be done to explore how blockchain-based healthcare systems can properly facilitate HIPAA compliance, and encourage patient engagement. Trade offs must be considered when using a blockchain compared to traditional centralized database solutions or current existing distributed database management systems such as scalability, governance, and latency issues. This could be done by exploring varying token incentive structures, data to be stored on chain versus off chain, estimating specific resource requirements, utilization of smart contracts, and constructing permissioning architectures. These modifications were outlined in further detail in subsection 4.4.1.

Bibliography

[1] Ashok Kay Kanagarajah, Peter Lindsay, Anne Miller, and David Parker. An exploration into the uses of agent-based modeling to improve quality of healthcare. In Ali Minai, Dan Braha, and Yaneer Bar-Yam, editors, *Unifying Themes in Complex Systems*, pages 471–478, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-85081-6.

[2] Brian L. Heath. The history, philosophy, and practice of agent-based modeling and the development of the conceptual model for simulation diagram. Master's thesis, Wright State University, 2010. URL https://etd.ohiolink.edu/pg_10?0::NO:10:P10_ACCESSION_NUM:wright1269176275.

[3] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008. URL https://bitcoin.org/bitcoin.pdf.

[4] Coinmarketcap.com, 2017. URL https://coinmarketcap.com/charts/.

[5] Jesse Damiani. Simplyvital health is using blockchain to revolutionize healthcare, 2017. URL https://www.forbes.com/sites/jessedamiani/2017/11/06/simplyvital-health-blockchain-revolutionize-healthcare/#3777e022880a.

[6] *Help - AnyLogic Simulation Software*, 2018. URL https://help.anylogic.com/index.jsp.

[7] Harsh Kupwade Patil and Ravi Seshadri. Big data security and privacy issues in healthcare. In Peter P. S. Chen and Hemant Jain, editors, *2014 IEEE International Congress on Big Data*, pages 762–765, Los Alamitos, California and Washington and Tokyo, June 2014. Conference Publishing Services, IEEE Computer Society. ISBN 978-1-4799-5057-7. doi: 10.1109/BigData.Congress.2014.112.

[8] Willem G Van Panhuis, Proma Paul, Claudia Emerson, John Grefenstette, Richard Wilder, Abraham J Herbst, David Heymann, and Donald S Burke. A systematic review of barriers to data sharing in public health. *BMC public health*, 14(1):1144, 2014. doi: 10.1186/1471-2458-14-1144.

[9] Ellen M. Schultz and Kathryn M. McDonald. What is care coordination? *International Journal of Care Coordination*, 17(1-2):5–24, 2014. doi: 10.1177/2053435414540615. URL https://doi.org/10.1177/2053435414540615.

[10] Bernard P. Zeigler. The role of modeling and simulation in coordination of health care. In *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*, pages IS–5–IS–16, 2014. URL http://ieeexplore.ieee.org/document/7094986/.

[11] Karla L Caballero Barajas and Ram Akella. Dynamically modeling patient's health state from electronic medical records: A time series approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 69–78. ACM, 2015. URL https://dl.acm.org/citation.cfm?id=2783289.

[12] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin net-

work. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE, 2013. ISBN 978-1-4799-0515-7. doi: 10.1109/P2P.2013.6688704.

[13] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. 2014. URL https://github.com/ethereum/wiki/wiki/White-Paper.

[14] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In Matteo Maffei and Mark Ryan, editors, *Principles of Security and Trust*, pages 164–186, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg. ISBN 978-3-662-54455-6.

[15] *hyperledger-fabricdocs Documentation*, 2018. URL https://readthedocs.org/projects/hyperledger-fabric/downloads/pdf/latest/.

[16] Meni Rosenfeld. Overview of colored coins. Technical report, bitcoil.co.il, 2012. URL https://bitcoil.co.il/BitcoinX.pdf.

[17] Marc Pilkington. Blockchain technology: Principles and applications. In F. Xavier Olleros and Majlinda Zhegu, editors, *Research Handbook on Digital Transformations*, Research Handbooks in Business and Management series, chapter 11, page 234. Edward Elgar Publishing, 2016. ISBN 9781784717766.

[18] bitcoin/validation.h at master bitcoin/bitcoin, 2018. URL https://github.com/bitcoin/bitcoin/blob/master/src/validation.h.

[19] Sha-256 - bitcoin wiki, 2016. URL https://en.bitcoin.it/wiki/SHA-256.

[20] Ameer Rosic. The science behind cryptocurrencies cryptography -

blockgeeks, 2017. URL https://blockgeeks.com/guides/cryptocurrencies-cryptography/.

[21] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16. ACM, 2016.

[22] Khaled Baqer, Danny Yuxing Huang, Damon McCoy, and Nicholas Weaver. Stressing out: Bitcoin "stress testing". In Jeremy Clark, Sarah Meiklejohn, Peter Y.A Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff, editors, *International Conference on Financial Cryptography and Data Security*, pages 3–18, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg. ISBN 978-3-662-53357-4. doi: 10.1007/978-3-662-53357-41.

[23] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *Proceedings, 2015 IEEE Symposium on Security and Privacy*, pages 104–121, Los Alamitos, California, 2015. IEEE Computer Society. ISBN 978-1-4673-6949-7. doi: 10.1109/SP.2015.14.

[24] Steven Goldfeder, Joseph Bonneau, J. A. Kroll, and E. W. Felten. Securing bitcoin wallets via threshold signatures. 2014. URL https://www.cs.princeton.edu/~stevenag/bitcoin_threshold_signatures.pdf.

[25] Florian Tschorsch and Bjorn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18 (3):2084–2123, 2016. doi: 10.1109/COMST.2016.2535718.

[26] Jordi Herrera-Joancomartí and Cristina Pérez-Solà. Privacy in bitcoin transactions: New challenges from blockchain scalability solutions. In Vicenç Torra, Ya-

suo Narukawa, Guillermo Navarro-Arribas, and Cristina Yañez, editors, *Modeling Decisions for Artificial Intelligence: 13th International Conference, MDAI 2016, Sant Julià de Lòria, Andorra, September 19-21, 2016. Proceedings*, pages 26–44. Springer International Publishing, Cham, 2016. ISBN 978-3-319-45656-0. doi: 10.1007/978-3-319-45656-03.

[27] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. On scaling decentralized blockchains. In Jeremy Clark, editor, *Financial cryptography and data security*, volume 9604 of *LNCS sublibrary. SL4 - Security and cryptology*, pages 106–125. Springer, Heidelberg, 2016. ISBN 978-3-662-53356-7. doi: 10.1007/978-3-662-53357-48.

[28] Jameson Lopp. Could spv support a billion bitcoin users? sizing up a scaling claim, 2017. URL https://www.coindesk.com/spv-support-billion-bitcoin-users-sizing-scaling-claim/.

[29] BitFury Group. Block size increase. Technical report, BitFury Group, 2015. URL http://bitfury.com/content/5-white-papers-research/block-size-1.1.1.pdf.

[30] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. 2016. URL https://lightning.network/lightning-network-paper.pdf.

[31] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling blockchain innovations with pegged sidechains. 2014. URL http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains.

[32] Pieter Wuille. Segregated witness and its impact on scalability, 2015. URL https://www.youtube.com/watch?v=NOYNZB5BCHM.

[33] Bitcoin cash - peer-to-peer electronic cash, 2017. URL https://www.bitcoincash.org/.

[34] Mike Belshe. Segwit2x final steps, 2017. URL https://lists.linuxfoundation.org/pipermail/bitcoin-segwit2x/2017-November/000685.html.

[35] Jimmy Song. Segwit2x bugs explained, 2017-11-20. URL https://bitcointechtalk.com/segwit2x-bugs-explained-8e0c286124bc.

[36] Testnet, 2018. URL https://en.bitcoin.it/wiki/Testnet.

[37] Stefan Bornholdt and Kim Sneppen. Do bitcoins make the world go round? on the dynamics of competing crypto-currencies. *arXiv preprint arXiv:1403.6378*, 2014. URL https://arxiv.org/pdf/1403.6378.pdf.

[38] Miles Carlsten, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167, 2016.

[39] Luisanna Cocco, Giulio Concas, and Michele Marchesi. Using an artificial financial market for studying a cryptocurrency market. *Journal of Economic Interaction and Coordination*, 12(2):345–365, 2017. URL https://arxiv.org/pdf/1406.6496.pdf.

[40] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. Decentralized execution of smart contracts: Agent model perspective and its implications. In *International Conference on Financial Cryptography and Data Security*, pages

468–477, 2017. URL https://link.springer.com/chapter/10.1007/978-3-319-70278-0_29.

[41] Pietro Terna, Marco Maggiora, and Luigi Battistoni. Emerging cryptocurrency trust in an agent–based model. Master's thesis, Universita di Torino, 2017. URL http://terna.to.it/tesi/battistoni.pdf.

[42] Marek Laskowski. A blockchain-enabled participatory decision support framework. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 329–334. Springer, 2017. URL https://papers.ssrn.com/Sol3/papers.cfm?abstract_id=2927692.

[43] Rajitha Yasaweerasinghelage, Mark Staples, and Ingo Weber. Predicting latency of blockchain-based systems using architectural modelling and simulation. In *2017 IEEE International Conference on Software Architecture*, pages 253–256, Los Alamitos, California, 2017. Conference Publishing Services, IEEE Computer Society. ISBN 978-1-5090-5729-0. doi: 10.1109/ICSA.2017.22.

[44] JULIA R. NORGAARD, HAROLD J. WALBERT, and R. AUGUST HARDY. Shadow markets and hierarchies: Comparing and modeling networks in the dark net. *Journal of Institutional Economics*, pages 1–23, 2018. URL https://www.researchgate.net/profile/Julia_Norgaard/publication/309421955_Shadow_Markets_and_Hierarchies_Comparing_and_Modeling_Networks_in_the_Dark_Net/links/580ff8b608ae009606bb8dfd.pdf.

[45] Announcing the blockchain challenge — newsroom — healthit.gov, 2016. URL https://www.healthit.gov/newsroom/blockchain-challenge.

[46] IBM Global Business Services Public Sector Team. Blockchain: The

chain of trust and its potential to transform healthcare – our point of view. Technical report, IBM Global Business Services Public Sector Team, 2016. URL https://www.healthit.gov/sites/default/files/8-31-blockchain-ibm_ideation-challenge_aug8.pdf.

[47] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *Open and Big Data (OBD), International Conference on*, pages 25–30. IEEE, 2016. URL http://ieeexplore.ieee.org/abstract/document/7573685/.

[48] Kefa Rabah. Challenges & opportunities for blockchain powered healthcare systems: A review. *Mara Research Journal of Medicine & Health Sciences-ISSN 2523-5680*, 1(1):45–52, 2017. URL https://medicine.mrjournals.org/index.php/medicine/article/view/6.

[49] Tsung-Ting Kuo, Hyeon-Eui Kim, and Lucila Ohno-Machado. Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association : JAMIA*, 24(6):1211–1220, 2017. doi: 10.1093/jamia/ocx068.

[50] Michael Dufel. A new paradigm for health information exchange. Technical report, Peer Health, 2016. URL http://peerhealth.io/download-whitepaper/.

[51] Peng Zhang, Michael A Walker, Jules White, Douglas C Schmidt, and Gunther Lenz. Metrics for assessing blockchain-based healthcare decentralized apps. In *e-Health Networking, Applications and Services (Healthcom), 2017 IEEE 19th International Conference on*, pages 1–4. IEEE, 2017. URL https://www.semanticscholar.org/paper/

Metrics-for-Assessing-Blockchain-based-Healthcare-Zhang-Walker/1bec1a803ca7d36a39952074aff4698d6826d9b1.

[52] Peng Zhang. *Design of Blockchain-Based Apps Using Familiar Software Patterns to Address Interoperability Challenges in Healthcare*. PhD thesis, Vanderbilt University, 2017. URL http://www.dre.vanderbilt.edu/~schmidt/PDF/PLoP-2017-blockchain.pdf.

[53] Mangosalad/mercantis-distributedhealth-2017-winner, 2018. URL https://github.com/MangoSalad/Mercantis-DistributedHealth-2017-Winner.

[54] acoravos/healthcare-blockchains github repository, 2018. URL https://github.com/acoravos/healthcare-blockchains.

[55] Bitcoin charts & graphs - blockchain, 2017. URL https://blockchain.info/charts.

[56] Historical data — tradeblock, 2017. URL https://tradeblock.com/bitcoin/historical/1h-f-tsize_per_avg-01011.

[57] Segwit charts, 2017. URL http://segwit.party/charts/.

[58] Johoe's mempool size statistics - 9/1/17 to 9/14/17, 2017. URL https://jochen-hoenicke.de/queue/#all.

[59] Jameson Lopp. Statoshi - bandwidth usage, 2017. URL https://statoshi.info/dashboard/db/bandwidth-usage.

[60] Average time to mine a block in minutes, 2017. URL https://data.bitcoinity.org/bitcoin/block_time/all?r=month&t=l.

[61] Akamai. State of the internet report. Technical report, Akamai, 2017. URL https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/.

[62] Anton Shilov. Market trends: Hdd capacities increase, average price flat - market views: Hdd shipments down 20% in q1 2016, hit multi-year low, 2016. URL https://www.anandtech.com/show/10315/market-views-hdd-shipments-down-q1-2016/3.

[63] Nhe fact sheet - centers for medicare and medicaid services, 2018. URL https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/nationalhealthexpenddata/nhe-fact-sheet.html.

[64] What is hipaa, 2018. URL http://www.dhcs.ca.gov/formsandpubs/laws/hipaa/Pages/1.00WhatisHIPAA.aspx.

Appendix A

Bitcoin Blockchain Simulation Interface

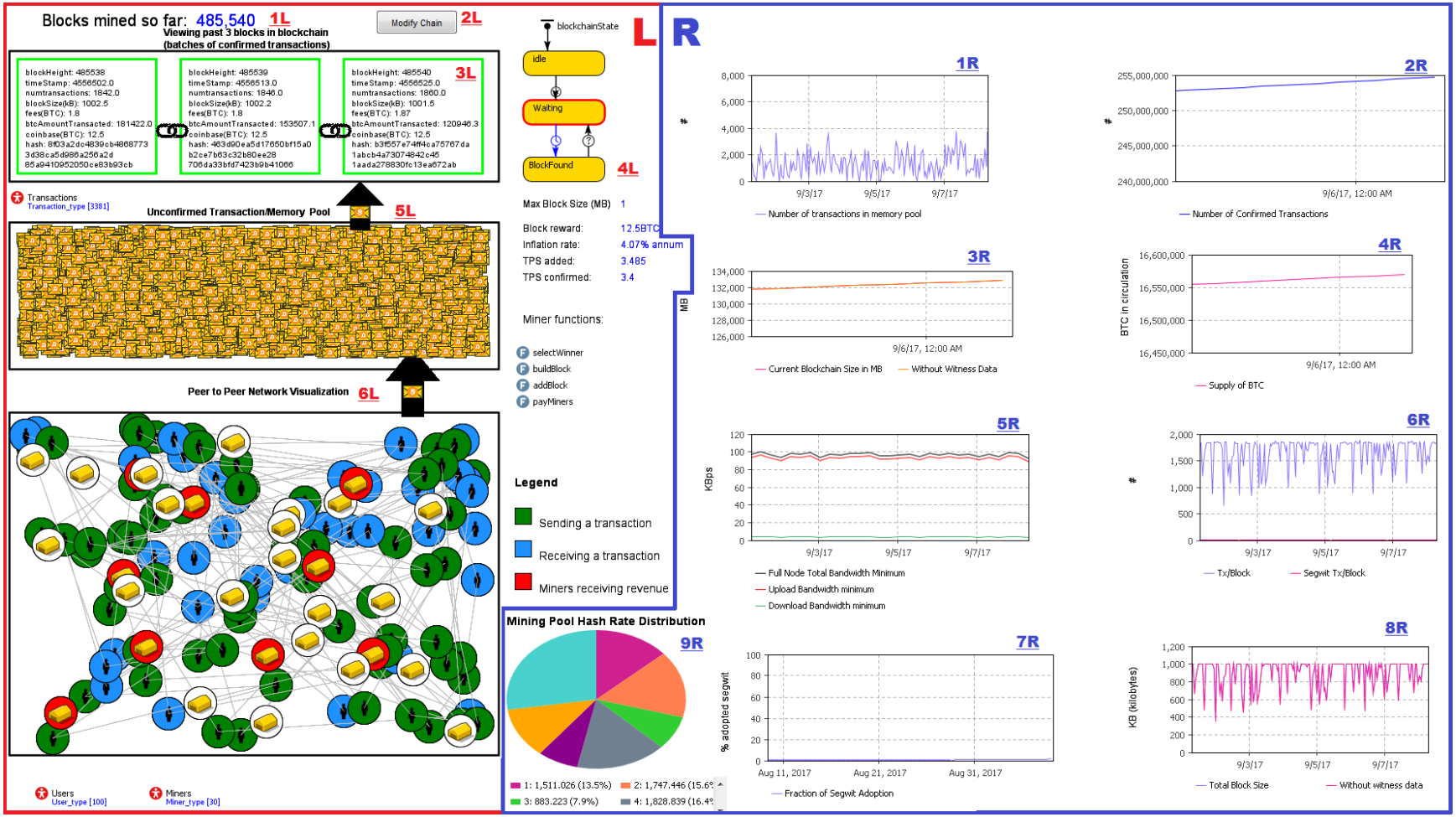


Figure A.1: Bitcoin Blockchain Simulation in Progress GUI

Appendix B

Segregated Witness Adoption Forecast

Segregated witness adoption was modeled as a sigmoid shaped adoption curve. Data from real Bitcoin network *segregated witness* adoption was used [57] for the first 68 days that *segregated witness* was active on the network to determine the fraction of transactions utilizing *segregated witness*.

A sigmoid adoption curve, specifically the logistic function, was used. The logistic function takes the form shown below in B.1.

$$f(x) = \frac{L}{1 + e^{-k*(x-x_0))}} \tag{B.1}$$

L represents the curve's maximum value. When determining the fraction of adoption, the maximum value would be 1.0.

e is the natural logarithm base.

x_0 is the value of x when the sigmoid function is equal to half of *L*.

k is a constant that defines the steepness of the curve.

A script was written in the R programming language that utilizes the data to first find x_0 by running a polynomial regression. The regression is then extrapolated to see when 50% adoption rate is reached.

The script then runs a logistic regression to determine the optimal *k* value, and outputs

the resulting logistic curve, as seen below in Figure B.1.

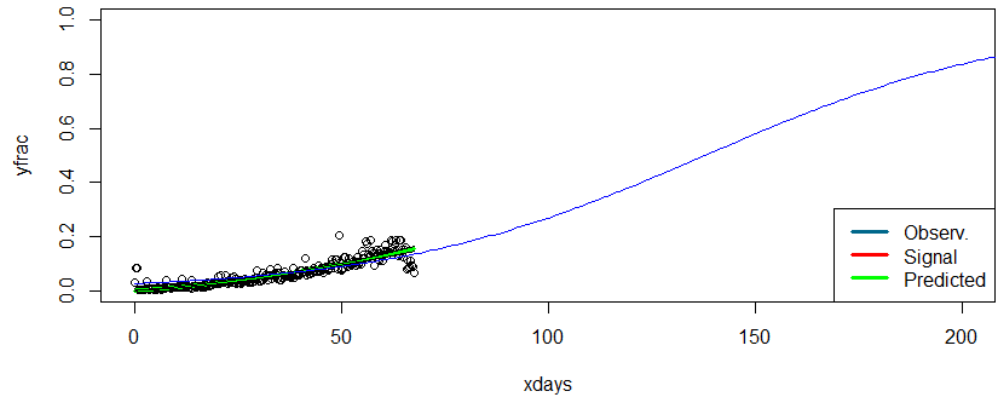


Figure B.1: Forecasted Segregated Witness Adoption Curve

The code for the script (segwitsigmoidregression.R) is shown below:

```
## Performing sigmoid regression on Segwit adoption based on
   ↪  data thru 10/23/2017 from Segwit.party
# data points were averaged over 36 block periods (~6 hours)


# First, a polynomial regression (x^2) is used to forecast
   ↪ the date of 50% adoption
# The date of 50% adoption is then used in determining the
   ↪ final logistic growth forecast


setwd("C:/Regressions")


library("dplyr")
```

```
library("lattice")
library("ggplot2")
library("rpart")
library("rpart.plot")
library("cluster")
library("rattle")
library("fpc")
library("vegan")



mysegwit<- read.csv("segwit_adoption_avg.csv", header= TRUE)



xdays = mysegwit$days
yfrac = mysegwit$adoptionfraction
x2 = seq(0,365,1)


#### POLYNOMIAL REGRESSION FIRST ####
# first step is to assume an aggressive x^2 growth rate for
   ↪ the adoption fraction
# we will fit y=x^2, and when the adoption fraction is at
   ↪ 0.5, that can be the maximum value used for the slope
```

```
# in the final logistic growth curve


# plot raw data
plot(xdays, yfrac,xlim=c(0,200),ylim=c(0,1))


k=rep(0,length(xdays))


model <- lm(yfrac ~ -1+xdays+I(xdays^2)+offset(k))
summary(model)


predicted.intervals <- predict(model,data.frame(x=xdays),
   ↪ interval='confidence',
                        level=0.99)
lines(xdays,predicted.intervals[,1],col='green',lwd=3)
lines(xdays,predicted.intervals[,2],col='black',lwd=1)
lines(xdays,predicted.intervals[,3],col='black',lwd=1)
legend("bottomright",c("Observ.","Signal","Predicted"),
      col=c("deepskyblue4","red","green"), lwd=3)


# great, so an r-squared value of 0.9318
# we'll solve for xdays when 50% adoption occurs, so 0.5
   ↪ =0.001003*xdays + 0.00001894*xdays^2
```

```
# our result is approximately 138 days
# so we will assume our maximum slope occurs at 138 days.
# find the derivative of the function when x is 138
# our solution is 0.00623044 %/day is the maximum growth
   ↪ rate and will occur at 138 days



#### LOGISTIC GROWTH FORECAST SECOND ####


# a standard sigmoid function holds the form f(x) = L/(1+ e
   ↪ ^(-k(x-x0)))
# in our case, L is known to be 1.0, x0 was determined as
   ↪ 138 days, which leaves us with just k to fit



# fitmodel2 <- nls(yfrac~SSlogis(xdays,1,138,1), data=xdays)
fitmodel2 <- nls(yfrac ~ 1/(1+exp(-k*(xdays-138))), start=
   ↪ list(k=0.00623), trace=TRUE)
summary(fitmodel2)


# found suitable model with k = 0.0264025
#predicted <- 1/(1+exp(-k*(xdays-138)))
```

```
sigmoid = function(x) {

  1/(1+exp(-0.0264025*(x-138)))

}

lines(x2, sigmoid(x2),col="blue")
```

Appendix C

Bitcoin Transaction Size Distribution

A custom distribution was utilized in the AnyLogic model to determine the size of transactions. The custom distribution was constructed using data collected from the real Bitcoin network [56] for transactions occurring between August 28th, 2017 and September 4th, 2017. The data is shown below in Table C.1.

Table C.1: Transaction size occurence on the Bitcoin network from 8/28/2017 to 9/4/2017 [56]

Size in bytes	**Occurrences**
0-200	35
200-400	18
400-600	696
600-800	176
800-1000	46
1000-1200	10
1200-1400	1
1400-1600	1
1600-1800	1

Appendix D

Bitcoin Transaction Fee Distribution

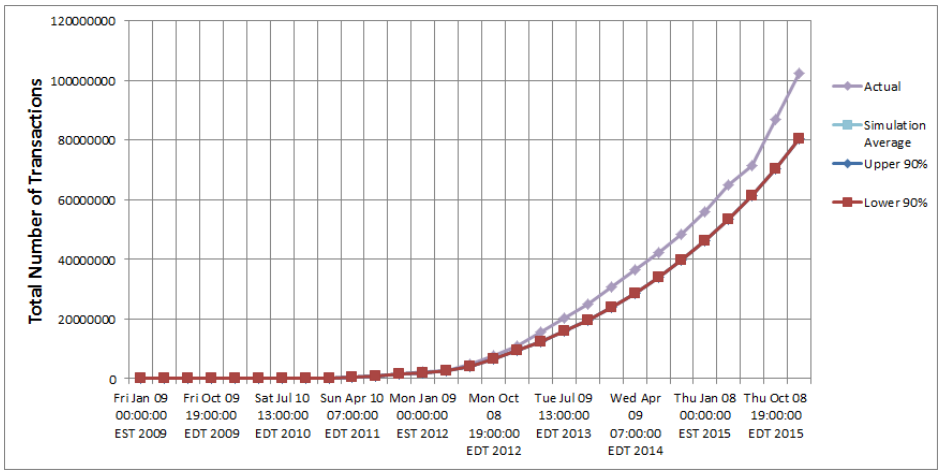
A custom distribution was utilized in the AnyLogic model to determine the transaction fee rate to utilize. The custom distribution was constructed using data collected from the real Bitcoin network [58] for transactions occurring between September 1st, 2017 and September 13th, 2017. The data is shown below in Table D.1.

Table D.1: Transaction fee occurence on the Bitcoin network from 9/1/2017 to 9/13/2017 [58]

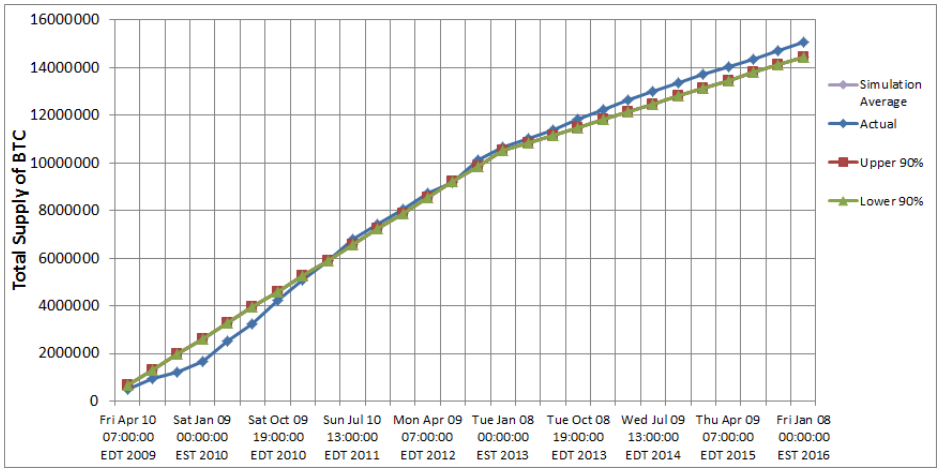
Satoshis/byte	**Occurrences**
0.0001-5	555407
5-10	528469
10-20	385257
20-30	334860
30-40	301890
40-50	274032
50-60	261178
60-70	248248
70-80	234063
80-90	224363
90-100	213018
100-120	201027
120-140	176514
140-160	154419
160-180	136921
180-200	120161
200-220	102091
220-240	85106
240-260	79336
260-280	69376
280-300	62353
300-350	55976
350-400	45316
400-450	23969
450-500	14896
500-550	6760
550-600	5017
600-650	4532
650-700	1218
700-750	1063
750-800	1009
800-850	943
850-900	849
900-950	484
950-1000	439
1000-1200	422

Appendix E

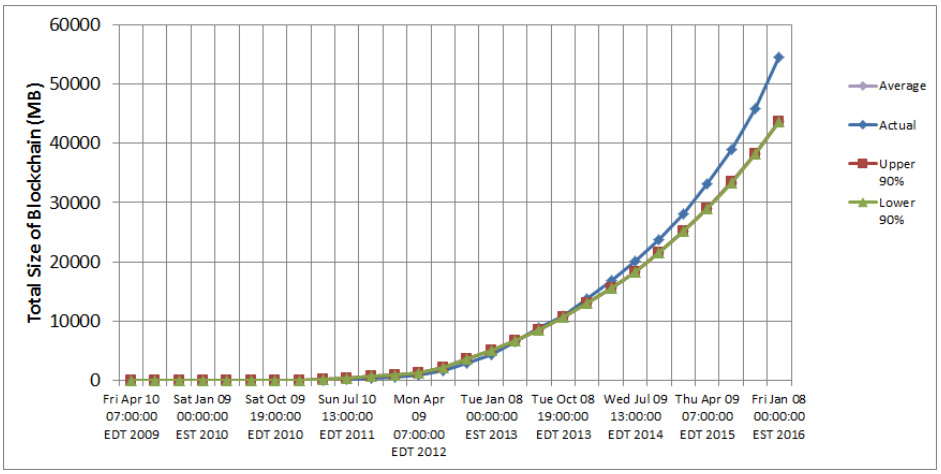
Bitcoin Simulation Results



(a) Total Number of Transactions

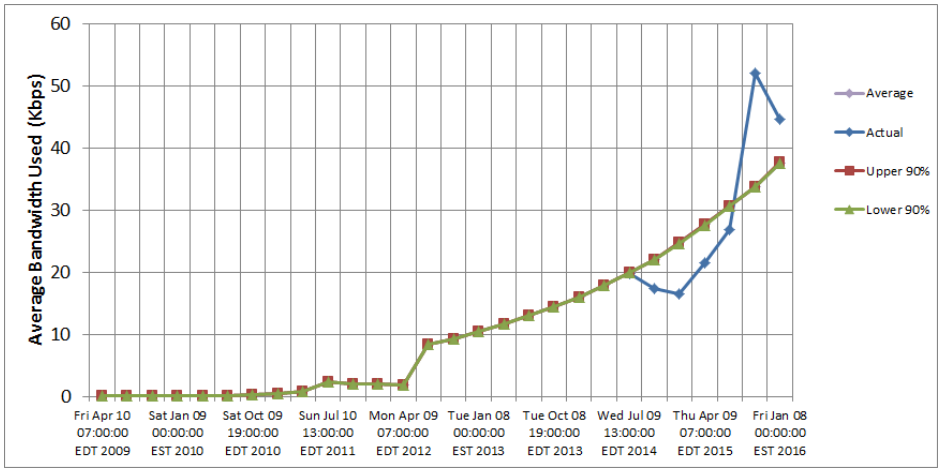


(b) Total Supply of BTC



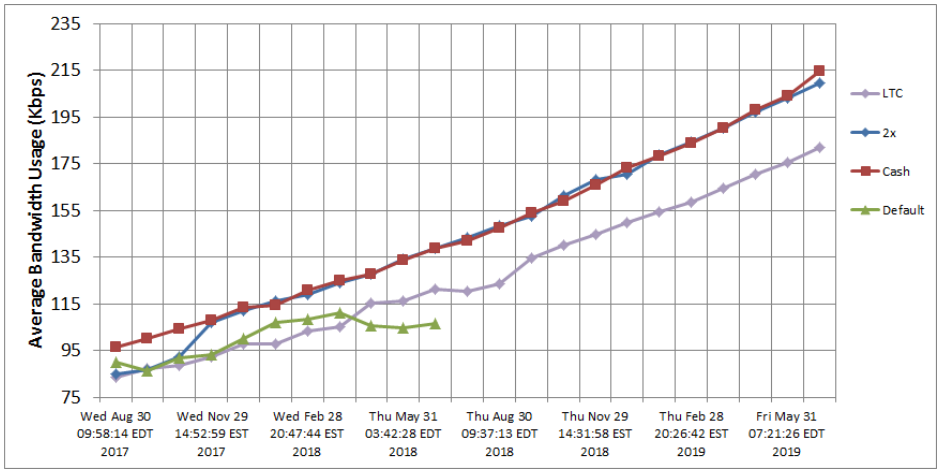
(c) Total Size of Blockchain

Figure E.1: Time series plots of validation simulation results and real data

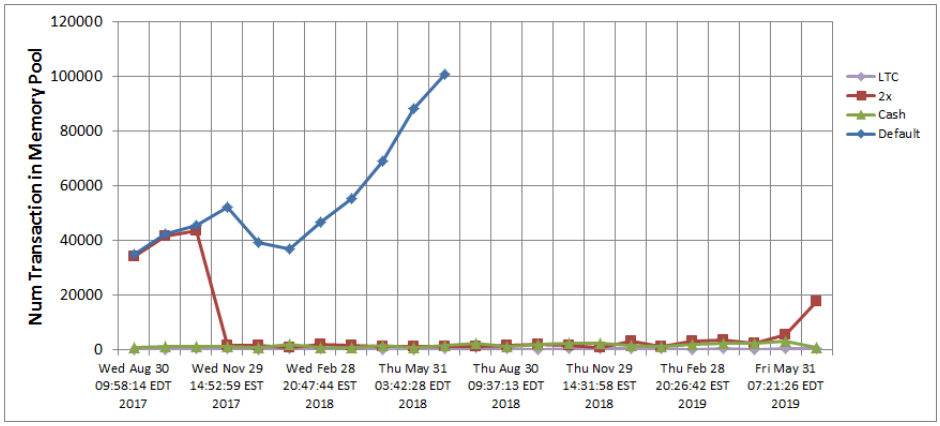


(d) Average Bandwidth Usage

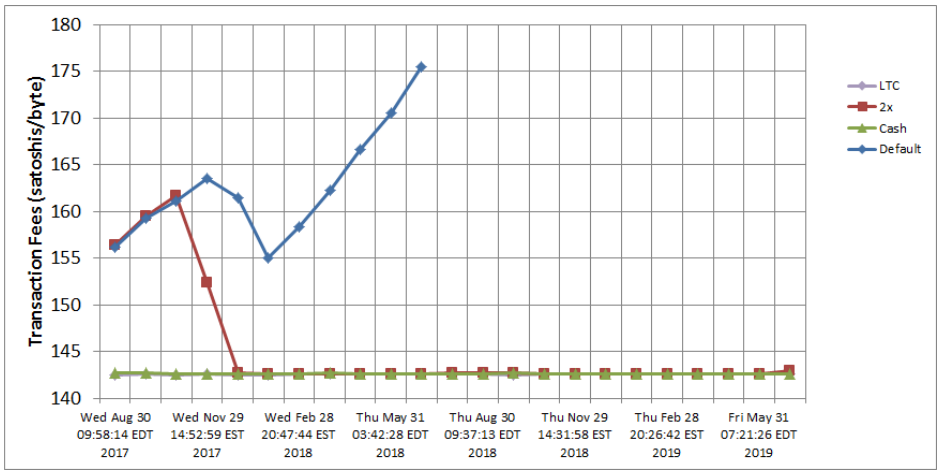
Figure E.1: Time series plots of validation simulation results and real data (cont.)



(a) Average Bandwidth Usage



(b) Transaction Fees



(c) Memory Pool Size

Figure E.2: Time series plots comparing varying configurations

Appendix F

Copyright