PREStO: A Systematic Framework for Blockchain Consensus Protocols

Stefanos Leonardos, Daniël Reijsbergen, and Georgios Piliouras
Singapore University of Technology and Design

Abstract—The rapid evolution of blockchain technology has brought together stakeholders from fundamentally different backgrounds. The result is a diverse ecosystem, as exemplified by the development of a wide range of different blockchain protocols. This raises questions for decision and policy makers: How do different protocols compare? What are their trade-offs? Existing efforts to survey the area reveal a fragmented terminology and the lack of a unified framework to reason about the properties of blockchain protocols.

In this paper, we work towards bridging this gap. We present a five-dimensional design space with a modular structure in which protocols can be compared and understood. Based on these five axes – Optimality, Stability, Efficiency, Robustness and Persistence – we organize the properties of existing protocols in subcategories of increasing granularity. The result is a dynamic scheme – termed the PREStO framework – which aids the interaction between stakeholders of different backgrounds, including managers and investors, and which enables systematic reasoning about blockchain protocols. We illustrate its value by comparing existing protocols and identifying research challenges, hence making a first step towards understanding the blockchain ecosystem through a more comprehensive lens.

Index Terms—Consensus Protocols, Cryptocurrency, Survey, Incentives, Equilibrium

I. INTRODUCTION

In the seminal Bitcoin paper [148], the pseudonymous Satoshi Nakamoto pioneered the use of blockchains as a secure way of maintaining a ledger of currency transfers in a trustless peer-to-peer network. In the ten years since, blockchains have grown [63] to underpin a $100 billion cryptocurrency market [52]. Meanwhile, their applicability is increasingly understood in a broad range of other contexts [44], e.g., the Internet of Things [80], supply chain management [124], healthcare [144], etc. This rapid growth has induced a considerable number of established market parties to invest in the sector [62], [64], or even develop their own platforms. Noteworthy examples of the latter include Quorum [164], which is developed by JPMorgan Chase, and the HyperLedger umbrella project [39], hosted by the Linux Foundation and supported by, inter alia, IBM and Intel. Applications of Quorum include JPMorgan’s internal digital currency [173] and the Interbank Information Network [46], [153], a platform for cross-border money transfers. Applications of HyperLedger include a project by the US retailer Walmart to track the movement of vegetables [108], [182]. IBM by itself had 1500 employees working on 500 blockchain-related projects in September 2018 [88]. Meanwhile, new multipurpose blockchain platforms developed by startups continue to emerge, e.g., Ethereum [36], Cardano [41], [119], Algorand [32], [94], and Zilliqa [206], [207].

This proliferation of blockchain technologies and applications has brought together stakeholders with fundamentally different degrees of technical expertise. So far, the discourse between these groups has been marked by the use of sometimes incongruous terminology, and the lack of a unified communication framework [163]. This hampers the ability of managers and investors to make business decisions, and of newly proposed protocols to be compared and understood. Particularly affected are one of the most fundamental technical aspects of blockchain platforms: the consensus protocols.

Consensus protocols fulfill, in a decentralized setting, the role that a single authority has in a centralized database or ledger. It is the mechanism to reach agreement among self-interested peers, and for making consistent decisions out of mutually exclusive alternatives. The choice of consensus protocol has a major impact on a platform’s performance, including its security and throughput, and is therefore important for anyone who is involved in blockchain development [137], particularly executives. This can be challenging if the differences between the alternatives are not well-understood.

Statement of Contribution and Managerial Relevance

In this paper, we address these difficulties by developing an accessible, yet technical and comprehensive framework to improve the communication between the diverse participants of the blockchain ecosystem. We assume only a basic...
understanding of mathematics and the high-level idea behind blockchains, and introduce technical terms related to blockchains and cryptocurrencies from the bottom up.

Our main contribution is the PREStO framework [49], [201], which is a dynamic tool to identify and classify properties of blockchain protocols. It is an acronym (in reverse order) of PREStO’s modular structure sets it apart from related efforts and enhances its value for managers. Initially, the five categories can be seen as a nesting doll of design goals, where each category considers a wider range of desirable properties than the previous, cf. Figure 1. We start at the very basic – i.e., optimal performance under ideal conditions – and gradually build up to the more advanced – e.g., recovery mechanisms to survive in the long run. Subsequently, the axes are organized into subcategories of increasing granularity, and PREStO develops into a dynamic tool to identify and group together challenges and research opportunities for the various blockchain protocols, cf. Table VI. We demonstrate its practical use via two running use cases, Bitcoin and Quorum, and conclude with a schematic illustration of the resulting classification in Figure 6. Furthermore, we extensively draw from the existing literature to motivate our framework.

**A Growing Ecosystem**

The consensus protocol introduced by the first blockchain platform – Bitcoin – is commonly called Nakamoto consensus [185]. It was designed to work in a permissionless setting, i.e., a setting in which any node in the network is allowed to add data to the blockchain. To prevent network overflow, nodes who seek to extend the blockchain must spend computational effort through a process called mining. In the presence of competing chains, honest nodes accept the chain with the most effort spent on creating it. Together, these rules ensure that if more than 50% of the computational power is in the hands of honest parties, then their chain will grow faster than all others. Nodes are compensated for the spent computational power through rewards in the form of tokens logged on the blockchain. Variations of Nakamoto consensus are currently implemented in over 600 cryptocurrencies [52], [204], including the Ethereum platform and various Bitcoin spin-offs.

In recent years, Nakamoto consensus has increasingly drawn criticism for its low transaction throughput and high energy consumption. A single Bitcoin transaction costs more energy than 100,000 Visa transactions, and the Bitcoin network as a whole consumes as much energy as a medium-sized country [66]. Furthermore, it is insecure in the sense that smaller platforms are vulnerable to attackers who seize a majority of the computational power, as witnessed by the recent 51% attacks on Ethereum Classic [113] and Bitcoin Gold [168]. Finally, research [75], [93], [172], [204] has shown that Nakamoto consensus can be incentive-incompatible, i.e., participants can increase their rewards by deviating from the protocol. To address these weaknesses, a multitude of new consensus protocols have been proposed that more closely follow traditional theory on permissioned (i.e., not open) networks. In particular, many approaches use variations of Byzantine fault tolerant (BFT) protocols [127] or other classical consensus protocols such as Paxos [126] and Raft [155]. Such approaches can achieve gains in efficiency and security at the cost of centralization. However, a precise description of this trade-off is complicated due to the differences between BFT protocol implementations, and the lack of alignment between the terminology used by different parties. This motivates the need for a formal framework to describe and compare different consensus protocols.

**Related Work**

The necessity of developing a unified communication framework for the blockchain ecosystem was already acknowledged in [202]. Accordingly, a brief outline of the PREStO framework was first introduced in [49]. While the five main axes remain the same, their organization into subcategories is first deployed in the present paper.

The rapid growth of the blockchain-related literature has also stimulated other projects that survey the area from different perspectives. Focusing exclusively on the Bitcoin blockchain, [55] provide a systematic review of Bitcoin’s underlying features, particularly its security and privacy-related threats and vulnerabilities, and discuss directions for future research. Their analysis extends initial analyses of the backbone protocols of the main cryptocurrencies [84], [85], [87], [157]. In [185], further insight is provided into the development and functionality of the Bitcoin blockchain, in addition to a non-exhaustive, yet interesting timeline of papers related to the analysis of Nakamoto consensus.

In a spirit closer to the present study, [196] acknowledge the lack of a comprehensive literature review on the various layers of blockchain technology, and provide a rigorous vision on the organization of blockchain networks. Their work extends to all aspects of the relevant technology and provides a central reference for future work. They define four layers for any blockchain system, from top to bottom: (1) the application layer, (2) the virtual machine layer, (3) the consensus layer, and (4) the network layer. In the present study, we focus on the third (i.e., consensus) layer. That is, application-layer properties, virtual-machine-layer properties (e.g., secure smart contract languages such as Scilla [178]), and network-layer properties (e.g., vulnerability to eclipse [103], BGP hijacking [4], or DoS [115] attacks) are treated only if and when they affect the consensus layer.

**TABLE I: The five axes of PREStO and their main purpose.**

| Dimension | Description | Section |
|-----------|-------------|---------|
| Optimality | Does the protocol maximize the quality of its core outcomes under normal circumstances? | III |
| Stability | Is the designed protocol an equilibrium? | IV |
| Efficiency | How does the protocol utilize its different resources, e.g., time, space, energy, network bandwidth? | V |
| Robustness | Does the protocol’s performance withstand perturbations to its parameters? | VI |
| Persistence | If the protocol is forced out of equilibrium, does it recover? | VII |

PREStO’s modular structure sets it apart from related efforts and enhances its value for managers. Initially, the five categories can be seen as a nesting doll of design goals, where each category considers a wider range of desirable properties than the previous, cf. Figure 1. We start at the very basic – i.e., optimal performance under ideal conditions – and gradually build up to the more advanced – e.g., recovery mechanisms to survive in the long run. Subsequently, the axes are organized into subcategories of increasing granularity, and PREStO develops into a dynamic tool to identify and group together challenges and research opportunities for the various blockchain protocols, cf. Table VI. We demonstrate its practical use via two running use cases, Bitcoin and Quorum, and conclude with a schematic illustration of the resulting classification in Figure 6. Furthermore, we extensively draw from the existing literature to motivate our framework.
The difficulty to conceptualize the dramatically evolving design landscape of blockchains is further supported by [18]. Similar to the present work, they focus on the consensus layer and discuss the various themes and key approaches that are exhibited by current blockchains. They systematize distinctive features and technical properties of existing consensus protocols and provide thorough comparisons, open questions, and directions for future research. Despite the common perspectives, our approach distinguishes itself from [18] due to its mathematical framework that allows for a description of properties from the ground up.

Using a practice-oriented focus, [69] develop BLOCKBENCH, a promising and publicly available software program that is designed to test and compare the performance of blockchain protocols. It applies to private blockchains and its findings are mainly associated with properties in the categories of Optimality and Efficiency of the PREStO framework, cf. Sections III and V. The paper features use cases of the blockchain protocols and provide thorough comparisons, open questions, and directions for future research. Despite the common perspectives, our approach distinguishes itself from [18] due to its mathematical framework that allows for a description of properties from the ground up.

A list of the symbols introduced in this section is given in Table II. Using Figure 2 and Table III, which display the state of a Bitcoin-like blockchain network at a given time \( t^* \), and this network’s evolution until and shortly after \( t^* \), respectively. A list of the symbols introduced in this section is given in Table II.

TABLE II: List of symbols.

| Symbol | Meaning |
|--------|---------|
| \( B \in \mathbb{N} \) | A block |
| \( T(B) \) | Timestamp of block \( B \) |
| \( P(B) \) | Predecessor of block \( B \) |
| \( W(B) \) | Proof-of-work difficulty of block \( B \) |
| \( g \) | Genesis block |
| \( C(B) \) | Chain between block \( B \) and \( g \) |
| \( N \) | Node; set of active nodes |
| \( i, R \) | Resource; set of protocol resources |
| \( r_i^t \in \mathbb{R}^+ \) | Amount of resource \( i \) owned by node \( n \) at time \( t \) |
| \( R_i^t \in \mathbb{R}^+ \) | Total amount of resource \( i \) in the network at time \( t \) |
| \( P_{ni}^t \in [0,1] \) | Fraction of resource \( i \) owned by node \( n \) at time \( t \) |
| \( V_n(t) \) | View of node \( n \) at time \( t \) |
| \( M_n^t \) | Memory pool of node \( n \) at time \( t \) |
| \( H_n^t \) | Head of node \( n \) at time \( t \) |
| \( f \) | Fork-choice rule |
| \( x, \mathcal{X} \) | State; state space |
| \( s, S, \mathcal{I} \) | Strategy; strategy profile; space of all strategy profiles |
| \( d \) | Default or follow-the-protocol strategy |
| \( v_n(S, x) \) | Long-term utility of node \( n \) given strategy profile \( S \) and initial state \( x \) |
| \( F_{\Pi} \) | Feature/property of protocol \( \Pi \) |
| \( U_{\Pi} \) | Performance measure of protocol \( \Pi \) |

A. Blockchains as Data Structures

At its core, a blockchain is a data structure that contains a sequence of elementary database operations called transactions. The semantics of the transactions depend on the platform [58] (e.g., they can represent token transfers, smart contract calls, sensor readings in an IoT context, etc). The transactions are grouped into blocks. Each block not only contains transactions, but also a header that contains summary information about the block. The header of a block typically contains a reference to the block’s transactions, a timestamp, a reference to the previous block, and some additional platform-specific data cf. [30]. We assume that each block can be identified by a unique integer in \( \mathbb{N} \) (e.g., via the hash of its header).

In our model, we represent the block’s timestamp as a function \( T : \mathbb{N} \rightarrow \mathbb{R}^+ \), i.e., if a block \( B \) is created at time \( t \) then \( T(B) = t \). The previous block of any block \( B \) is also represented via a function \( P : \mathbb{N} \rightarrow \mathbb{N} \), i.e., \( B \) points to \( P(B) \) as its previous block. For brevity, we write \( P^2(B) = P(P(B)) \), \( P^3(B) = P(P(P(B))) \), etc. The first block \( g \) is called the genesis or the genesis block, and is the only block to not have a previous block, i.e., \( P(g) = \emptyset \). Using \( P \), we can construct for each block \( B \) a chain of blocks \( C(B) \) to the genesis, i.e., \( C(B) = (B, P(B), P^2(B), \ldots, g) \). The chain is cryptographically secure, i.e., the relationship between a block \( B \) and its predecessor \( P(B) \) is given by encapsulating all information in \( P(B) \) in \( B \) via a cryptographic hash function. Essentially, if even a single bit of data in \( P(B) \) is

1For example, via a Merkle tree root [143].
2In practice, the block creator has considerable freedom in choosing the timestamp [186].

Outline

The current paper is structured as follows. We begin by defining an abstract, high-level model of a blockchain consensus protocol in Section II. In Sections III to VII, we describe defining an abstract, high-level model of a blockchain consensus. We use cases of the Ethereum, Parity and Hyperledger blockchains and concludes that these systems are still far from large-scale adoption. Finally, a non-exhaustive list of related surveys with focal points ranging from smart contract execution to general blockchain applications and research perspectives includes [10], [11], [29], [40], [44], [56], [68], [86], [117], [179], [189], [200], [205].

II. A MATHEMATICAL MODEL FOR BLOCKCHAIN CONSENSUS

To rigorously define the properties covered by the PREStO framework, we require a mathematical model to serve as a basis. In this section, we will present this model by describing the data on the blockchain (in Section II-A), the participating nodes as individuals (in Section II-B), the nodes as a network (in Section II-C), and the rewards and strategies (in Section II-D). In all cases we denote the consensus protocol by \( \Pi \). To account for the wide variety of existing protocols and the diversity of their technical features, we keep our model as general as possible. However, we illustrate our presentation using two running examples from the permissionless and permissioned settings, respectively. The first is Bitcoin [148], which is both the first and the (to our knowledge) simplest implementation of a permissionless protocol. The second is Quorum [164], which uses the Istanbul-BFT consensus protocol [111] – we chose this protocol in favor of the BFT features in Hyper Ledger because the latter were not well-documented at the time of writing, and in favor of other BFT-based blockchain protocols such as Tendermint because of Quorum’s business focus. We also visualize the core concepts.
The blockchain DAG

### Example 1 (Bitcoin)
In Bitcoin, transactions represent token transfers between users. Transactions also include fees that are paid to block creators. Among the block header fields, the proof-of-work, \( W : \mathbb{N} \to \mathbb{R}^+ \) (as given via the difficulty and the nonce) is also relevant.

### Example 2 (Quorum)
Quorum is based on Ethereum, and hence the transactions not only represent token transfers, but also smart contract calls and creations. The protocol messages of Instanbul-BFT, e.g., prepare and commit messages, are also seen as transactions, even though only the commit messages are included in the block (yet they are not referred to in the header).

### B. Nodes and Resources
The blockchain protocol is operated by a set \( \mathbb{N} \) of agents called nodes, which are identified by their index \( n \in \mathbb{N} \). To participate, each node \( n \) provides each of \( m \) distinct resources. The amount of resource \( i \in \mathcal{R} := \{1, \ldots, m\} \) contributed by node \( n \) is denoted by \( r_{n,i} \in \mathbb{R}_+ \). Since resources change over time, we will write \( \bar{r}_{n,i} = (r_{n,i,1}, \ldots, r_{n,i,m}) \) for the vector of resources of node \( n \) at timepoint \( t > 0 \). Accordingly, let \( \bar{R}^t = (R_{1}^t, \ldots, R_{m}^t) := \sum_{n \in \mathbb{N}} \bar{r}_{n,i} \) be the vector of total protocol resources at timepoint \( t > 0 \). We similarly define \( p_{n,i} := r_{n,i} / \bar{R}^t \) as the fraction of resource \( i \) owned by node \( n \) at timepoint \( t > 0 \).

The nation of being \( \alpha \)-strong can be generalized to multiple resources, but we omit this for the sake of brevity. We also omit the time superscript whenever it is unnecessary (like in the previous equation). Similarly, when only a single resource is critical to the consensus mechanism, we omit the subscript \( i \) entirely and write \( p_{n} \).

### Example 1 (Bitcoin)
In Bitcoin, we can identify three major “types” of nodes:

1. **Mining nodes**, who create new blocks by solving computational “puzzles”.
2. **Full nodes**, who verify blocks before accepting them (i.e., check whether all the included transactions are valid), and
3. **Light nodes**, who only verify block headers, and are only interested in checking the inclusion of individual transactions via Simplified Payment Verification (SPV).

The foremost resource \( r_{n,1} \) of node \( n \) is processing power:

- Typically a mining node will need a great amount of it (e.g., an ASIC rig), a full node a moderate amount (e.g., a high-end PC), and a light node very little (e.g., a smartphone).

The model for Bitcoin can further be extended by including bandwidth as a separate resource \( r_{n,2} \).
### Example 2 (Quorum).
In a permissioned blockchain, there can still be full nodes and light nodes, but mining nodes are unnecessary and processing power is less important (yet still required to, e.g., verify signatures). The main resource is access to the private keys that allow for the creation of blocks, i.e., *authority*. We model this in the following way: we assume that there are \( k \) private keys for block creation. Then for all \( n \in \mathbb{N} \) and \( i \in \{1, \ldots, k\} \), it holds that \( r_{n,i} = 1 \) if node \( n \) controls key \( i \), and \( r_{n,i} = 0 \) otherwise. Furthermore, \( r_{n,k+1} \) denotes the processing power of node \( n \) and \( r_{n,k+2} \) the bandwidth. Note that it is possible for more than one nodes to have access to the same private key, e.g., after a hack.

### C. Blockchains as Peer-to-Peer Networks
Each node \( n \in \mathbb{N} \) has incoming and outgoing connections to other protocol-running nodes. The resulting *peer-to-peer network* can be represented as a graph where the vertices represent nodes and edges represent connections – see Figure 2a for an illustration. Not every node is necessarily connected to all other nodes, however it is typically assumed that from each node a path to every other node exists in the graph. If this is not the case, then there is an ongoing network *partition*.

At each time \( t \geq 0 \), each node \( n \in \mathbb{N} \) is aware of a set \( V_{n} \subseteq \mathbb{N} \) of blocks: we call \( V_{n} \) the *view* of node \( n \) at time \( t \). The genesis block \( g \) is the only block that all nodes are aware of at time zero, i.e., \( V_{n}^{0} = \{g\} \) for all \( n \in \mathbb{N} \). In addition to \( V_{n}^{t} \), each node \( n \in \mathbb{N} \) is also aware of transactions that have not been included in any block at time point \( t > 0 \). This information is stored in the *memory pool*, denoted by \( M_{n}^{t} \), for \( n \in \mathbb{N} \) and \( t > 0 \). Due to the distributed nature of the network and the presence of network latency, there must exist points in time for which different nodes are aware of different sets of blocks or different information, i.e., there exist \( t > 0 \) and nodes \( n, m \in \mathbb{N} \) such that \( V_{n}^{t} \neq V_{m}^{t} \) or \( M_{n}^{t} \neq M_{m}^{t} \).

At any time \( t \), each node \( n \in \mathbb{N} \) that can create blocks has to decide which block in \( V_{n}^{t} \) to extend. This block is called the *head*, and is represented by the variable \( H_{n}^{t} \). A function \( f \) that selects a head from a view is called a *fork-choice rule*. A *network fork* is any period during which at least two (protocol-following) nodes have *incompatible* blocks as heads. Here, we mean by *incompatible blocks* “two blocks \( B \) and \( B' \) such that neither is in the chain of the other, i.e., \( B \notin C(B') \) and \( B' \notin C(B) \). The term “fork” is also commonly used in practice to refer to protocol changes. That is, if the protocol is changed from \( \Pi \) to \( \Pi' \) and blocks created under \( \Pi' \) are still considered valid by \( \Pi \), then this is referred to as a *soft fork*. If not, the change is called a *hard fork*.

Due to network forks, a block \( B \) can be *orphaned*, which occurs if at some point of time \( t \), there is no \( n \in \mathbb{N} \) such

| Time | Event Type | V1 | V2 | V3 | same view | same head | ongoing fork |
|------|------------|----|----|----|-----------|-----------|-------------|
| 0    | genesis    |    |    |    | ✓         | ✓         | ✓           |
| 160  | node 2 *mines* block B1 |    | ✓  |    | ✓         | ✓         | ✓           |
| 168  | node 3 *receives* block B1 |    |    | ✓  | ✓         | ✓         | ✓           |
| 196  | node 1 *receives* block B1 |    | ✓  |    | ✓         | ✓         | ✓           |
| 239  | node 1 *mines* block B2 |    |    | ✓  | ✓         | ✓         | ✓           |
| 241  | node 3 *mines* block B3 |    | ✓  |    | ✓         | ✓         | ✓           |
| 248  | node 2 *receives* block B3 |    |    | ✓  | ✓         | ✓         | ✓           |
| 272  | node 2 *receives* block B2 |    |    |    | ✓         | ✓         | ✓           |
| 275  | node 3 *receives* block B2 |    |    |    | ✓         | ✓         | ✓           |
| 281  | node 1 *receives* block B3 |    | ✓  |    | ✓         | ✓         | ✓           |
| 668  | node 1 *mines* block B4 |    |    | ✓  | ✓         | ✓         | ✓           |
| 702  | node 2 *receives* block B4 |    |    | ✓  | ✓         | ✓         | ✓           |
| 710  | node 3 *receives* block B4 |    | ✓  |    | ✓         | ✓         | ✓           |

TABLE III: An example of the evolution of blockchain network of Figure 2. Each row corresponds to an event: it displays its time \( t \) of occurrence and its type (mining or propagation). Furthermore, we display the view \( V \) of each node \( i \in \{1, 2, 3\} \) after the event, and note whether the nodes’ views and/or heads are consistent across the network and whether a fork is ongoing. Events related to the dissemination of transactions are not included for brevity.
that \( B \in C(H_n^t) \). We say that a block \( B \) is \emph{overturned} by node \( n \) at time \( t \) if \( B \in \lim_{t \to \infty} C(H_n^{t+i}) \) and \( B \notin C(H_n^t) \). For example, in Table III, a network fork occurs from time 241 until time 710. Block \( B_3 \) (the teal block) is overturned by node 2 at time 702 and by node 3 at time 710. In practice, blockchain users need either a formal or heuristic notion of \emph{finality} – i.e., a notion of when a block can be assumed to not be overturned. For example, an online retailer will need to decide when a block that contains a payment is safe enough from being overturned to dispatch the order.

**Example 1 (Bitcoin).** In Bitcoin, the fork-choice rule prescribes to select the block \( B \) with the highest accumulated proof-of-work, i.e.,

\[
   f(V) = \arg \max_{B \in V} \sum_{B' \in C(B)} W(B') .
\]

In case of ties, the block seen first is preferred. This can lead to soft forks that persist even when all nodes have the same view, as illustrated in Table III. In [75], it was suggested that adversarial behavior can be discouraged by using \emph{uniform tie breaking} – whenever a node learns of a new block that has as much proof-of-work as its head, it adopts the new block as its head with probability 0.5. As further discussed in [172], this can have either a positive or negative effect on attackers, depending on how well-connected they are within the network.

For finality, Bitcoin users typically use the six-confirmation rule [180]: i.e., a block \( B \) is considered final by \( n \) at time \( t \) if there exists a \( B' \in V_n^t \) such that \( B = \text{PoW}(B') \).

**Example 2 (Quorum).** In the Istanbul-BFT protocol used by Quorum, blocks are added to the blockchain if they are confirmed by more than 2/3 of the voters. In particular, let, for any block \( B \in \mathbb{N} \) and private key \( i \in \{1, \ldots, k\} \), \( 1_{M^i_t}(i, B) \) equal 1 if \( M^i_t \) contains a "commit" message for \( B \) signed with \( i \), and 0 otherwise. Then block \( B \) is considered a valid block by node \( n \) at time \( t \) if

\[
   \sum_{i=1}^k 1_{M^i_t}(i, B) \geq (2/3)k .
\]

This is also the finality rule: i.e., in Quorum, all blocks are either both valid and finalized, or neither. This is true for many other BFT protocols as well.

**D. Actions, Strategies, and Utilities**

Based on the above, the \emph{state} \( X_t \) of the blockchain at timepoint \( t \geq 0 \) is a vector

\[
   X_t := (V_n^t, H_n^t, M_n^t, r_n^t)_{n \in \mathbb{N}} .
\]

When time is not relevant, we simply write \( X \) instead of \( X_t \). The \emph{state space}, i.e., the set of all possible states \( X_t \) will be denoted by \( X \).

Transitions to new states typically occur through operations or \emph{actions} performed by the nodes. Each protocol II has its own set of actions, and conditions under which they are available. We denote by \( \mathcal{A} \) the set of all possible actions allowed by the protocol. Let a \emph{strategy} \( s : X \to \mathcal{A} \) be a function from the state space to the set of actions – i.e., during execution, a node uses its strategy to select which action to take given the state of the system (where ‘waiting’ can be an action). Let \( \mathcal{S} \) denote the set of all possible strategies. Particularly relevant to our presentation is the \emph{default strategy} or the strategy that prescribes to \emph{follow-the-protocol} as referenced, which we will denote by \( d \). We will call any other strategy \( s \in \mathcal{S} \setminus \{d\} \) a deviant or \emph{adversarial} strategy. Also, we will refer to nodes that follow \( d \) as \emph{honest} and to nodes that do not follow \( d \) as adversarial nodes. If a node takes an action, then the system state is changed after a random delay. Let \( A_t \) be the set of completed actions until time \( t \). This imposes a probability measure on system executions, i.e., given a strategy profile \( S = x_{n \in \mathbb{N}} s_n \), then for all \( y, z \in X \) and \( t, \delta \geq 0 \) we should be able to determine \( \mathbb{P}_S(X_{t+\delta} = y \mid X_t = z) \). Each agent \( n \in \mathbb{N} \) has a utility function \( u_n : X \to \mathbb{R} \), which indicates their satisfaction with a given network state, and a utility function \( u_n' : \mathcal{A} \to \mathbb{R} \), which indicates their satisfaction with a given action. Given an initial state \( x_0 \), the \emph{long-term average utility} \( v_n(S, x_0) \) with a given strategy profile \( S \) is then given by

\[
   v_n(S, x_0) = \mathbb{E}_S \left[ \lim_{t \to \infty} \frac{1}{t} \left( \int_0^t u_n(X_{t'}) dt + \sum_{a \in A_t} u_n'(a) \right) \right] .
\]

In the present study, rather than analyzing individual strategic behavior, we mainly focus on collective protocol performance and blockchain properties. To formalize these notions, we define a \emph{property} or \emph{feature} of a blockchain protocol II as a function \( F_{\Pi} : X \to \{0, 1\} \), with the following values

\[
   F_{\Pi}(X_t) = \begin{cases} 
   1, & \text{if } S \text{ satisfies property } F_{\Pi} \text{ at state } X_t, \\
   0, & \text{otherwise.} 
   \end{cases}
\]

Additionally, we define aggregate utilities or \emph{performance measures} of protocol II as functions \( U_{\Pi} : X \to \mathbb{R} \). We say that a performance measure is \emph{positive} when higher values indicate better performance, e.g., throughput rate and collective profits, and \emph{negative} when lower values indicate better performance, e.g., communication complexity and operational costs.

**Example 1 (Bitcoin).** The core actions that any node \( n \) in Bitcoin can perform are \emph{block creation}, \emph{block propagation}, \emph{transaction creation}, and \emph{transaction propagation}. Of these, block creation takes an amount of time that is approximately exponentially distributed [61], [169] with a mean that is determined by the node’s processing power. The time needed to create transactions is negligible. Block and transaction propagation times depend on the network latency, the node’s bandwidth, connectivity, and the message size. More generally, \emph{block validation} can be included as a separate action that consumes processing power, to model the profitability of so-called SPV mining (i.e., mining a block without validating its transactions). Also, if a node \( n \) represents a mining pool, then the \emph{entering or exiting} of \( n \) by other nodes can be considered as actions.

**Example 2 (Quorum).** Instanbul-BFT has the same core actions as Bitcoin, but also includes the propagation of protocol messages, e.g., prepares and commits. Unlike Bitcoin, block creation is nearly instantaneous, but a node’s block is only valid if it has been selected as the block proposer (e.g., via a round-robin scheme) for the current \emph{round}. Under certain
conditions, e.g., if a block proposer waits too long before proposing a block, the other nodes can request a round change. If a supermajority of nodes agree on a round change, or sign off on a block, the next round begins.

Throughout the text, we will frequently use the following terms (and abbreviations):

- **Proof of Work** (PoW): As already mentioned, this refers to Nakamoto consensus [148] in which nodes, also called miners, gain the right to participate in the block creation process by providing solutions to a computationally difficult and energy-consuming cryptographic puzzle.

- **Proof of Stake** (PoS): Often called Virtual Mining [22], PoS emulates the above process but saves on energy waste by requiring from participating nodes to provide proof of “virtual” resources such as the platform’s native tokens. We will call a protocol permissionless if the consensus-critical resources are not inherent to the blockchain (e.g., processing power in PoW). We will a protocol semi-permissionless if the consensus-critical resources are inherent to the blockchain, but freely divisible and transferable (e.g., tokens in PoS). We will call a protocol permissioned if the consensus-critical resources are inherent to the blockchain and indivisible (e.g., the private keys in Quorum).

We are now ready to define the 5 axes of the PREStO framework and describe their subcategories.

### III. Optimality

**Optimality** is the most basic property of a protocol, and generally refers to whether the protocol is optimal within its operational scope. In our setting, it concerns the question:

**Q:** Under normal conditions, does the protocol provide its core functionality in an optimal way?

By “normal conditions”, we mean that nodes do not act strategically or maliciously, and that there are no capacity constraints. However, we do consider network latency and nodes going offline. “Core functionality” primarily refers to the functionality of any distributed database, i.e., to correctly read and write to the database. However, some protocols also provide additional functionalities, e.g., a broader notion of transaction types, or a higher level of privacy.

#### A. Liveness and Safety

Since blockchains are essentially data structures, they must adequately perform the read and write operations that are required of any database. We focus on the data in the finalized blocks of the chain, since non-finalized blocks can be overturned, [180]. The “write” operation then consists of adding a transaction to a finalized block on the chain. The “read” operation consists of observing that a transaction has made it into a finalized block on the chain.

The ability to write and read correctly is formalized through the notions of liveness and safety. A liveness fault means that a node is unable to write to the blockchain. A safety fault means either that two honest nodes see different results when reading the database, or that a single node sees different results when reading the database at different times.\(^3\) This informs our two general definitions of liveness and safety below.

**Definition 1** (Liveness). We say that a protocol II is live if from any state \(X \in \mathcal{X}\), any protocol-following node \(n \in \mathbb{N}\) can take a sequence of actions \(a_1, a_2, \ldots, a_m\) that will lead to a valid transaction being added to a final block in its view.

**Definition 2** (Safety). We say that a protocol II is safe if the following holds: any protocol-following node \(n \in \mathbb{N}\) who considers a block \(B\) as final at some time point \(T \geq 0\), will also consider \(B\) as final at any time point \(t > T\).

In practice, most protocols satisfy these properties only under certain conditions. In particular, most require that the honest nodes control a given fraction of the consensus-critical resources. For example, Bitcoin is safe only if the honest nodes are over 50%-strong in terms of processing power. Even then, the property of safety is guaranteed only in a probabilistic sense.\(^4\) Quorum is live only if the honest and not permanently offline nodes are at least \(\frac{2}{3}\)-strong, and safe only if the adversarial nodes are less than \(\frac{2}{3}\)-strong in terms of authority.

During a network partition, protocols can either satisfy liveness or safety, but not both – this is known as the CAP theorem [95]. Different protocols resolve this trade-off in different ways. Liveness-oriented protocols such as Nakamoto consensus allow the chain to fork by providing an unambiguous rule of how to resolve such forks when the partition ends, e.g., the longest-chain rule. Safety-oriented protocols such as Tendermint [125] and most other Byzantine fault tolerant protocols [39], [194] require that a (super)majority of participants sign off on each block. This means that during a network partition, at least one of the branches of the chain stops growing. In other settings, different branches of the chain can grow during a fork, but only one of these branches can finalize blocks. Examples include a traditional proof-of-work chain with Casper the Friendly Finality Gadget as an overlay (‘hybrid’ Casper) [37], [38]. It is also possible for protocols to guarantee neither liveness nor safety, e.g., Tangaroa [40].

In the scientific literature, the definitions and terminology used for safety and liveness properties vary. Algorand [94] provides its own definitions of liveness (new transactions can be added to the final part of the blockchain) and safety (if a node accepts a transaction as final, then it will continue to do so). In [60], [84], [119], [157], safety is called persistence – in turn, persistence and liveness can be shown to follow from three properties: common prefix, chain quality, and chain growth. For the Snow White [24], [57] and Tortoise and Hares [23] protocols, chain growth, chain quality, and consistency are considered. Here, consistency is a combination of common prefix and future self-consistency. In [18], the properties of validity and agreement are discussed as liveness properties, whereas integrity and total order are used for safety.

---

\(^3\)These two types of safety faults are practically equivalent: if two nodes see different results, then either the network remains permanently forked, or at least one of them will read a different value at some point in the future.

\(^4\)However, this probability can be made arbitrarily high by increasing the number of confirmations required to make a block final.
Typically, liveness and safety are proven for a specific protocol through a bespoke mathematical proof. However, under some assumptions on the actions in the protocol, general proof techniques may be available, e.g., model checking [26].

B. Transaction Scope

Some protocols offer fundamentally different types of transactions than others. For example, Bitcoin only supports monetary transactions, which allows for the entire “state” of the system to be described using unspent transaction outputs (UTXOs). However, protocols that support smart contracts (e.g., Ethereum [36]) require that the clients also store the internal variables of the contracts [58]. This may have an impact of efficiency (see also Section V), both via reduced throughput due to slower transaction processing, and potentially less straightforward scalability (“state sharding” [207]).

C. Privacy

The choice to put data on a blockchain instead of a centralized database has implications for privacy. On one hand, permissionless blockchains such as Bitcoin do not require identity management, thus favoring privacy. On the other hand, the entire history of transactions is publicly accessible, which may allow for de-anonymization. In fact, Bitcoin transactions may be better described as pseudonymous than as anonymous [45]. Cryptographic techniques that improve privacy, e.g., zero-knowledge proofs [97] or ring signatures [167], are available, although they may impose additional computational overhead and therefore impact efficiency. Furthermore, usage pattern analysis can lead to user de-anonymization even in privacy-minded platforms such as Zcash [116].

IV. STABILITY

Since intended behavior cannot be enforced in decentralized settings, one of the core tasks of consensus protocols is to properly incentivize agents to behave appropriately. This will enable the network to reach an outcome that is both stable and desirable. Importantly, stability does not imply optimality; instead, it is concerned with the question:

Q: Does the protocol incentivize the intended behavior? Is implementing and following-the-protocol the best possible strategy for participating and prospective nodes?

Game theory and traditional economics provide numerous tools to analyze this setting. Yet, as consensus protocols become more elaborate, the incentives and the required stabilizing mechanisms also become more complicated. These issues are discussed separately below.

A. Incentive Compatibility

At its core, incentive compatibility entails that it is in the participants’ best interest to follow-the-protocol. In concrete terms, and using the notation of Section II, this means that the default strategy profile \(x_{n \in N} \) is a Nash equilibrium, [149], [193]. An equilibrium is an outcome that is optimal from the perspective of all decision makers involved. Formally,

**Definition 3 (Incentive Compatibility).** Let II denote a protocol with active nodes \(N \subseteq N\), set of available strategies \(\mathcal{S}\), and long-term average utility functions \(v_n\) as defined in Section II for each \(n \in N\). Also, let \(d \in \mathcal{S}\) denote the follow-the-protocol strategy, \(D = \times_{n \in N} d\) the strategy profile where all nodes follow \(d\), and \(D_{n,s}\) the profile where all nodes follow \(d\) except node \(n\) who follows \(s \in \mathcal{S}\). Then, II is incentive-compatible, if

\[
v_n(D, x_0) \geq v_n(D_{n,s}, x_0), \quad \text{for all } s \in \mathcal{S}, n \in N,
\]

and initial states \(x_0 \in X\). In words, II is incentive-compatible, if given that all other nodes follow the protocol, then it is optimal for an entering (or existing) node to also follow the protocol.

This definition relies on some assumptions that are not always satisfied in practice. It assumes that first, each node can take as given that all other agents do follow the protocol (strategy \(d\)) and second, that all agents are rational, i.e., utility maximizers. Also, it requires that utility functions are known for each node. Although this set of assumptions may seem restrictive, it is an essential first step in protocol design to establish stability of a protocol within this vanilla setting. It is within the scope primarily of robustness and to a lesser extent of persistence to explore what will happen if these assumptions are violated, cf. Sections VI and VII.

**Example 1** (Bitcoin). Nash equilibria in Bitcoin mining are discussed in [75] and [118]. These works show that the Bitcoin protocol is an equilibrium only if a potential adversary (which could just be a rational node) is not too strong in terms of precessing power. [123] find that there is a multitude of symmetric equilibria in the Bitcoin protocol. Still, the default strategy prevails by a focal-point argument [147], [174]. [75] and [172] show that if nodes are at least \(\alpha\)-strong, where \(\alpha\) depends on their connectivity, then they are incentivized to follow the adversarial selfish mining strategy. [150] combine selfish mining – a consensus-layer attack – with an eclipse attack – a network-layer attack – to augment the rewards of selfish mining. Despite their theoretical plausibility, such attacks have not been recorded in practice.

Based on the above, the task of the blockchain architect is to design the consensus protocol in a way to induce the desired behavior in practice. Differences between intended and observed behavior should be addressed at this point. The theoretical discipline that models and studies such settings is that of mechanism design [15], [132], [147]. Applied in the blockchain context, it aims to determine the rules of the protocol in such a way that individual incentives are perfectly aligned with societal goals.

The notion of incentive compatibility can be seen beyond just Nash equilibria. For example, depending on whether the majority is controlled by a single entity or not, one may discern between strong and weak incentive compatibility [28]. In practice, a consensus protocol of a public, permissionless blockchain needs to properly incentivize rational agents to perform the following actions:

- Participation: acquire protocol resources, e.g., bandwidth.
• Operations: perform core and auxiliary tasks such as proposal and creation of blocks, message propagation, transaction validation and execution, data storage, etc. [45].

• Applications: use the native cryptocurrency or blockchain related applications ("Dapps").

Although they are integral to viability of existing blockchains, not all of these actions are properly incentivized, and miners’ incentives may be at odds with the underlying protocol [183]. Additional concerns stem from the tension between short-term and long-term incentives [128]. In [165], a consensus protocol is proposed that motivates both ownership and participation, and which aims to develop blockchains for social interaction. In [90], it is shown that the core economic motives for miners – transaction fees and block rewards – are also inherent to the security of PoS protocols. Finally, recent works suggest reputation systems as possible solutions to improve the incentive mechanisms of consensus protocols [131], [152].

The diversity of entities that are involved in the blockchain ecosystem introduces additional complexity. Different groups ranging from investors, developers, token holders to participating nodes and end users often have conflicting incentives. This implies that apart from the need to incentivize certain operations, like the ones mentioned above, the blockchain protocol also needs to align potentially conflicting incentives of these groups. Similar concerns emerge in blockchain-based markets or applications that entail the interaction between infrastructure operators, cyber-security providers, entrepreneurs, and end users in a trustless environment [79].

The theory on social choice and public goods provides insight into misaligned blockchain incentives [175]. A notable instance is captured by the free-rider or pass-the-bucket problem [20], [184]. In simplified terms, it states that rational agents who benefit from the existence of a public good – in this case, the blockchain – may shift responsibility for its creation to their peers. In the resulting equilibrium, the public good is not created, to everyone’s detriment. In public, permissionless blockchains, this translates to nodes moving costly tasks to other nodes, leading to an improper functionality of the blockchain ecosystem and a deviation from its intended mission.

Example 1 (Bitcoin). To explain the lack of observed selfish mining incidents, [14] suggest natural incentives (i.e., mining rewards) and the high monetary value of Bitcoin as mitigating factors [83]. Yet, [73] identify incentives for attacks between miners and argue that the prevailing practice of not engaging in these attacks is fragile and if broken will lead to equilibria with dire consequences for the blockchain ecosystem.

In [13], it is argued that the Bitcoin reference protocol disincentivizes the propagation of information. In [9], it is demonstrated that the active usage of Bitcoin remains low, and largely restricted to speculation and illegal activity — this has a negative impact on the stability of Bitcoin’s exchange rate with fiat currencies, an issue also discussed in [123]. In [203], a scheme to incentivize miners to promptly propagate any blocks that they know of is proposed as a way to effectively defend against certain adversarial strategies.

Many blockchain platforms include transaction fees that are paid by the transaction creator to the node that creates a block that includes their transaction. Transaction fees also impact the incentives of Bitcoin users. Currently, Bitcoin transaction fees are low, yet non-zero [123]. However, in Turing-complete environments (e.g., Ethereum), transactions typically require more computation, which makes these environments particularly vulnerable to network-layer attacks [140]. Based on the resource utilized most, there are three main sources of cost for the miners: network, computation and storage. [48] propose a fee-paying scheme for memory consumption that is typical to cloud storage services. In contrast to the currently deterministic transaction fees, [96] put forward the idea of random payments to incentivize risk-neutral miners. [42] analyze the future of Bitcoin mining, when mining rewards will have diminished, and conclude that if transaction fees are the only incentive, then selfish mining will be profitable for miners with arbitrarily small wealth. In the case that transaction fees can be further reduced, [50] argue that cryptocurrencies have the potential to become viable alternatives to retail payment schemes.

Protocol Resources: Protocol stability is tightly linked to the way that participating nodes acquire and increment their resources, which is starkly different between, e.g., PoW and PoS. In PoW protocols such as Bitcoin, computational (CPU) power is the consensus-critical resource. This implies that the costs for participating nodes are mainly electricity and investment in mining equipment [65]. These resources can be acquired in fiat currencies, yet the mining rewards are distributed in the native cryptocurrency (Bitcoins).

PoS protocols generate different dynamics. Virtual miners acquire their resources by converting fiat currency to the native cryptocurrency, which they then use as a proof to participate in the consensus mechanism. Mining rewards are again distributed in the native currency, however, in this case, the rewards naturally contribute to the protocol resources. This creates conflicting motives for staking nodes, since wealth and resources coincide and may lead to unpredictable inflation or disincentives to use or spend one’s stake. These observations call for a re-evaluation of the economics of different protocols through the lens of novel macroeconomic tools. Integral are the questions about the distribution of resources, the corresponding entry barriers, and market dynamics — perfect or oligopolistic competition — that they induce [6], [130].

B. Decentralization

Decentralization lies at the core of blockchain design philosophy and is therefore integral for its long-term survival and sustainability [130], [141]. However, existing data shows that centralization plagues PoW (and PoS) cryptocurrencies of both high and low market values [31], [82]. Miners join centralized pools which efficiently distribute mining rewards among their participants. This is known to reduce the extreme variance of mining returns that discourages solo miners, [169], [176], [187]. Yet, the operation of mining pools introduces unpredictable dynamics in the consensus mechanism and incentivizes miners (or protocol participants) to behave dishonestly, especially under high transaction loads, and destabilize...
the system [133], [152]. For instance, staking pools – the equivalent of mining pools in PoS protocols – can potentially evolve to become institutions with arbitrary power over their cryptocurrency [31], [76].

In conventional market economics, market concentration is measured by the Herfindahl-Hirschman Index (HHI), [166]. In general, the HHI is defined as the sum of the squares of the market shares of the firms within the industry, where the market shares are expressed as percentages. As such, it can range from 0 to 10,000, with higher values indicating larger concentration\(^5\). In the blockchain context, it can be used to study the concentration of protocol resources. For a state \(X\) of protocol II with active nodes \(N \subset \mathbb{N}\), and distribution of consensus-critical resource fractions \((p_n)_{n \in N}\), the HHI is given by

\[
\text{HHI}_\Pi (X) := \sum_{n \in N} (p_n \cdot 100\%)^2
\]

The HHI is used in the context of antitrust management and also in applied social and political sciences to measure the concentration of political power [129], [190].

Example 2 (Bitcoin & Ethereum). Table IV shows the estimated distribution of mining power between the top 10 Ethereum mining pools (or accounts) by number of blocks.

| **Bitcoin** | Blocks % | **Ethereum** | Blocks % |
|-------------|-----------|--------------|----------|
| 1. BTC.com  | 20.1%     | Ethermine    | 26.5%    |
| 2. AntPool  | 14.5%     | Sparkpool    | 24.5%    |
| 3. F2Pool   | 13.1%     | F2Pool_2     | 11.8%    |
| 4. Shushu   | 8.8%      | Nanopool     | 11.2%    |
| 5. Poolin   | 8.8%      | MiningPoolHub_1 | 5.4% |
| 6. ViaBTC   | 8.3%      | Address_1    | 2.3%     |
| 7. BTC.TOP  | 6.1%      | Address_2    | 1.7%     |
| 8. BitFury  | 4.9%      | DwarfPool_1  | 1.7%     |
| 9. BitClub Network | 1.7% | zhizhu.top | 1.3% |
| 10. Bitcoin.com | 1.4% | firepool | 1.2% |
| **Total**   | 87.7%     | **HHI Index** | 1075.7  |
|             |           |              | 1610.5   |

TABLE IV: Concentration of mining power for the Bitcoin and Ethereum blockchains (as of 07 June 2019) and calculation of the Herfindahl-Hirschman Index (HHI). Sources: blockchain.com and etherscan.io.

The figures indicate a more decentralized market for Bitcoin than for Ethereum. Similar calculations indicate even higher centralization for smaller PoW platforms, for which 51% attacks – distributions in which a single entity owns more than 50% of the resources – are a reality [93], [104], [113]. These figures echo the concerns that the current structure of some of the major blockchain platforms is prone to centralization [4], [5], [92], [130].

From a stability perspective, [150] argue in favor of dispersing mining power since the follow-the-protocol strategy ceases to be a Nash equilibrium if a single node becomes too strong. Incentives to derive short-term profits from attacks on mining pools threaten the long-term viability of Bitcoin and negatively impact the Bitcoin ecosystem [128], [115] show that pool size and computational power are the main criteria when deciding whether to launch a network-level attack against a mining pool. These concerns are not only relevant to Bitcoin, but to other PoW blockchains as well [93].

Ideally, nodes should have no motive to band together at all. This informs the following definition:

**Definition 4 (Perfect Decentralization).** Let protocol II in any state \(X \in X\) of nodes \(N \subset \mathbb{N}\), such that each node \(n \in N\) controls a fraction \(p_n\) of the consensus-critical resource. Let the state \(X_{nm}\), for \(n, m \in N\), be the same state as \(X\) except with nodes \(n\) and \(m\) merged, and their resources combined. Also, let \(d \in D\) denote the follow-the-protocol strategy and \(D = \times_{n \in N} d\) the strategy profile where all nodes follow \(d\). Then a protocol satisfies perfect decentralization if

\[
v_n(D, X) \geq v_n(D, X_{nm}) \quad \text{for all } n, m \in N.
\]

Such a definition depends strongly on the utility functions: e.g., in Bitcoin, banding together always reduces the reward variance, but when the pools get too strong, trust in the system is undermined and Bitcoins will lose value against other (crypto)currencies. For example, the mining pool GHash.IO was forced to take action to reduce their pool size after they surpassed the 50% mark [77].

Mining pools are not the only threat to decentralization. Other sources involve the underlying network layer, the geographic or economic motives to concentrate mining rigs in countries with low energy cost, and the increasingly sophisticated technology that is required to participate in the block creation process [196], [53] study anti-trust policies in Turing-complete blockchains, i.e., blockchains that also support smart contract execution, and argue that although smart contracts mitigate information asymmetries and improve social welfare, they also encourage collusions, and hence generate a threat to decentralization. [87] develop a method to bootstrap the blockchain without a genesis block created by a trusted authority. In all cases, the threats of centralization and trust formation raise the closely related question of blockchain governance and sustainability in the long run which is addressed in Section VII, [27]. Various sources of centralization in the blockchain ecosystem are illustrated in Figure 3.

C. Fairness

An integral element of stability in non-permissioned protocols is fairness, which relies on the premise that participating nodes should be rewarded proportionally to their resource contribution, [47]. Recall that each node \(n \in N\) participates in the protocol by providing some consensus-critical resource \(r_n\) which corresponds to a proportion \(p_n\) of the total resources. If \(U_T\) denotes the total rewards \((U_T\) can be positive or negative) distributed by the protocol to nodes over a (long) period \(T\) of time, then we can formally define fairness as follows.

\(^5\)The USA Department of Justice considers a market with an HHI of less than 1,500 to be a competitive marketplace, an HHI of 1,500 to 2,500 to be a moderately concentrated marketplace, and an HHI of 2,500 or greater to be a highly concentrated marketplace. However, these thresholds refer to oligopolistic markets and should be much lower when studying decentralization in blockchains.
Achieving fairness seems challenging in practice. Message delays and network latency can cause a disproportional distribution of rewards [100]. Focusing on PoS protocols, [76] introduce the notion of equitability to quantify how much a proposer can amplify her stake compared to her initial investment. Even with everyone following the protocol (i.e., honest behavior), existing methods of allocating block rewards lead to poor equitability, as does the initialization of systems with small stake pools and/or large rewards relative to the stake pool. Consensus in distributed computing with weighted nodes and more general notions of fairness are studied in [7], [89], [158]. [195] extend this notion to environments with adaptive corruption by strengthening the definition of “ideal protocol quality” defined in [84] and [157].

Fairness in blockchains can be understood as a two dimensional notion that entails both the reward allocation and the block creation mechanism, as illustrated in Figure 4. Current protocols are based on the premise that proportional voting is fair, [131]. However, the simple and seemingly appealing axiom “one unit of resource (one computer or one coin), one vote” has been theoretically refuted in traditional voting systems [17], [198].

More importantly, node selection proportionally to their resources – as in current PoW and PoS protocols – does not necessarily imply fairness in the voting process, [78]. An illustration is provided in the following example.

**Definition 5 (Fairness [158]).** The reward allocation mechanism of a protocol $\Pi$ is said to be $(\alpha, \epsilon)$-fair for some $\epsilon > 0$, if in the presence of an $\alpha$-strong adversary, each node $n \in N$ can guarantee $(1 - \epsilon) \cdot p_n \cdot U_T$ of the rewards over any period of length $T$.

Efficiency exhibits a tradeoff between a modest (excessive) waste of resources and a high (low) risk of attacks. Nodes in PoW protocols provide proofs of the validity of created blocks...
via energy consumption, which creates a negative environmental externality, [123]. A promising alternative is offered by the PoS or Virtual Mining protocols which reduce this huge energy waste [22], [94], [120], [125], [171]. Since PoS protocols delegate decision power via proofs of coin (stake) ownership, one of their main advantages over Nakamoto’s PoW is their environmental sustainability [22].

Energy is not the only input that can be inefficiently used by a protocol. Space for data storage, bandwidth, and Random-Access Memory are only a few [48]. Other aspects of efficiency involve the times to process and finalize transactions and the communication complexity that is required for the distributed network to reach consensus [202]. Importantly, different applications introduce various degrees of uncertainty in the use of such resources and increase the challenge of designing efficient solutions. The processing power used for transaction processing and block propagation delay also determine the outcome of the mining competition [135]. Failing to address such issues demotivates agents from participating and leads to centralization. In this sense, efficiency is also related to stability [67].

Better ways to utilize the energy spent in PoW protocols may eliminate – if successful – the advantage of PoS over PoW protocols in terms of energy waste [16], [194] provide a classification of other early proposals and open questions in this direction. Still, all of these alternative proposals need to tackle the problem of scalability, described next.

B. Scalability

Scalability refers to the property that the consensus protocol – and hence the blockchain – benefits from the addition of nodes or resources [208]. Generally, a blockchain is scalable if it exhibits positive scale effects, i.e., if increased participation leads to (i) increased throughput and (ii) improved liveness, safety, stability and efficiency guarantees. Since these indicators may respond differently to variations in the number of nodes (or the amount of resources), it is more convenient to understand scalability as a property of performance measures rather than of the blockchain protocol II as a whole. Recall that a performance measure $U_\Pi : \mathcal{X} \to \mathbb{R}$ is called positive (negative) if increasing values indicate better (worse) performance, cf. Section II.

**Definition 6** (Scalability). Let state X have consensus-critical resources $r_n$, for nodes $n \in \mathbb{N}$, and state $X'$ resources $r'_n$ such that $\sum_{n \in \mathbb{N}} r'_n > \sum_{n \in \mathbb{N}} r_n$. Then II is scalable in the positive performance measure $U_\Pi$ if $U_\Pi (X) > U_\Pi (X')$ for any such $X, X' \in \mathcal{X}$.

Definition 6 states that a protocol II is scalable in the performance measure $U_\Pi$ if an increase in the resources of the current state implies an improved performance for $U_\Pi$. The definition for negative performance measures is similar.

**Example 1** (Bitcoin). To achieve its strong safety guarantees [180], the use of computational resources by the Bitcoin (PoW) protocol is not efficient: the maximum transaction throughput is the same as five years ago despite a dramatic increase in hash rate and energy consumption [154], [25] identify excessive spending and inefficiencies in the prevailing equilibrium of Bitcoin’s follow-the-protocol strategy. [50] suggest that partial or complete substitution of energy-costly mining activities with PoS mechanisms could benefit Bitcoin and make it more efficient in the long run.

Attacks on Bitcoin can inflict a significant energy cost on miners [139]. In general, by partitioning the network or by either censoring or delaying the propagation of blocks, network-layer attacks can cause a significant amount of mining power to be wasted, leading to revenue losses and enabling a wide range of attacks such as double-spending. To deal with these threats, [141] propose a mining pool that will run as a smart contract and show that this is a solution with good efficiency and scaling properties.

Currently, a broadly studied solution to scalability is *sharding*, see e.g., Elastico [138], OmniLedger [121] and Ethereum 2.0 [35], [72]. As an alternative approach, [91] model the concept of sidechains as a means to increase scalability and enable the interoperability of blockchains. Their construction features merged-staking which prevents Goldfinger attacks – attacks whose explicit goal is to undermine and destabilize the consensus protocol [27], [123] – and cross-chain certification based on novel cryptographic primitives. [19] study a similar combination of consensus protocols with PoS subchains linked to the PoW Bitcoin blockchain.

C. Throughput

Although throughput is closely related to scalability, a protocol can prioritize throughput even without making the protocol fundamentally more scalable. For example, by increasing the maximum number of transactions per block (e.g., Bitcoin Cash vis-à-vis Bitcoin), throughput is increased without essentially affecting scalability. The same is true for protocols such as EOS.IO and TRON, which achieve much higher throughput than, e.g., Bitcoin and Ethereum by curtailing the number of potential block proposers. In fact, a BFT protocol can easily achieve much higher throughput than a Nakamoto protocol if the number of nodes, denoted by $N$, is low. However, such protocols typically suffer from negative rather than positive scale effects when the number of nodes increases due to the $O(N^2)$ message complexity. So it is possible for a protocol change to have a positive effect on throughput yet a negative effect on scalability. This informs our definition below.

**Definition 7** (Throughput). Let the performance measure $U_{\Pi}(X)$ denote the long-term average transaction throughput, starting from state $X \in \mathcal{X}$. A blockchain protocol II with resources $r_n$, $n \in \mathbb{N}$ has a higher throughput than another protocol II' with the same resources if $U_{\Pi}(X) > U_{\Pi'}(X)$.

Fundamentally, scalability concerns the effects on the outputs when the resources are changed and the protocol is kept the same, whereas throughput (as a PREStO category) concerns the effects on the outputs when the protocol is changed and resources are kept the same.
D. Centralized Systems as Benchmarks

From a managerial perspective, the integral question in launching a blockchain project or application is whether a blockchain is indeed better than a centralized system for the intended purpose [137], [199]. Since blockchains eliminate trusted authorities to reach consensus via the coordination of distributed and self-interested entities, several questions come into play. How does the distributed system compare to a benchmark solution? Does it provide improved performance in terms of costs, efficiency and security?

Interestingly, related questions have been thoroughly researched in game theory. In particular, traffic routing, queueing theory and congestion networks explore precisely these tensions between equilibration and efficiency of centrally regulated systems, [51]. The trade-offs are quantified by the Price of Anarchy (PoA), which measures the sub-optimality caused by self-interested behavior relative to centrally designed and socially optimal outcomes [110], [160], [170]. PoA is defined as the ratio between the performance of the system at the worst-case equilibrium and that at a socially optimal state [122].

Studying this question in the current context requires us to quantify different aspects of blockchain performance and compare them to either an existing or a socially optimally (ideal) solution provided by a benevolent social planner or authority. To measure the effects of decentralizing a system when implementing it as a blockchain, we evaluate a derivative of Anarchy (PoA), which measures the sub-optimality caused by a system away from the vanilla setting. These concerns are expressed and rapidly the protocol’s properties degrade when we move away from the vanilla setting. These concerns are expressed by the following question:

Q: What is the resistance of the protocol to perturbations on its underlying assumptions?

VI. Robustness

Suppose that a protocol has provable performance guarantees within its scope (optimality) and that the follow-the-protocol strategy is an equilibrium (stability), at which the protocol resources are reasonably utilized (efficiency). The next natural step in protocol design is to explore how smoothly and rapidly the protocol’s properties degrade when we move away from the vanilla setting. These concerns are expressed by the following question:

Q: What is the resistance of the protocol to perturbations on its underlying assumptions?

In the case of a parametrizable protocol, this question may also be phrased in terms of the extent of the parameter variation that the system can tolerate [8]. Essentially, robustness tests the assumptions that were used to equilibrate and stabilize the system. The main challenge is to assess protocol performance under conditions that are not captured by the ideal setting of Definition 3, such as parameter fluctuations, collusion between nodes, and malicious or irrational behavior [107].

A. Alternative Equilibrium Concepts

The application of the Nash equilibrium as a stability concept in blockchains is not entirely uncontroversial [12], [101]. In particular, [101] and [59] discuss the following shortcomings of Nash equilibria in distributed computational systems: unexpected behavior (irrational players with out-of-system incentives), coalitional deviations, computational limitations (resource-bounded players), too much uncertainty or a lack of information (players are unaware of all the aspects of the game). To deal with these issues, [2], [101] propose the notion of robust strategy profiles which have two defining components. On one hand is the profit of deviating players. If $k$ agents simultaneously deviate from a given strategy profile but are not able to increase their profits, then the strategy profile is said to be $k$-resilient. On the other hand is the harm incurred to non-deviating players. If $t$ agents simultaneously deviate from a given strategy profile but are not able to decrease the profits of non-deviating agents, then the strategy profile is said to be $t$-immune. Combining these two elements yields the notion of $(k,t)$-robust equilibria as strategy profiles that are both $k$-resilient and $t$-immune.

Despite its theoretical appeal, [101] observes that the concept of $(k,t)$-robust equilibria has its own limitations and points to concepts of computational equilibria and particularly to the BAR-model – model with Byzantine, Altruistic and
Rational agents – as possible alternatives [3], [12]. Nevertheless, [99] provide strong arguments to support the use of Nash equilibria by showing that large games are innately fault-tolerant. In fact, anonymous games that can be used to model blockchain mining are shown to be resilient against irrational behavior (Byzantine faults), coalitions and asynchronous play.

In an approach that is particularly relevant to the blockchain setting, [136] define robustness of an equilibrium as the maximum proportion of malicious nodes that the desired equilibrium strategy can tolerate, in the sense that this strategy remains the best strategy for rational players. In this definition, robustness is understood as a local property, i.e., as a property of a specific strategy profile and against specific adversarial strategies. This definition overcomes the computational difficulties of defining robustness for the blockchain protocol as a whole, and utilizes the fact that in blockchains, the analysis of robustness mainly concerns the follow-the-protocol strategy.

B. Out-of-Protocol Incentives

In reality, an adversarial node may try to change the behavior of other nodes by influencing their utility functions through threats or rewards. One of the earliest examples of this is feather forking [145] in Bitcoin: in this case, a miner threatens to refuse to extend blocks if they contain a blacklisted transaction. Even if the expected impact of the threat is small, it may be high enough compared to the small cost of enforcing the blacklist to make it rational to comply with the threat. Similarly, bribery [27], [28], [142] or discouragement [34] attacks can be used to distort the incentives of rational nodes.

In protocols in which it is known how much consensus-critical resources are owned by each of the nodes (i.e., semi-permissionless or permissioned blockchains), it may be possible to predict which nodes are scheduled to propose blocks in the near future. Accordingly, [30] identify two complementary properties – recency and predictability – of all longest-chain PoS protocols and devise relevant attacks to show that all such protocols are susceptible to certain kinds of malicious behavior. Finally, [177] explore the trade-offs between PoW and PoS consensus and find that a combination of both may yield robust results. In particular, for small numbers of participants PoS exhibits better security properties against 51% attacks by mining pools but as the size of the network increases, they recommend reverting to PoW.

C. Resistance to Malicious Behaviour

Not all nodes are solely interested in protocol rewards – for example, they may be interested in performing a Goldfinger attack [123], in which one cryptocurrency platform is attacked to increase the value of others. One way of modeling this is to assign to each attacker a utility that is the inverse of the collective utility, and calculate the total losses under the new equilibrium. Another approach is to calculate bounds on the losses that attackers can do relative to their own losses. In [33], [34] this is made explicit through the grieving factor (GF). In particular, let $n, m \in \mathbb{N}$ be nodes, $x_0 \in \mathcal{X}$ a starting state, $D$ the strategy profile where all nodes play the default strategy, and $D_{n,s}$ the profile where all nodes play $d$ except $n$ who plays $s$. Then the grieving factor of between $n$ and $m$ is defined as

$$\text{GF}(n, m, s) = \frac{v_m(D_{n,s}, x_0) - v_m(D, x_0)}{v_n(D_{n,s}, x_0) - v_n(D, x_0)}$$

if the denominator is positive, and $\infty$ otherwise. Summing over all $m$, yields the network’s grieving factor (NGF). The value of the NGF reflects the relative loss that a participating node needs to incur in order to inflict a “unit” of loss on the whole network. For instance, a value of NGF equal to 1 implies that node $n$ needs to destroy 1 dollar for every 1 dollar loss that she inflicts to the network. Accordingly, larger values of the NGF correspond to the most harmful cases.

VII. PERSISTENCE

The four PRESIO categories discussed so far consider the protocol when it operates at or near to equilibrium conditions. However, what happens if the protocol is forced away from its equilibrium, for instance after a large-scale attack or catastrophic black swan event? Hence, to establish a protocol’s persistence property, we ask the following question:

Q: Does the protocol have mechanisms to recover from highly non-equilibrium conditions and return to stability in its optimal state? If so, then how fast, and at what cost?

These questions deal with the long-term sustainability of the blockchain platform. Whereas for robustness, we studied performance under perturbations of the stability assumptions, for persistence, we take this idea to its logical extreme. We assume that the ecosystem is under a large-scale or protracted attack, and study whether it is designed to recover and resume its desirable properties, at least sufficiently often. Hence, we want to assess to what extent a blockchain has the qualities to survive and evolve under extreme crashes, technology shocks or other rare events.

A. Weak & Strong Persistence

To understand protocols from this perspective, we formalize the notions of weakly and strongly persistent properties in the blockchain context. These ideas have been introduced within evolutionary game theory and in the study of biological systems, i.e., recovery of an ecosystem after infection from a virus [105], [181]. More relevant to the current context is the combination of these ideas with tools from optimization theory and algorithm design [21], [161]. Formally, recall that a property or feature of a protocol $\Pi$ is defined as a function $F_{\Pi} : \mathcal{X} \rightarrow \{0, 1\}$, cf. Section II.

Definition 8 (Weakly & strongly persistent properties [159]). Consider a protocol $\Pi$ and a property $F_{\Pi} : \mathcal{X} \rightarrow \{0, 1\}$. Let $(X_t)_{t \in \mathbb{Z}}$ be a protocol execution with initial state $x_0 \in \mathcal{X}$. We say that $F_{\Pi}$ is:

- weakly persistent for protocol $\Pi$, if for any $x_0$, and any $T > 0$, there exists $t > T$ such that $F_{\Pi}(X_t) = 1$.
- strongly persistent for protocol $\Pi$, if for any $x_0$, there exists $T > 0$, such that $F_{\Pi}(X_t) = 1$ for all $t > T$. 

Intuitively, a weakly persistent property will eventually be satisfied and become satisfied again infinitely often given any initial system condition, whereas a strongly persistent property will eventually be satisfied and stay satisfied given any initial system condition, [159]. These definitions capture the idea that a desirable property may not be satisfied by a system in equilibrium, but in a dynamic way. This allows for more flexibility between recovery/convergence time, “periodicity”, and the cost of implementation.

Example 5 (The Blockchain Trilemma). The idea of supporting two or more incompatible properties in a weakly persistent manner, as described above, can be exploited to address the challenging “Blockchain Trilemma” ([1], [54]) which is illustrated in Figure 5 and which we will discuss in more detail in Section VIII-A. The vertices of the triangle correspond to the three seemingly incompatible but desirable properties that blockchain consensus protocols may satisfy: decentralization, scalability and safety. Protocols can be thought of as points inside the triangle, with coordinates indicating the degree of satisfaction of each of these properties.

Designing an optimal protocol – i.e., a protocol which in equilibrium satisfies simultaneously all three properties – has been a formidable task for blockchain architects, [94]. Such a protocol is indicated by the green dot in Figure 5. However, the idea of weak persistence can be exploited for an alternative design: a protocol could solve the trilemma by constantly alternating between states that satisfy a non-conflicting subset of the otherwise incompatible properties. This is captured by the blue dot protocol and the dashed arrows in Figure 5 which show its transition between different states.

The idea of studying distributed computation through the lens of dynamical systems has been recently initiated by [112]. Based on their ideas, persistence can be also used to formulate a weaker definition of fairness, cf. Section IV-C. Namely, a protocol can be described as fair if each node gets to be selected in the block creation process infinitely often.

B. Recovery from Majority Attacks

One of the major challenges in blockchain consensus protocols is the recovery from attacks by malicious nodes who control the majority of protocol resources, [27]. Existing protocols establish their safety and liveness properties under the assumption of either a simple – 51% – or an enhanced – usually 67% – honest majority of nodes, [205]. Contrary to initial beliefs that these attacks are only of theoretical interests, recent studies have documented the contrary, [28]. An important insight from these studies is that it is sufficient to gain control for some short period of time, for instance by temporarily renting protocol resources.

A suggested mechanism to recover from majority or large-scale attacks on the Ethereum blockchain is the minority fork, proposed by [35]. In brief, a minority fork is a mechanism to recover the majority of the consensus-critical resources through a fork initiated by an honest minority. Because the majority cannot create blocks on both branches of the fork, they will be seen as offline on the minority-initiated branch, which may cause their share to shrink on this branch. Such a scheme is fundamentally impossible in permissionless blockchains.

C. Governance & Sustainability

Persistence is closely related to the decision processes that determine the structure and operation of the blockchain. The practical need for an optimal governance structure in the Bitcoin community has already been observed by [123]. In a different approach, [45] view the blockchain as a public good and discuss the role of intermediaries that will provide paid services of blockchain verification and monitoring which adds value to the entire blockchain ecosystem. With the exception of some tentative predictions, the formal governance structure of public, permissionless blockchains has yet to be determined, [106]. The issues of governance and long-term sustainability in blockchains are integral to their success, and therefore central themes in their evolution.

VIII. EVALUATION: USE CASES OF THE PRESTO FRAMEWORK

In this section, we evaluate the PRESTO framework’s ability to illustrate the fundamental differences between various protocols. In Section VIII-A, we begin by comparing the PRESTO framework to the Blockchain Trilemma, another well-known model of protocol properties. Next, we use the PRESTO framework to illustrate the properties of a range of 9 recently proposed protocols and protocol modifications. As can be seen from our analysis, the PRESTO framework reveals more detail than the blockchain trilemma. We conclude the section with an overview of research challenges in Section VIII-C. Throughout the next section we refer to the visual summary of the PRESTO displayed in Figure 6.

A. The Blockchain Trilemma

In Section VII, we referred to the Blockchain Trilemma as an example of how the theory of dynamical systems can offer a different perspective to one of the most long-standing problems in distributed computing. In particular, largely incompatible properties, like safety, decentralization and scalability, can be satisfied in a weakly or strongly persistent manner, i.e., in a regularly alternating but certainly recurrent manner. This offers a potential way out of the current deadlock in the search for better blockchain protocols.
Fig. 6: Visual representation of the PREStO framework. The Blockchain consensus protocol lies in the middle of a series of concentric circles. The inner cycle comprises the 5 major axes of PREStO and the outer cycles correspond to subcategories of increasing granularity. Starting from this setting, the framework can be extended in a dynamic way to integrate features of more elaborate blockchains in the future.

for an ideal protocol that will concurrently satisfy all of the three highly desirable properties.

But how do the properties – safety, decentralization and scalability – of the Blockchain Trilemma relate to the PREStO framework? Or put differently, does PREStO provide the right tool to parse the trilemma in its basic components, to communicate it to experts of different backgrounds and ultimately to reason about and try to resolve it? Before we can answer these questions, we discuss the three properties of the blockchain trilemma in more detail. We base our description on [71], which gives a concise, formal description of the three properties. In the following, we assume for simplicity that there is a single consensus-critical resource. We also introduce the scale parameter \( \bar{n} \), which represents the “size” of the network. The properties of the blockchain trilemma describe the behavior of the blockchain as it grows in size, i.e., in the limit \( \bar{n} \to \infty \). In the following, the total number of nodes in the network is given by \( n \), and \( R(\bar{n}) \), the total amount of resources in the network, is proportional to \( \bar{n} \) such that \( \lim_{\bar{n} \to \infty} R(\bar{n}) / \bar{n} = c \) for some constant \( c > 0 \). The three properties are then as follows.

Safety: The network is “secure” against an attacker whose resources are proportional to \( \bar{n} \). To make this formal: let the attacker’s resources be denoted by \( r_a(\bar{n}) \) when the network size is \( \bar{n} \). The attacker’s resources are proportional to the network size if \( \lim_{\bar{n} \to \infty} r_a(\bar{n}) / \bar{n} = \alpha \) for some constant \( \alpha > 0 \). The system must still be secure in this setting.

Decentralization: An honest node whose resources are proportional to \( c \) but not \( \bar{n} \) is still able to “participate” in the network. That is, if the honest node’s resources are given by \( r_h(\bar{n}) \), then even if \( \lim_{\bar{n} \to \infty} r_h(\bar{n}) / \bar{n} = 0 \), then the node should still be able to participate.

Scalability: The system is able to process a number of transactions that is proportional to \( \bar{n} \) rather than \( c \). That is, if the number of processed transactions in a network of size \( \bar{n} \) is given by \( U^*_\bar{n}(\bar{n}) \), then we require that \( \lim_{\bar{n} \to \infty} U^*_\bar{n}(\bar{n}) / \bar{n} = \beta \) for some \( \beta \in (0, \infty] \), and that \( \lim_{\bar{n} \to \infty} U^*_\bar{n}(\bar{n}) / c = \infty \).

As we can see, the blockchain trilemma describes an asymptotic regime in which certain properties must hold, but leaves these properties open to interpretation. In particular, “security” in the trilemma’s safety property can either refer to liveness or safety of optimality as displayed in Figure 6. However, a deeper analysis might also consider bribery or discouragement attacks, which are subcategories of robustness. The notion of “participating” in the protocol under the trilemma’s decentralization property can refer to the ability to perform the full range of operations, to the existence of entry barriers, or to the ability to be fairly rewarded. All of these are subcategories of the stability. Finally, the trilemma’s notion of scalability is clearly related to its namesake in PREStO, which is a subcategory of efficiency. However, unlike the trilemma, which requires that throughput is at least linear in term of the network size, we merely require that it is increasing (one extension of Figure 6 would be to expand the scalability category to include more subcategories for different rates of growth).

Hence, the PREStO framework can be used to fill the
gaps and ambiguities left in the definition of the blockchain trilemma. In general, safety relates to optimality (and perhaps robustness), decentralization to stability, and scalability to efficiency. This relationship is illustrated in Figure 7.

Turning to practical examples, Bitcoin is considered to be a safe blockchain because an attacker needs to control at least a given fraction of the total network hashrate for most attack to become possible or profitable (e.g., approximately one third for selfish mining). However, Bitcoin does not scale in its current form, and the emergence of pool-mining has lead to an undesirably high centralization, [5], [130] – i.e., it is practically impossible for smaller nodes to perform certain protocol operations such as block creation. There is also a trade-off between decentralization and safety: while solo mining outside of pools leads to higher degree of decentralization, the large variance of mining rewards may force solo miners to drop out of the mining process due to temporal losses and hence, ultimately decrease the network safety, [162].

On the other hand, EOS.IO, [70], promises both decentralization and scalability. However, although its minor nodes are able to vote, the ability to create blocks is concentrated in the hands of a small number of nodes. As such, it is unclear if EOS.IO can be described as decentralized. Its relatively small number of active nodes also constitutes a barrier to understanding its exact limits in terms of safety in a large-scale public setting [162]. Similarly, Litecoin and Bitcoin Cash [43], [134] aim to increase scalability – via an increased block frequency or size – but although this increases throughput by a constant factor, the network is still not scalable in the sense that throughput is linear (or even increasing) in terms of the network size. Hyperledger, [109] aims to provide both scalability and safety by operating as a private blockchain, i.e., in a less decentralized setting. Algorand, [94] and Ethereum, [197], are actively researching or developing ideas like Proof of Stake, sharding, side chains and more efficient Byzantine Fault Tolerant mechanisms that will lead to revolutionary solutions of the Blockchain Trilemma. The list of approaches does not end here, with protocols such as Zilliqa, [207], offering yet more ideas to this debate by using sharding to create a scalable but still decentralized protocol. Finally, while many claim that solving the trilemma is essentially not possible, co-existence of multiple, interoperable blockchains may be another approach to go past this bottleneck in the future, [192].

B. Evaluating Features of Blockchain Protocols

In this section, we use PREStO framework as a tool to compare and evaluate a range of recent protocol modifications. A summary of this comparison is presented in Table V, and we present a more detailed discussion in the following.

Partial solutions: Several recent works [158], [187] have proposed to modify Nakamoto consensus by allowing miners to include unsuccessful attempts to solve the PoW puzzle – partial solutions – in the blocks. These partial solutions contribute to the block’s likelihood to be selected by the fork-choice rule, and award rewards to the finders. Hence, weaker miners are rewarded more often, which reduces their barriers to entry and as such increases decentralization. In addition, the inherent advantage that big miners have when confronted with network latency is reduced, which improves fairness. Finally, preliminary experiments suggest that it is harder for selfish miners or attackers with a minority of the hash power to overturn blocks, which helps incentive compatibility and attack resistance. The downside is that new data is added to the blocks, which consumes bandwidth and therefore harms throughput.

Smart contracts: The main innovation of the Ethereum platform [36] was to extend the functionality of blockchain from token transfers to the creation and execution of Turing-complete programs called smart contracts. As such, the transaction scope is much wider. This comes at a cost to throughput as nodes need to expend considerably more processing power to execute the transactions and update the global state. Finally, smart contract platforms have a more complicated global state than those with just token transfers – this has an impact on the applicability of certain sharding techniques (e.g., transaction vs. state sharding) [71], [114] and therefore potentially reduced scalability.

Checkpointing: Casper the Friendly Finality Gadget (FFG) [37], [38], a checkpointing protocol for Ethereum, introduces a formal scheme for nodes to create finalized blocks that cannot be overturned without a manual reset. This increases the attack resistance of protocols. Furthermore, Casper FFG enables the “minority fork” mechanism described in Section VII-B, which increase the recoverability from majority attacks. However, the voting mechanism that is used for finalization consumes bandwidth, and hence reduces throughput.
Weighted voting: In [131], a modification to PoS protocols was proposed that weighted the consensus power of nodes not just by their staked deposits, but also by their voting history. For example, if they regularly fail to vote for the blocks on the main chain in a timely manner, then their voting power is reduced. This increases throughput because offline nodes are less likely to be elected as block proposers, which means that less time is wasted. By contrast, no new data is added to the blocks and the processing power required for clients to update the voting profiles is negligible. Furthermore, attack resistance is improved as attackers who seek to harm liveness by not voting (correctly) rapidly lose consensus power.

Zero-knowledge proofs: In Bitcoin, tokens are protected through a requirement that any token transfer (through the spending of UTXOs) can only be done if correct signatures are produced for the addresses of the sent tokens. These addresses (and the values of individual UTXOs) can be masked using zero-knowledge proofs, which have been implemented in the Zcash platform. This improves the privacy of users, but at the cost of additional computing power required to process transactions.

Increased block size/frequency: By increasing either the size or frequency of blocks (as done in Bitcoin Cash and Litecoin, which are forks/spin-offs of Bitcoin), the throughput in terms of the maximum number of transactions per second can be increased. However, this also requires that more resources (especially bandwidth) are consumed per second. Since it is more difficult for more smaller parties to handle this additional load, the barriers to entry are increased which leads to worse decentralization. Furthermore, the impact of network latency is increased, and since larger nodes have an advantage in high-latency situations, fairness is reduced.

Microblocks: The Bitcoin-NG proposal [74] simplifies the leader election process in Bitcoin by dividing time into a sequence of epochs, and keeping the same block proposer for all blocks within the same epoch. This eliminates latency effects within the epoch, and therefore allows for increased throughput. The reduction of latency effects also improves fairness. However, the smaller number of nodes that participate in node creation harms decentralization. Although the ability of slot leaders to perform double-spend attacks within epochs is potentially increased, Bitcoin-NG includes measures to counter this such as “poison transactions” and “proofs-of-fraud”, so the total impact on attack resistance is not entirely clear.

Sharding: In Sharding, the requirement that each node in the network maintains the full transaction ledger is relaxed. Systems that successfully implement sharding have potentially much higher scalability and throughput. However, in most existing sharding proposals it is much easier to attack a single shard than the entire system, leading to reduced attack resistance. Despite some important recent work in this area [121], [206], practical implementation remains an active research field with a high potential for future improvements.

In conclusion, the PREStO framework allows for a detailed comparison of different protocols and protocol features that goes far deeper than the safety/decentralization/scalability triad of the blockchain trilemma. As we can see from Table V, this allows us to create a “menu” for protocol designers from which they can choose protocol features that complement or offset each other. As the blockchain ecosystem evolves, this table can be both expanded (in terms of protocols) and refined (in terms of subcategories), making the PREStO a truly dynamic tool for the comparison and evaluation of blockchain protocols.

| Feature                  | Example Protocol | Privacy | Decentralization | Scalability | Throughput | Attack resistance | Resilience |
|--------------------------|------------------|---------|------------------|-------------|------------|------------------|------------|
| Partial solutions        | FruitChains [158], StrongChain [187] | +       | +                | -           | +          | +                | -          |
| Smart contracts          | Ethereum [36]     | +       | -                | -           | +          | -                | -          |
| Checkpointing            | Casper FFG [37], [38] | +       | -                | +           | -          | -                | -          |
| Weighted Voting          | [131]            | +       | +                | -           | +          | -                | -          |
| Zero-knowledge proofs    | Zcash            | +       | +                | -           | -          | -                | -          |
| Increased block size     | Bitcoin Cash     | -       | -                | +           | -          | -                | -          |
| Increased block frequency| Litecoin         | -       | -                | +           | -          | -                | -          |
| Microblocks              | Bitcoin-NG [74]  | -       | +                | +           | -          | -                | -          |
| Sharding                 | Zilliqa [206], OmniLedger [121] | -       | +                | +           | -          | -                | -          |

**C. Identifying Research Challenges & Opportunities**

To provide some insights into possibilities for future work, we summarize the categories and subcategories of PREStO and use them to identify research challenges and opportunities in Table VI. We elaborate on this in the following.

Optimality: In its current stage, the blockchain ecosystem strives to transition to alternative consensus mechanisms that will retain the success of Nakamoto consensus (which includes PoW) while reducing its energy footprint, [102]. Providing formal guarantees of safety and liveness and testing these new consensus mechanisms in large-scale practical settings is an ongoing challenge. The enhanced ability of the next generation of blockchains to enable and secure the widespread execution of smart contracts only adds to the complexity of this already difficult puzzle.
### PRESTO Framework

| Optimality | Stability | Efficiency | Robustness | Persistence |
|------------|-----------|------------|------------|-------------|
| - Liveness | - Incentive compatibility: Participation, Operations, Applications | - Scalability: positive scale effects | - Tolerance of perturbed assumptions/irrational behaviour | - Weak/strong persistent properties |
| - Safety   | - Decentralization: Entry barriers, Distribution of resources | - Throughput rate | - Out-of-Protocol Incentives | - Large scale or majority attacks |
| - Scope    | - Fairness: reward allocation, voting-decision making | - Economy of resources/ energy consumption | - Resilience to attacks: Incentives – Network – Cryptographic level | - Recovery mechanisms: rare events |
| - Privacy features: public/private, permissioned/less | - Benchmarking to centralized solutions | - Benchmarking to centralized solutions | - Governance & long-term sustainability | - Governance & long-term sustainability |

### Design of Blockchain Consensus Layer

Research Challenges – Opportunities

- Selection of design/architecture & Sybil protection (PoW, PoS etc.).
- Exploring the trade-off between safety and liveness.
- Secure execution of smart contracts.
- Design of incentive compatible mechanisms.
- Protection against adversarial behavior.
- Design of scalable properties.
- Reduction of energy footprint.
- Compare blockchain to conventional solutions.
- Protection against collusions, Goldfinger attacks.
- Equilibrate in elaborate adversarial models.
- Defense against 51% attacks, large network partitions
- Design of sustainable blockchains
- Decision of governance schemes

### Summary of Challenges and Current Research

**Stability:** A large part of the recent blockchain literature has focused on analyzing the incentives in traditional PoW protocols. However, many questions still remain unanswered. Are (virtual) miners motivated to support the network’s safety during its ups and downs? What are the vulnerabilities – at an incentives level – of the newly proposed PoS protocols and what are the optimal reward schemes that will safeguard their success? Among others, recent advances in theoretical research highlight that centralization and irregular supply of mining power are two threats that are inherent to PoW protocols, [5], [81], [98]. Partially, these problems stem from cost asymmetries and economies of scale that manifest in energy-consuming consensus mechanisms. Does PoS remedy these problems and is it indeed the next step in blockchain evolution?

**Efficiency:** One pressing – if not the single most important – challenge of the blockchain ecosystem is the issue of scalability. Nakamoto’s PoW proved unexpectedly successful, yet an important hurdle in Bitcoin’s mass adoption is that it does not scale. In particular, even with high energy consumption and increasing environmental externalities, the Bitcoin network cannot process a satisfactory number of transactions per second to compete with the established means of digital payments, like VISA and internet banking. The design of blockchains with scalable properties – in terms of communication overhead, transaction throughput and smart contract execution – remains the cornerstone of research in the Efficiency category.

**Robustness:** To address the problems that arise in terms of Robustness, the blockchain ecosystem first needs to establish solutions that satisfy Stability and Efficiency. The challenges for robustness will push these solutions to their limits and shift the research towards incorporating these systems to every day life. This is where the dynamic nature of PREStO shines most. Given the state of the art when Stability and Efficiency will have been achieved – if ever – the still open questions and challenges can appear as new categories in the framework’s Robustness axis. This perspective supports PREStO’s current development as a dynamic tool whose purpose is to aid the communication between the increasingly diverse stakeholders of the blockchain community.

**Persistence:** The open challenges in this category will also become more relevant at more mature stages in the mass adoption of blockchain technology. With the exact nature of next-generation blockchains and their main applications still unclear, the issues of governance and long-term sustainability will remain at the forefront of blockchain research. Are public blockchains indeed going to support global payment systems and currencies? And if blockchains and cryptocurrencies indeed succeed in this task, what will be the stance of governments, legislators, regulators and policy makers? How will the major technology firms react to the new technology and how will the existing banking system adapt to the changing reality?

The above provides a non-exhaustive list of the open problems that are currently puzzling the blockchain community. Yet, it highlights the comprehensive coverage of the proposed PREStO framework. Owning to its modular structure, PREStO can be expanded or modified to accommodate advances or additional research opportunities in the future of blockchain protocols. Accordingly, it can be used to track the evolution of the blockchain ecosystem and structure the communication between its diverse participants who range from protocol designers, technology experts and end users to academics, corporate managers and strategic investors.

### IX. Conclusions & Future Research

Summing up, the PRESTO framework sees protocols as multi-dimensional objects with the following cascade of goals. First, optimality requires that the protocol solves the problem that it is defined to address, otherwise there is no good reason to deploy it and the designer should go back to the drawing board.
features of future blockchain protocols.

Next, efficiency requires that resources are used as efficiently as possible (e.g. time, space, network bandwidth, energy, randomness, etc.). Given an optimal, stable and efficient protocol, the next steps are to consider more elaborate behavioral models from the perspective of the agents. These entail robustness and persistence which measure the resilience of the established equilibria in less idealized settings, and the performance of the blockchain in highly perturbed conditions, respectively.

The exploration of these trade-offs is an area for multidisciplinary research that relies on the synthesis of ideas from game theory, cryptography and theoretical computer science. In this direction, PREStO can be used as a dynamic framework to structure the communication between researchers with diverse backgrounds and to accommodate increasingly more elaborate features of future blockchain protocols.

REFERENCES

[1] J. Abadi and M. Brunnermeier. Blockchain Economics. Working Paper 25-407, National Bureau of Economic Research, December 2018.

[2] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed Computing Meets Game Theory: Robust Mechanisms for Rational Secret Sharing and Multiparty Computation. In Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing, PODC ’06, pages 53–62, New York, NY, USA, 2006. ACM.

[3] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth. BAR Fault Tolerance for Cooperative Services. In Proceedings of the Twentieth ACM Symposium on Operating Systems Principles, SOSP ’05, pages 45–58, New York, NY, USA, 2005. ACM.

[4] M. Apostolaki, A. Zohar, and L. Vanbever. Hijacking Bitcoin: Large-scale Network Attacks on Cryptocurrencies. CoRR, abs/1605.07524, 2016.

[5] N. Arnosti and S. M. Weinberg. Bitcoin: A Natural Oligopoly. In A. Blum, editor, 10th Innovations in Theoretical Computer Science Conference (ITCS 2019), volume 124 of LIPIcs, pages 5:1–5:11, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[6] N. Arnosti and S. M. Weinberg. Bitcoin: A Natural Oligopoly. In 10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA, pages 5:1–5:19, 2019.

[7] G. Asharov, R. Canetti, and C. Hazay. Toward a Game Theoretic View of Secure Computation. J. Cryptol., 29(4):879–926, Oct. 2016.

[8] K. J. Astrom and R. M. Murray. Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, Princeton, NJ, USA, 2010.

[9] S. Athey, I. Parashkevov, V. Sarukkai, and J. Xia. Bitcoin Pricing, Adoption, and Usage: Theory and Evidence. Research paper no. 16–42, Stanford University, Graduate School of Business, 2016.

[10] N. Atzei, M. Bartoletti, and T. Cimoli. A Survey of Attacks on Ethereum Smart Contracts SoK. In Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204, pages 164–186, New York, NY, USA, 2017. Springer-Verlag New York, Inc.

[11] S. Azouvi and A. Hicks. SoK: Tools for Game Theoretic Models of Security for Cryptocurrencies. arXiv preprint arXiv:1903.06595, 2019.

[12] S. Azouvi, A. Hicks, and S. J. Murdoch. Incentives in Security Protocols. In V. Matyáš, P. Svenda, F. Stajano, B. Christianson, and J. Anderson, editors, Security Protocols XXVI, pages 132–141, Cham, 2018. Springer International Publishing.

[13] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar. On Bitcoin and Red Balloons. In Proceedings of the 13th ACM Conference on Electronic Commerce, EC ’12, pages 56–73, New York, NY, USA, 2012. ACM.

[14] C. Badertscher, J. Garay, U. Maurer, D. Tschudi, and V. Zikas. But Why Does It Work? A Rational Protocol Design Treatment of Bitcoin. In J. B. Nielsen and V. Rijmen, editors, Advances in Cryptology – EUROCRYPT 2018, pages 34–65. Springer International Publishing, 2018.

[15] M. Balcan, S. Krehbiel, G. Piliouras, and J. Shin. Minimally invasive mechanism design: Distributed covering with carefully chosen advice. To appear in the Proceedings of the 51st IEEE Conference on Decision and Control (CDC), 2012.

[16] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan. Proofs of Useful Work. Cryptology ePrint Archive, Report 2017/203, 2017. [online].

[17] S. A. Banducci and J. A. Karp. Perceptions of Fairness and Support for Proportional Representation. Political Behavior, 21(3):217–238, 1999.

[18] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis. Consensus in the Age of Blockchains. arXiv e-prints, page arXiv:1711.03936, 2017.

[19] M. Bartoletti, S. Lande, and A. S. Padlla. A Proof-of-Stage Protocol for Consensus on Bitcoin Subchains. In M. Brenner, K. Rohloff, J. Bonneau, A. Miller, P. Y. Ryan, V. Teague, A. Bracci, M. Sali, F. Pintore, and M. Jakobsson, editors, Financial Cryptography and Data Security, pages 568–584, Cham, 2017. Springer International Publishing.

[20] W. Baumol. Welfare Economics and the Theory of the State. Harvard University Press, Cambridge, Massachusetts, 1952.

[21] A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization: analysis, algorithms, and engineering applications. SIAM, 2001.

[22] I. Bentov, A. Gabizon, and A. Mizrahi. Cryptocurrencies Without Proof of Stake. In J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, editors, Financial Cryptography and Data Security, pages 142–157, Berlin, Heidelberg, 2016. Springer International Publishing.

[23] I. Bentov, P. Hubáček, T. Moran, and A. Naor. Tortoise and Hares Consensus: the Meshwork Framework for Incentive-Compatible, Scalable Cryptocurrencies. IACR Cryptology ePrint Archive, 2017:300, 2017.

[24] I. Bentov, R. Pass, and E. Shi. Snow White: Provably Secure Proofs of Stake. IACR Cryptology ePrint Archive, 2016:919, 2016.

[25] B. Biais, C. Bissiére, M. Bouvard, and C. Casamatta. The blockchain folk theorem. IDEI Working Papers 873, Institut d’A&l’conomie Industrielle (IDEI), Toulouse, May 2017.

[26] A. Bierer, A. Cimatti, E. M. Clarke, O. Strichman, Y. Zhu, et al. Bounded model checking. Advances in computers, 58(11):117–148, 2003.

[27] J. Bonneau. Hostile blockchain takeovers (short paper). In Proceedings of the 5th IFCA Workshop on Bitcoin and Blockchain Research, 2018.

[28] J. Bonneau, E. Felten, S. Goldfeder, J. Kroll, and A. Narayanan. Why Buy When You Can Rent? Bribery Attacks on Bitcoin-Style Consensus. In Financial Cryptography Workshops, 2016.

[29] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. Felten. SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In 2015 IEEE Symposium on Security and Privacy, pages 104–121, 2015.

[30] J. Brown-Cohen, A. Narayanan, C.-A. Pomsas, and S. M. Weinberg. Formal Barriers to the Longest-Chain Proof-of-Stake Protocols. arXiv e-prints, 2018.

[31] L. Brünjes, A. Kiayias, E. Koutsoupias, and A.-P. Stouka. Reward Sharing Schemes for Stake Pools. arXiv e-prints, page arXiv:1807.11218, 2018.

[32] Business Wire. Algorand secures $62m in funding and announces appointment of executive team, 2018. Available [online]. [Accessed: 26-2-2019].

[33] V. Buterin. The triangle of harm, 2017. Available [online]. [Accessed: 3-9-2018].

[34] V. Buterin. Disencouragement Attacks, 2018. Available [online]. [Accessed: 13-6-2019].

[35] V. Buterin. Ethereum 2.0 spec – Casper and sharding, 2018. Available [online]. [Accessed: 30-10-2018].

[36] V. Buterin et al. A next-generation smart contract and decentralized application platform, 2014. Available [online]. [Accessed: 13-4-2018].

[37] V. Buterin and V. Griffith. Casper the friendly finality gadget. arXiv preprint arXiv:1710.09437, 2017.

[38] V. Buterin, D. Reisigeren, S. Leonards, and G. Piliouras. Incentives in Ethereum’s Hybrid Casper Protocol. In ICBC 2019, May 2019.

[39] C. Cachin. Architecture of the Hyperledger blockchain fabric. In Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.

[40] C. Cachin and M. Vukolić. Blockchain Consensus Protocols in the Wild. CoRR, abs/1707.01873, 2017.

[41] Cardano. [website]. [Accessed: 26-2-2019].

[42] M. Carlsen, H. Kalodner, S. M. Weinberg, and A. Narayanan. On the Instability of Bitcoin Without the Block Reward. In Proceedings of
[196] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, and Y. Wen. A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access*, 2019.

[197] E. Wiki. On sharding blockchains, 2019. Available [online]. [Accessed: 15-12-2019].

[198] T. Wong. An application of game theory to corporate governance. *Omega*, 17(1):59–67, 1989.

[199] K. Wüst and A. Gervais. Do you need a blockchain? In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 45–54. IEEE, 2018.

[200] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander. Where Is Current Research on Blockchain Technology? A Systematic Review. *PLoS ONE*, 11(10):e0163477, 2016.

[201] V. Zamfir. Personal communication.

[202] A. Zamyatin, N. Stifter, P. Schindler, E. R. Weippl, and W. J. Knott-tenbelt. Flux: Revisiting Near Blocks for Proof-of-Work Blockchains. *IACR Cryptology ePrint Archive*, 2018:415, 2018.

[203] R. Zhang and B. Preneel. Publish or Perish: A Backward-Compatible Defense Against Selfish Mining in Bitcoin. In *CT-RSA*, 2017.

[204] R. Zhang and B. Preneel. Lay Down the Common Metrics: Evaluating Proof-of-Work Consensus Protocols’ Security. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[205] Z. Zheng, S. Xie, X. Chen, and H. Wang. Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services (IJWGS)*, 14(4):352–375, 2018.

[206] Zilliqa. Zilliqa. [website]. [Accessed: 26-2-2019].

[207] Zilliqa Team. The Zilliqa Technical Whitepaper, 2017. Available [online]. [Accessed: 26-2-2019].

[208] A. Zohar. Securing and Scaling Cryptocurrencies. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 5161–5165, 2017.