From Concept to Measurement: A Survey of How the Blockchain Trilemma Can Be Analyzed

Mansur Aliyu Masama[1], Niclas Kannengießer[2], and Ali Sunyaev[3]

[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
Email: {mansur.masama}@partner.kit.edu

[2]Karlsruhe Institute of Technology, Karlsruhe, Germany
Email: {niclas.kannengiesser}@kit.edu

[3]Technical University of Munich, Campus Heilbronn, Germany
Email: {ali.sunyaev}@tum.de

***Abstract*—To meet non-functional requirements, practitioners must identify Pareto-optimal configurations of the degree of decentralization, scalability, and security of blockchain systems. Maximizing all of these subconcepts is, however, impossible due to the trade-offs highlighted by the blockchain trilemma. We reviewed analysis approaches to identify constructs and their operationalization through metrics for analyzing the blockchain trilemma subconcepts and to assess the applicability of the operationalized constructs to various blockchain systems. By clarifying these constructs and metrics, this work offers a theoretical foundation for more sophisticated investigations into how the blockchain trilemma manifests in blockchain systems, helping practitioners identify Pareto-optimal configurations.**

***Index Terms*—Benchmarking, blockchain technology, trade-offs, non-functional requirements.**

I. Introduction

Common non-functional requirements for blockchain systems relate to degree of decentralization (DoD), scalability, and security. According to the blockchain trilemma [1]–[4], simultaneous maximization of the DoD, scalability, and security of blockchain systems is, however, impossible. To still meet common non-functional requirements, practitioners need to find a Pareto-optimal configuration of the blockchain trilemma subconcepts (i.e., DoD, scalability, and security) by increasing one at the expense of another. For example, the Ethereum system transitioned from a consensus mechanism based on proof-of-work (PoW) to one based on proof-of-stake (PoS). That transition helped increase scalability regarding the transaction processing rate (i.e., throughput) to better meet requirements related to high scalability. Conversely, the change decreased the DoD because only a subset of nodes (i.e., validating nodes) can participate in consensus finding [5], [6]. To find Pareto-optimal configurations, practitioners must thoroughly understand the influences of blockchain system configurations on the blockchain trilemma subconcepts.

To help practitioners find Pareto-optimal configurations through quantification of the blockchain trilemma subconcepts, various analysis approaches, including *BBSF* [7], *Blockbench* [8], and *Diablo* [9], were developed. Such analysis approaches use different constructs, operationalized through different metrics[1], to measure the blockchain trilemma subconcepts. For example, practitioners could use the construct *availability* to investigate both scalability [9] and security [10], raising the question of how to effectively differentiate between these two constructs in the context of the blockchain trilemma. Moreover, practitioners are forced to decide between different operationalized constructs to quantify the same blockchain trilemma subconcepts, for example, between *wealth distribution* [2], [11], [12] and *block proposal randomness* [13], [14] to evaluate the DoD of blockchain systems.

Selecting suitable constructs and metrics for analyzing the blockchain trilemma is challenging due to the insufficient development of its theoretical foundations. Without such theoretical foundations, the suitability of constructs and their operationalization through metrics for investigations of the blockchain trilemma can hardly be justified. To tackle the lack of such a theoretical foundation, extant research presents mappings of the blockchain trilemma to established theories in distributed systems, such as the CAP theorem [3], [15], [16]. However, such mappings often do not cover all subconcepts or intermingle constructs of different subconcepts. Although such adjacent theoretical foundations are valuable, they only reflect parts of the blockchain trilemma, which hinders thorough suitability assessments of constructs for analyzing the blockchain trilemma. Due to inconsistencies in construct usage and a lack of a well-defined link between constructs and the blockchain trilemma subconcepts, practitioners must rely on individual discretion to identify constructs suitable for Pareto-optimal configurations that meet non-functional requirements. To assist practitioners in selecting appropriate constructs and associated metrics to analyze the blockchain trilemma, we pose the following research question: *What constructs and associated metrics are suitable to quantify the blockchain trilemma subconcepts?*

[1]A *metric* is a mathematical formulation that defines the relationship between input variables and an output variable. In this work, metrics are used to operationalize constructs to quantify the blockchain trilemma subconcepts.

We conducted a systematic literature search [17] to identify publications that propose constructs and metrics for analyzing the subconcepts of the blockchain trilemma. Using abductive thematic analysis [18]–[20], we iteratively analyzed the literature to develop and refine these constructs and their associated metrics for measuring DoD, scalability, and security. The process was guided by repeated interplay between empirical findings and theoretical framing, supplemented by targeted theoretical sampling to address identified gaps and validate emerging insights. Based on this iterative abductive analysis, we developed an overview of analysis approaches that apply these constructs and metrics to investigate the blockchain trilemma.

The primary purpose of this work is to enhance understanding of the constructs and associated metrics used to analyze the blockchain trilemma and its subconcepts. In particular, this work has three main contributions. First, by explaining common constructs and their operationalization through metrics used to quantify *DoD*, *scalability*, and *security*, their applicability, interpretability, and limitations, we offer a theoretical foundation for more targeted analyses of blockchain trilemma subconcepts. Second, by explaining those metrics, including their input variables, we support data collection in benchmarks. For example, we clarify blockchain system characteristics that need to be monitored to feed input variables of relevant metrics used in benchmarks. Third, by comparing analysis approaches based on the constructs and metrics used, we guide the selection of suitable approaches for investigating the blockchain trilemma.

The remainder of this work is divided into five sections. In section II, we introduce the foundations of blockchain technology by describing how the blockchain trilemma manifests in blockchain systems. Moreover, we describe how extant research attempts to operationalize the blockchain trilemma subconcepts from empirical and conceptual perspectives. The literature search and analysis are described in section III. Next, we present principal constructs and their associated metrics that can be used to operationalize the blockchain trilemma subconcepts in section IV. Moreover, we showcase extant analysis approaches with a focus on the constructs and metrics they use to operationalize the blockchain trilemma subconcepts. In section V, we discuss the key findings and present this work's contributions to practice and research. Moreover, we describe the limitations of this work and outline future research directions related to the blockchain trilemma. We conclude with a summary of this work and our key takeaways in section VI.

II. Theoretical Foundations and Related Research

This section explains the foundations needed to better understand how blockchain system configurations influence manifestations of the blockchain trilemma. In subsection II-A, we briefly explain the foundations of blockchain technology, introduce the blockchain trilemma subconcepts, and describe trade-offs between these subconcepts. Subsection II-C gives an overview of related research on the blockchain trilemma.

A. Blockchain Technology

Blockchain technology is a special form of distributed ledger technology that enables the operation of blockchain systems—distributed databases designed to record transactions securely and consistently [21]. Blockchain systems store replicas of a record of transactions on distributed computing devices called nodes. The record is structured into blocks, each containing a batch of transactions. Except for the genesis block, each block references its preceding block with that predecessor's hash value [22]. A sequence of blocks, each referencing its predecessor, forms a *blockchain*.

Most blockchain systems (e.g., the Bitcoin and Ethereum systems) operate as replicated state machines, where validating nodes maintain consistent local replicas of the record by redundantly executing the same protocol (see Figure 1), such as for validating and verifying transactions and blocks. The local state of a validating node is defined by its stored blockchain.

When a new transaction is created, a node broadcasts it to its adjacent validating nodes. Each validating node processes incoming transactions by executing the same protocol, ensuring redundant validation. Upon successful validation, the node's state transitions accordingly.

To maintain consistency among replicas, validating nodes use consensus mechanisms. Consensus mechanisms shape key properties of blockchain systems, including their permission models, finality models, and fault tolerance. In terms of permission models, a blockchain system can be *permissionless*, where all validating nodes have equal rights to participate (e.g., in the Bitcoin system), or *permissioned*, where participation is restricted based on predefined rules [21], [23], such as in systems using the Hyperledger Fabric protocol. Such permissions determine who can join the network and participate in consensus finding.

Finality models define when blocks appended to the blockchain are considered finalized, meaning they cannot be altered or reverted. In immediate (or deterministic) finality, a block is considered finalized as soon as it is added to the blockchain. In probabilistic finality, a block's finality increases as more blocks are added on top, reducing the likelihood of reorganization over time.

In terms of fault tolerance, blockchain systems can be omission-tolerant, crash fault-tolerant, and Byzantine fault-tolerant [1], [21], [24], [25]. *Omission tolerance* refers to blockchain systems that can compensate for network messages that are lost in transit. *Crash-fault tolerance* refers to the ability of a blockchain system to compensate for validating nodes that are (temporarily) unavailable, for example, because the validating nodes crashed or the network connection is unreliable. *Byzantine fault tolerance* [24], [25] extends crash-fault tolerance by the ability to compensate for accidental faults and deliberate attacks. Accidental faults include software bugs and misconfiguration, while adversarial attacks involve strategies, such as in selfish mining [26]–[28].

Consensus mechanisms strongly influence the performance of blockchain systems [1], [21], especially in terms of transaction processing rates (i.e., throughput). Voting-based consensus mechanisms with immediate finality, such as Practical Byzantine fault tolerance (PBFT) [29], [30], experience performance degradation as network size increases. This is mainly due to the higher communication complexity required for consensus among an increasing number of validating nodes. For consensus

mechanisms with probabilistic finality, such as Nakamoto consensus in the Bitcoin system [22], throughput is often less affected by changes in network size, but consistency assumptions are strongly relaxed. Instead of ensuring that all validating nodes store the same version of the replica at all times, they provide eventual consistency, where most validating nodes converge to a consistent state over time [21].

While blockchain technology [22] is a prominent example of distributed ledger technology, the underlying principles, such as consensus and replicated state machines, are not unique to blockchains. Many distributed systems achieve consistency and fault tolerance through similar mechanisms, even if they do not employ a strict blockchain data structure. For instance, Hyperledger Fabric utilizes traditional databases to store data, while IOTA organizes transactions in a directed acyclic graph (DAG) rather than a linear sequence of blocks.

Although the 'blockchain trilemma' was originally formulated to describe the trade-offs faced by blockchain systems, similar tensions arise in a broader class of distributed databases that rely on replication and consensus. Thus, the blockchain trilemma encapsulates a fundamental design challenge: achieving an optimal balance among DoD, scalability, and security is inherently difficult in various distributed database systems that seek consensus across nodes. This observation extends the relevance of the trilemma beyond blockchain systems to distributed databases that use replicated state machines and consensus. This work considers such distributed databases with a focus on blockchain systems. It focuses on the role of validating nodes or peers in general distributed databases, excluding client nodes, as indicated by the dashed border in Figure 1.

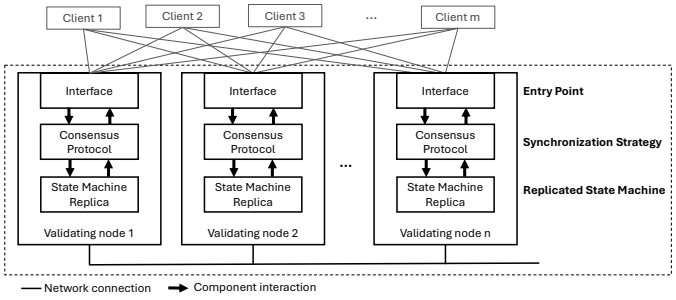

Fig. 1. Overview of blockchain system architectures (adapted from Leinweber et al. [31]). The dashed border encloses the part of a blockchain system in the focus of this work.

B. Blockchain Trilemma Subconcepts and Their Interrelationships

Optimizing blockchain system configurations to balance the blockchain trilemma subconcepts is essential to meet non-functional requirements. Achieving this balance requires a deep understanding of the interrelationships between DoD, scalability, and security in real-world blockchain systems. Despite the significance of the blockchain trilemma, extant literature presents multiple and sometimes conflicting definitions of its subconcepts, diluting the theoretical foundation of the blockchain trilemma. This delusion makes assignments of suitable constructs and corresponding metrics to blockchain trilemma subconcepts difficult, challenging empirical analyses for finding Pareto-optimal configurations of blockchain systems.

In this work, we use the term *construct* to refer to a dimension of a blockchain trilemma subconcept. Constructs are operationalized through metrics. A *metric* is a mathematically defined assignment of values (i.e., *input variables*) to objects (i.e., *output variables*) (cf. [32]). An *input variable* is an input to a metric and can often be manipulated in experiments as an *independent variable* (e.g., block size). In experiments, output variables can be *dependent variables* if they help operationalize a construct. Figure 2 illustrates the interrelationships between these key terms.

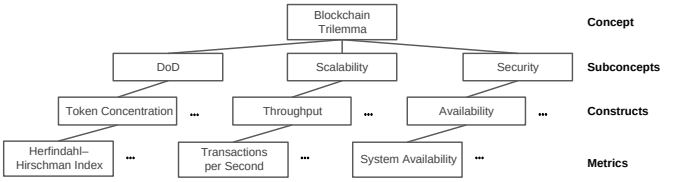

Fig. 2. Interrelationships between the blockchain trilemma, subconcepts, constructs, and metrics.

The following offers foundations for grasping the blockchain trilemma and its subconcepts and their interrelationships. These foundations form the basis for assigning constructs and corresponding metrics in section IV.

1) Blockchain Trilemma Subconcepts: Appropriately addressing the blockchain trilemma–the presumed impossibility of simultaneously maximizing DoD, scalability, and security of blockchain systems–poses a fundamental challenge in the development of blockchain systems that operate as replicated state machines [1], [2], [33]. Although the blockchain trilemma is widely recognized as a phenomenon, the definitions of its subconcepts remain inconsistent in existing literature. For example, DoD is described as the degree to which validating nodes equitably and (quasi-)autonomously participate in consensus finding [21], [34], [35], the extent to which validating nodes are geographically distributed and show different networks conditions (e.g., in terms of connectivity and synchrony due to bandwidth variations) [1], [36], [37], and the extent of equal wealth distribution [2], [11], [12], [35], [38]. Scalability is often associated with maximum transaction processing rates (i.e., throughput) [8], [12], [36], [39] or the maximum number of validating nodes that does not decrease throughput [21], [40]. Security is often related to availability, fault tolerance [7], [8], [12], and partition tolerance [8], [15], [36]. The following offers harmonized definitions of the blockchain trilemma subconcepts that we developed based on extant literature. In line with these definitions, we briefly outline common approaches to enhance blockchain systems along each subconcept. The results of this study will be grounded in these definitions.

a) Degree of Decentralization: DoD refers to the degree to which validating nodes equitably and (quasi-)autonomously contribute to consensus finding in a blockchain system [21], [34], [35], [41].

Blockchain systems achieve high DoD when validating nodes equitably participate in consensus finding. Such influences originate from actions including the proposal of blocks to be appended to the blockchain and (quasi-)autonomous acceptance or rejection of incoming blocks as in atomic broadcast protocols [1]. In an ideal Bitcoin system with high DoD, for example, validation nodes are equipped with equal computing power (e.g., CPU power), operate in similar settings (e.g., in terms of networks) and therefore can equitably participate in consensus finding by proposing new blocks and accepting or rejecting proposed blocks [1], [21], [34]. Such consensus mechanisms mimic a democratic decision process in which a proposal is made, debated, and finally accepted or rejected by people who can equitably contribute to the decision-making process.

b) Scalability: Scalability refers to the ability of a blockchain system to handle changing amounts (e.g., number of validating nodes and transactions per second) of workloads [1], [4], [21], [40], [42], [43].

High throughput, referring to 'amounts' of processed transactions in a specific timespan, is often achieved by reducing the number of validating nodes, as in leader-based consensus mechanisms (e.g., HotStuff, Paxos, and Raft [1], [44], [45]). In these mechanisms, a designated leader validates and propagates new blocks, significantly reducing message complexity and increasing transaction throughput [1], [46]. Similarly, in PoS-based consensus mechanisms, such as used in the Ethereum system [5], [6], a leader is (pseudo-)randomly selected from a pool of validating nodes that deposited tokens (i.e., a stake). The selected validating node proposes a block, which is then validated by all validating nodes before finalization. Although this design enhances scalability, it decreases DoD.

c) Security: Security refers to the degree to which a blockchain system remains operational and is resilient against faults, network partitions, and malicious attacks [3], [7], [8].

Consensus mechanisms, such as HotStuff, Nakamoto Consensus, and PBFT, ensure that blockchain systems can continue to operate correctly despite faulty or malicious validating nodes. Such fault–tolerant consensus mechanisms use redundancy and cryptographic techniques to achieve robustness against accidental faults and deliberate attacks, enhancing security.

The design of consensus mechanisms strongly influences availability and integrity, two major security characteristics of blockchain systems [21]. During synchronization, blockchain systems could be partitioned, which (temporarily) brings them to an inconsistent state. In such a state, validating nodes can respond differently to identical requests depending on their local blockchain replication, decreasing availability [47]–[49].

Blockchain systems protect the integrity of blockchains through the use of hash values, which link blocks, and economic deterrents. Each block references its predecessor via a hash value, making tampering computationally prohibitive. Furthermore, consensus mechanisms based on PoW and PoS introduce economic costs to adversarial behavior, deterring attacks. In PoW, an attacker must control a majority of computational power, while in PoS, they must stake a large number of tokens, making dishonest behavior financially less viable.

2) Interrelationships Between the Blockchain Trilemma Subconcepts: Understanding the interrelationships between the blockchain trilemma subconcepts and how different blockchain system configurations influence them is essential for identifying Pareto-optimal configurations. The following elucidates interrelationships between the blockchain trilemma subconcepts under consideration of established theories, such as the CAP theorem referring to the impossibility of simultaneously maximizing consistency, availability, and partition tolerance [47]–[49].

a) DoD vs Scalability: Blockchain systems with high DoD commonly involve all validating nodes to equitably and (quasi-)autonomously participate in consensus finding. Instead of a single validating node deciding the state of a blockchain system, all validating nodes collectively decide (e.g., based on majority votes). In a highly decentralized blockchain, all validating nodes must receive and process blocks to collectively agree on state transitions. This increases message complexity and communication overhead, ultimately reducing transaction throughput [21], [47].

To enhance scalability in terms of throughput, the number of validating nodes that must receive relevant blocks to collectively decide on global state transitions can be decreased [1], [21], [47]. For example, the Ethereum system transitioned from a PoW-based consensus mechanism to a PoS-based one, decreasing the number of validating nodes involved in consensus finding to increase transaction throughput. Reducing the number of validating nodes improves throughput but decreases DoD.

b) DoD vs Security: In blockchain systems with high DoD, such as an ideal Bitcoin system, all validating nodes can equitably participate in consensus finding (e.g., by proposing, accepting, and rejecting blocks). Equitable participation requires validating nodes to exchange network messages to synchronize and transition the blockchain system to a subsequent consistent state. Synchronization takes time during which blockchain systems are in an inconsistent state. Validating nodes in different network partitions store different versions of the blockchain. The time for synchronization is often influenced by the geographical distribution of validating nodes, heterogeneity of validating nodes, and different networks [1], [37], such as related to bandwidth and connectivity. Inconsistent states decrease availability [15], [47], [49] and ease successful double-spending [26], [28], decreasing security of blockchain systems.

To enhance security, a trusted party can operate multiple (geographically distributed) validating nodes that store replications of a blockchain and validate transactions. For synchronizing validating nodes in this setting, a crash-fault consensus mechanism can be used with immediate finality (e.g., Kafka and Raft [50], [51]). By limiting participation to known and trusted validating nodes, this approach decreases the risk of Byzantine nodes disrupting consensus finding. However, this also shifts the system toward a more centralized trust model, reducing the DoD. Attacks (e.g., double-spending and selfish-mining) caused by soft forks originating from network partitions can be mitigated by using a centralized consensus mechanism with immediate finality [21], [51] that is executed on validating nodes under the control of the trusted party. Because only a trusted party determines the state, such blockchain systems

have a low DoD.

c) Scalability vs Security: Blockchain systems can reach high scalability when only a few validating nodes with ideal network conditions participate in consensus finding [21], [52]. Moreover, to enhance communication speed, those validating nodes should be located in close proximity [1]. This approach, however, decreases the security of blockchain systems because system operation relies on a few validating nodes. Only a few validating nodes need to be compromised to take control over the blockchain system. Moreover, locating validating nodes in close proximity increases the likelihood that all validating nodes, thus the blockchain system, crash, for example, due to outages.

To enhance security, a sufficient number of validating nodes must be operational and store consistent replications of the blockchain [21]. To anticipate crashes, validating nodes should be geographically distributed. If a validating node crashes, a copy of the ledger can be retrieved from another. Synchronization of a large number of validating nodes that are geographically distributed, however, slows down transaction finalization, decreasing scalability [21].

C. Related Research on Measuring the Blockchain Trilemma Subconcepts

Related conceptual works offer definitions of the blockchain trilemma subconcepts and highlight trade-offs between these subconcepts [1], [3], [21], [53]–[55]. However, definitions of the blockchain trilemma subconcepts strongly vary across prior surveys. For example, Xu et al. [53] define DoD with respect to network size. In contrast, Xiao et al. [1] define DoD based on the geographical diversity of validating nodes. Other works emphasize autonomy and equal chances of validating nodes to contribute to consensus finding as an important aspect of DoD [3], [21], [34], [41]. By proposing disparate definitions of DoD, scalability, and security, prior surveys take different angles on the blockchain trilemma subconcepts. The construct definitions are, however, inconsistent and may not directly relate to the blockchain trilemma, which dilutes its concept, making it difficult to understand.

In addition to conceptual works, empirical studies propose multiple constructs and metrics to quantify the blockchain trilemma subconcepts. Focusing on DoD, scalability or security, in separation–not the entire blockchain trilemma–prior research [7]–[9], [26], [56] conducted experiments to better understand influences of blockchain system configurations on these subconcepts. Those works present multiple constructs for the same blockchain trilemma subconcepts. For example, changes in throughput and changes in confirmation latency due to faulty nodes were proposed to calculate fault tolerance, [7], [8] and stale block rate [10], [57] is used to operationalize the security subconcept. Similarly, throughput (i.e., transactions per second; [7], [8], [58]) and confirmation latency (e.g., [9], [57], [59], [60]) are used to estimate scalability subconcept. The same constructs are used to investigate different blockchain trilemma subconcepts. For example, availability was proposed to investigate scalability [9] and security [10]. Selecting suitable constructs remains difficult because there is insufficient justification for the validity of the proposed constructs and their operationalizations for measuring the blockchain trilemma subconcepts and how to interpret the constructs.

For blockchain systems using PoW-based consensus mechanisms, Nakai et al. [2] formalized the blockchain trilemma and demonstrated its existence in simulations. The authors used the number of transactions per block processed per time interval to estimate scalability, the inverse fork rate (number of soft forks) as a security construct, and token concentration to approximate DoD. Despite offering valuable insights, the proposed operationalized constructs only apply to blockchain systems using PoW-based consensus mechanisms.

For blockchain systems using PoS-based consensus mechanisms, Fu et al. [59] and Quattrocchi et al. [12] propose the wealth distribution and the token concentration as constructs to investigate DoD; they propose throughput and confirmation latency to investigate scalability. Fu et al. [59] used transaction fees as a construct to investigate security. They reveal that high transaction fees indicate a more secure blockchain system. This is mainly because transaction fees serve as an incentive to validating nodes. The authors propose average transaction fees as an operationalized construct for investigating security. Quattrocchi et al. [12] investigated security of blockchain systems based on the cost of attack. The higher the cost of attack, the more secure a blockchain system is assumed to be: it becomes increasingly hard for attackers to accumulate sufficient resources to successfully perform the attack. Mssassi et al. [36] propose a formalization of the blockchain trilemma for blockchain systems with PoW-based and PoS-based consensus mechanisms. They used the influence of token amount or hashing power owned by validating nodes to participate in consensus finding to assess DoD. Similarly, they used security thresholds (e.g., more than 50% of honest validating nodes participating in consensus finding), and metrics for throughput and confirmation latency constructs to measure scalability.

With a focus on permissioned blockchain systems, Wang et al. [15] mapped the blockchain trilemma to the CAP theorem [47], [48]. The authors make three assumptions: (1) eventual consistency of the blockchain system, (2) more than two-thirds of validating nodes are honest, and (3) validating nodes have the same computing power (e.g., memory) for the validity of their model. Building on these assumptions, the authors mapped consistency to security, availability to scalability, and partition tolerance to the DoD of a blockchain system. To calculate consistency, the authors use the probability of having forks in a blockchain system. To estimate availability, they propose computing throughput and, as a construct for DoD, the probability that a partitioned network cannot function properly. While the proposed constructs are helpful, their applicability to blockchain systems that do not meet those assumptions remains unclear.

In summary, prior research proposes useful constructs and associated metrics to quantify the blockchain trilemma subconcepts in blockchain systems with different designs. However, justification for the suitability of proposed constructs is often lacking, and literature on operationalized constructs is scattered across various sources. This leads to difficulty in selecting suitable constructs and metrics. To tackle this

issue, the suitability of constructs and associated metrics for measuring the blockchain trilemma subconcepts needs to be better understood.

III. Methods

We developed a set of constructs and associated metrics to operationalize the subordinate concepts of the blockchain trilemma in two steps. First, we conducted a systematic literature search [17] to compile an extensive set of relevant publications on the blockchain trilemma. Second, we analyzed the collected literature using abductive thematic analysis [18] to extract the constructs and associated metrics used to measure the blockchain trilemma subconcepts. The following subsections detail these two steps.

A. Literature Search

We conducted a systematic literature search [17] to identify publications that present constructs and associated metrics for analyzing blockchain trilemma subconcepts. To evaluate the relevance of publications, we applied five inclusion criteria: *English language*, *level of detail*, *peer-reviewed*, *topic fit*, and *uniqueness* (see Table I).

We used the search string: *("benchmarking" AND "blockchain trilemma")* to compile a set of publications on the blockchain trilemma via ACM Digital Library, IEEEXplore, ScienceDirect, and Scopus on March 26, 2024. This query was informed by a preliminary review of domain-specific terminology and indexing practices. The search returned 1,814 potentially relevant publications: 1,258 from ACM Digital Library, 546 from IEEE Xplore, 7 from ScienceDirect, and 3 from Scopus.

We screened all 1,814 publications based on title, keywords, and abstract against our inclusion criteria. This step excluded 436 publications: 4 were not in English, 348 were not peer-reviewed, 77 lacked topic fit, and 7 were duplicates, leaving 1,378 potentially relevant records.

We subsequently used the same inclusion criteria to assess the relevance of the 1,378 potentially relevant publications based on full texts. We excluded 1,211 publications due to insufficient detail. Moreover, we excluded 24 additional publications due to insufficient topic fit. The second relevance assessment yielded 143 relevant publications.

During our abductive analysis (Section III-B), we observed underrepresentation of constructs related to DoD and security. To address this and enhance theoretical sufficiency, we conducted targeted theoretical sampling [19], a purposive strategy used in abductive research to refine and deepen emerging conceptual insights. We used Google Scholar and re-applied the original search string to identify studies omitted in the initial search due to indexing limitations. This supplemental search yielded 12 additional publications that met our inclusion criteria (Table I), resulting in a final corpus of 155 publications.

B. Literature Analysis

We applied abductive thematic analysis [18]–[20] to identify constructs and metrics associated with the blockchain trilemma subconcepts. Abductive thematic analysis combines inductive coding and deductive theorizing in an iterative process, allowing researchers to move between data and theoretical constructs to generate conceptually rich themes. This approach enables theory development that is both grounded in the literature and informed by existing conceptual frameworks [19].

We adopted the blockchain trilemma and its subconcepts (see Section II-B) as a theoretical lens. These subconcepts are inherently broad and abstract; hence, we sought to enrich and refine them through inductive engagement with the literature. Guided by abductive reasoning, we iteratively moved between data and theory, adjusting our understanding of both as patterns emerged.

We began by inductively coding passages from the 155 publications that referenced constructs or metrics relevant to the trilemma. Initial codes (e.g., *fault tolerance*, *throughput*) were derived from the data. These were continuously refined through theoretical reflection, developing a two-way relationship between emerging empirical codes and conceptual understanding.

We defined a theme as a construct that was empirically grounded and associated with at least one metric. Themes thus captured both descriptive patterns and theoretical relevance within the blockchain trilemma. To reduce redundancy, overlapping constructs were merged. For example, *robustness* was subsumed under *fault tolerance*, and *availability* and *success rate* were grouped as indicators of scalability.

To ensure reliability and reduce subjective bias, multiple researchers independently coded subsets of the literature. We resolved discrepancies through discussion, refining our thematic structure iteratively. Through cycles of coding, comparison, and theoretical integration, we converged on a stable set of 14 final themes, distilled from 397 initial codes.

One construct was excluded due to conceptual inconsistency and lack of empirical support. Despite attempts to contact the original authors for clarification, the issue remained unresolved. In line with abductive logic, which emphasizes conceptual clarity and explanatory adequacy, we excluded this construct from our results.

We then mapped each theme to one of the three trilemma subconcepts based on both patterns in the literature and alignment with conceptual definitions (Section II-B). For instance, we assigned *throughput* and *confirmation latency* to scalability, and *fault tolerance* and *stale block rate* to security. Mapping disagreements were resolved through collaborative review and consensus.

In a final review step, we assessed the internal coherence and theoretical saturation of the thematic structure. No new themes emerged during the last coding iterations, indicating thematic saturation. This confirmed that the identified constructs were both empirically grounded and conceptually robust within the blockchain trilemma framework.

IV. Constructs, Metrics, and Analysis Approaches

This section first presents an overview of the constructs used to operationalize the blockchain trilemma subconcepts in section IV-A. We link the constructs to the blockchain trilemma subconcepts, explain the operationalization of the

TABLE I
INCLUSION CRITERIA USED IN THE LITERATURE SEARCH.

Criterion	Description
English Language	The publication must be in English.
Level of detail	The publication must present sufficient descriptions and explanations of the investigated blockchain trilemma subconcept(s) and used construct(s).
Peer-Reviewed	The publication is peer-reviewed.
Topic Fit	The publication focuses on measuring at least one of the blockchain trilemma subconcepts, and the constructs apply to core blockchain systems with no specialized hardware (e.g., trusted execution environments) and no peripheral software artifacts (e.g., payment channel networks).
Uniqueness	The publication must be the latest version and must not be a duplicate in the literature set.

constructs through merits, and point out the limitations of the operationalized constructs. Moreover, we offer examples of how the constructs can be used. Subsection IV-B showcases uses of the operationalized constructs in analysis approaches.

A. Constructs and Metrics to Measure the Blockchain Trilemma Subconcepts

We identified 14 constructs associated with the blockchain trilemma subconcepts (see Table II): 5 for DoD, 3 for scalability, and 6 for security. These constructs are operationalized through 16 metrics, which are detailed in the following. Supplementary material A offers an overview of the input variables used in the metrics.

1) Degree of Decentralization: To estimate the DoD of blockchain systems, we identified five metrics with the following constructs: *block proposal randomness*, *geographical diversity*, *hashing power distribution*, *token concentration*, and *wealth distribution*.

Block Proposal Randomness: *The degree to which it is uncertain what validating node will propose the next block.*

Depending on the consensus mechanism, a static (e.g., in Raft [50]) or randomly selected validating node (e.g., in Nakamoto consensus [22]) proposes the next block to be appended to the blockchain. To compute the extent to which the selection of such a validating node is random, Shannon entropy is often used [11], [13], [14], [35], [38]. Shannon entropy is a measure of uncertainty regarding the occurrence of discrete events, such as the proposal of a new block by validating nodes. It is defined as follows:

$$H(X) = H(p_1, p_2, ..., p_n) \tag{1a}$$

$$= -k \sum_{i=1}^{n} \left(\frac{b_i}{\sum_{j=1}^{n} b_j} \right) \log_2 \left(\frac{b_i}{\sum_{i=j}^{n} b_j} \right) \tag{1b}$$

Shannon entropy $H(X)$ represents the entropy of block proposal randomness. n is the total number of validating nodes. b_i refers to the number of blocks that a validating node i proposed. $p_i = \frac{b_i}{\sum_{j=1}^{n} b_j}$ corresponds to the probability of a validating node i proposing a block in a block creation interval. To adjust Shannon entropy, the scaling factor k can be used [14]; often, it is set to 1.

A higher Shannon entropy indicates more equal chances of validating nodes to propose the next block, indicating a high DoD [35]. Conversely, a lower Shannon entropy signifies uneven participation of validating nodes in the consensus finding, reflecting a lower DoD in the blockchain system [11].

The following exemplifies the use of Shannon entropy to estimate the DoD of a fictitious blockchain system with a PoW-based consensus mechanism. Suppose three validating nodes n_1, n_2, and n_3, which have proposed $n_1 = 1$, $n_2 = 1$, and $n_3 = 8$ blocks, respectively. The total number of proposed blocks is 10, leading to the following probabilities:

$$p_1 = \frac{1}{10}, \quad p_2 = \frac{1}{10}, \quad p_3 = \frac{8}{10}$$

Substituting into equation 1:

$$H(X) = -\left(\left(\frac{1}{10} \times \log_2 \frac{1}{10} \right) \times 2 + \left(\frac{8}{10} \times \log_2 \frac{8}{10} \right) \right)$$
$$= 0.922$$

The highest value for DoD that can be calculated using equation 1 is reached when all validating nodes have equal probabilities. If all validating nodes from the above example have the same probability of $\frac{1}{3}$ to propose the next block, maximum entropy corresponds to 1.585. Thus, the calculated entropy of 0.922 is moderate, indicating that the exemplary blockchain system is rather centralized.

Shannon entropy can be used to estimate DoD based on the probabilities of validating nodes to propose the next block in blockchain systems with random leader selection [38]. Shannon entropy only focuses on block proposals, neglecting which proposed blocks are actually finalized. Network effects, such as bandwidth variations, can influence block propagation speed and decrease DoD. Blocks in a partition with higher bandwidth propagate faster than those in lower-bandwidth partitions [1], [61]. In that setting, the likelihood that the blockchain system finalizes a block from a validating node in a partition with a higher bandwidth is higher than for the block that was first propagated in a partition with a lower bandwidth. Consequently, validating nodes of the partitions with different bandwidths cannot equitably participate in consensus finding, which centralizes the blockchain system [62]. Such aspects are not reflected in Shannon entropy when focusing on block proposals. Even if the block proposal probabilities per validating node are equal, corresponding to maximum Shannon entropy, the actual DoD of the blockchain system can be low.

TABLE II
OVERVIEW OF THE IDENTIFIED CONSTRUCTS ASSOCIATED WITH THE BLOCKCHAIN TRILEMMA SUBCONCEPTS (I.E., DEGREE OF DECENTRALIZATION, SCALABILITY, AND SECURITY).

	Construct	Description
Degree of Decentralization	Block Proposal Randomness	The degree to which it is uncertain what validating node will propose the next block.
	Geographical Diversity	The degree to which validating nodes in a blockchain system are distributed across different locations.
	Hashing Power Distribution	The extent to which hashing power is distributed among the validating nodes that compete to propose the next block.
	Token Concentration	The distribution of token shares that validating nodes own in a blockchain system.
	Wealth Distribution	The degree of inequality between validating nodes in terms of token ownership.
Scalability	Availability	The degree to which a blockchain system is operational and delivers up-to-date responses.
	Confirmation Latency	The timespan between the proposal of new blocks and their confirmation.
	Throughput	The highest number of transactions a blockchain system can process in a specified timeframe.
Security	Availability	The degree to which a blockchain system operates correctly at an arbitrary time.
	Consistency	The degree to which all validating nodes are in a shared and agreed-upon state.
	Cost of Attack	The cost in fiat currency to gain control of a blockchain system through an attack.
	Fault Tolerance	The degree to which a blockchain system operates correctly despite experiencing accidental or Byzantine faults.
	Reliability	The continuity of a blockchain system to offer correct service.
	Stale Block Rate	The number of blocks that have been propagated in a blockchain system but not finalized in the mainchain in a specified timespan.

Geographical Diversity: *The degree to which validating nodes in a blockchain system are located in different locations.*

Blockchain systems are distributed systems where nodes are often physically distributed across different locations. Equation 3 can be used to calculate geographical diversity GD of blockchain systems [37]:

$$GD = \frac{(GD_{excl} - GD_{target}) - GD_{equal}}{GD_{excl} - GD_{equal}} \tag{3}$$

GD_{excl} denotes the geographical diversity when all validating nodes of a blockchain system are operated in one location. GD_{target} signifies the actual geographical diversity of a blockchain system to be quantified. GD_{equal} denotes the geographical diversity when the number of validating nodes is equally distributed across locations. Both GD_{target}, GD_{excl}, and GD_{equal} are calculated using the auxiliary geographical diversity GD_{aux}, as defined in equation 4. GD_{aux} defines the standard deviation of the distribution of validating nodes for GD_{target}, GD_{excl}, and GD_{equal}.

$$GD_{aux} = \left(2 - \frac{\log_{(|N|+1)} |N_t| - \log_{(|N_t|+1)} |N_t|}{\log_2 |N_t| - \log_{(|N_t|+1)} |N_t|}\right) \tag{4a}$$

$$\times \sqrt{\frac{\sum_{i=1}^{|N_t|} (|n_i| - \mu)^2}{|N_t|}} \tag{4b}$$

$|N_t|$ denotes the total number of locations considered in the calculation. $|N|$ signifies the number of locations where validating nodes of a blockchain system operate. μ represents the mean of the validating nodes (i.e., the number of validating nodes divided by the number of operating locations). $|n_i|$ is the number of validating nodes operating in the i_{th} location. The constant values in the GD_{aux} are used as scaling factors. Equation 5 defines μ with n denoting the total number of validating nodes:

$$\mu = \frac{n}{|N_t|} \tag{5}$$

High geographical diversity means validating nodes are spread across multiple locations, avoiding the dominance of validating nodes in a single location in consensus finding [1]. High geographical diversity reduces the influence of local laws, enhancing regulatory neutrality [37], which contributes to a higher DoD [1].

The following example illustrates how equation 3 can be used. Suppose there are ten locations out of which validating nodes operate in four locations only. Suppose the total number of validating nodes in a blockchain system equals 100. Using equation 5, $\mu = \frac{100}{10} = 10$. The number of validating nodes operating in each location equals $\frac{100}{4} = 25$. Using equation 4 to calculate GD_{target} leads to:

$$GD_{target} = \left(2 - \frac{\log_{(4+1)} 10 - \log_{(10+1)} 10}{\log_2 10 - \log_{(10+1)} 10}\right)$$
$$\times \sqrt{\frac{(25-10)^2 \times 4 + (0-10)^2 \times 6}{10}} = 22.05$$

If all 100 validating nodes operate in one location, using equation 4, GD_{excl} equals:

$$GD_{excl} = \left(2 - \frac{\log_{(1+1)} 10 - \log_{(10+1)} 10}{\log_2 10 - \log_{(10+1)} 10}\right)$$

$$\times \sqrt{\frac{(100-10)^2 \times 1 + (0-10)^2 \times 9}{10}} = 30.00$$

If the 100 validating nodes are equally distributed across 10 locations, 10 validating nodes are located in each location, GD_{equal} equals:

$$GD_{equal} = \left(2 - \frac{\log_{(10+1)} 10 - \log_{(10+1)} 10}{\log_2 10 - \log_{(10+1)} 10}\right) \times \sqrt{\frac{(10-10)^2 \times 10}{10}} = 0$$

With the results obtained from calculations of GD_{target}, GD_{excl}, and GD_{equal} in three forms, the geographical diversity of the blockchain system is as follows:

$$GD = \frac{(30 - 22.05) - 0}{30 - 0} = 0.250$$

DoD can be increased in terms of geographical diversity if validating nodes are equally distributed across all possible locations compared to when they are located in a few locations.

A major limitation of this metric is the lack of a clear definition of 'location.' The criteria for determining when a location is considered 'new' are unclear. New locations could be defined based on factors such as physical distance, cultural distinctions, or regulatory differences. The selection of relevant locations is left to the discretion of practitioners, which can substantially influence the measured geographical diversity. Validating nodes located in different locations can have different bandwidths and influence consensus finding (e.g., hashing power or token stakes). In blockchain systems that use leader selection based on PoW (e.g., the Bitcoin system), for example, validating nodes with little bandwidth are at a disadvantage. Such disadvantages are neglected in equation 3, but could centralize blockchain systems to validating nodes with wide bandwidth [62]. Moreover, to use equation 3, it is assumed that the locations of all validating nodes are known, which hardly applies to public-permissionless blockchain systems like the Bitcoin system. For example, validating nodes can obfuscate their actual location through virtual private networks or TOR network [63]. In such cases, equation 3 will produce inaccurate results. Furthermore, package routing through central internet service providers could decrease the DoD of blockchain systems [64] but is also neglected in equation 3. While being useful as a construct of DoD, geographical diversity likely positively correlates with security: a high geographical diversity can increase resilience to network disturbances and robustness against outages [1]. Thus, to quantify trade-offs between blockchain trilemma constructs, equation 3 could be unsuitable.

Hashing Power Distribution: *The extent to which hashing power is distributed among all validating nodes that compete to propose the next block.*

In most blockchain systems using PoW-based consensus mechanisms, such as the Bitcoin system, validating nodes compete to produce the next block by computing hash values that meet a specified requirement, such as being smaller than a hash target value. The process of computing hash values from random nonces in concatenation with block data is called 'mining.' Nodes with more hashing power (mining power) are more likely to propose the next block than those with less hashing power.

Equation 9 defines how Nakamoto coefficient is calculated to estimate the hashing power distribution in blockchain systems [2], [11], [12], [35], [59]. Nakamoto coefficient is the minimum number of validating nodes needed to surpass a threshold in hashing power required to control a blockchain system. For example, an attacker must control more than 50% of the hashing power to control a blockchain system with a PoW-based consensus mechanism to succeed in 51% attacks [1], [12], [26].

$$NC = min\left\{k \in [1, 2, \ldots, n] : \sum_{i=1}^{k} p_i \geq threshold\right\} \quad (9)$$

n denotes the number of validating nodes. p_i denotes the resources (e.g., hashing power) of the i_{th} validating node.

A high Nakamoto coefficient is an indicator of a high DoD of a blockchain system. Hashing power is not concentrated in a few validating nodes, but most nodes possess similar hashing power. Thus, attackers must control a relatively large portion of validating nodes to have sufficient hashing power to control consensus finding. In contrast, a low Nakamoto coefficient indicates that a small portion of validating nodes possesses a large fraction of resources. Attackers must only gain control over a few validating nodes to control consensus finding [2], [59].

Nakamoto coefficient can be used to estimate the DoD of blockchain systems that employ mining for leader selection. In the Bitcoin system, the top two mining pools—*Unknown* and *AntPool*—possess approximately 60.63% of the total hashing power [65]. Using equation 9, the hashing power distribution in the Bitcoin system is 2 because the two mining pools control more than 50% of the hashing power. This indicates a low DoD of the Bitcoin system.

Nakamoto coefficient offers a simple estimation of the DoD of a blockchain system. However, the Nakamoto coefficient only indicates the minimum number of validating nodes required to compromise consensus in a blockchain system, neglecting overall resource distributions across all validating nodes in a blockchain system. In a blockchain where all validating nodes have equal hashing power, for example, Nakamoto coefficient will be high but will not enable inferences of the hashing power distribution in the blockchain system. Moreover, Nakamoto coefficient neglects factors influencing the success of compromising a blockchain system, such as the position of a validating node in a network, the number of validating nodes, as well as bandwidth and network quality [26], [28].

Token Concentration: *The distribution of token shares that validating nodes own in a blockchain system.*

In many public blockchain systems, including the Bitcoin and Ethereum systems, nodes are incentivized to participate in consensus finding by rewards in the form of tokens: if a node produces a block that is (probabilistically) finalized in the blockchain system, that node is rewarded by a defined

token amount. The share of tokens a validating node owns can reflect its success in proposing finalized blocks. In blockchain systems with consensus mechanisms that select leaders based on their shares of staked tokens, such as in the BitShares and Cosmos Tendermint systems [1], [66], [67], the token shares of validating nodes can also indicate the nodes' influence on consensus finding. In both cases, the concentration of token ownership can be a helpful estimate of the DoD of blockchain systems.

To calculate token concentration, Herfindahl-Hirschman Index HHI is often used [2], [11], [59]. HHI is a metric to calculate the concentration of token ownership (e.g., with a focus on received token rewards or staked tokens) by validating nodes in a blockchain system [59]. HHI can be formalized as follows [2], [11], [59]:

$$HHI = \sum_{i=1}^{n} \left(\frac{t_i}{t_{total}} \times 100 \right)^2 \tag{10a}$$

$$= 10,000 \times \sum_{i=1}^{n} \left(\frac{t_i}{t_{total}} \right)^2 \tag{10b}$$

n denotes the number of validating nodes under investigation. t_i represents the token amount owned (or staked) by the i_{th} validating node. $t_{total} = \sum_{j=1}^{n} t_j$ signifies the total token amount owned (or staked) by all validating nodes considered in the investigation. $\frac{t_i}{t_{total}}$ is multiplied by 100 to translate the token shares owned per node into percentage values. According to the literature examined on the use of HHI in blockchain research [11], a blockchain system has a low DoD if $HHI > 2,500$. In economics (e.g., [68], [69]), $HHI < 1,000$ indicates a low market concentration (i.e., high DoD), $1,000 \leq HHI < 1,800$ a medium DoD, and $1,800 \geq HHI$ indicates high market concentration (i.e., low DoD).

As equation 10 produces results influenced by n [70], HHI needs to be normalized to be comparable across evaluations of blockchain systems with different n. To compute the normalized HHI_{norm}, equation 11 is commonly used [70]:

$$HHI_{norm} = \frac{n}{n-1} HHI - \frac{1}{n-1} \tag{11}$$

HHI and HHI_{norm} help estimate the DoD of blockchain systems based on the concentration of token ownership [11]. A high HHI indicates that a few validating nodes own most tokens, suggesting they dominate block proposals and often receive rewards. For stake-weighted influences on consensus finding, such as in BitShares' consensus mechanism [1], [66] and stake-weighted quality of service in Solana [71], this reflects centralization, as nodes with more staked tokens have greater influence. Only staked tokens matter, not total token holdings. A low HHI indicates a more equal token distribution across validating nodes. This suggests that validating nodes have been similarly successful in gaining rewards, indicating equitable participation in consensus finding and, thus, a high DoD. In blockchain systems with stake-weighted influences on consensus finding, a low HHI moreover suggests that consensus power is more evenly distributed among validating nodes, reinforcing decentralization.

Equation 10 can be used to investigate blockchain systems that involve tokens in consensus mechanisms (e.g., weighted PoS-based consensus mechanisms or distribution of block rewards). The following offers an example of how to use HHI to estimate the DoD of a fictitious blockchain system with a PoS-based consensus mechanism that distributes block rewards. Suppose eleven validating nodes n_0, n_1, ..., and n_{10} owned 5 tokens each. The total token amount owned by the two validating nodes is 55. Using equation 10, HHI of the fictitious blockchain system can be calculated as follows:

$$HHI = 10,000 \times \left(\left(\frac{5}{55} \right)^2 \times 11 \right) \approx 909.1$$

$HHI = 909.1$ and $HHI_{norm} = 999.91 (< 1,000)$ indicate a high DoD.

If n_0 owned 37 tokens, ten validating nodes owned 1.8 tokens each, and the remaining seven validating nodes owned no tokens, $HHI \approx 4,622$ and $HHI_{norm} \approx 5,084 (> 2,500)$, indicating a low DoD.

HHI focuses on the token amounts owned (or staked) by validating nodes, neglecting whether the tokens were received as block rewards or were simply transferred to the addresses of the validating nodes for other reasons. Because owned token amounts may not only correspond to earned block rewards, using any token amounts owned by validating to estimate DoD could bias HHI. Moreover, a one-to-one mapping of addresses and nodes is assumed to calculate HHI. This could lead to misinterpretations of HHI in sociotechnical settings where single users own multiple nodes and addresses (e.g., in Sybil attacks) [1], [11]. The cumulative token amounts of such users could be very high, leading to strong token concentration, while HHI still indicates low token concentration. To mitigate resulting misinterpretations, HHI should be applied to cumulative token amounts per user instead of addresses or validating nodes. As HHI offers a simple estimate of token concentration, it may not appropriately capture such more complex settings.

Wealth Distribution: *The degree of inequality between validating nodes in terms of token ownership.*

In many public blockchain systems, such as Bitcoin and Ethereum, nodes receive token rewards as an incentive to participate in consensus finding. When a node successfully proposes a block that gets confirmed, it earns a predefined token amount. As a result, the distribution of tokens to validating nodes can reflect their success in block proposals.

$Gini$ coefficient can be used to calculate wealth distribution [2], [11], [12], [35], [38], [59]. $Gini$ coefficient is a measure of inequalities (e.g., in terms of token ownership of validating nodes in a blockchain system) and is formalized as follows [11], [35], [59]:

$$Gini = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |t_i - t_j|}{2n \sum_{i=1}^{n} t_i} \tag{12}$$

n represents the total number of validating nodes. t_i denotes the number of tokens owned by a validating node i. t_j is the number of tokens owned by a validating node j.

The co-domain of $Gini$ coefficient ranges from 0, indicating equal wealth distribution among validating nodes, to 1, indicating high inequality in wealth distribution [11]. A high $Gini$ coefficient signifies significant wealth inequality among validating nodes, which may result from unequal opportunities to earn rewards for proposing blocks. Less-wealthy validating nodes may be hampered from participating in consensus finding, indicating a low DoD. In contrast, a low $Gini$ coefficient points to a more balanced wealth distribution among validating nodes. This suggests that nodes have more equal chances to participate in consensus finding, which corresponds to a high DoD.

Equation 12 can be used as follows. Suppose five validating nodes n_1, n_2, n_3, n_4, n_5, own the following tokens: $n_1 = 2$, $n_2 = 3$, $n_3 = 5$, $n_4 = 6$, and $n_5 = 2$. The mean differences of tokens owned by validating nodes can be calculated in pairs as follows:

$$pair1 = (|2-2|) \times 2 + (|2-3|) + (|2-5|) + (|2-6|) = 8,$$
$$pair2 = 7, pair3 = 9, pair4 = 12, pair5 = 8$$

Using equation 12, the wealth distribution in the exemplary blockchain system is calculated as follows:

$$Gini = \frac{8+7+9+12+8}{2 \times 5 \times 18} \approx 0.244$$

If a single validating node owned 18 tokens while other validating nodes owned none, the $Gini$ coefficient equals 0.8, indicating a lower DoD compared to 0.244 shown in the previous example. If all validating nodes owned equal amounts of tokens $Gini = 0$. The value of the $Gini$ coefficient obtained in the earlier example (i.e., 0.244) indicates that validating nodes have fairly equitable wealth, indicating a higher DoD.

The $Gini$ coefficient is an economic metric that offers a simple estimate of the DoD based on wealth distribution. The metric, however, underrepresents other considerations (e.g., use of reputation in weighted PoS) [72] sometimes used in consensus finding in addition to token ownership. Amounts of tokens owned by validating, therefore, do not guarantee their actual participation in consensus finding. Moreover, the metric neglects other factors that can influence consensus in addition to wealth distribution (e.g., bandwidth). Validating nodes with high bandwidth often have more influence on consensus finding than those with low bandwidth [1], [61]. Additionally, for the metric, it is assumed that each validating node is independently controlled. In practice, however, a single entity may operate multiple nodes and hold their tokens. To correctly use $Gini$ coefficient, it is essential to account for such social factors.

2) Scalability: We identified five metrics to operationalize the constructs *availability*, *confirmation latency*, and *throughput* of scalability.

Availability: *The degree to which a blockchain system is operational and delivers up-to-date responses.*

Before transaction data is available from all validating nodes, the nodes must have synchronized to be in a consistent state [1], [48], [49]. Prior to synchronization, states of validating nodes in blockchain systems with probabilistic finality are inconsistent. Validating nodes from different network partitions may respond differently to identical responses.

Equation 15 can be used to calculate availability $ASca$ as a construct of the scalability of blockchain systems [9]:

$$ASca = \frac{NumOfConfTr_{t_2}}{NumOfTr_{t_1}} \times 100\% \tag{15}$$

$NumOfConfTr_{t_2}$ signifies the number of confirmed transactions until time t_2. The number of transactions issued to a blockchain system at time t_1, with $t_1 < t_2$, is denoted by $NumOfTr_{t_1}$.

Availability of blockchain systems strongly depends on the throughput and time for synchronization of validating nodes [12]—key indicators of scalability of blockchain systems [8], [21], [61] In blockchain systems with high availability, validating nodes rapidly process transactions and synchronize, ensuring that all validating nodes quickly transition to a consistent and up-to-date state in a short time. After synchronization, all validating nodes provide identical and up-to-date responses to requests. Conversely, in blockchain systems with low availability, validating nodes slowly process transactions and synchronize, delaying the time until all validating nodes deliver up-to-date responses. Depending on the network partition a requested node is part of, users could receive different responses to identical requests. In blockchain systems with probabilistic finality, such as the Bitcoin system, slow synchronization can entail network partitions, where validating nodes temporarily have inconsistent states in different network partitions. As a result, depending on the network partition a validating node is part of, it may return up-to-date or outdated responses, indicating lower availability.

Equation 15 applies to most blockchain systems, including the Algorand, Bitcoin, and Ethereum systems [9], as illustrated in the following simplified example. The Ethereum system confirms approximately 1.326MM transactions in an exemplary day [73]. We assume that 1.5MM transactions were issued to the Ethereum system in one day. Using equation 15, the availability of the Ethereum system is 88.4% on that exemplary day.

Equation 15 can be used to estimate scalability by focusing on workload. However, it only incorporates confirmed and issued transactions. Transactions that were processed but later not finalized are ignored. Although this simplification is often sufficient, more detailed analyses may require distinguishing between different outcomes of transaction processing (e.g., verification or drop of transactions). For example, a high number of dropped transactions among those processed could bias results, as the number of potentially confirmable transactions may be underrepresented.

Confirmation Latency: *The timespan between the proposal of new blocks and their confirmation.*

When new blocks are added to a blockchain, validating nodes must process them and decide whether to include them permanently. In blockchain systems with probabilistic finality, where newly added blocks may still be removed due to forks, a block b is considered confirmed when it is added to the blockchain and additional blocks are appended to it. With

an increasing number of subsequently appended blocks, the likelihood to exclude b from the mainchain decreases, and b is assumed to be finalized. In Nakamoto consensus [22], for example, at least six additional blocks must be appended to a block until it is considered confirmed. In contrast, blockchain systems with immediate finality directly finalize blocks. For example, in systems using practical Byzantine fault tolerance (PBFT), at least one-third of the validating nodes must accept a block for it to be finalized [29]. As confirmations are not essential in such blockchain systems, the following metrics mainly apply to blockchain systems with probabilistic finality.

Equation 16 offers a simple estimate of the confirmation latency CL_1 of blockchain systems with immediate and probabilistic finality (e.g., [7], [15], [56], [74]):

$$CL_1 = BConfTime - BPropTime \tag{16}$$

$BConfTime$ represents the timestamp at which a newly added block is confirmed in a blockchain system. $BPropTime$ denotes the timestamp when a new block is issued to a blockchain system.

Security confirmations (i.e., blocks appended to a block in focus) are particularly important in blockchain systems with probabilistic finality, where the finalization of blocks can only be assumed. With a focus on security confirmations, equation 17 offers another approach to calculate the confirmation latency CL_2 of blockchain systems [10]:

$$CL_2 = SecConf \times BCI \tag{17}$$

$SecConf$ is the number of blocks that must be added for a block to be confirmed in the mainchain of a blockchain system. BCI is the time between the creation of consecutive blocks that are added to the mainchain of a blockchain system.

High confirmation latency often entails low throughput, thus low scalability of a blockchain system. For example, block confirmation latency in the Bitcoin system is high due to a majority of validating nodes required to reach consensus and a minimum of six subsequent blocks for confirmation [21]. In contrast, lower confirmation latency often entails higher scalability because blocks are processed and confirmed in a short time [36], [61]. Equation 16 can be adapted to blockchain systems with immediate finality by replacing $BConfTime$ with $BlockFinalizationTime$—the timestamp at which a block is finalized.

The following example illustrates how equation 16 can be used. From Etherscan [75], we obtained block confirmation timestamps for five blocks in the Ethereum system from block number $21,744,430$:

$$b_{ct,0} = 1,738,325,567, b_{ct,1} = 1,738,325,579,$$
$$b_{ct,2} = 1,738,325,591, b_{ct,3} = 1,738,325,603,$$
$$b_{ct,4} = 1,738,325,615$$

The corresponding block proposal timestamps are as follows:

$$b_{pt,0} = 1,738,325,207, b_{pt,1} = 1,738,325,207,$$
$$b_{pt,2} = 1,738,325,591, b_{pt,3} = 1,738,325,591,$$
$$b_{pt,4} = 1,738,325,591$$

We assume that the epoch times are in seconds throughout the manuscript. Using equation 16, the confirmation latencies of the five blocks are $CL_{1,0} = 360$, $CL_{1,1} = 372$, $CL_{1,2} = 0$, $CL_{1,3} = 12$, and $CL_{1,4} = 24$ seconds. This means consecutive blocks in the Ethereum system can have different confirmation latencies.

The following example illustrates how equation 17 can be used. Blockchain Explorer [76] displays the block confirmation timestamps for six blocks, starting from block number 879,319 in the Bitcoin system:

$$b_{ct,0} = 1,736,963,315, b_{ct,1} = 1,736,963,430,$$
$$b_{ct,2} = 1,736,964,423, b_{ct,3} = 1,736,965,070$$
$$b_{ct,4} = 1,736,965,163, b_{ct,5} = 1,736,965,455$$

The block creation intervals $b_{ci,0}, ..., b_{ci,4}$ in seconds of five blocks from block number 879,320 are as follows:

$$b_{ci,0} = 115, b_{ci,1} = 993, b_{ci,2} = 647, b_{ci,3} = 93, b_{ci,4} = 292$$

Since the number of security confirmations is 6 in the Bitcoin system [77], using equation 17, confirmation latencies showcase variations between consecutive blocks as follows:

$$CL_{2,0} = 690, CL_{2,1} = 5,958, CL_{2,2} = 3,882,$$
$$CL_{2,3} = 558, CL_{2,4} = 1,752$$

The higher the number of required security confirmations and the longer the block creation interval, the longer it takes to probabilistically finalize blocks. Thus, the larger CL_2, the lower the scalability of a blockchain system can be assumed.

Both metrics for estimating confirmation latency focus on the time it takes for a block to be confirmed in a blockchain system, but neglect the number of transactions included per block. Additional blockchain characteristics, such as block size, must be considered, in addition to confirmation latency, to quantify the scalability of blockchain systems. A large block size enables the inclusion of more transactions per block. Thus, large blocks that include many transactions can mitigate long confirmation latency [78], [79]. This is, however, not reflected in either metric for the confirmation latency construct.

Throughput: *The highest number of transactions a blockchain system can process in a specified timeframe.*

Transaction processing is at the core of blockchain systems. Transaction processing involves transaction propagation between validating nodes, validation of transaction data, batching valid transactions in blocks, and appending blocks to the mainchain. Depending on the type of finality of the consensus mechanisms used in a blockchain system, transactions included in the mainchain are either finalized (immediate finality) or need to be confirmed by subsequent blocks (probabilistic finality).

Equation 22 can be used to calculate throughput TPS_1 of blockchain systems (e.g., [7]–[10], [12], [59], [80], [81]):

$$TPS_1 = \frac{NumOfConfTr}{t_1 - t_0} \tag{22}$$

$NumOfConfTr$ denotes the number of transactions that are added to the blockchain and are assumed to be confirmed (e.g., they will not be excluded in an attack). t_1 denotes the system timestamp at the end of an observation. t_0 denotes the system timestamp at the beginning of an observation.

As an alternative to equation 22, equation 23 was proposed to calculate throughput TPS_2 of the Ethereum system [82]:

$$TPS_2 = \frac{min(N_B, MemPoolSize)}{BCI} \tag{23}$$

N_B denotes the number of confirmed transactions per block. $MemPoolSize$ denotes the maximum possible number of pending transactions buffered in a MemPool (usually 1,024) [82]. BCI denotes the average system time between the production of consecutive blocks included in a blockchain system. To calculate N_B, equation 24 can be used:

$$N_B = \frac{|G_{limit}|}{|G_{cost}|} \tag{24}$$

$|G_{limit}|$ denotes the block gas limit, which is the maximum amount of computational effort a user is willing to extend in confirming transactions. $|G_{cost}|$ denotes the block gas cost per transaction, which is usually 21,000 gas.

Throughput can be used to quantify the scalability of blockchain systems in terms of transactions per second. High throughput has been proposed as a key indicator of scalability of blockchain systems [2], [3], [52], [59], [81], [83]. In contrast, low throughput of a blockchain system indicates low scalability because fewer transactions can be confirmed in a blockchain system within a specified time.

Equation 22 can be used to measure the scalability of most blockchain systems. The five consecutive blocks of the Bitcoin system, starting from block number 879,320 [76] stored 16,457 transactions with a creation interval of 2,025 seconds. Using equation 22, the throughput of the Bitcoin system equals approximately 8 transactions per second.

The five consecutive blocks in the Ethereum system, starting from block number 2,1744,430 stored 947 transactions [75]. The timespan between the inclusion of the first and fifth blocks equals 48 seconds. Using equation 22, the throughput equals 19 transactions per second. At the time of writing, there are an estimated 160,000 average pending transactions per hour in the Ethereum system [84]; it would take approximately 2 hours to process them in addition to incoming transactions. This showcases that the Ethereum system faces a backlog of incoming transactions, highlighting scalability challenges despite a better throughput.

Equation 23 can be used to estimate the throughput of the Ethereum system. We obtained the transaction data of five blocks, starting from block number 21,744,430 from Etherscan [75] to quantify availability. The five blocks reveal the following block gas limits:

$$b_{gl,0} = 30,115,832, b_{gl,1} = 30,086,424, b_{gl,2} = 30,115,804,$$
$$b_{gl,3} = 30,145,212, b_{gl,4} = 30,115,775$$

Using equation 24 and assuming a block gas cost of 21,000 gas, the number of confirmed transactions for the five blocks is as follows:

$$N_{B,0} = 1,434, N_{B,1} = 1,433, N_{B,2} = 1,434,$$
$$N_{B,3} = 1,435, N_{B,4} = 1,434$$

The average number of transactions for the five blocks is $1,434$. Because the $MemPoolSize$ is the minimum value compared to N_B, using equation 23, the throughput of the Ethereum system equals 85 transactions per second.

Equations 22 and 23 apply only to throughput, neglecting scalability in terms of the number of validating nodes in blockchain systems. The number of validating nodes, however, influences communication complexity in consensus finding, thus the time until transactions are confirmed [85]. Moreover, the metrics only focus on confirmed transactions, neglecting overhead to process transactions that are not included in the mainchain. Neglecting such transactions may cause equations 22 to underestimate maximum throughput.

3) Security: To quantify the security of blockchain systems, we identified six metrics that operationalize the constructs: *availability*, *consistency*, *cost of attack*, *fault tolerance*, *reliability*, and *stale block rate*.

Availability: *The degree to which a blockchain system operates correctly at an arbitrary time.*

To remain operational at all times, blockchain systems typically strongly rely on replication and redundancy [21]. However, according to fundamental principles in distributed systems theory—such as the CAP theorem [48] and the PACELC model [49]—there exists an inherent trade-off between availability and consistency, especially in the presence of network partitions. Blockchain systems inherit this trade-off [15], [21], making it difficult to simultaneously maximize both availability and consistency. As a result, it cannot always be expected that a blockchain system will provide correct and up-to-date transaction data at any arbitrary time, especially in the presence of failures in the system.

Equation 27 can be used to calculate availability $ASec$ in the context of security of blockchain systems [10]:

$$ASec = \frac{MTBF}{MTBF + MTTR} \times 100\% \tag{27}$$

$MTBF$ (i.e., mean time between failures) denotes the average time a blockchain system operates correctly before failures. A failure may correspond to validating nodes failing to reach consensus or the throughput of the blockchain systems drops below a minimum threshold. The $MTBF$ encompasses both repair and restoration times. $MTTR$ (i.e., mean time to repair) denotes the average time required to diagnose, repair, and restore a blockchain system to full functionality after a failure. The original equation for the $ASec$ metric uses $MTTF$ (i.e., mean time to failures), which is commonly used for estimating the availability of non-repairable systems. To make the metric better suitable for blockchain systems (i.e.,

repairable systems), we replaced $MTTF$ with $MTBF$ after discussion with the authors of the metric [10].

To calculate the $MTBF$, equation 28 can be used:

$$MTBF = \frac{TotalOperationalTime}{NumberOfFailures} \tag{28}$$

$TotalOperationalTime$ denotes the total time that a blockchain system operates correctly. $NumberOfFailures$ denotes the total number of failures in a blockchain system (e.g., exceeding a specified timespan to reach consensus due to crashed validating nodes).

To calculate $MTTR$, equation 29 can be used:

$$MTTR = \frac{TotalRepairTime}{NumberOfRepairs} \tag{29}$$

$TotalRepairTime$ denotes the total time that a blockchain system took to recover from failures. The total number of repairs in a blockchain system is denoted by $NumberOfRepairs$.

In blockchain systems with low availability, validating nodes slowly synchronize, which can lead to inconsistencies. Such inconsistencies lead to multiple partitions in a blockchain system. Partitions in blockchain systems can be exploited by attackers to perform double spending (i.e., using the same asset twice or more into different partitions) [21], [28]. In blockchain systems with high availability, validating nodes' states transition to the same subsequent state quickly to maintain consistency. Quickly reaching consistency across all validating nodes increases availability and can help mitigate attacks that exploit inconsistencies, such as double-spending attacks.

Equation 27 can be used to quantify the availability of most blockchain systems, as illustrated in the following example. Assuming that a fictitious blockchain system has been subjected to two major failures, where validating nodes could not reach consensus due to Byzantine attacks. The first failure lasted for 10 minutes, the second for 20 minutes. The total failure time amounts to 30 minutes. Using equation 29, $MTTR$ (e.g., the average time to recover from failure) equals:

$$MTTR = \frac{10 + 20}{2} = 15$$

Suppose the blockchain system had been operating continuously for a year, assuming that one year approximately equals 365 days, for 24 hours per day and 60 minutes per hour, it would have been up for approximately $525,600$ minutes. Subtracting the total time during the two failures, the $TotalOperatingTime$ equals 525,570 minutes. Given that two failures occurred, equation 28, $MTBF$ equals:

$$\frac{525,570}{2} = 262,785$$

Using equation 27, availability of the blockchain system can be estimated as follows:

$$ASec = \left(\frac{262,785}{262,785 + 15}\right) \times 100 = 99.99\%$$

Equation 27 can be used to estimate the security of blockchain systems by computing the probability with which a blockchain system operates correctly. A major limitation of the availability metric in the context of blockchain technology is unclear guidance on what type of failure should be considered to estimate it. Multiple types of failure exist (e.g., failures in uptime, in preserving the integrity of a blockchain, in reaching consensus, or in maintaining performance higher than a specified threshold), presenting challenges to comparability for practitioners who use the metric to estimate security.

Consistency: *The degree to which all validating nodes are in a shared and agreed-upon state.*

Blockchain systems are envisioned to synchronize the state of validating nodes to achieve consistency. Depending on the consensus mechanism used, blockchain systems can reach immediate consistency or eventual consistency.

Equation 30 can be used to calculate the consistency $Const$ of blockchain systems with eventual consistency [10]:

$$Const = \frac{1}{|N_c|} \times \sum_{i=1}^{|N_c|} (BConfTime_i - BPropTime_i) \tag{30}$$

$|N_c|$ denotes the number of confirmed blocks. $BConfTime_i$ denotes the system time when a new block i is confirmed in a blockchain system. $BPropTime_i$ denotes the system time when the block i is issued to a blockchain system.

A short timespan to reach consistency in blockchain systems indicates that transactions are processed quickly. This facilitates validating nodes to offer a consistent view of the ledger and mitigate Byzantine behavior of nodes as in double spending [86]. Fast synchronization of validating nodes in blockchain systems with probabilistic finality makes it difficult for attackers to exploit forks (i.e., conflicting versions of a ledger), for example, in 51% attacks. In contrast, low consistency indicates the presence of forks across validating nodes in a blockchain system. Conflicting versions can facilitate double-spending [21].

The following is a simplified demonstration of how equation 30 can be used to estimate the consistency of the Ethereum system. Block confirmation and proposal timestamps [2] of five blocks starting from block number 21,744,430 [75]: (1,738,325,567; 1,738,325,207), (1,738,325,579; 1,738,325,207), (1,738,325,591; 1,738,325,591), (1,738,325,603; 1,738,325,591), and (1,738,325,615; 1,738,325,591). The confirmation latencies for the five blocks are 360, 372, 0, 12, and 24 seconds. To calculate consistency in a better-readable format, we converted the timestamps from Etherscan into epoch time using epoch converter [87]. In this example, the consistency in the Ethereum system equals approximately 153.6 seconds.

Equation 30 can be used to quantify the time until consistency is reached in blockchain systems. For blockchain systems with immediate finality, CL_1 needs to be adapted to finalization instead of confirmation. CL_2 does not apply because $SecConf \equiv 0$.

[2]Block confirmation time on the Etherscan is usually on the *timestamp* field, whereas the block proposal timestamp is on the *proposed on* field.

Cost of Attack: *The cost in fiat currency to gain control of a blockchain system through an attack.*

Blockchain systems are prone to different attacks (e.g., 51% attacks and selfish mining attacks) that help attackers (temporarily) gain control over the system. Such attacks have different attack vectors and exploit different characteristics of blockchain systems. In 51% attacks and selfish mining attacks, for example, attackers exploit the probabilistic finality of blockchain systems. Attacks to control blockchain systems are often expensive—especially in public-permissionless blockchain systems like Bitcoin. High cost of successful attacks can discourage Byzantine behavior by rendering successful attacks economically unprofitable or unfeasible.

Offering an indicator of the vulnerability of blockchain systems for specific attacks, equation 31 can be used to calculate the cost of an attack CoA on a blockchain system [12]:

$$CoA = t * c * \sum_{i=1}^{n} s_i \tag{31}$$

To calculate CoA, a threshold t must be chosen. t indicates the minimum amount of resources needed to compromise the blockchain system, such as 51% of the hashing power in the Bitcoin system. In consultation with the authors of the article [12], we added t to the equation to better explain the use of the metric. c refers to the cost of one unit of required resources in fiat currency, such as USD. n represents the number of validating nodes in a blockchain system. s_i is the amount of resources possessed by a validating node i that an attacker needs to control in order to dominate the blockchain system. Exemplary resources are hashing power or tokens for staking.

Equation 31 can be used to estimate the security of blockchain systems based on two key assumptions: the required resources (e.g., funds) must be available, and the attacker's gains must exceed the cost of the attack. High cost of attack makes it difficult for attackers to acquire sufficient resources due to economic constraints. High cost can make attacks unfeasible to perform, enhancing the security of the corresponding blockchain system. Low cost of attack implies that attackers can compromise the blockchain system with low financial investment. This can enable a broader range of users, wealthy and less wealthy ones, to perform attacks. Moreover, even little gains may exceed the cost of attack, which can motivate users to attack the blockchain system. Thus, a low cost of attack can indicate low security of a blockchain system.

Equation 31 applies to blockchain systems with PoW-based (e.g., the Bitcoin system) and proof-of-stake–based consensus mechanisms (e.g., the Ethereum system). In the Bitcoin system, for example, attackers must accumulate at least 51% of hashing power to gain control of the blockchain system. Given the total hashing power of the Bitcoin system of approximately 781.25 EH/s [88] and assuming that each validating node operates one ASIC miner that calculates 234 TH/s [3], the attacker needs to control at least 1,702,725 validating nodes to gain 398,437,650 TH/s in a 51% attack. In this simplified example, a unit of one *hash* costs about USD. 2.709×10^{-11} (excluding additional expenses such as energy cost). The cost of a 51% attack in this simplified example amounts at least to approximately USD 46MM, which is very expensive and thus unlikely.

In the Ethereum system, attackers can gain control over multiple validating nodes to dominate consensus finding. For each validating node, the attacker is assumed to pay 32 ETH [6], which amounts to USD 98,835.52 per validating node at the ETH token price of USD 3,138.18 [89]. At the time of writing, the Ethereum system comprised 1,762,190 validating nodes [90], of which the attacker must at least control one-third (i.e., 587,397 validating nodes) to succeed in a Sybil attack [12]. Using equation 31, the cost of attack amounts to 18,796,704 ETH (i.e., approximately USD 58,987,440,559). As only a few people own that much money, it is very unlikely that single attackers perform such a Sybil attack.

Equation 31 can be used to estimate security based on the cost of an attack to gain control of a blockchain system. However, the metric only offers a rough estimate of the cost with a focus on only one selected attack. To estimate the overall security of a blockchain system using equation 31, the minimum cost of any possible attack would need to be calculated and compared. The cheapest attack could be interpreted as the most likely one, thus the greatest vulnerability of a blockchain system. Another major shortcoming of this approach is the neglect of the severity of attacks. Moreover, for equation 31, it is assumed that the attacker buys resources that already exist in the blockchain system. Attackers that set up new validating nodes to compromise the blockchain system increase the overall hashing power. For such scenarios, t needs to be adapted to correspond to 51% of the resulting overall hashing power of the system after the attacker nodes join the system.

Fault Tolerance: *The degree to which a blockchain system operates correctly despite accidental or Byzantine faults.*

Blockchain systems are subject to faults, such as omission, crashes, or Byzantine behavior of nodes [24], [25]. To ensure successful synchronization despite faults, different consensus mechanisms are robust against (i.e., tolerate) different types of faults. For example, Raft handles compensates faults [1]; PBFT tolerates Byzantine faults [29].

Equation 32 can be used to calculate the fault tolerance FT with a focus on performance change [7], [8]:

$$FT = \{\Delta ThroughputDiff, \Delta ConfLatDiff\} \tag{32}$$

$ThroughputDiff$ denotes the change in throughput due to faulty validating nodes. $ConfLatDiff$ signifies the change in confirmation latency during the failure of validating nodes in a blockchain system.

$ThroughputDiff$ is calculated using equation 33:

$$ThroughputDiff = |Throughput_N - Throughput_F| \tag{33}$$

$Throughput_N$ denotes the throughput when a blockchain system operates normally. $Throughput_F$ denotes the throughput when a fault occurs in the blockchain system.

[3]We assume that each validating node operates one Bitmain Antminer S21 Pro 234TH, each at the cost of USD 6,339.

To calculate $ConfLatDiff$ equation 34 can be used:

$$ConfLatDiff = |ConfLat_N - ConfLat_F| \tag{34}$$

$ConfLat_N$ denotes the confirmation latency when a blockchain system operates normally.
$ConfLat_F$ denotes the confirmation latency when a blockchain system experiences faults. Input variables in equations 33 and equations 34 can be obtained using equations 22, 23, 16, and 17.

Equation 32 can be used to quantify the security of blockchain systems in terms of fault tolerance [7], [8], [10], [57]. Blockchain systems with high fault tolerance demonstrate stable throughput and latency even when some validating nodes are faulty [7]–[9]. For example, with 1,702,725 validating nodes in the Bitcoin system (see illustration in equation 31), up to 51% of faulty validating nodes can be tolerated with a good throughput and confirmation latency. Conversely, the performance of blockchain systems with low fault tolerance degrades in the case of faults.

Equation 32 can be used to quantify the security of most blockchain systems with respect to fault tolerance. The following example illustrates the use of equation 32 to estimate the fault tolerance of the fictitious blockchain system. Suppose a blockchain system has a throughput of 10 transactions per second and a confirmation latency of 5 seconds in regular operation. The throughput of this blockchain system must not drop below 8 transactions per second. Suddenly, a large portion of validating nodes crashes. When the crashed validating nodes start recovering, the throughput of the blockchain system decreases to 8 transactions per second with a confirmation latency of 10 seconds. Using equation 32, the fault tolerance of the fictitious blockchain can be estimated as follows:

$$ThroughputDiff = 10 - 8 = 2$$

$$ConfLatDiff = |5 - 10| = 5$$

$$FT = 2, 5$$

The fault tolerance of the fictitious blockchain system equals FT 2 transactions per second and 5 seconds. This means that due to faults, the throughput of the blockchain system decreased by 2 transactions per second and increased by 5 seconds for confirmation latency.

Equation 32 mainly reflects changes in blockchain system performance caused by crash faults [7], [8], [10], [57]. Byzantine behavior of validating nodes (e.g., double spending; [26], [86]) is hardly reflected in the metric. Despite stable throughput and confirmation latency, attacks such as double-spending and selfish mining can be successfully performed without influencing the results of equation 32.

Moreover, such performance changes reflected in fault tolerance are hardly useful in benchmarks of a single blockchain system to identify its Pareto-optimal configuration: fault tolerance is, from a theoretical perspective, set with a maximum value (e.g., one-third of validating nodes). This maximum may diverge from the actual system behavior, which surely is important to be evaluated. However, to measure trade-offs between fault tolerance and constructs of DoD and scalability, different consensus mechanisms with different theoretical fault tolerance should be used.

Reliability: *The continuity of a blockchain system to offer correct service.*

Blockchain systems employ validating nodes to operate replicas that often store critical data, such as financial transactions in decentralized exchanges [91]. Especially when used as a critical digital infrastructure (e.g., in finance [92]), blockchain systems must continuously offer the correct service to meet reliability requirements.

Equation 35 can be used to calculate the reliability $R(t)$ of blockchain systems [10]:

$$R(t) = e\left(-\frac{t}{MTBF}\right) \tag{35}$$

$R(t)$ denotes the probability that a blockchain system remains functional at time t. t denotes the duration of quantifying the reliability of a blockchain system. $MTBF$ denotes the average time until failures occur (e.g., system performance drops below a specified threshold). Equation 28 defines how to calculate the $MTBF$. e denotes Euler's number (which is approximately 2.718), representing continuous decay. To make the metric better suitable for repairable systems such as blockchain systems, we consulted the authors [10] and agreed to update $MTTF$ by $MTBF$.

In the context of security, the reliability construct is used to express how long a blockchain system operates correctly without failures. Blockchain systems with high reliability operate correctly without failure for a long time, thus increased security. Low reliability indicates frequent failures, thus rather low security.

The following example illustrates the use of equation 35. Imagine two failures in one year in a blockchain system, totaling 30 minutes. The $TotalOperatingTime$ equals 525,570 minutes for a year with 365 days (525,600 minutes). Due to the two failures, using equation 28, $MTBF$ equals 262,785 minutes. Using equation 35, the reliability of the blockchain system for a year $R(525,600)$ is calculated as follows:

$$R(t) = e\left(-\frac{525,600}{262,785}\right) = e^{-2.14} = 11.77\%$$

The fictitious blockchain system has a probability of approximately 11.77% to operate unreliably within one year. The blockchain system is approximately 88.23% reliable.

Equation 35 can be used to estimate the reliability of blockchain systems based on failures. Such estimates can be used to forecast how a blockchain system can function correctly based on a history of failures. However, the metric definition leaves practitioners to decide on what type of failure to focus on when estimating the security of the blockchain system. This could entail issues in terms of the comparability of results if not clearly articulated.

Stale Block Rate: *The number of blocks that have been propagated in a blockchain system but not finalized in the mainchain in a specified timespan.*

In blockchain systems with probabilistic finality, not all blocks proposed by validating nodes are included in the mainchain. In blockchain systems with probabilistic finality, if multiple nodes propose blocks simultaneously, only the fastest-propagating block is accepted, while the others become stale blocks [26]. The possibility of stale blocks often hints at the possibility of network partitions in blockchain systems. Such partitions facilitate attacks, such as double spending and selfish mining [26], [28]. To account for attacks facilitated by stale blocks, estimating the stale block rate is important to know [26].

Equation 36 can be used to calculate the stale block rate SBR of blockchain systems [10], [57], [93]:

$$SBR = \frac{NumberOfStaleBlocks}{NumberOfConfirmedBlocks} \tag{36}$$

$NumberOfStaleBlocks$ denotes the number of valid blocks proposed but not included in the mainchain. The number of blocks stored in a blockchain system is denoted by $NumberOfConfirmedBlocks$.

The possibility of stale blocks can facilitate attacks, where adversaries use a fork to overrule the mainchain [26], [27]. Inconsistencies between states of validating nodes facilitate successful attacks such as double spending [26], [86] and selfish mining [27], [28] The higher the fraction between the number of stale blocks and the number of blocks confirmed around the current block height, the more vulnerable a blockchain system is to such attacks [8], [21], [57], thus the less secure. If the fraction of stale blocks is low, the blockchain system is assumed to be less vulnerable to such attacks.

Equation 36 applies to blockchain systems with eventual consistency, such as the Bitcoin system. Imagine a blockchain system that has 879,320 blocks that are confirmed and stored in the mainchain. Due to propagation delays (e.g., due to network congestion) and concurrent block proposals at the same block height, 2 stale blocks occurred. The stale block rate of the imaginary blockchain system equals approximately 2.27×10^{-6} according to equation 36. Suppose the blockchain system has a very low stale block; it is unlikely for attackers to compromise the blockchain system due to the longer creation interval.

Equation 36 can be used to estimate the stale block rate of blockchain systems to infer the security of blockchain systems. The metric, however, only offers a simplified estimate of the vulnerability of blockchain systems with probabilistic finality. The metric neglects blockchain system characteristics related to the propagation delays that facilitate stale blocks (e.g., block creation interval and block size) [26]. Moreover, the severity of exploiting stale blocks in attacks is neglected. Moreover, while research offers estimates on how many stale blocks in the past may lead to vulnerabilities [94], [95], it is up to the description of the analyst to decide how many blocks in the past are considered in the analysis.

B. Overview of Selected Analysis Approaches for Investigating the Blockchain Trilemma Subconcepts

With a focus on the Bitcoin and Ethereum systems, a few analysis approaches involve at least two blockchain trilemma subconcepts (e.g., [2], [12]). These analysis approaches include Bitnodes, Etherscan, Google BitQuery, and SimBlock. Table III presents an overview of selected analysis approaches that are used to quantify at least two blockchain trilemma subconcepts. Supplementary material B offers an overview of all analysis approaches considered in this work.

Analysis approaches for investigating multiple blockchain trilemma subconcepts mostly use throughput (equation 22) as a metric for scalability; hashing power distribution (equation 9) and wealth distribution (equation 12) as metrics for the DoD. For security, different metrics are used, such as CoA (equation 31), fault tolerance (equation 32), and stale block rate (equation 36).

V. Discussion

The vast number of constructs and their operationalization make selecting the most suitable ones for identifying Pareto-optimal blockchain system configurations under consideration of the blockchain trilemma a challenging task. We conducted a systematic literature review to assess the suitability of commonly used constructs in analysis approaches. In section V-A, we present our key findings on constructs for evaluating DoD, scalability, and security in blockchain systems. Subsection V-B explains the main contributions of this work to research and practice. We then elucidate this study's limitations in section V-C, followed by a discussion of promising future research directions in section V-D.

A. Principal Findings

The literature review shows that constructs used to assess the scalability of blockchain systems are generally straightforward, with clear metrics and evaluation methods. Five key constructs emerged, with *throughput* and *confirmation latency* used most frequently. Multiple metrics are available to operationalize these constructs, and their suitability depends on the blockchain system design—for instance, equation 17 is appropriate for systems with probabilistic finality.

In contrast, assessing security is more complex, involving numerous interdependent constructs and metrics. Because each captures a critical aspect, combining them into a unified measure remains a significant challenge. Until such a combination is possible, estimating blockchain system security will likely require relying on multiple distinct constructs.

We identified six constructs for security, with *fault tolerance* and *stale block rate* being the constructs most often proposed in the analyzed literature. Despite the importance of availability and reliability constructs for analyses of general software systems, the examined literature does not emphasize those constructs in analyses of blockchain systems. Being essential constructs of the security subconcept, this suggests further investigation of their usefulness in the context of the blockchain trilemma.

Unlike scalability and security, measuring DoD poses a different challenge that stems from the lack of a clear, unified concept. The main concepts influencing DoD—the degree of autonomy and equity of each element in a system—are hardly quantifiable as they are very broad and context-dependent. Moreover, DoD involves social and technical aspects

TABLE III
SELECTION OF ANALYSIS APPROACHES FOR INVESTIGATING THE BLOCKCHAIN TRILEMMA SUBCONCEPTS AND EXAMPLES OF BLOCKCHAIN SYSTEMS ANALYZED USING THE APPROACHES.

	Constructs and Metrics Used to Analyze the Blockchain Trilemma Subconcepts			
Analysis Approach	**DoD**	**Scalability**	**Security**	**Analyzed Blockchain Systems**
BBSF [7]		Confirmation latency (eq. 16), Throughput (eq. 22)	Fault tolerance (eq. 32)	Ethereum, Quorum
Blockbench [8]		Confirmation latency (eq. 16), Throughput (eq. 22)	Fault tolerance (eq. 32)	Ethereum, Hyperledger Fabric, Parity
BlockSim [96]		Confirmation latency (eq. 16), Throughput (eq. 22)	Stale block rate (eq. 36)	Bitcoin, Ethereum
DIABLO [9]		Confirmation latency (eq. 16), Throughput (eq. 22), Availability (eq. 15)	Fault tolerance (eq. 32)	Algorand, Ethereum, Hyperledger Fabric, Quorum, Redbelly, Solana, Zcash
Fu et al. [59]	Token concentration (eq. 10), Hashing power distribution (eq. 9), Wealth distribution (eq. 12)	Confirmation latency (eq. 16), Throughput (eq. 22)		Algorand, Ethereum
SimBlock [2]	Token concentration (eq. 10), Hashing power distribution (eq. 9), Wealth distribution (eq. 12)	Throughput (eq. 22)	Fault tolerance (eq. 32), Stale block rate (eq. 36)	Bitcoin
Quattrocchi et al. [12]	Hashing power distribution (eq. 9), Wealth distribution (eq. 12)	Throughput (eq. 22)	Cost of attack (eq. 31)	Bitcoin, Cardano, Ethereum, Polygon, Solana
Thakkar et al. [97]		Confirmation latency (eq. 16), Throughput (eq. 22)	Fault tolerance (eq. 32)	Hyperledger Fabric
Gräbe, et al. [10]		Confirmation latency (eq. 17), Throughput (eq. 22)	Availability (eq. 27), Consistency (eq. 30), Fault tolerance (eq. 32), Reliability (eq. 35)	Ethereum, Hyperledger Indy, Tezos

eq.: equation

[21], [98]. Many operationalized constructs collated in this work are proposed to approximate DoD but often focus on either economic aspects, such as *token concentration* and *wealth distribution*, or technical aspects, such as participation possibilities of validating nodes in consensus finding reflected in *block proposal randomness*. Although capturing important aspects of DoD, the identified constructs are mainly treated in separation. We could not identify a single construct that fully captures DoD from a sociotechnical perspective, highlighting the need for theoretical foundations that inform more exhaustive measurement approaches for DoD.

While the presented constructs can be linked to the blockchain trilemma subconcepts, not all constructs show clear interrelationships. For example, geographical diversity does not seem directly interrelated with *cost of attack*. Practitioners need to carefully select constructs that have direct interrelationships with each other, particularly negative correlations, to capture trade-offs between the blockchain trilemma subconcepts. For example, the trade-off between fault tolerance (equation 32) for security and throughput (equation 22) from scalability can be of interest by examining how increasing or decreasing either of the constructs can influence hashing power distribution (equation 9) and vice versa. An inappropriate selection of constructs can easily lead to the misperception that a Pareto-optimal configuration of the blockchain system has been found. For reasonable combinations of operationalized constructs, practitioners could choose constructs focusing on similar aspects of blockchain systems.

B. Contributions

The blockchain trilemma is complex to analyze due to various constructs that capture important but only selected aspects of the blockchain trilemma subconcepts. Our main ambition with this work is to offer a theoretical foundation for more thorough analyses of the blockchain trilemma that help practitioners find Pareto-optimal configurations of blockchain systems that meet common non-functional requirements. Specifically, this work has three key contributions. First, the work offers an overview of common constructs and their operationalization through metrics for measuring the blockchain trilemma subconcepts (i.e., DoD, scalability, and security). By examining the meaning, applicability to blockchain systems with different designs, and limitations of these constructs and associated metrics, this work helps to better understand what operationalizations of constructs are available and offers guidance for selecting suitable ones. This facilitates better-informed decisions about the appropriate constructs. For example, practitioners can better understand to what extent they capture the DoD and security of blockchain systems and identify aspects that may have been overlooked in their analyses. This is useful for more sophisticated analyses of blockchain system behaviors through the lens of the blockchain trilemma and lays a cornerstone for better comparability of such analyses.

Second, by explaining the metrics, including their input variables, this work offers a foundation for planning benchmarks. The defined input variables (e.g., number of validating nodes, number of transactions issued to a blockchain system) guide data collection efforts by helping practitioners identify which aspects of blockchain systems need to be monitored. This also helps identify potentially relevant manipulations of blockchain system configurations to investigate their influence on the blockchain trilemma.

Third, by comparing various analysis approaches based on their used constructs and associated metrics, this work offers a foundation for selecting suitable approaches for studying the blockchain trilemma. Practitioners can better understand the operationalized constructs and their limitations. Combined

with the overview of operationalized metrics, practitioners can customize existing analysis approaches for better analyses.

C. Limitations

Although this work offers guidance on navigating the complexity of the blockchain trilemma, the results presented are subject to several limitations. We used a generic search string to capture a broad range of relevant literature proposing constructs to analyze the blockchain trilemma. Despite reviewing a large number of publications, we may have excluded constructs introduced in sources that are not peer-reviewed. Additional constructs may exist that could further support investigations of the blockchain trilemma. Furthermore, the applicability of the identified constructs to blockchain systems is illustrative rather than exhaustive. In assessing suitability, we focused on the Bitcoin and Ethereum systems because they are commonly analyzed blockchain systems in investigating the blockchain trilemma. The constructs may be useful for analyzing the blockchain trilemma subconcepts in additional blockchain systems.

This study focuses on blockchain systems that use the concept of replicated state machines and broadcast-based consensus mechanisms (see Figure 1). Blockchain systems that rely on specialized hardware, such as trusted execution environments, or additional software components, such as payment channel networks, were excluded from this study. Although the identified constructs may still apply to these types of systems, they fall outside the scope of this study. Furthermore, this study is limited to blockchain systems. Other distributed ledger technologies, such as transaction-based directed acyclic graphs [21], fall outside its scope.

In the comparison of analysis approaches, we present exemplary tuples of constructs that are used to operationalize the blockchain trilemma based on existing literature. However, we could not validate what tuples reasonably capture the blockchain trilemma based on clear interrelationships between the constructs that uncover trade-offs between the blockchain trilemma subconcepts. Moreover, different blockchain systems may require additional tuples for analysis.

D. Future Work

Commonly used constructs capture only selected aspects of the blockchain trilemma subconcepts, such as geographical diversity and hashing power distribution of DoD and availability and fault tolerance of security. To capture the blockchain trilemma subconcepts more exhaustively, approaches to combine different constructs for the same blockchain trilemma subconcepts are needed. For example, security analyses of blockchain systems could produce combined indices that reflect multiple constructs related to security, such as availability, consistency, and fault tolerance. Moreover, additional constructs may be needed to exhaustively capture the blockchain trilemma subconcepts. For example, the constructs presented for DoD do not account for social influences stemming from individuals and organizations operating the validating nodes. Such influences might include social relationships between individuals and organizations that facilitate collusion, leading to the emergence of covert power structures that centralize information within blockchain systems.

Future research should aim to better understand which construct tuples are most useful for investigating the blockchain trilemma across blockchain systems with different consensus mechanisms and broadcasting protocols. To assess the validity of construct tuples, benchmarks should be conducted in which the interrelationships between constructs are quantified. To not only observe interrelationships but also thoroughly explain them, such benchmarks should be informed by explicit theorems (e.g., the CAP theorem [47]) and hypotheses derived from assumed interrelationships between the blockchain trilemma subconcepts. Insights generated from such benchmarks will not only benefit a better understanding of interrelationships between blockchain system characteristics but also help foster the theoretical foundation of the blockchain trilemma.

Based on the results presented in this work, the blockchain trilemma may also apply to distributed systems with different architectures that exhibit DoD, scalability, and security. This expansion could help better understand the extent to which the blockchain trilemma applies not only to blockchain systems but also to other distributed systems that use consensus mechanisms. Such investigations could pave the way for a new theorem for distributed systems that complements the CAP theorem, enriching the theoretical foundations for understanding consensus mechanisms.

VI. Conclusion

Existing constructs and their metrics helped generate valuable insights into the blockchain trilemma subconcepts from different perspectives (e.g., socioeconomic and distributed systems perspectives). However, lack of clear guidance on the applicability of constructs and their metrics to blockchain systems and how they relate to the blockchain trilemma subconcepts limits practitioners to finding Pareto-optimal configurations of blockchain systems that meet common non-functional requirements.

This study presents 14 constructs operationalized through 16 metrics to quantify DoD, scalability, and security and explains how they apply to different blockchain systems. The overview of metrics can help practitioners select suitable metrics for investigating the blockchain trilemma. Moreover, this work offers a theoretical foundation for developing analysis approaches that support better investigations of how blockchain system configurations influence the blockchain trilemma subconcepts. We hope that this work helps develop tooling that supports finding Pareto-optimal configurations of blockchain systems that meet common non-functional requirements.

References

[1] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[2] T. Nakai, A. Sakurai, S. Hironaka, and K. Shudo, "A formulation of the trilemma in proof of work blockchain," *IEEE Access*, vol. 12, pp. 80 559–80 578, 2024.

[3] J. Werth, M. H. Berenjestanaki, H. R. Barzegar, N. El Ioini, and C. Pahl, "A review of blockchain platforms based on the scalability, security and decentralization trilemma." *International Conference on Enterprise Information Systems*, pp. 146–155, 2023.

[4] G. D. Monte, D. Pennino, and M. Pizzonia, "Scaling blockchains without giving up decentralization and security: A solution to the blockchain scalability trilemma," in *Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, New York, NY, USA, 2020, pp. 71–76.

[5] V. Buterin, *Proof of stake: The making of Ethereum and the philosophy of blockchains*. Seven Stories Press, 2022.

[6] E. Kapengut and B. Mizrach, "An event study of the ethereum transition to proof-of-stake," *Commodities*, vol. 2, no. 2, pp. 96–110, 2023.

[7] K. Ren, J. F. Van Buskirk, Z. Y. Ang, S. Hou, N. R. Cable, M. Monares, H. F. Korth, and D. Loghin, "Bbsf: blockchain benchmarking standardized framework," in *Proceedings of the 1st Workshop on Verifiable Database Systems*, New York, NY, USA, 2023, pp. 10–18.

[8] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*, New York, NY, USA, 2017, pp. 1085–1100.

[9] V. Gramoli, R. Guerraoui, A. Lebedev, C. Natoli, and G. Voron, "Diablo: A benchmark suite for blockchains," in *Proceedings of the Eighteenth European Conference on Computer Systems*, New York, NY, USA, 2023, pp. 540–556.

[10] F. Gräbe, N. Kannengießer, S. Lins, and A. Sunyaev, "Do not be fooled: Toward a holistic comparison of distributed ledger technology designs," in *Proceedings of the 53rd Hawaii international conference on system sciences*, 2020, pp. 6297–6306.

[11] M. Juodis, E. Filatovas, and R. Paulavičius, "Overview and empirical analysis of wealth decentralization in blockchain networks," *ICT Express*, vol. 10, no. 2, pp. 380–386, 2024.

[12] G. Quattrocchi, F. Scaramuzza, and D. A. Tamburri, "The blockchain trilemma: an evaluation framework," *IEEE Software*, vol. 41, no. 6, pp. 101–110, 2024.

[13] Y. Liu, M. H. Cheung, and J. Huang, "Incentive mechanism for throughput enhancement in blockchain-based energy trading system," in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications*, Kyoto, Japan, 2023, pp. 153–160.

[14] K. Wu, B. Peng, H. Xie, and Z. Huang, "An information entropy method to quantify the degrees of decentralization for blockchain systems," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication*, Beijing, China, 2019, pp. 1–6.

[15] H. Wang, H. Li, A. Smahi, M. Xiao, and S.-Y. R. Li, "Gbt-chain: A system framework for solving the general trilemma in permissioned blockchains," *Distributed Ledger Technologies: Research and Practice*, vol. 3, no. 2, pp. 1–15, 2024.

[16] H. Li and H. Wang, *Principles and Applications of Blockchain Systems: How to Overcome the CAP Trilemma in Consortium Blockchain*. John Wiley & Sons, 2025.

[17] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future," *MIS Quarterly*, vol. 26, no. 2, p. xiii–xxiii, 2002. [Online]. Available: http://www.jstor.org/stable/4132319

[18] L. Vila-Henninger, C. Dupuy, V. Van Ingelgom, M. Caprioli, F. Teuber, D. Pennetreau, M. Bussi, and C. Le Gall, "Abductive coding: Theory building and qualitative (re) analysis," *Sociological Methods & Research*, vol. 53, no. 2, pp. 968–1001, 2024.

[19] A. Dubois and L.-E. Gadde, "Systematic combining: an abductive approach to case research," *Journal of business research*, vol. 55, no. 7, pp. 553–560, 2002.

[20] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[21] N. Kannengießer, S. Lins, T. Dehling, and A. Sunyaev, "Trade-offs between distributed ledger technology characteristics," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–37, 2020.

[22] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," jan 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[23] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *Ieee Access*, vol. 7, pp. 22 328–22 370, 2019.

[24] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, no. 3, pp. 382–401, 1982.

[25] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[26] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC conference on Computer and Communications Security*, Vienna, Austria, 2016, pp. 3–16.

[27] I. Eyal and E. G. Sirer, "Majority is not enough: bitcoin mining is vulnerable," *Commun. ACM*, vol. 61, no. 7, p. 95–102, Jun. 2018. [Online]. Available: https://doi.org/10.1145/3212998

[28] Y. Sproll, R. Heinrich, L. B. Q. Le, and N. Kannengießer, "Sm-sim: A simulator for selfish-mining attacks in blockchain systems," in *2025 IEEE International Conference on Blockchain and Cryptocurrency*, Pisa, Italy, 2025.

[29] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, BOSTON, MA, USA, 1999, pp. 173–186.

[30] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

[31] M. Leinweber, N. Kannengießer, H. Hartenstein, and A. Sunyaev, *Leveraging Distributed Ledger Technology for Decentralized Mobility-as-a-Service Ticket Systems*. Wiesbaden: Springer Fachmedien Wiesbaden, 2023, p. 547–567.

[32] S. S. Stevens, "On the theory of scales of measurement," *Science*, vol. 103, no. 2684, pp. 677–680, 1946.

[33] M. Touloupou, M. Themistocleous, E. Iosif, and K. Christodoulou, "A systematic literature review towards a blockchain benchmarking framework," *IEEE Access*, pp. 70 630–70 644, 2022.

[34] L. Zhang, X. Ma, and Y. Liu, "Sok: blockchain decentralization," aug 2023. [Online]. Available: https://arxiv.org/abs/2205.04256

[35] Q. Lin, C. Li, X. Zhao, and X. Chen, "Measuring decentralization in bitcoin and ethereum using multiple metrics and granularities," in *2021 IEEE 37th International Conference on Data Engineering Workshops*, Chania, Greece, 2021, pp. 80–87.

[36] S. Mssassi and A. Abou El Kalam, "The blockchain trilemma: A formal proof of the inherent trade-offs among decentralization, security, and scalability," *Applied Sciences*, vol. 15, no. 1, p. 19, 2024.

[37] J. Lee, B. Lee, J. Jung, H. Shim, and H. Kim, "Dq: Two approaches to measure the degree of decentralization of blockchain," *ICT Express*, vol. 7, no. 3, pp. 278–282, 2021.

[38] Y. Jia, C. Xu, Z. Wu, Z. Feng, Y. Chen, and S. Yang, "Measuring decentralization in emerging public blockchains," in *2022 International Wireless Communications and Mobile Computing*, Dubrovnik, Croatia, 2022, pp. 137–141.

[39] A. Ahmad, M. Saad, J. Kim, D. Nyang, and D. Mohaisen, "Performance evaluation of consensus protocols in blockchain-based audit systems," in *2021 International Conference on Information Networking*, Jeju Island, Korea, 2021, pp. 654–656.

[40] F. Gräbe, N. Kannengießer, S. Lins, and A. Sunyaev, "Do not be fooled: Towards a holistic comparison of distributed ledger technology designs," in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, Maui, Hawaii, USA, 2020, p. 6297–6306.

[41] A. Sunyaev, N. Kannengießer, R. Beck, H. Treiblmaier, M. Lacity, J. Kranz, G. Fridgen, U. Spankowski, and A. Luckow, "Token economy," *Business & Information Systems Engineering*, vol. 63, p. 457–478, 2021. [Online]. Available: http://link.springer.com/10.1007/s12599-021-00684-1

[42] G. Voron and V. Gramoli, "Planetary scale byzantine consensus," in *Proceedings of the 5th workshop on Advanced tools, Programming Languages, and PLatforms for Implementing and Evaluating Algorithms for Distributed Systems*, New York, NY, USA, 2023, pp. 1–6.

[43] G. Wang, Q. Wang, and S. Chen, "Exploring blockchains interoperability: A systematic survey," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–38, 2023.

[44] G. Danezis, L. Kokoris-Kogias, A. Sonnino, and A. Spiegelman, "Narwhal and tusk: a dag-based mempool and efficient bft consensus," in *Proceedings of the Seventeenth European Conference on Computer Systems*, New York, NY, United States, 2022, pp. 34–50.

[45] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, New York, NY, USA, 2019, pp. 347–356.

[46] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 USENIX Annual Technical Conference*, Philadelphia, PA, 2014, pp. 305–319.

[47] E. Brewer, "Cap twelve years later: How the" rules" have changed," *Computer*, vol. 45, no. 2, pp. 23–29, 2012.

[48] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *Acm Sigact News*, vol. 33, no. 2, pp. 51–59, 2002.

[49] D. Abadi, "Consistency tradeoffs in modern distributed database system design: Cap is only part of the story," *Computer*, vol. 45, no. 2, pp. 37–42, 2012.

[50] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, ser. USENIX ATC'14, Philadelphia, PA, USA, 2014, p. 305–320.

[51] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, New York, NY, USA, 2018, pp. 1–15.

[52] A. I. Sanka and R. C. Cheung, "A systematic review of blockchain scalability: Issues, solutions, analysis and future research," *Journal of Network and Computer Applications*, vol. 195, p. 103232, 2021.

[53] J. Xu, C. Wang, and X. Jia, "A survey of blockchain consensus protocols," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–35, 2023.

[54] S. Leonardos, D. Reijsbergen, and G. Piliouras, "Presto: A systematic framework for blockchain consensus protocols," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1028–1044, 2020.

[55] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A survey of iot applications in blockchain systems: Architecture, consensus, and traffic modeling," *ACM Computing Surveys*, vol. 53, no. 1, pp. 1–32, 2020.

[56] B. Nasrulin, M. De Vos, G. Ishmaev, and J. Pouwelse, "Gromit: Benchmarking the performance and scalability of blockchain systems," in *2022 IEEE International Conference on Decentralized Applications and Infrastructures*, Newark, CA, USA, 2022, pp. 56–63.

[57] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE transactions on knowledge and data engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[58] N. R. Pradhan, A. P. Singh, N. Kumar, M. M. Hassan, and D. S. Roy, "A flexible permission ascription (fpa)-based blockchain framework for peer-to-peer energy trading with performance evaluation," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2465–2475, 2021.

[59] Y. Fu, M. Jing, J. Zhou, P. Wu, Y. Wang, L. Zhang, and C. Hu, "Quantifying the blockchain trilemma: A comparative analysis of algorand, ethereum 2.0, and beyond," in *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*, Hong Kong, China, 2024, pp. 97–104.

[60] C. Wang and X. Chu, "Performance characterization and bottleneck analysis of hyperledger fabric," in *2020 IEEE 40th International Conference on Distributed Computing Systems*, Singapore, Singapore, 2020, pp. 1281–1286.

[61] V. Gramoli, *Blockchain scalability and its foundations in distributed systems*. Springer, 2022.

[62] S. P. Gochhayat, S. Shetty, R. Mukkamala, P. Foytik, G. A. Kamhoua, and L. Njilla, "Measuring decentrality in blockchain based systems," *IEEE Access*, vol. 8, pp. 178 372–178 390, 2020.

[63] M. C. K. Khalilov and A. Levi, "A survey on anonymity and privacy in bitcoin-like digital cash systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2543–2585, 2018.

[64] C. S. Leberknight, M. Chiang, H. V. Poor, and F. Wong, "A taxonomy of internet censorship and anti-censorship," in *Fifth International Conference on Fun with Algorithms*, Iscia, Italy, 2010, pp. 52–64.

[65] E. Blockchain, "Hashrate distribution," feb 2025. [Online]. Available: https://web.archive.org/web/20250220081409/https://www.blockchain.com/explorer/charts/pools

[66] ioBanker and Abit, "Bitshares whitepaper," oct 2023. [Online]. Available: https://github.com/bitshares/whitepaper

[67] K. Jae and B. Ethan, "Cosmos," jan 2019. [Online]. Available: https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md

[68] V. Nocke and M. D. Whinston, "Concentration thresholds for horizontal mergers," *American Economic Review*, vol. 112, no. 6, pp. 1915–1948, 2022.

[69] D. o. J. U.S., "Herfindahl-hirschman index," jan 2024. [Online]. Available: https://web.archive.org/web/20250407100816/https://www.justice.gov/atr/herfindahl-hirschman-index

[70] D. Cracau and J. E. D. Lima, "On the normalized herfindahl-hirschman index: a technical note," *International Journal on Food System Dynamics*, vol. 7, no. 4, pp. 382–386, 2016.

[71] F. Solana, "A guide to stake-weighted quality of service on solana," mar 2024. [Online]. Available: https://solana.com/de/developers/guides/advanced/stake-weighted-qos

[72] S. Motepalli and H.-A. Jacobsen, "How does stake distribution influence consensus? analyzing blockchain decentralization," in *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Dublin, Ireland, 2024, pp. 343–352.

[73] I. YCharts, "Ethereum transactions per day," feb 2025. [Online]. Available: https://web.archive.org/web/20250213095640/https://ycharts.com/indicators/ethereum_transactions_per_day

[74] F. C. Geyer, H.-A. Jacobsen, R. Mayer, and P. Mandl, "An end-to-end performance comparison of seven permissioned blockchain systems," in *Proceedings of the 24th International Middleware Conference*, New York, NY, USA, 2023, pp. 71–84.

[75] Etherscan, "The ethereum blockchain explorer," jan 2025. [Online]. Available: https://web.archive.org/web/20250131122615/https://etherscan.io/blocks?ps=100

[76] E. Blockchain, "Latest btc blocks," jan 2025. [Online]. Available: https://web.archive.org/web/20250115045625/https://www.blockchain.com/explorer/blocks/btc?page=1

[77] Cryptomus, "How many confirmations are needed for transaction," nov 2024. [Online]. Available: https://web.archive.org/web/20250207095012/https://cryptomus.com/blog/how-many-confirmations-are-needed-for-transaction

[78] S. Ahmadjee, C. Mera-Gómez, R. Bahsoon, and R. Kazman, "A study on blockchain architecture design decisions and their security attacks and threats," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 2, pp. 1–45, 2022.

[79] J. Göbel and A. E. Krzesinski, "Increased block size and bitcoin blockchain dynamics," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, Melbourne, Australia, 2017, pp. 1–6.

[80] T. Q. Ban, B. N. Anh, N. T. Son, and T. Van Dinh, "Survey of hyperledger blockchain frameworks: case study in fpt university's cryptocurrency wallets," in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, New York, NY, USA, 2019, pp. 472–480.

[81] A. I. Sanka, M. Irfan, I. Huang, and R. C. Cheung, "A survey of breakthrough in blockchain technology: Adoptions, applications, challenges and future research," *Computer communications*, vol. 169, pp. 179–201, 2021.

[82] D. Son, S. Al Zahr, and G. Memmi, "Performance analysis of an energy trading platform using the ethereum blockchain," in *2021 IEEE International Conference on Blockchain and Cryptocurrency*, 2021, pp. 1–3.

[83] H. Ozkul, E. Celiker, M. Aydos, and A. Ozsoy, "Comparison of top 10 well-known blockchain consensus algorithms," in *2023 16th International Conference on Information Security and Cryptology*, Ankara, Turkey, 2023, pp. 1–6.

[84] Etherscan, "Pending transactions," feb 2025. [Online]. Available: https://web.archive.org/web/20250203120918/https%3A%2F%2Fetherscan.io%2FtxsPending%3Fps%3D100

[85] N. Kannengießer, M. Pfister, M. Greulich, S. Lins, and A. Sunyaev, "Bridges between islands: Cross-chain technology for distributed ledger technology," in *Proceedings of the 53rd Hawaii International Conference on System Sciences*, Maui, Hawaii, USA, 2020, pp. 5298–5307.

[86] M. Iqbal and R. Matulevičius, "Exploring sybil and double-spending risks in blockchain systems," *IEEE Access*, vol. 9, pp. 76 153–76 177, 2021.

[87] Misja.com, "Epoch & unix timestamp conversion tools," mar 2025. [Online]. Available: https://www.epochconverter.com/

[88] L. T. Corp., "Bitcoin hashprice index," jan 2025. [Online]. Available: https://web.archive.org/web/20250131092153/https://hashrateindex.com/rigs/bitmain-antminer-s21-pro?ref=hashrateindex.com

[89] CoinMarketCap, "Ethereum markets," jan 2025. [Online]. Available: https://web.archive.org/web/20250129232113/https://coinmarketcap.com/currencies/ethereum/

[90] Beaconscan, "The number of validators being run on the mainnet beacon chain," jan 2025. [Online]. Available: https://web.archive.org/web/20250128032442/https://beaconscan.com/stat/validator

[91] D. Kirste, N. Kannengießer, R. Lamberty, and A. Sunyaev, "How automated market makers approach the thin market problem in cryptoeocnomic systems," mar 2025. [Online]. Available: https://arxiv.org/abs/2309.12818

[92] R. Lamberty, D. Kirste, N. Kannengießer, and A. Sunyaev, "Hybcbdc: A design for central bank digital currency systems enabling digital cash," *IEEE Access*, 2024.

[93] R. Paulavičius, S. Grigaitis, and E. Filatovas, "A systematic review and empirical analysis of blockchain simulators," *IEEE access*, vol. 9, pp. 38 010–38 028, 2021.

[94] L. Kiffer, D. Levin, and A. Mislove, "Stick a fork in it: Analyzing the ethereum network partition," in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, Palo Alto, CA, USA, 2017, pp. 94–100.

[95] L. Kiffer, R. Rajaraman, and A. Shelat, "A better method to analyze blockchain consistency," in *Proceedings of the 2018 acm sigsac conference on computer and communications security*, Toronto, Canada, 2018, pp. 729–744.

[96] M. Alharby and A. van Moorsel, "Blocksim: An extensible simulation tool for blockchain systems," *Frontiers in Blockchain*, vol. 3, p. 28, 2020.

[97] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of computer and Telecommunication Systems*, Milwaukee, WI, USA, 2018, pp. 264–276.

[98] P. E. Agre, "P2p and the promise of internet equality," *Communications of the ACM*, vol. 46, no. 2, p. 39–42, feb 2003. [Online]. Available: https://dl.acm.org/doi/10.1145/606272.606298

[99] D. Saingre, T. Ledoux, and J.-M. Menaud, "Bctmark: a framework for benchmarking blockchain technologies," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications*, Antalya, Turkey, 2020, pp. 1–8.

[100] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE International conference on Blockchain*, Atlanta, GA, USA, 2019, pp. 536–540.

[101] M. Shuaib, N. H. Hassan, S. Usman, S. Alam, N. A. A. Bakar, and N. Maarop, "Performance evaluation of dlt systems based on hyper ledger fabric," in *2022 4th International Conference on Smart Sensors and Application*, Kuala Lumpur, Malaysia, 2022, pp. 70–75.

[102] W. Choi and J. W.-K. Hong, "Performance evaluation of ethereum private and testnet networks using hyperledger caliper," in *2021 22nd Asia-Pacific Network Operations and Management Symposium*, Tainan, Taiwan, China, 2021, pp. 325–329.

[103] H. Pan, X. Duan, Y. Wu, L. Tseng, M. Aloqaily, and A. Boukerche, "Bbb: A lightweight approach to evaluate private blockchains in clouds," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, Taipei, Taiwan, 2020, pp. 1–6.

[104] K. Qian, Y. Liu, Y. Han, and K. Wang, "Bcadvisor: Enabling green blockchain systems through resource-oriented benchmarking," in *ICC 2022-IEEE International Conference on Communications*, Seoul, South Korea, 2022, pp. 4031–4036.

[105] I. Alom, M. S. Ferdous, and M. J. M. Chowdhury, "Blockmeter: An application agnostic performance measurement framework for private blockchain platforms," *IEEE Transactions on Services Computing*, vol. 16, no. 6, pp. 3879–3891, 2023.

[106] A. Kassab, E. Rivière, G. Rosinosky, R. Sadre, and V. H. Tran, "C2b2: a cloud-native chaos benchmarking suite for the hyperledger fabric blockchain," in *2022 18th European Dependable Computing Conference*, Zaragoza, Spain, 2022, pp. 89–96.

[107] M. Dabbagh, M. Kakavand, M. Tahir, and A. Amphawan, "Performance analysis of blockchain platforms: Empirical evaluation of hyperledger fabric and ethereum," in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology*, Kota Kinabalu, Malaysia, 2020, pp. 1–6.

[108] M. Imran, B. Yao, W. Ali, A. Akhunzada, M. K. Azhar, M. Junaid, and U. Iqbal, "Research perspectives and challenges of blockchain for data-intensive and resource-constrained devices," *IEEE Access*, vol. 10, pp. 38 104–38 122, 2022.

[109] T. Nakaike, Q. Zhang, Y. Ueda, T. Inagaki, and M. Ohara, "Hyperledger fabric performance characterization and optimization using goleveldb benchmark," in *2020 IEEE International Conference on Blockchain and Cryptocurrency*, Toronto, Canada, 2020, pp. 1–9.

[110] J. A. Chacko, R. Mayer, and H.-A. Jacobsen, "Why do my blockchain transactions fail? a study of hyperledger fabric," in *Proceedings of the 2021 International Conference on Management of Data*, New York, NY, USA, 2021, pp. 221–234.

[111] Y. Liu, K. Qian, K. Wang, and L. He, "Effective scaling of blockchain beyond consensus innovations and moore's law: Challenges and opportunities," *IEEE Systems Journal*, vol. 16, no. 1, pp. 1424–1435, 2021.

[112] D. K. Meena, R. Dwivedi, and S. Shukla, "Preserving patient's privacy using proxy re-encryption in permissioned blockchain," in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security*, Granada, Spain, 2019, pp. 450–457.

[113] A. Sharma, F. M. Schuhknecht, D. Agrawal, and J. Dittrich, "Blurring the lines between blockchains and database systems: the case of hyperledger fabric," in *Proceedings of the 2019 International Conference on Management of Data*, Amsterdam, Netherlands, 2019, pp. 105–122.

[114] L. Kuhring, Z. István, A. Sorniotti, and M. Vukolić, "Streamchain: Building a low-latency permissioned blockchain for enterprise use-cases," in *2021 IEEE International Conference on Blockchain*, Melbourne, Australia, 2021, pp. 130–139.

[115] D.-Y. Tsai, S. A. Harding, M.-F. Sie, and S.-w. Liao, "Testbed design and performance analysis for multilayer blockchains," in *2021 IEEE International Conference on Blockchain and Cryptocurrency*, Seoul, Korea, 2021, pp. 1–5.

SUPPLEMENTARY MATERIAL

APPENDIX A
OVERVIEW OF INPUT VARIABLES

Table A1 briefly describes 22 input variables used in the metrics to operationalize constructs related to the blockchain trilemma subconcepts.

TABLE A1
OVERVIEW OF INPUT VARIABLES OF METRICS THAT OPERATIONALIZE CONSTRUCTS OF THE BLOCKCHAIN TRILEMMA SUBCONCEPTS.

Input Variable	**Description**	**Used in Equations**
Block Confirmation Time	The timestamp in milliseconds when a new block is (assumed to be) confirmed in a blockchain system.	16, 30
Block Creation Interval	The average time in milliseconds between the proposal of consecutive blocks that are included in a blockchain.	17, 23
Block Gas Cost	The computation cost required to execute a transaction in a blockchain system.	23
Block Gas Limit	The maximum amount of gas available in a block to process a transaction in a blockchain system.	23
Block Proposal Time	The timestamp in milliseconds when a new block is proposed to a blockchain system.	16, 30
Hashing Power	The number of hashes per second used to produce a new block.	9, 31
MemPool Size	The maximum possible number of pending transactions buffered in a mempool.	23
Number of Confirmed Blocks	The number of (probabilistically) finalized blocks stored in a blockchain system.	30, 36
Number of Confirmed Transactions	The total number of transactions processed and included in a block that has been (probabilistically) finalized into the mainchain of a blockchain system.	15, 22
Number of Failures	The number of failures in a blockchain system within a given observation timespan.	27
Number of Locations	The number of locations where validating nodes in a blockchain system operate.	3
Number of Proposed Blocks	The number of blocks proposed by validating nodes to a blockchain system.	1
Number of Repairs	The number of repairs, including recovery, a blockchain system experiences after failures within a defined timespan.	27
Number of Stale Blocks	The number of valid blocks that are proposed but eventually not included in the mainchain.	36
Number of Tokens	The number of tokens owned by an individual validating node.	10, 12, 31
Number of Transactions	The total number of transactions issued to a blockchain system.	15
Number of Validating Nodes	The number of validating nodes in a blockchain system.	1, 3, 10, 9, 31, 12
Resource Cost	The value of tokens or hashing power in fiat currency.	31
Security Confirmation	The required minimum number of blocks that must be appended to a specific block to achieve sufficiently high probabilistic finality for that block.	17
Total Number of Locations	The number of possible locations where validating nodes could operate.	3
Total Operational Time	The timespan a blockchain system operates correctly within a define observation time.	27, 35
Total Repair Time	The timespan a blockchain system takes to recover from failure.	27

APPENDIX B
OVERVIEW OF ANALYSIS APPROACHES AND THEIR OPERATIONALIZED CONSTRUCTS

Table B1 illustrates an overview of analysis approaches used to quantify the blockchain trilemma subconcepts. For example, Blockchain Benchmark Standardized Format (BBSF) [7], BCAdvisor [99], Caliper [100]–[102], and COCONUT [74].) can be used to investigate the scalability construct. *Throughput* (equation 22) and *confirmation latency* (equation 16) are predominant operationalized constructs used to quantify scalability with analysis approaches. The constructs apply to most blockchain systems, including the Algorand, Ethereum, and Hyperledger Fabric systems.

Similarly, Google BitQuery, BitInforCharts, and Etherscan [11], [14], [35], [38] have been used to generate benchmark data to quantify constructs related to the DoD construct. Block proposal randomness (equation 1) and wealth distribution (equation 12) are common constructs focusing on investigating the DoD. Analysis of DoD mainly focuses on permissionless blockchain systems (e.g., the Bitcoin and Ethereum systems) because the data to analyze such blockchain systems is publicly available due to their permission models compared to permissioned blockchain systems (e.g., blockchain systems based on Hyperledger Fabric protocol).

TABLE B1

OVERVIEW OF ANALYSIS APPROACHES FOR INVESTIGATING THE BLOCKCHAIN TRILEMMA SUBCONCEPTS AND EXAMPLES OF BLOCKCHAIN SYSTEMS ANALYZED USING THE APPROACHES.

	Operationalized Constructs used to Quantify the Blockchain Trilemma Subconcepts			
Analysis Approach	**DoD**	**Scalability**	**Security**	**Analyzed Blockchain Systems**
BBB [103]		Confirmation latency (eq. 16), Throughput (eq. 22)		Ethereum
BBSF [7]		Confirmation latency (eq. 16), Throughput (eq. 22)	Fault tolerance (eq. 32)	Ethereum, Quorum
BCadvisor [104]		Confirmation latency (eq. 16), Throughput (eq. 22)		Ethereum, Hyperledger Fabric, Parity
Blockbench [8]		Confirmation latency (eq. 16), Throughput (eq. 22)	Fault tolerance (eq. 32)	Ethereum, Hyperledger Fabric, Parity
Blockmeter [105]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Fabric, Hyperledger Sawtooth
BlockSim [96]		Confirmation latency (eq. 16), Throughput (eq. 22)	Stale block rate (eq. 36)	Bitcoin, Ethereum
C2B2 [106]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Fabric
Caliper [107]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Fabric
COCONUT [74]		Confirmation latency (eq. 16),Throughput (eq. 22)		BitShares, Corda Enterprise, Corda Open Source, Hyperledger Fabric, Quorum, Sawtooth
Debug-Bench [108]		Confirmation latency (eq. 16), Throughput (eq. 22)		Ethereum
DIABLO [9]		Confirmation latency (eq. 16), Throughput (eq. 22), Availability (eq. 15)	Fault tolerance (eq. 32)	Algorand, Ethereum, Hyperledger Fabric, Quorum, Redbelly, Solana, Zcash
Fu et al. [59]	Token concentration (eq. 10), Hashing power distribution (eq. 9), Wealth distribution (eq. 12)	Confirmation latency (eq. 16), throughput (eq. 22)		Algorand, Ethereum
GoLevelDB [109]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Fabric
Gromit [56]		Confirmation latency (eq. 16), Throughput (22)		Algorand, Bitshares, Ethereum, Hyperledger Burrow, Hyperledger Fabric, Stellar
HyperLedgerLab [110]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Fabric
SimBlock [2]	Token concentration (eq. 10), Hashing power distribution (eq. 9), Wealth distribution (eq. 12)	Throughput (eq. 22)	Fault tolerance (eq. 32), Stale block rate (eq. 36)	Bitcoin
Prism [111]		Confirmation latency (eq. 16), Throughput (eq. 22)		EOS, Ethereum, Hyperledger Burrow, Hyperledger Fabric, Hyperledger Iroha, Hyperledger Sawtooth

eq.: equation

TABLE B1
OVERVIEW OF ANALYSIS APPROACHES FOR INVESTIGATING THE BLOCKCHAIN TRILEMMA SUBCONCEPTS AND EXAMPLES OF BLOCKCHAIN SYSTEMS ANALYZED USING THE APPROACHES (CONTINUED).

	Operationalized Constructs used to Quantify the Blockchain Trilemma Subconcepts			
Analysis Approach	**DoD**	**Scalability**	**Security**	**Analyzed Blockchain Systems**
PTE [112]		Confirmation latency (eq. 16), throughput (eq. 22)		Hyperledger Fabric
Quattrocchi et al. [12]	Hashing power distribution (eq. 9), Wealth distribution (eq. 12)	Throughput (eq. 22)	Cost of attack (eq. 31)	Bitcoin, Cardano, Ethereum, Polygon, Solana
SmallBank [113]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Fabric
StreamChain [114]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Fabric
Thakkar et al. [97]		Confirmation latency (eq. 16), Throughput (eq. 22)	Fault tolerance (eq. 32)	Hyperledger Fabric
Tsai et al. [115]		Confirmation latency (eq. 16), Throughput (eq. 22)		Quorum, Stellar
Wu et al. [14]	Block proposal randomness (eq. 1)			Bitcoin, Ethereum
Lin et al. [35]	Block proposal probability (eq. 1), Hashing power distribution (eq. 9), Wealth distribution (eq. 12)			Bitcoin, Ethereum
Jia et al. [38]	Block proposal randomness (eq. 1), Wealth distribution (eq. 12)			Binance Smart Chain, Cardano, Elrond, Fantom, Polkadot, Polygon, Tron
Juodis et al. [11]	Block proposal randomness (eq. 1), Token concentration (eq. 10), Hashing power distribution (eq. 9), Wealth distribution (eq. 12)			Bitcoin, Ethereum
Ban et al. [80]		Confirmation latency (eq. 16), Throughput (eq. 22)		Hyperledger Burrow, Hyperledger Fabric, , Hyperledger Indy, Hyperledger Iroha, Hyperledger Sawtooth,)
Quattrocchi et al. [12]	Hashing power distribution (eq. 9), Wealth distribution (eq. 12)	Throughput (eq. 22)	Cost of attack (eq. 31)	Bitcoin, Cardano, Ethereum, Solana, Polygon
Gräbe et al. [10]		Confirmation latency (eq. 17), Throughput (eq. 22)	Availability (eq. 27), Consistency (eq. 30), Fault tolerance (eq. 32), Reliability (eq. 35)	Ethereum, Hyperledger Indy, Tezos

eq.: equation