**Overcoming Blockchain Adoption Problems in Software Engineering**

**Abstract**

Since blockchains immergence it has become a wide topic of discussion. Yet, a lot of people don’t understand quite what it is, where it came from, and where it is heading. Let alone is it popular enough in common nomenclature for wide scale adoption. What are these factors that have limited blockchains wide scale adoption? This is a problem because blockchain is a revolutionary construct that has many new principles benefiting from the concept of decentralization. In this discussion I will include relevant and significant topics on how blockchain has created emerging trends in software engineering, while these trends have also created problems for blockchains adoption. Building from that we will delve into why blockchain is not always the best choice for situations and guide into when and where to implement one. From that we will lead into issues and problems that affect blockchain security in regards to popular consensus algorithms. Although blockchain has mainly been associated with innovating financial services, it also has roles in other fields such as e-government, supply chain management, and cyber security. This will ultimately lead to a more wide scale adoption of blockchain technology. The research methodology of this paper will be sources gained from academic journals with professional insight to resolutions. This paper will begin to discuss and wide scale adoption issue of blockchain in software engineering with a goal of how to resolve them.

**Chapter 1: Introduction.**

To begin let’s start by defining what blockchain is. This has been a very confusing concept for most because it typically is synonymous with the cryptocurrency Bitcoin.

Although, Bitcoin and other cryptocurrencies use blockchain, it’s not blockchains only real world use case. In its most basic from Blockchain is simply a digital leger. This digital ledger, is a distributed database that is constantly reconciling new information know as blocks. These blocks are appended onto the end of the data set. This effectively creates a blockchain. The data is then stored in multiple locations in contrast of one central location. This makes blockchain pretty difficult to manipulate considering multiple copies are stored on a machine simultaneously and can be verified from multiple nodes. This is what makes blockchain public, decentralized, and verifiable.

From its conception blockchain snowballed to the market essentially piggy backing off the popularity of Bitcoins financial revolutionary concepts but behind these concepts driving this new technology was the decentralized ledger itself. Of course, the history of this new technology is still argued and somewhat shrowded in mystery simply for the fact we really don’t know who or where this technology came from. When Bitcoin was implemented it also implemented the world’s first public blockchain database. The white paper for this technology was authored by a pseudonymous entity names Satoshi Nakamoto. This could also lead to the argument of the validity of blockchains security. Why, so secretive? Would want to hide their identity from getting credit to a revolutionary new concept? This alone could make someone wary from a security standpoint and is a valid standpoint on what hindered blockchaind full scale adoption. The fact that its author is unknown does not make the public trust in it.

Bitcoins conception was the start of blockchains breakthrough phase in its technology innovation cycle. (38.1txtbooks)Next would come the replicator phase, the birth of alternate coins. Originally bitcoin was a store of value on a decentralized digital ledger only to be used to verify transactions between two parties in a peer to peer network. From this the next generation of blockchain advancement came, the advancement of decentralized applications and smart contracts from a Proof of Staking consensus. This technology allowed blockchain to grow from a just a store of value to building full scale decentralized applications. At a high level smart contracts are back end server code ran on the decentralized blockchain network. A decentralized application will have a front end code that makes calls to back end code blockchain powered code. This has lead the deployment of apps on the blockchain in a publicly controlled decentralized manner. This concept has become popularized by crypto currencies like Etherium and its smart contract programing language “solidity”. Smart contract are not without fault and come with their own set of security vulnerabilities to keep in mind. We will later discuss how these Proof of Stake vulnerabilities have slowed wide scale adoption.

As the rise of Blockchain and distributed ledger technology continues to grow and mature, we will see it settle into the new global economy. No one can predict the future but we do have some educated insights into what blockchain technology can evolve into. Most notably e-government as it will allow government entities to easily track information on goods, items, services, people, voting, or even militaristic reasons.

As more time is spent critiquing and perfecting blockchain architecture more vulnerabilities will be discovered while more advancements will be made in its field.

This will ultimately lead to a more wide scale adoption.

**Chapter 2: Blockchain: when and where to use it .**

Now that we have a better understanding on what blockchain is, where it came from, and the directions it can head, we will dive into when and why we should implement a blockchain solution along with the problems of implementing one at the improper time.

There are certain questions one must ask yourself when implementing a blockchain solution. Such as, if the data I am using going to be shared across multiple parties? Decentralized ledgers are records that are stored on multiple nodes with different parties agreeing to changes. This creates a situation where any one can read or make changes to the database. In a centralized operation, if you were to want to keep a database with all your top secret information off of a network on a single system then a blockchain solution would not be ideal and traditional database would be recommended. That way only you would have access to the confidential data inside. The beauty of sharing information between parties is it eliminates the distrust between them because data is transparently stored on the decentralized ledger. A far use case would be if all big business was stored on a decentralized ledger then fraud would be exponentially more difficult because all transactions in and out would be monitored by all parties on the blockchain. If one company where to try and manipulate the data other companies could review their digital ledgers to the point where communication error occurred and effectively point the finger back at fraudulent company. This has been a hurdle for blockchains adoption because companies are not so quick to change from their legacies systems. These concepts are factors to think about when considering a blockchain solution and whether information should be central governed or not.

Next question is whether data should be dynamic and needs and auditable history. Blockchains are immutable, meaning that once information is added to the ledger it cannot be changed. This immutable data is left as an audit trail for other entities to verify.so if you don’t want your transactions to have a paper trail or want the contents of its history to be changed then a blockchain is not a solid option.

Another issue on when choosing to deploy a blockchain solution is speed. If a high performance that is dependent on millisecond transactions then it is best to lean towards a centralized system. Blockchains are typically still pretty slow in comparison to traditional model-client architecture. If you are customer waiting to verify a debit card transaction you probably are not willing to wait 15 minutes for the transaction to go through. This obviously creates the problem of speed and is one of the heavy hitters when identifying issues that have stagnated blockchains adoption.

**Chapter 3: Problems in Blockchain Security.**

When implementing new technology you must keep in mind the concept of zero day exploits, because the technology is so knew there could be a multitude of issues still to be discovered. Still for the most part if implemented correctly and adhere to secure practices and will find its place in the world. In the security realm nothing is impenetrable, even multilayered security can have its flaws and this certainly holds true in the case of blockchain. Being a new construct always comes with some kickback. Due to its rapid development many crucial mistakes were taken advantage of in the crypto currency market. Although blockchain itself was secure, the way businesses utilized it was questionable. One of the most infamous examples of a cryptocurrency hack was the incident that happened at Mt.Gox. The Mt.Gox hack at a high level was due to poor software development methodologies involving the development of blockchain applications. Another issue was that certain standards were not yet created in the blockchain community to adhere to security. It is still contested on what truly happened in the Mt.Gox hack but the underlying basis is that wallet private keys were not yet encrypted at the time, so someone was able to access wallets private keys in clear text. This in turn led to the standard practice in blockchain to encrypt wallet private keys when at rest and is a prime example of how the blockchain space and software in general evolves to meet the needs of security. The Mt.Gox hack was one of the most infamous of all of cryptocurrency history and has easily left a stain of distrust on the public and corporate sectors, thus hindering blockchains development to the masses.

Another topic of discussion is the concept of environment costs, mining takes extreme power consumption through the use of electricity and the raw materials used to create mining hardware that supports the network. In its current state blockchain solves complex algorithms with large amounts of computing power to provide security. This could cause a problem if you are intending to deploy a large network. Each node verifying hashes is using equipment and energy which can quickly add up. Bitcoins mining nodes have been known to use more electricity than some small countries. So if you are trying to be environmentally conscious with your network, you would not deploy a large scale blockchain application. Because early blockchains were environmentally taxing they were not widely adopted.

One of the biggest issues facing blockchain adoption today is its complexity for end users to understand. At its core users must understand public key infrastructure (PKI).The concept of wallets and having a public address seem foreign to the everyday person. It will take time for society to understand that your routing and account number are your public keys in a traditional banking environment, while your password to your account is essentially your private key in that situation. Of course this the same in crypto with your public address as your public key, and then your private key often being a mnemonic phrase or password. Certain growth has been made in this field such as cloud wallets to store your coins with ease, disconnected physical wallets for security, and updated software wallets with nice graphical user interfaces. The initial command line interface architectures were hard to learn and lead to a slow adoption growth. On top of the deep underlying technical architecture behind blockchain the average person fails to realize its real world use potential. The most popular concept the hit the ground running during blockchains conception is that it would be a disruptor in the traditional banking system.Most people who have even heard the terms bitcoin or blockchain is its theoretic real world use of being able to create a worldwide decentralized ledger for financial transactions. Going beyond being a bank killer most do not realize its true technological core or other real world use cases. One could argue that blockchains lack of public thorough understanding and common nomenclature is a problem it is still facing today.

The next issue with blockchain technology adoption is that since it is a new idea, its interoperability and standardization comes into question. With so many new players coming into the blockchain market, there needs to be a standardization of technology and how they interact. This has been in issue moving forward in blockchain design. Creating blockchains that can communicate freely with other blockchains becomes a cumbersome design along with getting a wide spread community of players to agree on a single standard. Creating standardization could help with application development, validate proof of concepts, as well as helping with integration. The lack of interoperability and standardization is a problem in blockchain adoption.

One of the more recognizable issues in blockchain adoption is that it is relatively slow compared to some legacy transactions systems. As a network grows with more miners validating transactions it then takes more confirmation to make a change on the ledger. This effectively creates a more secure network at the expense of speed.

So the larger a network is the more secure it will become, while simultaneously becoming slower. This creates a huge scalability problem when creating a large blockchain applications for instant transactions. Of Couse advancements have been made in this field and every day companies are working toward making blockchains more instant without giving up speed. One example of this is the development of the Proof of Stake algorithm compared to the tradition Proof of Work consensus mechanisms. The Idea that consensus algorithms still need to be improved for speed and security is just another problem facing blockchains mass adoption.

**Chapter 4: Issues with Proof of Work Consensus.**

Upon blockchains conception was the first time the world would see the proof of work algorithm deployed on a public network. Arguably the beauty of the proof of work algorithm is it takes up resources just for the sake of taking up resources. As in it is designed to take up a lot of energy and computational power. The hashing algorithms are designed in a way that it is very taxing on the system. Network miners will use time and energy in hopes of solving a hashing function. If they solve it they add a block to the chain and get a reward. This is why it is called proof of work because you are working for a reward. Proof of Work was initially intended to be decentralized but is not as decentralized as it is intended. To begin on this issue special computer chips were designed specifically for calculating these hashing functions. These specialized chips are called application specific integrated chips and have dominated the Proof of Work market. This forces everyday people who want to mine to invest in high end equipment and effectively raises the difficulty rate for mining new blocks. Because Proof of Work algorithms can be dominated by ACIC miners it can lead us to the next issue with PoW. The 51% problem. If 51% of the network agrees on lie, then it becomes the truth. This would not be an issue if it was truly decentralized but considering there are large mining pools who own good chunks of the network, they could theoretically team up to create false transactions. This of course is an adherent flaw to the PoW algorithm and is a good example of how this new concept didn’t out way the advancements in physical hardware. This is an adherent issue in adoption.

**Chapter 6 Proof of stake**

Moving forward from the PoW the Proof of Stake consensus came into being.

PoS allows validators to lock up funds in escrow. After that they start validating blocks. If they think a block should be appended they will add it to the chain. If it is appended successfully they will get a reward. If they are caught lying they will lose their funds in escrow and their validating positions. This is the concept of Staking your funds in an escrow account to be able to validate blocks on the network. This makes validation way more resource conservative. The PoS does come with its own issue too called the “Nothing at Stake” exploit. If there is a true primary chain and a faux branched chain a validator can put its escrow on both chains effectively winning either outcome. Because it can get a guaranteed pay out from either chain this is why it is called the nothing to stake issue. The Proof of Work algorithm mitigates this because miners will mine the longest chain because it is more profitable and risk free to do so. So here we see how the Proof of Stake protocol can be susceptible to malicious forks hindering adoption, although advancements in this architecture are still underway.

**Chapter 8 Delegated byzantine fault tolerance.**

Another popular consensus algorithm is Delegate byzantine fault tolerance. This builds from the staking of the PoS algorithm. DBFT makes nodes that are staking to vote for a speaker to represent its changes to the main chain. Delegates are chosen and they choose a speaker to communicate with the main chain. The delegates are the book keepers who communicate to the speaker who communicates to the main chain. If the delegates all vote that the request and response between all parties are correct they will add the transaction to their records. If a speaker acts maliciously they are voted off their speaker position. If a node is caught acting nefariously they lose their stake. This is a great advantage to speed and scalability but it is not truly decentralized in nature since you need to have a node with an escrow account to be able to vote for delegates who vote for a speaker. Making it have a high entry barrier creating a buffer for decentralization adoption by everyday users.

**Work Cited.**

**Certification of Authorship**



Submitted to: Professor Yair Levy.

Student’s Name: Eric Webb

Date of Assignment: 10/27/2019

Title of Assignment: Assignment No. 2

Certification of Authorship: I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for this course.

Student's Signature: ERIC WEBB