Blockchain-Based Software Systems: Taxonomy Development

Lamia Alashaikh
College of Computer and Information Sciences
King Saud University
Riyadh, Saudi Arabia
lamia.alashaikh@outlook.sa

***Abstract*— At an era of new emerging technologies, and spotting lights to blockchain technology specifically; many promises take place. Blockchain has attracted many researchers and practitioners from various professions and disciplines, due to the powerful characteristics behind adopting such technology. And in order to formulate a better understanding of it, many researchers are interested in classifying the Blockchain-Based Software Systems (BBSSs). Unfortunately, none of the available taxonomies in the literature considers the various Software Engineering (SWE) aspects for building a BBSS. A blockchain subject matter expert (SME) may absorb a specific aspect understanding from each of the related taxonomies, and in order to have a complete understanding of the various options available for building a BBSS, they will have to flip from a resource to another in order to collect the scattered information they need. So, this paper aims to assist the blockchain SMEs to articulate a comprehensive understanding of the various and most recent concepts and design decisions available, in order to implement or propose a BBSS solution. The taxonomy is derived from the fundamental knowledge and the major SWE aspects which a BBSS implementer or researcher has to consider, and thus is not biased to a specific SWE aspect.**

Keywords*— *blockchain, Blockchain-Based Software System (BBSS), taxonomy, Subject Matter Expert (SME), Software Engineering (SWE) aspect.

I. Introduction

Could it ever be possible that we reach to a time when the trust between entities is so solid, to the extent that the existence of intermediaries is no longer needed? This is what blockchain existed for; and even more. As the name suggests, blockchain is made-up of a chain of blocks. Those blocks are shared by all nodes in the distributed blockchain network, and cannot be edited or removed once they are validated and added to the chain, and that chain is called the distributed ledger. Which leads us to two significant benefits of blockchain: immutability, and transparency.

Blockchain has been known as the technology behind Bitcoin, which is a cryptocurrency system that emerged in 2009 and bloomed out the interest in blockchain ever since. That is just a drop of water in an ocean of blockchain applications. Blockchain can store anything of value, such as: financial transactions, medical records, or land titles. And before that value gets added to the blockchain, it must be validated by other peers. In the case of Bitcoin, the validation process is called mining, and the validators are called miners, who are rewarded for their job. Those miners represent the concept behind Proof-of-Work (PoW) protocol. They are key players when it comes to blockchain security and integrity.

The evolution of blockchain technology might remind us of the beginnings of the cloud computing, and the various debates around its non-functional characteristics (such as: security, reliability, scalability). By time, cloud computing became a must, and so would blockchain. So, in order for organizations and businesses (like: eBay, Uber, and Airbnb) to guarantee their continuous success, and fit with the evolution of such technology, they should rethink of why they exist and what value they offer [1]. As the data they hold and maintain in their centralized databases is no longer their strength point, because it is going to be shared amongst the nodes in the distributed public ledger. In fact, having a single authority in control of the data represents a single point of failure, which would not happen in the case of blockchain.

In this work, the importance to execute a taxonomy of the blockchain technology in BBSS environments is highlighted. The term "Taxonomy" is derived from the Greek taxis "arrangement" and nomos "law", and it is defined by A. J. Cain [2] as: "the methodology and principles of systematic botany and zoology and sets up arrangements of the kinds of plants and animals in hierarchies of superior and subordinate groups".

Most of the academic researchers who have done a blockchain taxonomy shifted their focus towards specific SWE aspects, and limited their classification results to the perspective of that aspect. The main SWE aspects discussed in this paper are: architecture, security, deployment, latency, and implementation. As an outcome of this work, a balanced taxonomy (i.e. a taxonomy which covers all the fundamental SWE aspects) of BBSSs is proposed, which should cover the basics and fundamentals of the different SWE aspects, and would hopefully be a great addition to support the interested SWE researchers and practitioners, and help the continuous evolution of the blockchain technology. The objective here is to help the blockchain SMEs (i.e. researchers, designers, developers, architects, and any SWE researcher or practitioner who has the experience in blockchain technology) to understand the state-of-the-art of BBSSs, identify the gaps, and implement or propose a BBSS solution.

II. Background

A. Blockchain; A Chain of Blocks

The magical masterpiece behind the existence of blockchain technology is its distributed ledger technology, which holds many advantageous characteristics within it. First, the ledger is distributed amongst all the peers in the blockchain network, so it is available to everyone with no central authority in control, and thus it avoids single-point-of-failure issue and supports system availability. This would increase people trust in using such technology, as they are actually building their trust on the software itself instead of a singular third party. Moreover, the distribution makes the blockchain network a transparent technology, as all the nodes hold the last copy of its ledger, and they can access all\part of its data. Second, the ledger is immutable, so any newly added block to the chain (i.e. the ledger) cannot be edited or removed

2021 IEEE International Conference on Blockchain (Blockchain) | 978-1-6654-1760-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/BLOCKCHAIN53845.2021.00075

978-1-6654-1760-0/21/$31.00 ©2021 IEEE
DOI 10.1109/Blockchain53845.2021.00075

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.

from it. So, by time, the size of the distributed ledger gets bigger and bigger, which serves as an archival repository of the past, and a current repository of the present. All the data since the blockchain application existed is stored, and you can trace it back\forward. Moreover, the public ledger is made-up of sequential blocks, and each block contains its data, hash value of the block, and hash value of the previous block. The data stored in a block could be of any type: image, audio, video, text, or any digital content. For example, the content of a block in Bitcoin is a list of transactions. In Ethereum, the content is an executable code which is used to execute the contract [3]. In addition, the strength behind blockchain immutability is the cryptographic hash which prevents fraud, because if a block changes, then its hash is programmed to change accordingly. Last but not least, in order for a block to be added to the blockchain, a consensus mechanism takes place, which is defined by Swanson [4] as: ''the process in which a majority (or in some cases all) of network validators come to agreement on the state of a ledger. It is a set of rules and procedures that allows maintaining coherent set of facts between multiple participating nodes''. And in order to support this mechanism, many protocols have emerged, on top of them: Proof-of-Work (PoW). To sum up, blockchain positively shines due to many fundamental characteristics, such as: distribution, availability, transparency, immutability, and traceability [5], [6].

B. Taxonomy in Software Engineering

The concept of taxonomy existed since the 1750s, when its founder Carolus Linnaeus, who is a Swedish naturalist, started to set up the rules for naming the plants and animals, by writing books which are considered as the modern reference of botanical and zoological nomenclature. Linnaeus's taxonomy was driven by the science of logic, which was invented by the Greek scientist Aristotle. Aristotle's concept was based on natural grouping and classifying the living things based on their nature [2]. The term "*Taxonomy*" is derived from the Greek taxis "*arrangement*" and nomos "*law*", and it is defined by A. J. Cain [2] as: "the methodology and principles of systematic botany and zoology and sets up arrangements of the kinds of plants and animals in hierarchies of superior and subordinate groups".

The idea of a classification taxonomy is found to be very useful in the discipline of Software Engineering. There are various examples to consider, such as *the Guide to the Software Engineering Body of Knowledge (SWEBOK)* [7], which characterizes the SWE discipline and provides a structural representation to its body of knowledge. Another example is a structural taxonomy of the research in SWE: *Research in Software Engineering: an Analysis of the Literature*, done by Glass et al. [8]. Generally speaking, achieving a taxonomy can follow two approaches: top-down or bottom-up [9], and can have different graphical representations, either a tree-based or table-based representation. As specified by [9], the benefit of adopting a top-down approach is the ability of reusing existing definitions and categories in order to create an objective classification. On the other hand, the benefit of adopting a bottom-up approach is the emergence of new characteristics which would improve and enhance the taxonomy.

III. Related Work

In this work, the author is interested in classifying the BBSSs with a balanced set of different SWE aspects. This taxonomy research is derived from various related works, which are chosen to be from different SWE aspects and viewpoints, in order to build a classification which is balanced and not biased to one SWE aspect.

Xu et al. have published a blockchain taxonomy entitled: "A Taxonomy of Blockchain-Based Systems for Architecture Design" [10]. Their taxonomy covers different architectural design decisions: decentralization, configuration, storage and computation. The analysis is achieved by comparing the fundamental properties, cost, performance, failure points, and flexibility of the design. Although that their work has a very rich content, the comparison for achieving a classification is architecture driven only.

Wieninger et al. [11] present another taxonomy of blockchain, by developing a morphology. The aim of this research is to formulate an enhanced understanding of blockchain technology and support further research. The morphology is presented by a morphological box, which includes 11 features from 3 categories: participation, application, and technology. Each feature is assigned with characteristics of that feature, concluding a total of 27 characteristic. Comparative to this work, the morphological box is driven by the blockchain features without considering any of the SWE aspects.

Zhang and Lee [12] have classified the main consensus protocols of the blockchain technology into two broad categories: the probabilistic-finality consensus protocols and the absolute-finality consensus protocols. The different consensus protocols underneath each category are discussed, including their strengths and weaknesses and the blockchain types (i.e. public, private, or consortium blockchains) they are used for. As the paper title suggests, the taxonomy is only related to the consensus protocols of a BBSS.

Ahmadjee and Bahsoon [13] present a security technical debts focused taxonomy of blockchain-based systems. They emphasize on the importance for security software engineers to understand the security risks accompanied with the architectural design decisions of BBSS. The authors argue that the design decisions of blockchain elements and their configuration may result in a security technical debt. The taxonomy helps software architects to proactively prevent potential security risks, by evaluating the design decisions using the taxonomy. Again, this paper is focused on the security aspect only.

Tasca and Tessone [14] have developed a detailed blockchain taxonomy tree by first analyzing the current blockchain systems, then building a hierarchical taxonomy tree, including main, sub and sub-sub components. Finally, the authors have identified the layouts for the components on the lowest level, and as the blockchain technology keeps evolving and the number of layouts is progressively increasing, the authors has limited the number of layouts into two to three main layouts per each sub or sub-sub component. The resulted blockchain taxonomy tree is rich in content, yet it is not derived from the fundamental SWE aspects.

A. Problem Statement

As derived from the literature, and illustrated in Table 1, none of the existing works considered the various SWE aspects when building a BBSS. A blockchain SME may absorb a specific aspect understanding from each of the related taxonomies, and in order to have a complete understanding of

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.

the various options available for building a BBSS, they will have to flip from a resource to another in order to collect the scattered information they need.

TABLE I. RELATED WORK COMPARISON

Title/reference	**A Taxonomy of Blockchain-Based Systems for Architecture Design**	
[10]	***SWE aspect***	***Identified gap***
	Architecture	other aspects are not considered
Title/reference	**Development of a Blockchain Taxonomy**	
[11]	***SWE aspect***	***Identified gap***
	n/a	not driven by SWE aspects
Title/reference	**Analysis of the main consensus protocols of blockchain**	
[12]	***SWE aspect***	***Identified gap***
	Security	other aspects are not considered
Title/reference	**Taxonomy for Understanding the Security Technical Debts in Blockchain Based Systems**	
[13]	***SWE aspect***	***Identified gap***
	Security	other aspects are not considered
Title/reference	**Taxonomy of blockchain technologies. Principles of identification and classification**	
[14]	***SWE aspect***	***Identified gap***
	n/a	not driven by SWE aspects

IV. TAXONOMY OF BLOCKCHAIN-BASED SOFTWARE SYSTEMS

The proposed solution is composed in the development of a balanced taxonomy that covers all the fundamental SWE aspects, which has a direct affect into the understanding of the BBSS SMEs, who are supposed to be building or proposing a solution for the development and innovation of the blockchain technology.

This taxonomy research is derived from various related works, which are chosen to be from different SWE aspects and viewpoints, in order to build a classification which is balanced and not biased to one SWE aspect. Additionally, some important knowledge and aspects which are not found in the literature and considered to be fundamental as part of the taxonomy tree are added to the taxonomy.

The criteria followed for developing such a taxonomy is described in *section A.* Afterwards, the taxonomy tree is explained in *section B.* A given example of a classified BBSS using the proposed taxonomy is described in *section C.*

A. Taxonomy Criteria

The goal here is a development of the taxonomy, which is planned to be visualized as a taxonomy tree. The traceability of the tree should start by the root node, which in our case is the BBSS, followed by a sequence of nodes as needed, until the sequence ends by a leaf node. The different SWE aspects will fall underneath the root node, along with the details of the different categories of each aspect as illustrated in Fig. 1.

The implemented taxonomy criteria is a three steps process, where the first two steps are inspired by the work of Tasca and Tessone [14], where the author of this work develops a taxonomy tree with a **bottom-up approach**, by first identifying all the existing categories, based on the state-of-the-art of the existing studies, as well as by adding the missing pieces. The result was a total of 20 categories, as illustrated in Fig. 2. Afterwards, as illustrated in Fig. 3, those categories were grouped into five SWE aspects: deployment, implementation, architecture, security, and latency. The other categories which are considered as general ones are grouped into a sixth aspect: business need, to make more sense in regard to their value, since they are actually driven by the business need. Finally, as a third step, following a **top-down approach**, the categories are broken down into an N-level of sub-categories to formulate the resulted taxonomy tree in its final version, which is described in detail in the next *section (B).*

Wherever applicable, the facts of the other aspects (such as: the performance of the different consensus protocols, which fall under the security aspect) which would be distributed across the tree and cannot be collected into some sort of a coherent aspect, will be explained in the description of each category. Additionally, since we cannot classify neither on the level of the aspects nor the categories, they will be broken down into one to three levels of sub-categories, until a clear classification can be driven out from the leaf nodes.

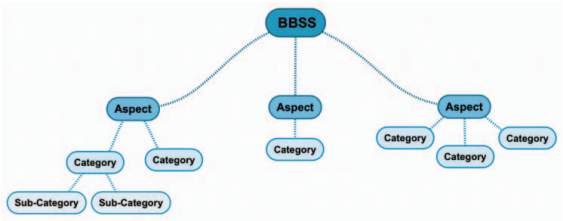

Fig. 1. Taxonomy Tree Structure

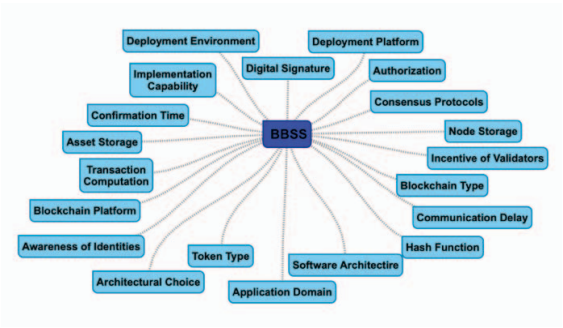

Fig. 2. Criteria Step 1

B. Taxonomy Tree

This section explains the third step of the criteria (i.e. the top-down approach), where the taxonomy tree is developed and explained in detail. It includes six aspects, and twenty categories, with a minimum of three and a maximum of five levels, as illustrated in Fig. 4.

1. Deployment

1.1. Deployment Environment

A blockchain solution can be deployed into two main environments; either on premises in the data centers of an organization, or on the cloud. There are variety of options when it comes to cloud deployment. Three of the most popular public clouds worldwide are: Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP). A BBSS offered over the cloud is called Blockchain as a Service

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.

(BaaS). BaaS comes in three different flavors: private, public, or hybrid cloud, depending on the use case.

1.2. Deployment Platform

Containerization concept has emerged recently, and it is usually compared against virtual machines (VMs) as a better platform with unique benefits, such as less operating system (OS) costs (i.e. there is no need for a guest OS for each container), and more operational automation for IT operations. When it comes to the deployment of a BBSS, there are mainly two options; whether to go with containerization or traditional deployment. Traditional deployment could be on a virtual machine or a bare metal. One of the most popular container platforms is the Red Hat OpenShift Container Platform. And the most popular VM is VMWare.

1.3. Blockchain Platform

A BBSS solution can be deployed on top of an existing blockchain platform, such as Bitcoin or Ethereum. One of the most popular blockchain platforms is the open source Hyperledger Fabric, which many of the blockchain solutions are based on, such as IBM Blockchain Platform (IBP). An SME can either choose one of those platforms or start building a new blockchain platform.

2. Implementation

2.1. Implementation Capability

Whenever an organization plans to start a blockchain project, the given capabilities and skill set within this organization shall be considered. Then, a decision can be made on whether to start the solution implementation from the scratch, or to go with the vendor choice.

- new

Whenever the organization decides to start a new implementation, they can decide whether to build an open-source code or not, depending on the organization strategy and the sensitivity of the solution [11], [14]. Additionally, there are different blockchain development platforms to choose from, some are free of charge, such as: Visual Studio Code and some are commercial.

- vendor

The vendor choice comes in different software license types, depending on the vendors offerings. Generally, those software licenses can be divided into proprietary or subscription based. Each choice has its own positives and drawbacks. For example, the subscription based helps an organization capitalize on their Return on Investment (ROI). Moreover, some vendors offer an open-source solution, while others offer closed-source ones [11], [14]. The main benefit of an open-source solution is to avoid vendor lock-in, which is a huge obstacle facing organizations today from moving to cloud.

2.2. Transaction Computation

Computation in a blockchain context can be performed either on-chain or off-chain. An example of on-chain computation is smart contracts. Another example for off-chain computation is to perform it on a private or third party cloud. The benefits provided with on-chain computation are better fundamental blockchain properties: immutability, non-repudiation, integrity, transparency, and equal rights. On the other hand, off-chain computation is more cost efficient and has better performance [10].

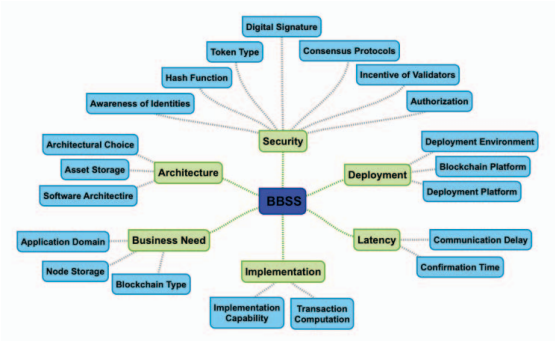

Fig. 3. Criteria Step 2

3. Architecture

3.1. Architectural Choice

There are mainly three architectural choices to build a BBSS solution: fully centralized, partially centralized and partially decentralized, and fully decentralized. Those choices are mainly related to the type of the blockchain. For example, permissionless blockchains are always fully decentralized. On the other hand, permissioned blockchains with a single provider in charge (e.g., governments and courts) are fully centralized. Partially centralized and partially decentralized blockchain example is a permissioned blockchain with permissions to create an asset or write a transaction, but no permission to read the ledger. Fully centralized blockchains are more favorable when it comes to performance and cost efficiency, but they suffer from a single-point-of-failure. Whereas fully decentralized blockchains avoid single-point-of-failure but are less favorable in terms of cost and performance. Additionally, fully decentralized blockchains are better in terms of maintaining the fundamental blockchain properties: immutability, non-repudiation, integrity, transparency, and equal rights [10].

3.2. Asset Storage

A transaction is a transfer of an asset from an entity to another. This asset is not necessarily stored within the blockchain. If an asset exists outside of the blockchain, the procedural link can represent a security risk. There are mainly two options for an asset storage: on-chain or off-chain. An example for on-chain storage is Bitcoin assets: the tokens. Another example for off-chain storage is the trading of diamonds [11]. An off-chain storage is more favorable in terms of performance, whereas on-chain storage is more favorable in terms of fundamental blockchain properties: immutability, non-repudiation, integrity, transparency, and equal rights [10].

3.3. Software Architecture

The software architecture represents a high-level structure of a BBSS, which is composed of the elements and the relationships between them. There are two software architecture designs: monolithic and polylithic. In monolithic design, all the application elements are composed as a one single-tier software application. The drawback of this choice is the difficulty of extending the application with additional elements in the future. Two examples here are Bitcoin and Ethereum. The polylithic design decouples the software

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.

elements from each other, and those elements communicate with each other through simple Application Programming Interfaces (APIs), which increases the interoperability between them. So, two elements written in two different programming languages can easily and seamlessly communicate together. An example of this design is the Hyperledger Fabric [14].

4. Security

4.1. Consensus Protocols

Since the existence of Bitcoin, the researches have been very active in inventing new consensus protocols. Nowadays, many consensus protocols exist. This project covers the main consensus protocols of blockchain: Proof-of-Work (PoW), Proof-of-Stake (PoS), Delegated Proof-of-Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), and Ripple. PoW, PoS, and DPoS are probabilistic-finality protocols and they are more suitable for permissionless blockchains. Whereas PBFT and Ripple are absolute-finality protocols and they are more suitable for permissioned blockchains. PoW, PoS, and DPoS have very high fault tolerance, which equals 50%. So, an attacker has to control 50% of the blockchain network in order to attempt an attack. On the other hand, PBFT has a lower percentage which is 33% fault tolerance, and Ripple holds the lowest percentage which equals 20%. The drawback of PoW is the huge consumption of power, compared to the other four consensus protocols. Compared to PoW and PoS, DPoS has a lower cost and higher efficiency. Although PBFT has a high performance but it has limited scalability, since it is suitable for a small number of nodes. Additionally, PBFT does not guarantee anonymity since the identity of the participating nodes are known. Ripple has a very high performance which makes it suitable for the payment scenarios, but it does not support a fully decentralized architecture [6], [12].

4.2. Awareness of Identities

Identities in a BBSS can be known, anonymous, or pseudonymous, depending on the purpose of that system and the existence of a need to know the identities of the participating nodes. For example, Ripple has known identities in order to be able to verify users information to perform some financial services. This increases the transparency in terms of network participants. Pseudonymous means that identities can be derived from initially unknown identities, by tracing the history of transparent transactions and driving conclusions about the identities in the network [11], [14].

4.3. Incentive of Validators

To make sure a validation process always takes place; validators need to have incentives to do so. In case of Bitcoin, miners who participate in the consensus mechanism are rewarded for their work with Bitcoins, so it has a financial incentive. Not all BBSS have a financial incentive for validating the blocks. Hence, incentives can be generally divided into financial or non-financial incentives [11].

4.4. Authorization

The authorization to participate in a BBSS is mainly divided into who can view, propose, and validate transactions. Each authorization of those levels is different from the other; for example, some BBSS have public read authorization but not necessarily public proposal or public validation authorization [11].

- to view

The authorization to view is mainly divided into public and restricted. For example, Bitcoin has a public read authorization, so any user in the BBSS network can read transactions with full transparency. On the other hand, some BBSSs have a restricted authority to view the records on a need-to-know basis. One example here is the authorization to view patients records in a healthcare sector, which should be restricted.

- to propose

This is the authorization to propose a transaction which is different from the authorization to validate. A participant can propose a transaction, which then can be validated, whether that participant is part of the validation process or not. An example here is the usage of blockchain in supply chain: the end customer has a transparent visibility over the history of transactions (i.e., authorization to view), but they don't have the authorization to propose transactions. So, the authorization to propose can be either public or restricted.

- to validate

The authorization to validate circulates around the consensus mechanism. In case of Bitcoin, it is a public authorization to validate, so any node can participate in the PoW consensus with no permissions required. Whereas Corda blockchain has a restricted group of validators, who are called notary nodes. A third scenario is when the authorization to validate is granted to a single authority, who is responsible to validate all transactions, such as a bank or a court. As a result, the validation authorization can be either public, restricted group, or central authority.

4.5. Token Type

A token, as defined by S. Wieninger et al. [11], is: "A digital unit whose ownership is documented on the Blockchain. It can represent different values or can be the value itself. Not every Blockchain has a token. Not all tokens have the same purpose.". There are three different kinds of tokens covered in this project:

- cryptocurrency token: a token here acts as an asset in a payment system
- utility token: a token which serves as an admission ticket to access an application
- asset token: a token used for profit sharing or share rights for an asset

Other tokens rather than the above can be categorized as "other token". And in case there is no token used, it is categorized as "no token".

4.6. Hash Function

The primary hash functions used so far in a BBSS are: Secure Hash Algorithm 2 (SHA-2), SHA-3, Message Digest 5 (MD5), and BLAKE2. Any other hash functions rather than the above mentioned ones are categorized as "other".

BLAKE2 is fast, secure, and simple. It is faster than SHA-2, SHA-3, and MD5, and as secure as SHA-3 [15]. When SHA-1 was first attacked, SHA-3 was created in order to overcome the weakness of SHA-1 and boost the strength of SHA-2. Compared to its predecessor, SHA-3 is found to be stronger than SHA-2 against the attacks [16].

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.

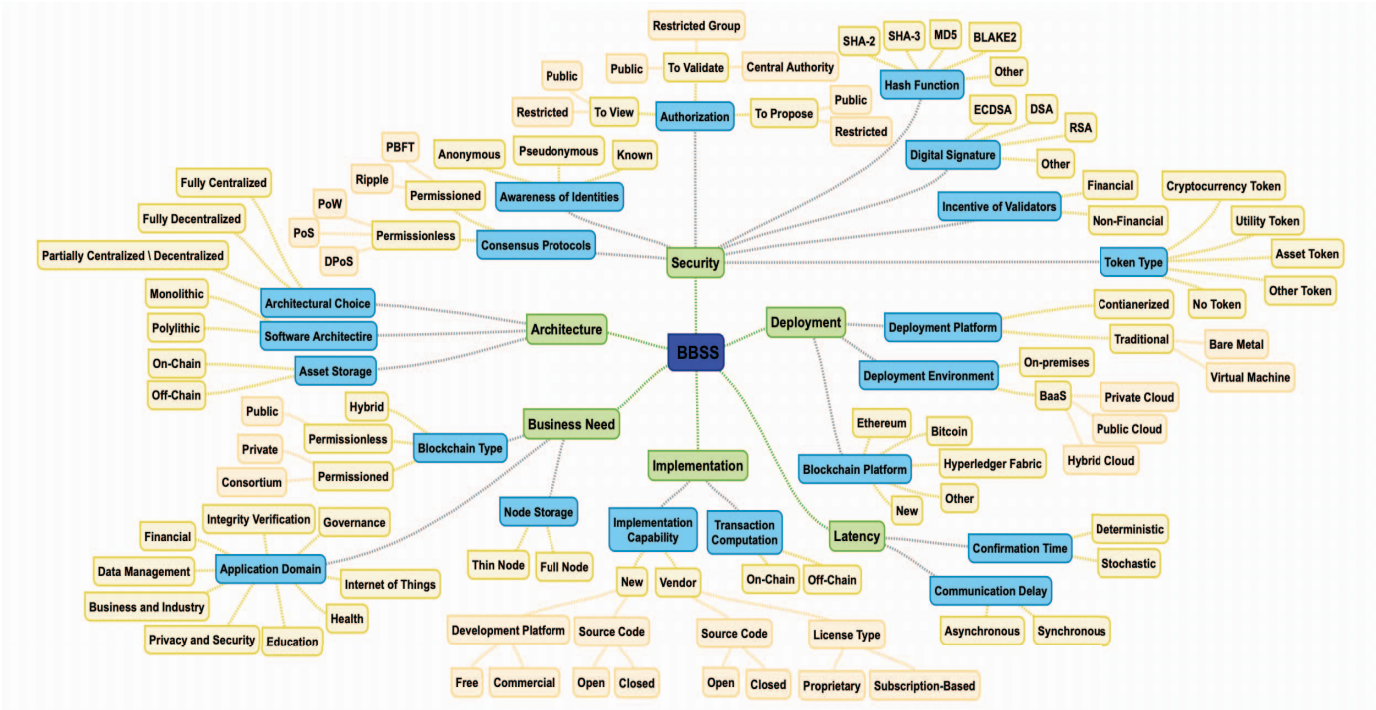

Fig. 4. Taxonomy Tree

4.7. Digital Signature

The most used digital signature in the BBSSs is Elliptic Curve Digital Signature Algorithm (ECDSA), due to many advantages of it against Digital Signature Algorithm (DSA) and RSA (named after its inventors Rivest, Shamir and Adleman):

- stronger security. 160-bit ECDSA is of the same security strength as 1024-bit RSA and DSA
- lower computation and faster processing speed
- smaller storage space
- lower bandwidth requirements

Moreover, as mentioned by Fang et. al. [17]: “with the same key length, DSA (with extended support) decrypts the ciphertext faster and the encryption is slower; RSA is just the opposite, and generally, the decryption times are more than the encryption times.” Any other digital signatures are grouped under the “other” category [13], [17].

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.

5. Latency

5.1. Communication Delay

BBSSs which set an upper bound for communication delay, so that each message arrives within a certain predefined time interval are called synchronous. All delays are considered, including exogenous network latency. Any message which takes longer than the upper bound is discarded. Two examples of BBSSs using synchronous communication are Bitcoin and Ripple. In Ripple, "LastLedgerSequence" parameter confirms that a transaction is either validated or rejected within a matter of seconds. On the other hand, any BBSS which does not set an upper bound for communication delay so that each message can take an indefinite time to arrive is called asynchronous. The advantage here is that nodes don't have to be active all the time, but the disadvantage is that we cannot predict how long it will take to receive a response. An example of asynchronous communication is Synereo [14].

5.2. Confirmation Time

The time it takes a transaction to be confirmed depends purely on the time needed to validate it and add it to the blockchain. There are two kinds of confirmation time: deterministic, based on some given time intervals, and stochastic, which is a random confirmation time [14].

6. Business Need

6.1. Blockchain Type

There are three primary types of blockchain: permissionless, permissioned, and a hybrid of both:

- permissionless blockchains: participants can join the network with no permissions needed. A disadvantage of a permissionless blockchain is the low efficiency, as the consensus mechanism limits the number of TPS.
- permissioned blockchains: participants need to be invited in order to be able to join the network. Permissioned blockchains are categorized into two types: private and consortium (or community) blockchains. The difference between the two is that the participation in a private blockchain is controlled by a single organization, whereas in consortium it is controlled by a group of organizations.
- hybrid blockchains: a hybrid blockchain combines the advantages of both the permissionless and permissioned blockchains.

In Bitcoin and Ethereum, anyone can join the network, read the ledger, make transactions, and become a miner, and thus are permissionless blockchains. Whereas participants of a Hyperledger Fabric need to be invited which is why it is a permissioned blockchain [6].

6.2. Application Domain

Any BBSS solution has to have a specific purpose of using it, and thus has to have a specific application domain. The different blockchain application domains are visualized in Fig. 5 of [18].

6.3. Node Storage

Different nodes have access to different layers of information. There are mainly two types [14]:

- full nodes: all nodes are of the same kind, and all of them contain the same information, which increases information redundancy and system resiliency.
- thin nodes: some nodes contain only a subset of all information contained in the network, which increases system scalability in terms of the number of nodes, but may drop the system resiliency, as only a fraction of nodes have the full information.

C. Classification Example

Let us consider the example of Bitcoin, classified using our taxonomy tree in Fig. 4. The classification applies on the leaf nodes, and below is the explanation:

- consensus protocol: the consensus protocol used for Bitcoin is PoW
- awareness of identities: since identities can be known from initially unknown identities, Bitcoin is classified as pseudonymous
- authorization:
  - to view: public, anyone can join the network and view transactions
  - to validate: public, any participant can become a miner and validate transactions
  - to propose: public, any participant can propose new transactions
- hash function: Bitcoin is based on double SHA-256, which is a subset of SHA-2
- digital signature: Bitcoin is based on ECDSA
- incentive of validators: financial, as miners are rewarded with Bitcoins
- token type: Bitcoin tokens are classified as cryptocurrency tokens
- deployment platform: Bitcoin is deployed on a VM [19]
- deployment environment: as the public nature of Bitcoin, it is therefore hosted on a public cloud
- blockchain platform: Bitcoin
- confirmation time: Bitcoin has a stochastic confirmation time
- communication delay: Bitcoin has a synchronous communication delay
- transaction computation: Bitcoin's computation of transactions happens on-chain
- implementation capability:
  - development platform: no information was found
  - source code: Bitcoin was released as an open source software
- node storage: all Bitcoin nodes are the same, and hence it is classified as full node
- application domain: Bitcoin is a cryptocurrency application, so it is classified under financial application domain
- blockchain type: Bitcoin is a permissionless blockchain
- asset storage: all assets are stored on-chain in the public ledger
- software architecture: Bitcoin has a monolithic software architecture
- architectural choice: Bitcoin is a fully decentralized blockchain

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.

V. Conclusion, Limitations and Future Work

As technology grows; new innovations emerge, and blockchain is a trending topic in this context, so the objective of this work is to help the blockchain SMEs to understand the state-of-the-art of BBSSs, identify the gaps, and implement or propose a BBSS solution. The taxonomy is derived from the fundamental knowledge and the major SWE aspects which a BBSS implementer or researcher has to consider, and thus is not biased to a specific SWE aspect.

The limitation of this work is that the taxonomy has no defined boundaries in terms of the number of tree levels, with the first level including all the aspects, and the second level including all the categories, then the sub-categories are extended from the third to the fifth level. The reason here is that in order to classify a given BBSS, some sub-categories on the third level have to be broken down into the fourth or even the fifth; so that a classification can be derived from the leaf node. In the future, a mechanism of identifying the leaf nodes and tree length boundaries could be established.

References

[1] M. Ferguson, "Preparing for a blockchain future," MIT Sloan Manag. Rev., vol. 60, no. 1, 2018.

[2] A. J. Cain, "Taxonomy," Encycl. Br., Jun. 2020, Accessed: 11-Jul-2020. [Online]. Available: https://global.britannica.com/science/taxonomy.

[3] M. C. Benton and N. M. Radziwill, "Quality and Innovation with Blockchain Technology," vol. 20, pp. 35–44, 2017.

[4] T. Swanson, "Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems," vol. 151, pp. 10–17, 2015, doi: 10.1145/3132847.3132886.

[5] D. Tapscott and A. Tapscott, "How Blockchain Will Change Organizations," MIT Sloan Manag. Rev., vol. 58, no. 2, 2017.

[6] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung, "Decentralized Applications: The Blockchain-Empowered Software System," IEEE Access, vol. 6, pp. 53019–53033, 2018, doi: 10.1109/ACCESS.2018.2870644.

[7] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, and L. Tripp, "Guide to the software engineering body of knowledge," IEEE Softw., vol. 16, no. 6, pp. 35–44, 1999, doi: 10.1109/52.805471.

[8] R. L. Glass, I. Vessey, and V. Ramesh, "Research in software engineering : an analysis of the literature," Inf. Softw. Technol., vol. 44, pp. 491–506, 2002.

[9] M. Unterkalmsteiner, R. Feldt, and T. Gorschek, "A taxonomy for requirements engineering and software test alignment," ACM Trans. Softw. Eng. Methodol., vol. 23, no. 2, 2014, doi: 10.1145/2523088.

[10] X. Xu et al., "A Taxonomy of Blockchain-Based Systems for Architecture Design," Proc. - 2017 IEEE Int. Conf. Softw. Archit. ICSA 2017, pp. 243–252, 2017, doi: 10.1109/ICSA.2017.33.

[11] S. Wieninger, G. Schuh, and V. Fischer, "Development of a Blockchain Taxonomy," Proc. - 2019 IEEE Int. Conf. Eng. Technol. Innov. ICE/ITMC 2019, 2019, doi: 10.1109/ICE.2019.8792659.

[12] S. Zhang and J. H. Lee, "Analysis of the main consensus protocols of blockchain," ICT Express, no. xxxx, pp. 1–5, 2019, doi: 10.1016/j.icte.2019.08.001.

[13] S. Ahmadjee and R. Bahsoon, "A Taxonomy for Understanding the Security Technical Debts in Blockchain Based Systems," 2019, [Online]. Available: http://arxiv.org/abs/1903.03323.

[14] P. Tasca and C. J. Tessone, "Taxonomy of blockchain technologies. Principles of identification and classification," arXiv, 2018, doi: 10.5195/ledger.2019.140.

[15] "BLAKE2." https://www.blake2.net/ (accessed Apr. 17, 2021).

[16] F. Wang et al., "An Experimental Investigation into the Hash Functions Used in Blockchains," IEEE Trans. Eng. Manag., vol. 67, no. 4, pp. 1404–1424, 2020, doi: 10.1109/TEM.2019.2932202.

[17] W. Fang, W. Chen, W. Zhang, J. Pei, W. Gao, and G. Wang, "Digital signature scheme for information non-repudiation in blockchain: a state of the art review," Eurasip J. Wirel. Commun. Netw., vol. 2020, no. 1, 2020, doi: 10.1186/s13638-020-01665-w.

[18] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," Telemat. Informatics, vol. 36, pp. 55–81, 2019, doi: 10.1016/j.tele.2018.11.006.

[19] S. Todd, ROCK AROUND THE BLOCKCHAIN WITH DELL TECHNOLOGIES. 2017.

Authorized licensed use limited to: UNIVERSIDADE ESTADUAL DE MARINGA. Downloaded on July 14,2025 at 17:03:57 UTC from IEEE Xplore. Restrictions apply.