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

Mansur Masama Aliyu[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*—The blockchain trilemma highlights the difficulty of simultaneously achieving a high degree of decentralization (DoD), scalability, and security in blockchain systems. While numerous constructs and metrics have been proposed to analyze these subconcepts, existing guidance is fragmented and inconsistent, limiting comparability across studies. This lack of clarity hinders practitioners in identifying Pareto-optimal blockchain system designs that meet common non-functional requirements. We systematically reviewed literature on the blockchain trilemma and blockchain benchmarks to synthesize constructs and their operationalizations through metrics to analyze the trilemma's subconcepts. We identified 12 constructs, operationalized through 15 metrics, that capture DoD, scalability, and security. We explain how these constructs apply across different blockchain systems and provide a structured overview that supports benchmarking and blockchain system design. Beyond blockchain, the findings offer insights for distributed database systems that rely on consensus and state machine replication. This work contributes a harmonized foundation for quantitative analyses of the blockchain trilemma, guiding both researchers in developing analysis approaches and practitioners in evaluating real-world systems.**

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

I. Introduction

Blockchain systems are commonly designed and evaluated against non-functional requirements spanning degree of decentralization (DoD), scalability, and security–the subconcepts of the blockchain trilemma concept. The 'blockchain trilemma' posits that these subconcepts cannot be maximized simultaneously; in practice, productive blockchain systems are designed toward Pareto-optimality that balances these subconcepts to meet requirements [1]–[4].

For a general intuition about the trade-offs: increasing DoD typically involves more independent validating nodes and more evenly distributed influence on consensus, but this raises coordination and network overhead, which can increase latency and reduce throughput. Pursuing scalability via larger blocks, shorter block intervals, or higher hardware baselines can improve throughput and latency, yet it tends to favor well-resourced participants, thereby reducing DoD. Strengthening security (e.g., through larger committees, additional verification, or redundancy) can harden the system against faults and attacks, but often adds computational and networking costs that also pressure scalability. In short, improving one subconcept can impose costs on the others.

Identifying Pareto-efficient configurations, therefore, requires a systematic approach to quantifying DoD, scalability, and security, and to assessing the strengths of trade-offs between these subconcepts [5]. A variety of analysis approaches, such as *BBSF* [6], *BLOCKBENCH* [7], *Diablo* [8], and the simulator *SimBlock* [2], [9], operationalize these subconcepts using different constructs and metrics.[1] For DoD, examples include the construct of *block-proposal randomness* [10]–[12] and the construct of *wealth distribution* [2], [13]–[15], each with multiple candidate metrics. However, heterogeneous operationalizations hinder comparability across studies and leave practitioners with limited guidance when selecting constructs and metrics aligned with their questions.

Selecting suitable constructs and metrics is further complicated by conceptual ambiguities around the trilemma. Some works map the trilemma's subconcepts to distributed systems, such as the CAP theorem [16], [17], but such mappings are necessarily partial, as they address only subsets of the subconcepts or blend constructs across them. This makes it difficult to argue, in a principled way, why a particular construct or metric is appropriate for a given analysis. To support more consistent, defensible choices, we pose the following research question: *Which constructs and associated metrics are suitable to quantify the blockchain trilemma's subconcepts?*

We conducted a systematic literature search to identify publications that propose constructs or metrics for analyzing DoD, scalability, or security [18]. Using abductive thematic analysis, we iteratively synthesized constructs and their associated metrics [19]–[22]. This process was guided by the interplay between empirical findings and conceptual framing, supplemented by targeted theoretical sampling to address gaps

[1] A *metric* is a mapping from inputs (e.g., counts, shares, or times) to a quantitative output used to operationalize a construct.

and validate emerging insights. Based on this analysis, we developed an overview of analysis approaches that apply these constructs and metrics to investigate the blockchain trilemma.

This work contributes to purposeful evaluation of blockchain systems under the trilemma in three ways. First, we synthesize common constructs and their operationalization through metrics, explaining their applicability, interpretability, and limitations for analyzing DoD, scalability, and security. Second, by clarifying the input variables of these metrics, we support data collection in benchmarks (e.g., which system characteristics must be monitored to feed relevant metrics). Third, by comparing analysis approaches based on their choice of constructs and metrics, we guide the selection of suitable approaches for future investigations.

The remainder of this work is structured as follows. Section II introduces the foundations of blockchain technology, including the manifestation of the blockchain trilemma and existing conceptual and empirical attempts to operationalize its subconcepts. Section III describes our literature search and analysis. Section IV presents principal constructs and metrics for operationalizing the blockchain trilemma's subconcepts, alongside analysis approaches that implement them. Section V discusses our findings, contributions to practice and research, limitations, and open research directions. Finally, Section VI concludes with key takeaways.

II. Theoretical Foundations and Related Research

The blockchain trilemma has been identified for blockchain systems, in which a high DoD, scalability, and security are desirable. This section introduces the system model to which the blockchain trilemma applies and explains how design choices shape its manifestation. Subsection II-A briefly explains the foundations of such distributed databases, including a system model upon which this work mainly relies. Using the system model, we introduce the blockchain trilemma's subconcepts and describe trade-offs between them. Subsection II-C gives an overview of related research on the blockchain trilemma.

A. Blockchain Technology and System Model

Blockchain technology enables the operation of blockchain systems, a class of replicated databases with consensus. The database is maintained by multiple *validating nodes*, which contribute to one or more consensus-critical tasks, such as processing transactions and maintaining the ledger state. We use *validating node* as a generic umbrella term rather than a protocol-specific role name. Not all validating nodes necessarily perform every function: some participate only in ordering, whereas others also validate blocks, execute transactions to update the ledger state, and store the resulting data.

Across blockchain system designs, the specialization of validating nodes differs, particularly in terms of how they are involved in executing the consensus protocol and ledger replication. In the Bitcoin system, *miners* (block producers) assemble candidate blocks by selecting and ordering pending transactions; the canonical global order of blocks then emerges from proof-of-work (PoW) competition and the longest-work fork-choice rule [23], [24]. Full nodes, which may or may not be miners, validate blocks and maintain the ledger, based on unspent transaction outputs (UTXOs). In the Ethereum system[2], *validators* are assigned to propose blocks in time slots, while other validators in committees attest to proposed blocks. Ethereum nodes typically run *paired consensus and execution clients*: the consensus client manages fork choice and finality, while the execution client processes transactions and updates state. Validators operate both clients in tandem [25]. The Solana system uses a leader schedule derived from proof-of-history (PoH) to provide a verifiable ordering source; PoH is not a consensus protocol itself but feeds a Byzantine fault-tolerant voting protocol—Tower Byzantine Fault Tolerance (BFT)—that provides safety and fast confirmations [26]. Polygon Proof-of-Stake (PoS) distinguishes between *Bor* nodes (block production) and *Heimdall* nodes (checkpoints that finalize state on the Ethereum system); confirmations on the Bor chain are probabilistic until checkpointed by Heimdall [27]. In permissioned systems, such as those based on Hyperledger Fabric, roles are explicitly separated: *orderers* provide total ordering of transactions (e.g., via Raft) with deterministic finality, while *peers* endorse, validate, and store the ledger and world state. Peers—*not* orderers—operate a replicated state machine [28].

In contrast to a validating node, a *transaction originator* is any external entity (e.g., client/application, end-user wallet, or service) that creates and submits transactions to the blockchain system. Transaction originators do not participate in consensus or maintain the ledger; they interact with the system only through the *entry and propagation* interfaces of validating nodes such as JavaScript Object Notation–Remote Procedure Call (JSON–RPC), Google–Remote Procedure Call (gRPC), and Quick User Datagram Protocol Internet Connections (QUIC). Figure 1 presents the abstract layered model; Table I instantiates it for the platforms considered in this work.

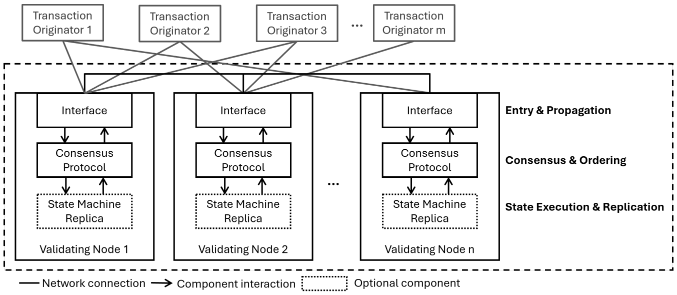

Fig. 1. Layered blockchain system model (adapted from Leinweber et al. [29]). The dashed border delineates the *replicated database system* composed of *validating nodes*; *transaction originators* (clients/applications, end-user wallet, or service) submit transactions from outside this boundary. Validating nodes implement one or more functional layers: *Entry & Propagation* (interfaces and dissemination), *Consensus & Ordering* (canonical ordering/finality), and *State Execution & Replication* (deterministic validation/execution and ledger updates). Not all validating nodes implement every layer; for example, Hyperledger Fabric *orderers* participate only in Consensus & Ordering and do not execute transactions or store the application ledger, whereas Fabric *peers* and validators in permissionless systems implement the replicated state machine.

[2] Unless otherwise specified, "Ethereum" refers to the post-Merge system (often called "Ethereum 2.0" in earlier literature).

System	Entry & Propagation	Consensus & Ordering	State Execution & Replication
Bitcoin [23], [30], [31]	JSON-RPC, mempool, gossip	PoW + longest-work fork choice; probabilistic finality	Full nodes validate blocks and update the UTXO set (miners typically also run full nodes)
Ethereum [25]	JSON-RPC, gossip	Gasper (combining LMD-GHOST fork choice with Casper-FFG epoch finality)	Validators (consensus and execution clients) validate transactions, execute state transitions, and update the ledger state
Hyperledger Fabric [28], [32]	Client→peer endorsement gRPC; client→orderer submit	Ordering service (Raft/BFT) establishes total order; deterministic finality	**Peers** endorse, validate, execute chaincode deterministically, and commit ledger/world state; **orderers do not store the ledger**
Polygon PoS [27]	RPC/mempool, gossip	Bor block production + Heimdall (Tendermint-style) checkpoint finality (to the Ethereum system)	Bor/validators execute/commit locally; Heimdall checkpoints finalize
Solana [26], [33]	RPC, QUIC, Turbine	PoH leader schedule + Tower BFT voting; fast confirmations	Validators replay/execute and commit ledger (shreds→blocks)

TABLE I

Instantiation of the layered model for selected platforms. Each row maps platform-specific components to (i) *Entry & Propagation*, (ii) *Consensus & Ordering*, and (iii) *State Execution & Replication*. Only validating nodes that execute transactions and update the ledger implement the replicated state machine (e.g., peers, full nodes, validators); some validating nodes (e.g., Fabric orderers) do not.

Once a transaction is issued by a transaction originator, validating nodes synchronize by executing a consensus protocol. The consensus protocol determines canonical ordering and when entries are considered final. Consensus protocols differ across blockchain systems: the Bitcoin system employs Nakamoto consensus [23]; the Ethereum system combines a Latest Message-Driven Greedy Heaviest-Observed Sub-Tree (LMD-GHOST) fork choice with Casper the Friendly Finality Gadget (Casper-FFG), collectively known as Gasper [34]; the Solana system integrates PoH with BFT voting (Tower BFT) [26]; Polygon PoS applies hybrid block production with Tendermint-style checkpoint finality [27]; and Hyperledger Fabric uses pluggable ordering such as Raft (and BFT options) [32]. Despite these differences, all aim to ensure consistency among replicas under faults, often including adversarial behavior [1], [35], [36].

Consensus protocols shape key properties of blockchain systems, including their permission models, finality models, and fault tolerance.

a) Permission models: Blockchain system can be *permissionless*, where identity is open and consensus participation is Sybil-resisted by economic or resource costs (e.g., Algorand [37], Bitcoin [23], Cardano [38], and Ethereum [39]), or *permissioned*, where participation is restricted to authenticated members under predefined rules [40], [41] (e.g., BitShares [42], [43], Hyperledger Fabric [28], [32], and Red Belly [44]).

b) Finality models: Finality models define when appended blocks are considered finalized (i.e., cannot be reverted under the assumptions of the consensus protocol). With *deterministic* (immediate) finality, a committed block is final once agreed (e.g., Algorand [37] and Hyperledger Fabric systems [32]). With *economic* or *probabilistic* finality, finality is achieved after further confirmations or epochs. For example, the Ethereum system finalizes epochs via Casper-FFG once sufficient attestations are observed (typically minutes) [45], while the Bitcoin system increases the probability of irreversibility as more blocks are built on top of a block. The Solana system [26] provides rapid confirmations via Byzantine fault-tolerant voting but may require validator restarts under certain failure conditions, which can temporarily reduce availability; Polygon PoS attains checkpoint finality on Heimdall (to the Ethereum system), while Bor-chain confirmations before checkpointing are effectively probabilistic.

c) Fault tolerance: Blockchain systems can be omission-tolerant, crash fault-tolerant, and Byzantine fault-tolerant [1], [41]. *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. *Byzantine fault tolerance (BFT)* extends crash-fault tolerance by the ability to compensate for accidental faults and deliberate attacks [46], [47]. Accidental faults include software defects and misdesign, while adversarial attacks involve strategies, such as selfish mining [48]–[50]. Most blockchain systems—both permissionless (e.g., Bitcoin and Ethereum) and permissioned (e.g., Hyperledger Fabric and Red Belly)—exhibit multiple forms of fault tolerance.

d) Performance: Consensus protocols strongly influence the performance of blockchain systems [1], [41], [51], especially in terms of throughput. Voting-based *consensus protocols* with immediate finality, such as practical Byzantine Fault Tolerance (PBFT) [52], [53], degrade as network size increases [54]. This is due to the higher communication complexity required for consensus among more validating nodes. In contrast, protocols with probabilistic finality, such as Nakamoto consensus in Bitcoin [23], are less sensitive to network size, but relax consistency assumptions and provide only eventual consistency [41].

Although the blockchain trilemma originated in blockchain discourse, similar tensions occur in consensus-based replicated databases (e.g., CockroachDB, ZooKeeper) that implement replicated state machines without a strict 'blockchain' data structure [55]–[57].

B. Blockchain Trilemma's Subconcepts and Their Interrelationships

Optimizing blockchain system designs to balance the blockchain trilemma's subconcepts is essential to meet non-functional requirements. Achieving this balance requires a clear conceptual foundation of *DoD*, *scalability*, and *security*, as well as an understanding of how these subconcepts interact in real-world blockchain systems. Existing literature presents multiple and sometimes conflicting definitions of these subconcepts, which dilutes the conceptual foundation of the blockchain trilemma and complicates empirical analyses aimed at identifying Pareto-optimal designs.

One reason for the blockchain trilemma concept being diluted is that its subconcepts comprise multiple, interrelated constructs, and grasping all relevant constructs of a subconcept is often difficult. Most previous definitions neglect important constructs or overemphasize selected ones, making it difficult to understand the meaning of the individual subconcepts. In this work, we use the term *construct* to refer to a dimension of a blockchain trilemma's 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. [58]). Input variables can often be manipulated in experiments as *independent variables* (e.g., block size), whereas output variables are *dependent variables* when they represent measurable effects of design choices. Figure 2 illustrates the interrelationships between the blockchain trilemma, its subconcepts, constructs, and metrics.

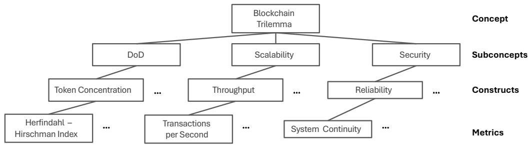

Fig. 2. Interrelationships between the blockchain trilemma, subconcepts, constructs, and metrics.

1) Blockchain Trilemma Subconcepts: The blockchain trilemma highlights the presumed impossibility of simultaneously maximizing DoD, scalability, and security of blockchain systems that are consistent with the system model described in Section II-A [1], [14]. Although widely recognized, the subconcepts are inconsistently defined in the literature. To provide a harmonized foundation, we synthesize extant definitions and clarify their scope.

a) Degree of Decentralization: DoD has been defined from multiple perspectives: the equitable and quasi-autonomous participation of validating nodes in consensus [41], [59], the geographical and network diversity of validating nodes [1], [60], [61], and the distribution of economic resources such as tokens or mining power [2], [13], [14], [62]. These perspectives highlight different but related facets of DoD. We define *DoD as the extent to which validating nodes participate equitably and (quasi-)autonomously in consensus*. Geographic distribution and economic concentration are incorporated as influencing factors, rather than core definitional elements, to provide a clear operational boundary while recognizing their impact on consensus participation.

b) Scalability: Scalability has been defined in multiple ways. Some studies have associated it with the maximum transaction processing rate (i.e., maximum possible throughput) [7], [14], [60], [63], while others define it more broadly as the efficiency of the system as scale and workload increase [36]. It also depends on how system performance scales with the number of consensus-critical participants [41], [64]. We therefore define *scalability as the ability of a blockchain system to handle changing workloads, including the number of validating nodes per decision and the number of transactions per second* [1], [4], [65]. Consensus protocols such as HotStuff, Paxos, and Raft [66]–[68] reduce message complexity by designating leaders or committees, thereby increasing throughput. Similarly, PoS systems such as Ethereum use slot-based proposers and committees of attesters [39], [69], and Algorand pseudo-randomly selects subsets of validators [37]. These mechanisms improve scalability but may reduce DoD by concentrating decision power per step, even if committees rotate or membership changes over time [70].

c) Security: Security has likewise been defined inconsistently. Some works emphasize availability and fault tolerance [6], [7], others highlight partition tolerance [16], [17], while others stress adversarial resistance [3]. We harmonize these by defining *security as the degree to which a blockchain system remains operational and resilient against faults, partitions, and malicious attacks* [3], [6], [41]. Security manifests at multiple layers: protocol robustness (e.g., BFT or crash-fault tolerance), data integrity (e.g., cryptographic hash chaining of blocks), and economic deterrence (e.g., the resource costs of PoW or the staking requirements of PoS). Consensus protocols such as HotStuff [66] and PBFT [52] provide BFT, while Nakamoto consensus [23] achieves probabilistic finality under majority-honest assumptions. These protocols combine resilience, consistency, and economic deterrence in different ways to sustain operation despite faults or attacks.

2) Interrelationships Between the Blockchain Trilemma's Subconcepts: At first glance, the trade-offs between DoD, scalability, and security may appear to re-label established theorems such as CAP. The CAP theorem formalizes the impossibility of simultaneously maximizing consistency, availability, and partition tolerance [71]–[73]. However, the analogy is limited: CAP theorem assumes partitioned networks and focuses on distributed databases, whereas the blockchain trilemma also involves economic incentives and consensus participation. To clarify the blockchain trilemma, the following describes how trade-offs under the blockchain trilemma manifest in blockchain systems.

a) DoD vs. Scalability: High DoD requires a large share of (or all) validating nodes in a blockchain system to equitably and autonomously participate in consensus finding. This commonly increases message complexity and communication overhead, reducing throughput [41], [74], for example, when validating nodes need to negotiate and agree on changes to the blockchain. To increase throughput, blockchain systems often employ fewer validating nodes per decision, for example, in leader- or committee-based protocols [5], [75]. The Ethereum system's transition from PoW to PoS (the Merge) illustrates this trade-off. By introducing slot-based proposers and attesting committees, throughput and energy efficiency improved. However, these design changes also concentrated decision-making power per slot, which may reduce the actual DoD depending on validator diversity and participation.

b) DoD vs. Security: A high DoD can be achieved by increasing the number and heterogeneity of validating nodes that must coordinate to reach consensus. As the number of validating nodes grows and becomes geographically dispersed, message propagation slows and network assumptions weaken, making it harder to preserve safety and liveness under adversarial or faulty conditions [**?**], [48]. A large number of validating nodes can, for example, elongate temporary inconsistencies that facilitate double spending in probabilistic-finality systems [48]. While broader participation can reduce the risk of collusion and system failure, it also enlarges the attack surface and complicates synchronization, which can reduce effective security. Permissioned (or committee-based) blockchain systems mitigate these risks by restricting membership and relying on deterministic finality consensus protocols (e.g., Raft in Hyperledger Fabric [76]), thereby strengthening safety under controlled conditions. However, this limits open, equitable participation and autonomy, reducing certain dimensions of decentralization.

c) Scalability vs. Security: Scalability is enhanced when only a few well-connected validating nodes participate in consensus. However, concentrating participation reduces security, because compromising a small number of validating nodes suffices to disrupt operation. Geographical colocation of validating nodes may improve throughput, but increases the risk of correlated failures. Conversely, distributing validating nodes across regions improves resilience against crashes and attacks, but synchronization across diverse validating nodes slows down transaction finalization and reduces throughput [41]. For example, the Solana system prioritizes high throughput by limiting block propagation delays through leader scheduling and high-bandwidth requirements. While this improves scalability, it reduces resilience to validating node churn and can lead to liveness issues under overload, illustrating a scalability–security trade-off [26], [77], [78].

In summary, the blockchain trilemma reflects inherent tensions in distributed consensus design. Each subconcept can be strengthened only by weakening another, and different system designs instantiate these trade-offs in different ways.

C. Related Research on Measuring the Blockchain Trilemma's Subconcepts

Research on the blockchain trilemma has proceeded along two primary lines: conceptual works that define the subconcepts and trade-offs, and empirical works that propose constructs and metrics to operationalize them. Both lines are essential, but they differ in scope, precision, and consistency.

Conceptual works highlight the blockchain trilemma's subconcepts but use disparate definitions. Xu et al. [36] define DoD primarily with respect to network size, whereas Xiao et al. [1] emphasize the geographical diversity of validating nodes. Other authors stress autonomy and equal participation of validating nodes in consensus [3], [41], [59], [79]. These inconsistent definitions illustrate how surveys adopt different perspectives, sometimes mixing subconcepts or omitting dimensions, which dilutes the conceptual clarity of the blockchain trilemma.

Empirical works complement these surveys by proposing constructs and metrics for DoD, scalability, and security, but typically investigate subconcepts in isolation rather than the blockchain trilemma as a whole. Benchmarks such as BLOCKBENCH [7], BBSF [6], and Diablo [8] provide experimental setups to evaluate blockchain systems, but employ different constructs. To measure security, for example, stale block rate has been used [80], [81], while other works have used fault tolerance approximated by changes in throughput and confirmation latency under faulty validating nodes [6], [7]. Scalability has been estimated through maximum throughput (transactions per second) [6], [7], [82] and confirmation latency [8], [70], [80], [83]. This diversity of constructs makes cross-comparison between studies difficult, as the same subconcept may be quantified in multiple, non-equivalent ways.

For PoW-based blockchain systems, Nakai et al. [2] formally modeled the blockchain trilemma and investigated it through simulations. Their operationalizations rely on hashing power distribution, token concentration, wealth distribution, and throughput. These constructs offer rigor but remain tailored to PoW-based consensus protocols. Economic concentration measures such as token distribution serve as indirect proxies for equitable participation in consensus.

For PoS-based blockchain systems, Fu et al. [70] and Quattrocchi et al. [14] use constructs such as wealth distribution and token concentration for DoD, and throughput and confirmation latency for scalability. For security, Fu et al. [70] argue that higher transaction fees can strengthen economic deterrence, since reverting blocks with high fees becomes increasingly costly for attackers. Quattrocchi et al. [14], in contrast, introduce the cost of attack as a direct proxy for system resilience. Mssassi et al. [60] generalize these approaches, applying both token-based influence and security thresholds (e.g., a majority of honest validating nodes) to assess DoD and security, respectively.

Permissioned blockchain systems have also been studied in relation to the blockchain trilemma. Wang et al. [16] map the blockchain trilemma to the CAP theorem [71], [72]. Under the assumption that the majority of validating nodes are honest, they align consistency with security, availability with scalability, and partition tolerance with DoD. They operationalize consistency via fork probability, availability via throughput, and DoD via the probability that a partitioned network ceases to function. While these constructs are conceptually appealing, their applicability to heterogeneous permissioned systems is limited by the restrictive assumptions of the system model, such as eventual consistency, equal computational power, and at least two-thirds honest validating nodes, which constrain generalizability.

Overall, related research provides a rich but fragmented landscape of constructs and metrics for measuring the blockchain trilemma's subconcepts. Conceptual works often lack precision or conflate dimensions, while empirical works employ metrics that are not always comparable across settings. This fragmentation complicates the task of selecting suitable constructs and justifying their operationalization in benchmarks or theoretical models. A systematic synthesis is therefore required to clarify which constructs and associated metrics are appropriate for quantifying DoD, scalability, and security across diverse blockchain systems.

III. Research Approach

We identified a set of constructs and associated metrics to analyze the subordinate concepts of the blockchain trilemma in two main steps. First, we conducted a systematic literature search [18] to compile an extensive set of relevant publications on the blockchain trilemma. Second, we analyzed the collected literature using abductive thematic analysis [19] to extract the constructs and associated metrics used to measure the blockchain trilemma's subconcepts. The following subsections detail these two steps.

A. Literature Search

We conducted a systematic literature search [18] to identify publications that present constructs and associated metrics for analyzing blockchain trilemma's 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 II).

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 1,210 publications due to insufficient detail. Moreover, we excluded 24 additional publications due to insufficient topic fit. The second relevance assessment yielded 144 relevant publications.

During the abductive thematic analysis (Section III-B), we observed underrepresentation of constructs related to DoD and security. To address this and improve conceptual sufficiency, we complemented the systematic search with purposive sampling guided by abductive reasoning. This approach, inspired by the principle of *theoretical sampling* in grounded theory but applied in a broader abductive sense [19], [20], allowed us to deliberately extend the corpus to address conceptual blind spots and capture up-to-date blockchain systems (e.g., Ethereum 2.0 rather than Ethereum 1.0). 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 17 additional publications that met our inclusion criteria (Table II), resulting in a final corpus of 161 publications.

B. Literature Analysis

We applied abductive thematic analysis [19]–[22] to synthesize constructs and metrics associated with the blockchain trilemma's 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 results. This approach enables theory development that is both grounded in the literature and informed by existing conceptual frameworks [20].

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 abductively coding passages from the 161 publications that referenced constructs or metrics relevant to the blockchain trilemma. Initial codes (e.g., *block-proposal randomness*, *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 recurring and empirically grounded pattern that was associated with at least one metric. In the context of this study, themes were interpreted as *constructs* of the blockchain trilemma's subconcepts. To reduce redundancy, overlapping constructs were merged. For example, *robustness* was subsumed under *fault tolerance*, and *availability* and *success rate* were grouped as a single construct of scalability.

To enhance reliability and reduce subjective bias, we coded subsets of the literature independently and resolved discrepancies through discussion. This iterative process led to a stable structure of 12 constructs, distilled from 410 initial codes, indicating conceptual consistency across the corpus.

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 the final results.

Each construct was then mapped to one of the blockchain trilemma's subconcepts based on patterns in the literature and alignment with conceptual definitions (Section II-B). For instance, *maximum possible throughput* and *confirmation latency* were assigned to scalability, while *fault tolerance* and *stale block rate* were associated with security. Mapping disagreements were resolved through collaborative review and unanimous consensus.

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

To further assess the robustness of the set of extracted constructs and their mapping to the blockchain trilemma's subconcepts, we contacted the author teams of publications included in our corpus. We reached out to 27 author teams and asked them to review the descriptions of their constructs for accuracy. We sent the descriptions of the constructs

TABLE II
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's 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., state channel networks).
Uniqueness	The publication must be the latest version and must not be a duplicate in the literature set.

in a PDF of the manuscript, invited written feedback via email, and offered virtual meetings for discussion. Four author teams responded. Their feedback led to minor refinements of construct descriptions and mappings. For example, we refined *confirmation latency* to reflect the design of Ethereum 2.0, which was not fully represented in the analyzed literature, and rephrased *throughput* to *maximum possible throughput*. Moreover, to remain consistent with the scope of this work, we rephrased *trusted third party* to *trusted validating node*. We sent the refined descriptions back to the authors for verification. As no additional concerns were raised, we consider the construct descriptions and mappings validated through author feedback.

IV. Constructs, Metrics, and Analysis Approaches

This section first presents an overview of the constructs used to operationalize the blockchain trilemma's subconcepts in subsection IV-A. We link the constructs to the blockchain trilemma's subconcepts, explain the operationalization of the constructs through metrics, and offer examples of how the constructs can be used. Moreover, we point out limitations of the operationalized constructs. Subsection IV-B showcases uses of the operationalized constructs in analysis approaches.

A. Constructs and Metrics to Measure the Blockchain Trilemma's Subconcepts

We identified 12 constructs associated with the blockchain trilemma subconcepts (see Table III): 5 for DoD, 3 for scalability, and 4 for security. These constructs are operationalized through 15 metrics, which are detailed in the following. Table IV 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 that operationalize the constructs: *block-proposal randomness*, *geographical diversity*, *hashing power distribution*, *token concentration*, and *wealth distribution*.

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

Depending on the consensus protocol, either a static validating node (e.g., in Raft [84]) or a (pseudo-)randomly selected validating node (e.g., a Bitcoin *miner* under Nakamoto consensus [23] or an Ethereum system *validator* under Gasper [34]) proposes the next block. Block-proposal randomness is commonly computed using Shannon entropy [10]–[13], [15], [62], [85]–[87], which measures uncertainty in discrete events [88] such as which validating node proposes the next block:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i, \qquad p_i = \frac{b_i}{\sum_{j=1}^{n} b_j}, \tag{1}$$

where n is the number of validating nodes and b_i the number of blocks proposed by node i (with $0 \log 0 := 0$). Let $k = \left|\{i : p_i > 0\}\right|$ be the number of nodes that proposed at least one block; then $H \leq \log_2 k$, achieving $\log_2 n$ only if all nodes have equal nonzero proposal probability.

For cross-configuration comparability, a unitless normalization is useful:

$$H_{\text{norm}} = \begin{cases} \dfrac{H}{\log_2 k}, & k \geq 2, \\ 0, & k < 2, \end{cases} \tag{2}$$

which lies in $[0, 1]$ and equals 1 under perfectly even proposer probability across the k active proposers.

For illustration, suppose three validating nodes n_1, n_2, and n_3 proposed 1, 1, and 8 blocks, respectively, out of 10 total. The resulting probabilities are: $p_1 = \frac{1}{10}$, $p_2 = \frac{1}{10}$, $p_3 = \frac{8}{10}$, yielding $H = 0.922$ bits from (1). The maximum entropy for three equally likely proposers is $\log_2(3) = 1.585$ bits, so $H_{\text{norm}} \approx 0.922/1.585 \approx 0.58$. This reflects intermediate dispersion of block-proposal probability, meaning DoD is neither fully concentrated nor fully uniform.

Shannon entropy is applicable to blockchain systems with random leader selection, but has limitations. It focuses only on block proposals, ignoring which proposed blocks are finalized. Network conditions (e.g., bandwidth) can bias propagation speeds, meaning blocks from high-bandwidth partitions are more likely to be accepted [1], [74]. Such conditions can reduce effective DoD even when block-proposal randomness is high. A quick fix can be to compute block-proposal randomness based on finalized blocks. Moreover, Shannon entropy does not distinguish between different validating node roles: in some systems (e.g., Bitcoin, Ethereum), the same validating node proposes and validates blocks, whereas in others (e.g., Hyperledger Fabric), distinct validating nodes act as proposers (orderers) and executors (peers). These nuances can distort DoD estimates if proposal randomness is considered in isolation.

TABLE III
OVERVIEW OF THE IDENTIFIED CONSTRUCTS ASSOCIATED WITH THE BLOCKCHAIN TRILEMMA'S SUBCONCEPTS (I.E., DOD, SCALABILITY, AND SECURITY).

	Construct	**Description**
Degree of Decentralization	Block-Proposal Randomness	The degree to which it is uncertain which validating node will propose the next block.
	Geographical Diversity	The degree to which validating nodes in a blockchain system are located in different locations.
	Hashing Power Distribution	The extent to which hashing power is distributed among all validating nodes that compete to propose the next block.
	Token Concentration	The distribution of token shares that validating nodes hold in a blockchain system.
	Wealth Distribution	The degree of inequality between validating nodes in terms of token holding.
Scalability	Availability	The degree to which a blockchain system is operational and delivers consistent, timely responses.
	Confirmation Latency	The timespan from the proposal of new blocks to their confirmation.
	Maximum Possible Throughput	The highest number of transactions a blockchain system can process in a specified timeframe.
Security	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 consistently and correctly despite 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 main chain 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, and the physical or jurisdictional placement of validating nodes influences equitability in consensus participation and resilience against correlated risks (e.g., outages). Geographical diversity (GD) can be computed as [61]:

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

GD_{excl} denotes the dispersion measure when all validating nodes operate in a single location, GD_{equal} the measure when validating nodes are equally distributed across all available locations, and GD_{target} the measure for the observed distribution. These quantities are derived from an auxiliary dispersion term GD_{aux}:

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

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

with

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

where n is the total number of validating nodes, $|N|$ the number of possible locations, $|N_t|$ the number of occupied locations, and $|n_i|$ the number of validating nodes in location i.

A higher GD indicates a more even spread of validating nodes across locations, which corresponds to higher DoD: no single location disproportionately shapes network conditions or regulatory exposure influencing consensus.

Suppose n=100 validating nodes, $|N|$=10 possible locations, and $|N_t|$=4 occupied locations with 25 validating nodes each. Using equations (4) and (3), one obtains $GD \approx 0.250$. Distributing the same 100 validating nodes evenly across all ten locations would yield $GD = 1$.

The GD captures a key influence on DoD by discouraging dominance of a single location and improving regulatory neutrality. However, 'location' is ambiguous (e.g., geography, jurisdiction, or network domain), and the choice of discretization can materially affect results. Moreover, GD does not account for heterogeneous link bandwidths [1], latencies, or stake/hash within locations, so high GD need not imply equitable participation; conversely, low GD does not automatically imply low DoD if other equalizing mechanisms offset locality effects.

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

In blockchain systems using PoW-based consensus protocols, such as Bitcoin, validating nodes compete to produce the next block by computing hash values that satisfy a difficulty target. This process, called *mining*, favors validating nodes with greater hashing power, which increases their probability of successfully proposing a block.

Hashing power distribution is commonly estimated using the *Nakamoto coefficient*, defined as the minimum number of validating nodes that together control enough resources (e.g., hashing power) to exceed a compromise threshold [2], [10], [12]–[15], [70], [85]:

$$NC = \min\left\{k \in [1, \ldots, n] : \sum_{i=1}^{k} p_i \geq threshold\right\}, \tag{6}$$

where p_i denotes the resources (e.g., hashing power) controlled by validating node i. For PoW blockchain systems, a common threshold is 50%, as attackers must control more than half of the hashing power to perform a 51% attack [1], [14], [48].

TABLE IV
OVERVIEW OF INPUT VARIABLES OF METRICS THAT OPERATIONALIZE CONSTRUCTS OF THE BLOCKCHAIN TRILEMMA'S SUBCONCEPTS.

Input Variable	**Symbol**	**Description**	**Used in Equations**
Accumulated Resources	s	The amount of resources possessed by a validating node that an attacker needs to control in order to dominate the blockchain system.	19
Block Confirmation Time	$BConfTime$	The timestamp in milliseconds when a new block is (assumed to be) confirmed in a blockchain system.	12
Block Creation Interval	BCI	The average time in milliseconds between the proposal of consecutive blocks that are included in a blockchain.	13, 17
Block Gas Cost	$\|G_{cost}\|$	The computational resources (e.g., in gas) required to execute a transaction in a block.	17
Block Gas Limit	$\|G_{limit}\|$	The maximum amount of computational resources (e.g., in gas) available in a block to process a transaction.	17
Block-Proposal Time	$BPropTime$	The timestamp in milliseconds when a new block is proposed to a blockchain system.	12
Elapsed Time	t	The duration of quantifying the reliability of a blockchain system.	22
Epoch Length	e_l	The number of slots in an epoch in a blockchain system.	15
Epoch to Finality	e_f	The number of epochs required for a block to be considered finalized in a blockchain system.	14
Hashing Power	p	The number of resources (e.g., hashing power) used to produce a new block.	6
Number of Confirmed Blocks	$\|N_c\|$	The number of (probabilistically) finalized blocks stored in a blockchain system.	23
Number of Confirmed Transactions	$NumOfConfTr$	The total number of transactions processed and included in a block that has been (probabilistically) finalized into the main chain of a blockchain system.	11, 16
Number of Failures	$NumberOfFailures$	The number of failures in a blockchain system within a given observation timespan.	22
Number of Occupied Locations	$\|N_t\|$	The number of locations where validating nodes in a blockchain system operate.	4
Number of Possible Locations	$\|N\|$	The number of possible locations where validating nodes could operate.	4
Number of Proposed Blocks	b	The number of blocks proposed by validating nodes to a blockchain system.	1
Number of Subsequent Blocks	b'	The required minimum number of blocks that must be appended to a block to achieve sufficiently high probabilistic finality for that block.	13
Number of Stale Blocks	$NumberOfStaleBlocks$	The number of valid blocks that are proposed but eventually not included in the main chain.	23
Number of Tokens	τ	The number of tokens held by an individual validating node.	7, 9
Number of Transactions	$NumOfTr$	The total number of transactions issued to a blockchain system.	11
Number of Validating Nodes	n	The number of validating nodes in a blockchain system.	1, 5, 7, 6, 19, 9
Resource Cost	c	The monetary cost per unit of network resources (token or hashing power).	19
Threshold	t_h	The minimum amount of resources required to gain control of consensus.	19
Total Operational Time	$TotalOperationalTime$	The timespan a blockchain system operates correctly within a defined observation time.	21
Transaction Size	tx	The average size of a blockchain transaction in bytes.	17

A high Nakamoto coefficient indicates higher DoD, since many validating nodes must collude to surpass the threshold. Conversely, a low coefficient indicates that a small number of validating nodes control most hashing power, making the system more vulnerable to capture [2], [70]. In practice, large mining pools in the Bitcoin system collectively control a substantial share of total hashing power; when the top k pools together exceed 50%, equation 6 yields a Nakamoto coefficient of k, indicating lower DoD. Point-in-time figures fluctuate, so k should be computed from contemporaneous measurements.

While the Nakamoto coefficient provides a straightforward measure of compromise resistance, it has limitations. It captures only the minimum set of validating nodes required to exceed the threshold and neglects the full distribution of resources. In a blockchain system where hashing power is evenly distributed, the Nakamoto coefficient may be large but will not reflect finer differences in distribution. Moreover, it does not consider factors such as the network position of a validating, bandwidth, or latency, which can influence the actual feasibility of attacks [48], [50].

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

In many public blockchain systems, including Bitcoin and Ethereum, validating nodes are incentivized to participate in consensus finding through token-based rewards. When a validating node successfully proposes a block that is finalized, it receives a predefined token reward. In systems where leaders are selected according to their staked tokens (e.g., BitShares [43] and Cosmos Tendermint [89]), token holding directly influences the probability of proposing the next block. In both settings, the concentration of token holding provides insight into DoD: when a few validating nodes hold large token shares, they can disproportionately affect consensus finding.

Token concentration is typically measured using the *Herfindahl–Hirschman Index* (HHI) [2], [10], [12], [13], [15], [70], a standard economic indicator of concentration [90], [91]:

$$HHI_{prop} = \sum_{i=1}^{n} \left(\frac{\tau_i}{\tau_{total}} \right)^2, \tag{7}$$

where $HHI_{prop} \in \left[\frac{1}{n}, 1\right]$ uses *proportional* shares (no scaling).

Competition practice often reports $HHI_{10k} = 10{,}000 \times HHI_{prop}$. For comparability across systems with different n, a common normalization is

$$HHI_{\text{norm}} = \frac{HHI_{prop} - \frac{1}{n}}{1 - \frac{1}{n}} = \frac{n}{n-1} HHI_{prop} - \frac{1}{n-1}, \tag{8}$$

which maps equal shares to 0 and monopoly to 1. If using the 10,000-scaled variant, apply $HHI_{prop} = HHI_{10k}/10{,}000$ in (8).

A high HHI signals that a small number of validating nodes control most token shares (low DoD); a low HHI indicates more even dispersion (high DoD). For example, with eleven validating nodes each holding five tokens (total 55), $HHI_{prop} = 1/11 \approx 0.0909$, so $HHI_{10k} \approx 909$ and $HHI_{\text{norm}} = 0$. By contrast, if one node holds 37 tokens and ten hold 1.8 each (still 55 total), then $HHI_{prop} \approx 0.4623$, $HHI_{10k} \approx 4623$, and $HHI_{\text{norm}} \approx (0.4623 - 0.0909)/(1 - 0.0909) \approx 0.408$, reflecting lower DoD.

Although straightforward, HHI has limitations. It assumes a one-to-one mapping between validating nodes and token holding, which may not hold if a single entity controls multiple validating nodes (e.g., through Sybil strategies [92]). This can obscure the true concentration of influence. Furthermore, HHI does not differentiate between tokens staked for consensus participation and tokens held for other purposes. As a result, while HHI provides a useful estimate of token concentration, it should be interpreted with caution and ideally complemented with additional information on holding structures and staking activity.

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

In many public blockchain systems, such as Bitcoin and Ethereum, validating nodes are rewarded with tokens for successful block proposals. Over time, these rewards accumulate unevenly, leading to differences in token holding among validating nodes. Wealth distribution, therefore, reflects how equitably rewards are spread across the validating node set and, by extension, how evenly consensus influence is distributed in practice.

Wealth distribution is commonly measured using the *Gini coefficient* [2], [10], [12]–[15], [62], [70], [85], [87], a standard inequality metric [93]–[95]. It is defined as:

$$Gini = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |\tau_i - \tau_j|}{2n \sum_{i=1}^{n} \tau_i}, \tag{9}$$

τ_i and τ_j denote the token holdings of validating nodes i and j, respectively. The Gini coefficient ranges from 0 (perfect equality, all validating nodes hold the same number of tokens) to 1 (perfect inequality, one validating node holds all tokens). A higher value indicates stronger inequality in token holdings and thus a lower DoD.

The normalized $Gini$ is

$$\text{Gini}_{\text{norm}} = Gini \cdot \frac{n}{n-1}, \tag{10}$$

which lies between $[0, 1]$, 0 means equal token holdings, and near 1 means highly unequal.

For illustration, suppose five validating nodes $\{n_1, \ldots, n_5\}$ hold tokens as follows: $n_1 = 2$, $n_2 = 3$, $n_3 = 5$, $n_4 = 6$, $n_5 = 2$. Substituting into equation 9 yields $Gini \approx 0.244$, suggesting relatively equal wealth distribution and therefore high DoD. By contrast, if one validating node holds all 18 tokens and the others none, $Gini = 0.8$, with $\text{Gini}_{\text{norm}} \approx 0.8 \cdot \frac{5}{5-1} \approx 1$ reflecting low DoD.

Wealth distribution and token concentration measure related but distinct aspects of DoD. Token concentration (via HHI) captures how strongly token holding is concentrated in a few validating nodes (e.g., stake shares), while wealth distribution (via $Gini$) captures the overall degree of inequality across all validating nodes. A system may exhibit moderate concentration (e.g., several validating nodes with large equal stakes) yet still display high inequality, or vice versa. Considering both measures together provides a more comprehensive view of how token holding affects DoD.

Despite its usefulness, the Gini coefficient also has limitations. It is assumed that token holding directly translates into consensus influence, which may not hold in systems where only staked tokens matter (e.g., stake-weighted PoS). It further ignores social factors such as one entity operating multiple validating nodes or external determinants like bandwidth and reputation that affect consensus finding [1], [74]. For accurate application, Gini-based estimates should be contextualized with knowledge of system design and governance.

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

Availability: *The degree to which a blockchain system is operational and delivers consistent, timely responses.*

Following prior benchmarks, we operationalize availability as a scalability construct because it captures whether the system sustains responsiveness under varying workloads. We acknowledge, however, that availability is also often treated as a security property in fault-tolerance literature. In the context of scalability, availability reflects whether validating nodes remain responsive in processing transactions and maintaining access to the replicated ledger under varying workloads. A highly available blockchain system can sustain operation during validating node crashes or message losses and continue to provide service at scale, ensuring that increasing demand does not significantly degrade system responsiveness.

Availability is often measured using the ratio of confirmed transactions $NumOfConfTr$ to issued transactions $NumOfTr$ over a given observation period [8]:

$$Availability = \frac{NumOfConfTr}{NumOfTr}. \tag{11}$$

A higher ratio indicates that the system successfully processes most issued transactions, reflecting both scalability and robustness. For example, in a blockchain system under high workload, if 950 out of 1,000 issued transactions are confirmed, availability is 0.95.

This metric is straightforward to interpret and directly links to user experience: a system that consistently confirms issued transactions is both available and scalable. However, it abstracts away important nuances. First, it does not consider confirmation latency: transactions may eventually be confirmed, but only after a significant delay, which reduces practical scalability. Second, availability is sensitive to network partitions and failures. If partitions prevent transactions from propagating, availability will appear low, even though local subsystems may still function. Third, the metric does not distinguish between temporary and persistent failures; both reduce the ratio equally, although their implications for scalability differ. Thus, while availability provides a useful first-order estimate of scalability, it should be complemented with other constructs, such as confirmation latency and throughput, to capture the full picture.

Confirmation Latency: *The timespan from the proposal of new blocks to their confirmation.*

When new blocks are proposed, validating nodes must process them and decide whether to include them permanently. In blockchain systems with probabilistic finality, a block b is considered confirmed when a sufficient number of additional blocks are appended on top of it. The more subsequent blocks are added, the lower the probability that b will be excluded from the main chain. For example, in Nakamoto consensus [23], at least six additional blocks are typically required before a block is regarded as confirmed. In contrast, systems with immediate finality, such as those using PBFT, finalize a block once at least two-thirds of validating nodes accept it [52]. In such systems, confirmation latency is network-bound at the protocol level–bounded by a small number of message rounds, rather than dependent on a variable number of subsequent blocks.

Equation 12 gives a simple estimate of confirmation latency $\mathrm{CL}_{\mathrm{emp}}$ in systems with either immediate or probabilistic finality, usually through direct empirical measurement (e.g., [6], [16], [96], [97]):

$$\mathrm{CL}_{\mathrm{emp}} \;=\; BConfTime - BPropTime. \tag{12}$$

In probabilistic-finality systems, confirmations matter most because they secure the block against forks. Equation 13 calculates confirmation latency $\mathrm{CL}_{\mathrm{conf}}$ using the number of security confirmations b' required and the average block creation interval BCI [81]:

$$\mathrm{CL}_{\mathrm{conf}} \;=\; b' \times BCI. \tag{13}$$

Equation 14 models confirmation latency in the Ethereum system, where finality requires that a block be justified and then finalized in the next epoch [45]

$$\mathrm{CL}_{\mathrm{epoch}} \;=\; e_f \times TTJ, \tag{14}$$

with

$$TTJ \;=\; e_l \times \mathit{slot\_time}, \tag{15}$$

where e_f is the number of epochs until finalization, e_l is the number of slots per epoch, and $\mathit{slot\_time}$ is the duration of a slot (12 seconds in the Ethereum system).

High confirmation latency typically indicates lower throughput and thus reduced scalability, while low latency reflects faster processing and higher scalability. Equation 12 can also be adapted to immediate-finality systems by replacing $BConfTime$ with the exact $BlockFinalizationTime$.

Using equation 12, suppose a block is proposed at timestamp $1,755,428,614,000$ ms and finalized at $1,755,429,214,000$ ms. The resulting confirmation latency is 600,000 ms (10 minutes).

For equation 13, Bitcoin typically requires six confirmations [98]. If average block creation intervals vary between 93 and 993 seconds, the resulting confirmation latency ranges from 558 to 5,958 seconds, showing how both b' and BCI influence scalability.

For the Ethereum system, equation 14 with $e_f = 2$, $e_l = 32$, and $\mathit{slot\_time} = 12$ s yields $\mathrm{CL}_{\mathrm{epoch}} = 768$ seconds (12.8 minutes).

All three metrics capture the time to confirmation but ignore how many transactions are included per block. Block size, therefore, interacts with confirmation latency: larger blocks may contain more transactions but also prolong the time until confirmation [99], [100]. Hence, confirmation latency alone cannot fully represent scalability.

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

Transaction processing in blockchain systems involves propagating transactions, validating them, batching valid transactions into blocks, and appending those blocks to the main chain. Depending on the consensus/finality model, blocks are either finalized immediately (deterministic finality) or become irreversible after sufficient confirmations (probabilistic/economic finality). Throughput is therefore a key indicator of scalability.

First, the empirical measure computes realized throughput over an observation window:

$$\mathrm{TPS}_{\mathrm{emp}} \;=\; \frac{\mathrm{NumOfConfTr}}{t_1 - t_0}\,, \tag{16}$$

where $\mathrm{NumOfConfTr}$ is the number of transactions that meet the system's notion of 'committed' in $[t_0, t_1]$ (e.g., finalized for immediate finality; k-confirmed for probabilistic finality).[3] For example, in Bitcoin, five consecutive blocks containing $16{,}457$ transactions over $2{,}025$ seconds yield $\mathrm{TPS}_{\mathrm{emp}} \approx 8$.

A complementary, capacity-bound upper limit follows from per-block constraints and the block interval. Let BCI be the block creation interval and

$$N_B \;=\; \min\!\left(\left\lfloor \frac{|G_{limit}|}{|G_{cost}|} \right\rfloor,\; \left\lfloor \frac{B_{\mathrm{byte}}}{s_{\mathrm{tx}}} \right\rfloor \right), \tag{17}$$

be the maximum number of transactions that can fit in a block, where $|G_{limit}|$ is the per-block gas limit (if applicable), $|G_{cost}|$ the average gas cost per transaction, B_{byte} an optional per-block byte-size limit, and s_{tx} the average transaction size in bytes. Under saturated load (enough ready transactions at each block boundary), the gas/byte-bound throughput is

$$\mathrm{TPS}_{\mathrm{cap}} \;=\; \frac{N_B}{BCI}\,. \tag{18}$$

[3] Analysts should align "committed" with the chain's finality model and report k where applicable. Ensure $t_1 - t_0$ is in seconds to yield TPS.

If the arrival rate λ of ready transactions is lower than $\mathrm{TPS}_{\mathrm{cap}}$, realized throughput is supply-limited: $\mathrm{TPS} \leq \min(\mathrm{TPS}_{\mathrm{cap}}, \lambda)$. Mempool *capacity* affects queuing and drops, not the per-block packing limit N_B.

Illustratively, on a gas-limited chain with $|G_{limit}| =$ 45,000,000 gas, value-transfer $|G_{cost}| =$ 21,000 gas, and $BCI = 13\,\mathrm{s}$, one has $N_B \approx$ 2,143 and $\mathrm{TPS}_{\mathrm{cap}} \approx 165$. Realized $\mathrm{TPS}_{\mathrm{emp}}$ is typically below this bound due to headers/consensus overheads, empty space, propagation/validation delays, and workload mix.

Both formulations are informative but incomplete on their own. Neither explicitly models the effect of the number of validating nodes on communication complexity (which can increase latency and reduce realized throughput), and both ignore resources consumed by transactions that never finalize. Throughput should therefore be interpreted alongside latency and failure-mode metrics for a fuller picture of scalability.

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

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

Blockchain systems are vulnerable to different attacks (e.g., 51% attacks and selfish mining) that allow adversaries to (temporarily) dominate consensus. Such attacks exploit characteristics of probabilistic finality in blockchain systems and differ in their resource requirements. Public-permissionless blockchain systems like Bitcoin and Ethereum are especially targeted because attacks can be attempted by any participant. A high cost of attack discourages Byzantine behavior by making it economically unprofitable or infeasible, whereas a low cost reduces the barrier to launching attacks.

Equation 19 offers an indicator of the vulnerability of blockchain systems for specific attacks by calculating the cost of attack CoA [14]:

$$CoA \;=\; t_h \times c \times \sum_{i=1}^{n} s_i, \tag{19}$$

t_h denotes the minimum fraction of network resources (e.g., 50% of hashing power or one-third of stake) required to gain control of the consensus. c denotes the monetary cost per unit of resource, and s_i is the amount of resources controlled by validating node i.

High CoA values indicate that attackers must invest large sums to compromise consensus, enhancing security. Conversely, low CoA suggests that consensus can be compromised with modest resources, reducing security.

This metric applies to blockchain systems with PoW-based leader election (e.g., Bitcoin) and to blockchain systems with PoS-based consensus (e.g., Ethereum). In the Bitcoin system, for example, an attacker would need to control at least 51% of the system's hashing power. As a rough order-of-magnitude illustration, if total network hash rate were about 781.25 EH/s [101] and one assumed ASIC devices delivering 234 TH/s at USD 6,339 per unit, acquiring $\approx 0.51 \times \frac{781{,}250{,}000\ TH/s}{234\ TH/s} \approx 1.7$ million devices would alone cost ~USD 10–11B, excluding energy and infrastructure. In the Ethereum system, control over one-third of validating nodes (at 32 ETH per validating node) would entail costs on the order of tens of billions of USD at prevailing ETH prices [102], [103]. These magnitudes make such attacks economically prohibitive in typical conditions.

Equation 19 provides only a rough estimate for one chosen attack vector. A comprehensive security assessment would require comparing the minimum cost across all possible attacks. Moreover, the metric neglects the severity of attacks and assumes that adversaries purchase existing resources rather than adding new ones.

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

Blockchain systems are subject to faults such as crashes, omission, or Byzantine behavior of validating nodes [46], [47]. Consensus protocols are designed to tolerate some level of faults: for example, Raft handles crash faults, whereas PBFT tolerates Byzantine faults.

Equation 20 quantifies fault tolerance FT as the performance degradation observed during faults [6], [7], [78]:

$$FT \;=\; \{\Delta ThroughputDiff, \Delta ConfLatDiff\}, \tag{20}$$

$ThroughputDiff = |Throughput_N - Throughput_F|$ is the drop in maximum possible throughput under failures. $ConfLatDiff = |ConfLat_N - ConfLat_F|$ is the increase in confirmation latency under failures. Normal-operation values are computed using equations 16 and 12, while fault-operation values reflect conditions with failed validating nodes.

High fault tolerance means that throughput and confirmation latency remain stable despite faults, indicating strong security. For example, in a system with 10 transactions/s throughput and 5 s latency under normal conditions, a crash fault reducing throughput to 8 transactions/s and increasing latency to 10 s yields $FT = \{2, 5\}$.

Equation 20 captures performance degradation but not all security-relevant faults. Byzantine misbehavior, such as double spending or selfish mining, may succeed without significant changes in throughput or confirmation latency. Furthermore, fault tolerance is theoretically bounded (e.g., one-third of validating nodes in Byzantine fault-tolerant protocols), which may diverge from empirical system behavior. To fully assess trade-offs, blockchain systems with different consensus protocols and fault tolerance guarantees must be compared.

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

Blockchain systems replicate data across validating nodes, often in high-stakes contexts such as finance [104], [105]. Reliability reflects the probability that the system continues to operate correctly without interruption.

We assume exponentially distributed inter-failure times (homogeneous Poisson failures): a constant hazard rate λ, so $R(t) = \Pr[T > t] = e^{-\lambda t}$ with $\lambda = 1/\mathrm{MTBF}$.

Equation 21 expresses reliability $R(t)$ as a function of the mean time between failures (MTBF):

$$R(t) \;=\; e^{-\,t/\mathrm{MTBF}}. \tag{21}$$

Here, $R(t)$ denotes the probability that the next failure occurs after time t. We estimate MTBF from observations as

$$\mathrm{MTBF} = \frac{\mathrm{TotalOperationalTime}}{\mathrm{NumberOfFailures}}. \tag{22}$$

High reliability indicates that failures are rare, enhancing security. Low reliability signals frequent failures and reduced security.

If two failures totaling 30 minutes occur in a year (525,600 minutes), then $\mathrm{MTBF} = (525{,}600 - 30)/2 = 262{,}785$ minutes. The probability of experiencing no failure over the full year is $R(525{,}600) = e^{-525{,}600/262{,}785} \approx e^{-2.00} \approx 13.5\%$. Equivalently, there is an $\approx 86.5\%$ probability that at least one failure occurs within that year.

A limitation is that analysts must define which failure types to count, which affects comparability across studies. Moreover, the exponential assumption implies a constant hazard; if failures exhibit aging/correlation, $R(t)$ will deviate from (21). When repair time is non-negligible, reporting steady-state availability $A_\infty = \mathrm{MTBF}/(\mathrm{MTBF} + \mathrm{MTTR})$ alongside $R(t)$ is recommended.

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

In blockchain systems with probabilistic finality, not all blocks proposed by validating nodes are included in the main chain. When multiple blocks are proposed concurrently, only one may be finalized, and the others become stale blocks [48]. Frequent stale blocks can indicate network partitions, which facilitate attacks such as double spending or selfish mining [48], [50].

Equation 23 calculates the stale block rate SBR [80], [81]:

$$SBR = \frac{NumberOfStaleBlocks}{NumberOfConfirmedBlocks}. \tag{23}$$

A higher stale block rate may increase the potential for forks and chain reorganizations, which can reduce security in probabilistic-finality blockchain systems. Conversely, a low rate suggests fewer opportunities for adversaries to exploit such conditions, although actual exploitability depends on network conditions and attacker resources. For example, if a system with 879,320 confirmed blocks records 2 stale blocks, then $SBR \approx 2.27 \times 10^{-6}$, indicating a very low security risk from stale blocks.

$NumberOfStaleBlocks$ denotes the number of valid blocks proposed but not included in the mainchain. The total number of blocks recorded on the blockchain is denoted by $NumberOfConfirmedBlocks$. The metric is simple but backward-looking, measuring past stale blocks without directly predicting attack feasibility. It also neglects system-specific factors (e.g., block size, block creation interval) that influence stale block propagation [48]. Moreover, the severity of attacks facilitated by stale blocks is not reflected: in Bitcoin, for instance, rewriting history requires producing a longer chain, which is computationally intensive despite occasional stale blocks [49], [50].

B. Overview of Selected Analysis Approaches for Investigating the Blockchain Trilemma's Subconcepts and Examples of Blockchain Systems Analyzed Using the Approaches

Several analysis approaches operationalize at least two subconcepts of the blockchain trilemma, particularly in Bitcoin- and Ethereum-class systems (e.g., [2], [14]). Table V provides an overview of selected approaches– including benchmarks, simulators, and empirical studies– and highlights the constructs and metrics they employ to assess DoD, scalability, and security.

Analysis approaches that addresses multiple subconcepts most commonly use maximum possible throughput (eq. 16) to measure scalability; hashing power distribution (equation 6) and wealth distribution (equation 9) to assess DoD; and, for security, cost of attack (equation 19), fault tolerance (equation 20), or stale block rate (equation 23).

V. Discussion

The diversity of constructs and their operationalizations makes it difficult to decide which are most suitable for identifying design trade-offs under the blockchain trilemma. To address this challenge, we systematically reviewed the literature and assessed how constructs have been defined, measured, and applied. This section summarizes the main findings, highlights contributions to research and practice, outlines limitations, and points to promising future research directions.

A. Principal Findings

Constructs for assessing scalability are relatively straightforward, with well-defined metrics and evaluation methods. Three constructs are common, with *maximum possible throughput* and *confirmation latency* used most frequently. Metric suitability depends on the finality model and workload; for example, equation 13 is appropriate for probabilistic finality.

Security encompasses several interrelated constructs that together describe a system's robustness under adversarial or fault conditions. No single measure captures all aspects of security. Empirical measurements typically combine multiple indicators, each reflecting a critical dimension of resilience. Among the four identified constructs, *fault tolerance* and *stale block rate* appear most frequently. Notably, *reliability*—central in general software engineering—and *availability* (often treated under scalability in blockchain benchmarks) receive comparatively little attention in security-focused analyses, which is a promising direction for further work.

Measuring DoD is challenging for different reasons. Unlike scalability and security, DoD lacks an established and clear conceptual foundation. Core ideas such as autonomy and equity of participants are broad and highly context-dependent. Existing constructs either focus on economic dimensions (e.g., *token concentration*, *wealth distribution*) or technical participation opportunities (e.g., *block-proposal randomness*). While each captures an important facet, they are often treated in isolation.

TABLE V
SELECTION OF ANALYSIS APPROACHES FOR INVESTIGATING THE BLOCKCHAIN TRILEMMA'S SUBCONCEPTS AND EXAMPLES OF BLOCKCHAIN SYSTEMS ANALYZED USING THE APPROACHES. EQUATION REFERENCES CORRESPOND TO DEFINITIONS IN SECTION IV-A. THE MAPPING OF CONSTRUCTS TO APPROACHES IS BASED ON THE REPORTED APPROACH; CONCRETE METRICS USED IN SPECIFIC EXPERIMENTS MAY VARY BY CONFIGURATION OR LOAD.

	Constructs and Metrics Used to Analyze the Blockchain Trilemma's Subconcepts			
Analysis Approach	**DoD**	**Scalability**	**Security**	**Analyzed Blockchain Systems**
BBSF [6]		Confirmation latency (eq. 12), Maximum possible throughput (eq. 16)	Fault tolerance (eq. 20)	Ethereum, Quorum
BLOCKBENCH [7]		Confirmation latency (eq. 12), Maximum possible throughput (eq. 16)	Fault tolerance (eq. 20)	Ethereum, Hyperledger Fabric
BlockSim [106]		Confirmation latency (eq. 12), Maximum possible throughput (eq. 16)	Stale block rate (eq. 23)	Bitcoin, Ethereum
Diablo [8]		Confirmation latency (eq. 12), Maximum possible throughput (eq. 16), Availability (eq. 11)	Fault tolerance (eq. 20)	Algorand, Avalanche, Ethereum, Hyperledger Fabric, Red Belly, Solana
Fu et al. [70]	Token concentration (eq. 7), Hashing power distribution (eq. 6), Wealth distribution (eq. 9)	Confirmation latency (eq. 12), Maximum possible throughput (eq. 16)		Algorand, Ethereum
SimBlock [2], [9]	Token concentration (eq. 7), Hashing power distribution (eq. 6), Wealth distribution (eq. 9)	Maximum possible throughput (eq. 16)	Stale block rate (eq. 23)	Bitcoin
Quattrocchi et al. [14]	Hashing power distribution (eq. 6), Wealth distribution (eq. 9)	Maximum possible throughput (eq. 16)	Cost of attack (eq. 19)	Bitcoin, Cardano, Ethereum, Polygon, Solana
Thakkar et al. [107]		Confirmation latency (eq. 12), Maximum possible throughput (eq. 16)	Fault tolerance (eq. 20)	Hyperledger Fabric
Gräbe, et al. [81]		Confirmation latency (eq. 13), Maximum possible throughput (eq. 16)	Fault tolerance (eq. 20), Reliability (eq. 21)	Ethereum, Hyperledger Indy, Tezos
TezBed [5]	Block-proposal randomness (eq. 1), Token concentration (eq. 7), Wealth distribution (eq. 9)	Confirmation latency (eq. 13), Maximum possible throughput (eq. 16)		Tezos

eq.: equation

To the best of our knowledge, no operationalized construct provides a comprehensive sociotechnical measure of DoD, highlighting the need for deeper theoretical foundations.

Interrelationships between constructs are uneven. Some pairs, such as throughput and fault tolerance, show clear trade-offs, whereas others, such as geographical diversity and cost of attack, appear weakly coupled in observed studies. Practitioners should therefore select construct tuples deliberately to surface meaningful tensions across subconcepts; otherwise, analyses may overstate apparent optimality.

B. Contributions

This study makes three main contributions to research and practice. First, it synthesizes the literature on constructs and their operationalization, explaining their applicability, interpretability, and limitations for evaluating the blockchain trilemma. By clarifying definitions, aligning constructs with metrics, and highlighting interrelationships, we contribute a coherent theoretical framework that structures the trilemma into measurable subconcepts. This framework provides a foundation for more rigorous benchmarking and comparative analysis. For example, practitioners can use the synthesis to select construct tuples that reveal design trade-offs, while researchers can employ the framework to position empirical results within a shared conceptual space.

Second, by explaining the metrics in detail and defining their input variables, this work offers a foundation for systematic benchmarks. The input variables (e.g., number of validating nodes, number of confirmed transactions) point directly to the data that must be collected, thereby supporting both measurement and experimental design. This also facilitates planning experiments by clarifying what blockchain system characteristics (e.g., maximum block size) could be manipulated to investigate system behaviors in focus (e.g., maximum possible throughput).

Third, by comparing analysis approaches in terms of their constructs and metrics, we provide a basis for tailoring and refining existing methods. Practitioners can draw on this overview to adapt approaches to their specific systems, while researchers can use it as a starting point for methodological innovation. Moreover, the comparison of analysis approaches shows opportunities for development of more advanced analysis approaches that cover subconcepts and their interrelationships in more detail.

C. Limitations

The scope of this study is bounded by conceptual harmonization rather than empirical validation. The definitions of DoD, scalability, and security synthesize multiple perspectives into constructs, but they are abstractions derived from prior literature and may omit system-specific nuances, especially in architectures that decouple execution, data availability, and consensus. As a result, the proposed boundaries should be read as a practical lens for measurement, not as an exhaustive theory of the blockchain trilemma.

The metric catalog presents operationalizations developed under heterogeneous assumptions, workloads, and system models. Consequently, convergent validity (that different metrics for the same construct behave consistently) and discriminant validity (that metrics for different constructs do not capture the same signal) are not established here. Normalization choices (e.g., entropy and HHI scaling) also influence interpretation across systems with different numbers of validating nodes; while these choices improve comparability, they introduce sensitivity to distributional tails and sampling windows (e.g., [5]).

Data and sampling choices may bias the synthesis. We focused on peer-reviewed, English-language publications indexed in ACM, IEEE, ScienceDirect, and Scopus at a fixed collection date. This focus supports rigor but may underrepresent recent preprints, system documentation, or non-English venues, and it may overweight research prototypes relative to production deployments.

This work does not benchmark systems nor estimate metric values from live networks. Examples are illustrative and intended to clarify operationalization, not to claim external validity for any specific platform or configuration. Consequently, any comparative application of the catalogue requires careful alignment of workloads, network conditions, and threat models. Finally, while parallels to non-blockchain replicated databases are conceptually justified by shared consensus and state-machine properties, our synthesis is grounded in blockchain literature. Such generalization should therefore be interpreted cautiously unless underlying assumptions about faults, partitions, and participation demonstrably align.

D. Open Research Challenges

While this work aims to support in-depth analyses of the blockchain trilemma, several open challenges remain, offering opportunities for future research.

a) Utility of Constructs and Metrics.: Existing constructs capture only selected aspects of each blockchain trilemma subconcept. More integrative approaches are needed, such as composite indices that combine fault tolerance, stale block rate, and reliability to represent security more comprehensively. Similarly, DoD constructs should be extended to incorporate social factors, such as relationships among validating node operators and potential collusion. Future research should also validate construct tuples through controlled benchmarks that explicitly test interrelationships, ideally aligned with formal impossibility and lower-bound results (e.g., CAP and FLP [71], [108]).

b) Impact of Additional Software Layers.: Layer-2 solutions and peripheral artifacts, such as sidechains and state channels, fundamentally alter the trade-offs described by the blockchain trilemma. Studying such extensions may require new or adapted constructs. Here, non-peer-reviewed sources such as white papers and community documentation may play an important role.

c) Beyond Blockchain Systems.: The tensions described by the blockchain trilemma are not unique to blockchain systems. Other replicated database systems that rely on consensus and state machine replication face similar challenges. Extending the analysis to these domains may enrich theoretical understanding and could inform refinements or analogies to existing distributed systems theorems, such as CAP [71].

d) Sociotechnical Dimensions and Broader Systems.: The identified constructs primarily capture technical or economic properties, such as hashing power distribution or token concentration, yet the blockchain trilemma is inherently sociotechnical. Relationships among validating node operators, governance structures, and organizational affiliations can shape consensus participation and enable covert collusion, undermining decentralization despite favorable technical indicators. Similar tensions arise in other distributed infrastructures, such as federated learning, decentralized identity, and collaborative data-sharing systems, where consensus and replication interact with human incentives and governance. Extending the trilemma to these contexts could yield a generalized framework for distributed coordination that complements existing theorems like CAP by explicitly incorporating human and organizational factors.

VI. Conclusion

This study synthesizes 12 constructs, operationalized through 15 metrics, to quantify oD, scalability, and security in blockchain systems. By clarifying applicability, inputs, and limitations, the synthesis helps practitioners assemble construct tuples that surface meaningful trade-offs and supports researchers in designing more rigorous, comparable analyses.

A central insight is that practice primarily captures technical and economic facets, while sociotechnical influences (e.g., validating-node ownership/affiliations, governance participation, and potential collusion) remain underrepresented. Extending DoD and security constructs to make these concepts better observable is a promising direction, as is combining latency/throughput with reliability and availability for a fuller view of security and performance.

The tensions we describe are not unique to blockchains: they arise in consensus-based replicated databases more broadly. Applying and adapting these constructs beyond blockchain may strengthen connections to established results (e.g., CAP and FLP) and sharpen expectations about what can–and cannot–be optimized simultaneously.

Overall, this work provides a common vocabulary, explicit metric formulations with inputs, and a map of analysis approaches that together enable more reproducible, comparable evaluations and more purposeful system design under the blockchain trilemma.

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] F. von Normann, M. Aliyu, N. Kannengießer, L. B. Q. Le, R. Heinrich, and A. Sunyaev, “On the trade-off between degree of decentralization and scalability in proof-of-stake–based blockchain systems,” in *2025 7th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, 2025.

[6] 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.

[7] 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.

[8] 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.

[9] Y. Aoki, K. Otsuki, T. Kaneko, R. Banno, and K. Shudo, “Simblock: A blockchain network simulator,” in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 325–329.

[10] N. Chemaya, L. W. Cong, E. Jorgensen, D. Liu, and L. Zhang, “A dataset of uniswap daily transaction indices by network,” *Scientific Data*, vol. 12, no. 1, p. 93, 2025.

[11] 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.

[12] T. Yan, S. Li, B. Kraner, L. Zhang, and C. J. Tessone, “A data engineering framework for ethereum beacon chain rewards: From data collection to decentralization metrics,” *Scientific Data*, vol. 12, no. 1, p. 519, 2025.

[13] 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.

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

[15] C. Ovezik, D. Karakostas, M. Milad, D. W. Woods, and A. Kiayias, “Sok: measuring blockchain decentralization,” in *International Conference on Applied Cryptography and Network Security*, Munich, Germany, 2025, pp. 184–214.

[16] 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.

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

[18] 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

[19] 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.

[20] 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.

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

[22] J. Thompson, “A guide to abductive thematic analysis,” *The Qualitative Report*, vol. 27, no. 5, pp. 1410–1421, May 2022.

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

[24] A. Sakurai and K. Shudo, “Tie-breaking rule based on partial proof of work in a blockchain,” *IEEE Access*, 2024.

[25] ethereum.org, “Node architecture,” jul 2025. [Online]. Available: https://ethereum.org/en/developers/docs/nodes-and-clients/node-architecture/

[26] F. Solana, “Tower bft: Solana’s high performance implementation of pbft,” jul 2019. [Online]. Available: https://solana.com/de/news/tower-bft--solana-s-high-performance-implementation-of-pbft

[27] L. Polygon, “Architecture,” n.d. 2025. [Online]. Available: https://docs.polygon.technology/pos/architecture/overview/#bor-block-production-layer

[28] Hyperledger, “What’s new in hyperledger fabric,” n.d 2025. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/whatsnew.html

[29] 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.

[30] P. Bitcoin, “Developer guides,” n.d. 2020. [Online]. Available: https://developer.bitcoin.org/devguide/

[31] OSL, “What is bitcoin’s architecture?” jan 2025. [Online]. Available: https://www.osl.com/hk-en/academy/article/what-is-bitcoins-architecture

[32] T. Guggenberger, J. Sedlmeir, G. Fridgen, and A. Luckow, “An in-depth investigation of the performance characteristics of hyperledger fabric,” *Computers & Industrial Engineering*, vol. 173, p. 108716, 2022.

[33] I. Helius, “How to mitigate spam quickly: All you need to know about solana and quic,” sep 2023. [Online]. Available: https://www.helius.dev/blog/all-you-need-to-know-about-solana-and-quic

[34] J. Neu, E. N. Tas, and D. Tse, “Two more attacks on proof-of-stake ghost/ethereum,” in *Proceedings of the 2022 ACM Workshop on Developments in Consensus*, New York, NY, USA, 2022.

[35] 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.

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

[37] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, “Algorand: Scaling byzantine agreements for cryptocurrencies,” in *Proceedings of the 26th symposium on operating systems principles*, Shanghai, China, 2017, pp. 51–68.

[38] IOHK, “Cardano architecture,” n.d. 2025. [Online]. Available: https://docs.cardano.org/about-cardano/explore-more/cardano-architecture

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

[40] 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.

[41] 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.

[42] B. F. BitShares, “Introduction & architectures,” n.d. 2025. [Online]. Available: https://docs.bitshares.dev/en/latest/intro/architectures.html

[43] ioBanker and Abit, “Bitshares whitepaper,” oct 2023. [Online]. Available: https://github.com/bitshares/whitepaper

[44] T. Crain, C. Natoli, and V. Gramoli, “Red belly: A secure, fair and scalable open blockchain,” in *2021 IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, 2021, pp. 466–483.

[45] A. Asgaonkar, F. D’Amato, R. Saltini, L. Zanolini, and C. Zhang, “A confirmation rule for the ethereum consensus protocol,” Nov 2024. [Online]. Available: https://arxiv.org/abs/2405.005499

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

[47] 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.

[48] 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.

[49] 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: 10.1145/3212998

[50] Y. Sproll, R. Heinrich, L. B. Quang Le, and N. Kannengießer, “Sm-sim: A simulator for analyzing selfish mining attacks in blockchain systems,” in *2025 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2025, pp. 1–9.

[51] M. Esmaili and K. Christensen, “Performance modeling of public permissionless blockchains: A survey,” *ACM Computing Surveys*, vol. 57, no. 7, pp. 1–35, 2025.

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

[53] 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.

[54] 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.

[55] M. T. Özsu, P. Valduriez *et al.*, *Principles of distributed database systems*. Springer, 1999, vol. 2.

[56] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, "{ZooKeeper}: Wait-free coordination for internet-scale systems," in *2010 USENIX Annual Technical Conference (USENIX ATC 10)*, 2010.

[57] P. Ruan, T. T. A. Dinh, D. Loghin, M. Zhang, G. Chen, Q. Lin, and B. C. Ooi, "Blockchains vs. distributed databases: Dichotomy and fusion," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1504–1517.

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

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

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

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

[68] 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.

[69] 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.

[70] 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.

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

[72] 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.

[73] 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.

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

[75] 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.

[76] Hyperledger, "The ordering service," n.d 2025. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/orderer/ordering_service.html

[77] 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

[78] V. Gramoli, R. Guerraoui, A. Lebedev, and G. Voron, "Evaluating blockchain fault tolerance with stabl," in *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, 2025, pp. 182–183.

[79] 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

[80] 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.

[81] 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.

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

[87] 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.

[88] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

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

[90] 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

[91] S. A. Rhoades, "The herfindahl-hirschman index," *Fed. Res. Bull.*, vol. 79, p. 188, 1993.

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

[93] C. Gini, "Measurement of inequality of incomes," *The economic journal*, vol. 31, no. 121, pp. 124–125, 1921.

[94] R. Dorfman, "A formula for the gini coefficient," *The review of economics and statistics*, vol. 61, no. 1, pp. 146–149, 1979.

[95] C. Dagum, "A new approach to the decomposition of the gini income inequality ratio," *Empirical economics*, vol. 22, no. 4, pp. 515–531, 1997.

[96] 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.

[97] 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.

[98] 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

[99] 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.

[100] 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.

[101] 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

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

[103] 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

[104] D. Kirste, N. Kannengießer, R. Lamberty, and A. Sunyaev, “Automated market makers in cryptoeconomic systems: A taxonomy and archetypes,” vol. forthcoming, 2025. [Online]. Available: https://arxiv.org/abs/2309.12818

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

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

[107] 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.

[108] 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, p. 374–382, Apr. 1985.