(**a preprint version to be submitted to preprint servers**)
By Shymaa M. Arafat
Independent Associate Professor in Computer Science & Engineering
shar.academic@gmail.com, shymaa.arafat@gmail.com

A Study of Blockchain Consensus Protocols

Abstract. When Nakamoto invented Bitcoin, the first generation of cryptocurrencies followed it in applying *POW (Proof of Work)* consensus mechanism; due to its excessive energy consumption and heavy carbon footprints, new innovations evolved like *Proof of Space*, *POS (Proof of Stake)*, and a lot more with many variants for each. Furthermore, the emergence of more blockchain applications and kinds beyond just cryptocurrencies needed more consensus mechanisms that is optimized to fit requirements of each application or blockchain kind; examples range from *IoT (Internet of Things)* blockchains for sustainability applications that often use variants of *BFT (Byzantine Fault Tolerance)* algorithm, and consensus needed to relay transactions and/or assets between different blockchains in interoperability solutions. Previous studies concentrated on surveying and/or proposing different blockchain consensus rules, on a specific consensus issue like attacks, randomization, or on deriving theoretical results. Starting from discussing most important theoretical results, this paper tries to gather and organize all significant existing material about consensus in the blockchain world explaining design challenges, tradeoffs and research areas. We realize that the topic could fit for a complete textbook, so we summarize the basic concepts and support with tables and appendices. Then we highlight some case examples from interoperability solutions to show how flexible and wide the design space is to fit both general and special purpose systems. The aim is to provide researchers with a comprehensive overview of the topic, along with the links to go deeper into every detail.
Keywords: Blockchains, consensus, consistency, slashing, BFT, POW, POS, VRFs, VDF, Quasi-Permissionless.

1 Introduction

Blockchains inherit the ***Byzantine Generals Problem*** from distributed information systems that is usually addressed using ***state machine replication (SMR)*** approach; when a distributed information system replicates its servers to tolerate malfunctioning or malicious servers (referred to as *Byzantine*), it is supposed to answer enquiries and execute concurrent update requests to its replicated servers in a ***consistent*** and ***live*** manner [1,2]. When more than one transaction competes to write on a distributed database record, this is quite similar to when block proposers each constructs a block with all needed certificates and references to previously delivered blocks and then compete to append their constructed block into the blockchain. Also similar to those competing transactions, choosing a certain block may change the Blockchain status in a way that renders other competing blocks invalid.

However, the replicated nodes that performs the validation are now 1) anonymous, 2) dynamic, and 3) scattered around the world with different communication times

[3,4]; this increases the possibility of a node being malicious[1] and makes the protocols used to communicate and negotiate a decision more complicated. Fig.1 is a modified adoption from the presentation of [3]; the paper contains detailed figures about the geo-distribution ratios of Ethereum validators.

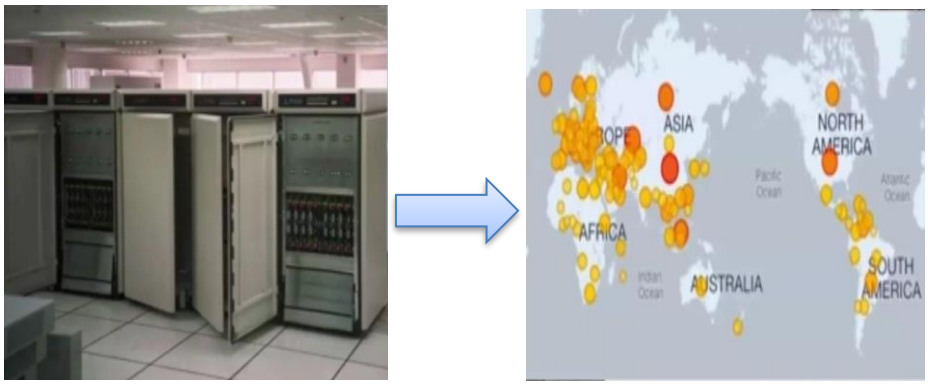

Fig.1: Instead of known servers locked in a room, we now have anonymous nodes that go on/off at any time with different capabilities and different communication times between them (adopted from [3], where the authors geographically distributed the nodes in their experiments)

The mechanism and criteria on which a block is selected from all proposed competing blocks is called the ***consensus protocol***. Typically, a consensus protocol involves a ***fair*** selection criterion between valid blocks (like heaviest computation for POW or probabilistic stakes ratio for POS). Also, an ***efficient reliable*** message exchange protocol[2] to negotiate the selection between participating nodes where a malicious node may broadcast a wrong message or not broadcast at all hoping to cause a DoS attack. Finally, the usual *lock-commit* paradigm known in distributed databases [5] maybe mapped to forks handling strategies in single blockchains, but it **is** only part of cross-chain consensus protocols when things get more complicated (more conflict possibilities due to transactions dependencies) [6].

The literature holds extensive review studies that either surveys different existing blockchain consensus rules or systemizes available material on a specific consensus issue like attacks, randomization source, and a lot more. In this paper we try to

[1] Byzantine now refers to malicious nodes not to possibility of malfunctioning (crashing) as in distributed databases.

[2] Although beyond the scope of this paper, the interested reader may find new research on theoretical bounds for *message exchange complexity*, which usually involves a broadcast, and different *validity definitions* on https://arxiv.org/abs/2301.04920 (last version June 2023). Unless mentioned otherwise, in this paper we assume a fixed set of nodes exchange messages using a variant of the Dolev Strong protocol (https://decentralizedthoughts.github.io/2019-12-22-dolev-strong/, and [7/lec2&3]); also that broadcasting is done using *Byzantine Reliable Broadcast* (*BRB*) which will mention some exceptions in section 4.1.

present a consolidated view of the complete picture, naturally focusing on only some of its details, and at the same time guiding the reader to all available material we have encountered. We also encountered many important papers with theoretical nature that prove different fault tolerance thresholds and impossibility results; we recognized the need to collect and summarize those results explaining the difference and significance of each. Hence, section 2 explains more formally the main differences between consensus settings in different environments and summarizes the most important theoretical results. Then, section 3 summarizes most important consensus criteria and/or protocols kinds used in different blockchain applications, while section 4 discusses other research directions in the consensus area that evolved with time. After that, we put some focus on consensus in blockchain interoperability solutions in section 5 by discussing design issues and introducing some examples of consensus in interoperability solutions; finally, section 6 concludes the paper. We have also added some formal definitions in Appendix A, some extra deeper details about leader/committee election process and the randomization involved in Appendix B, and Appendix C is a summary table of discussed criteria in most famous blockchains.

2 Consensus From Distributed Information Systems into Blockchains

This section delves into problematic research areas that appear in blockchain consensus as compared to distributed information systems, which could be roughly viewed as permissioned system clock synchronized blockchain. This mostly theoretical research has been the concern of many researchers [3,7,8,9]; more research mixing theoretical foundations and commercial protocols will be mentioned in the details, and we also recommend watching Vitalik Buterin talk [10] about some misconceptions in the field. Fig.2 is a step-by-step explanatory diagram, while Table.1 at the end of the section summarizes some thresholds and impossibility/possibility results; formal definitions of the used terms can be found in Appendix A.

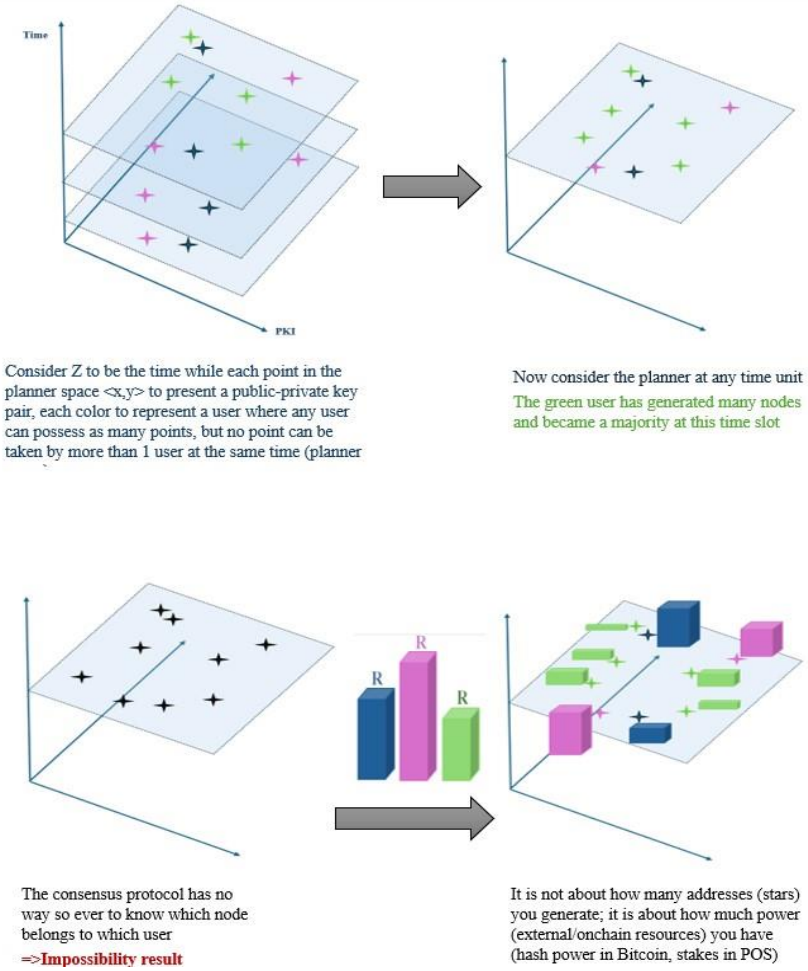

Fig.2: a graphical illustration of the effect of anonymity, time, and resources on blockchain consensus protocols. Note that the black stars diagram (impossibility result) applies also to DA and QP settings because the protocol still has no way of telling how many registered IDs belong to the same individual, so it is the resource restrictions that makes it "possible".

2.1 Participating Nodes

The dynamically varying number of participating nodes makes the consensus problem more complicated in blockchains [7/lec9, 8,9,11] since the committee that should agree upon a block is not predefined…

1) Decentralization could be compromised if the number of online nodes is not large enough, while very large numbers could result in excessive message passing and scalability issues[3].
2) we cannot possibly guarantee a threshold ratio for Byzantine nodes if nodes can freely go on and off at any time (so called the ***Fully Permissionless (FP)*** setting if completely anonymous like Bitcoin, or the ***Dynamically Available*** (***DA)*** setting if conditioned by a pre-registration like POS as will be further explained in detail), Therefore, either a new innovation is deployed, like *POW longest chain* consensus in Bitcoin, or a selection phase is first applied to select a committee (group) and then a proprietary *BFT* protocol is applied to the fixed size committee (we refer to those blockchains as working in the ***Quasi Permissionless*** (***QP***) settings); in all cases consensus is only possible through resource restriction as shown in Fig.2, and a well-studied economic incentive model to keep the committee active and maintain a certain byzantine threshold. Example strategic selection phases vary from POW *Keyblock* competition like in Byzcoin [12,13], to an application based criteria like the Helium blockchain [14] consensus protocol where nodes compete in submitting *Proof of Coverage* (internet coverage) [15] to select a group then a variant of BFT is applied (*Honeybadger BFT* then migrated with Solana which uses a version of PBFT). Also, the use of the Tendermint protocol in Cosmos and Terra ecosystems [16].

2.2 Resources

As shown in Fig.2, consensus in blockchains is only possible through resource restriction; naturally users' resources are constrained by their money value cost.
-In Bitcoin and similar POW blockchains the hash power is the resource; in Proof of Space blockchains storage is the resource. Those are ***external resources*** that costs money, and the probabilistic security depends on the infeasibility of an adversary controlling more than 50% of total existing hash power. However, the fact that even with >50% monopoly, double spending or sybil attacks will decrease the coin price, and hence is not profitable, ignores the possibility of a malicious scorched ground attack; hence, safety is always probabilistic [9].
-Staked values in POS blockchains (still restricted by the coin cap) are the most famous ***on-chain resource***; those blockchains that use POW leader election pre-phase (again restricted by hash power) like Byzcoin [13], Hybrid [17], and Solida [18] are also considered an on-chain resources [8].
-Another categorization we notify the reader to keep in mind when discussing the wide variety of consensus criteria is ***resource reusability***; it is true the same resource can only be used once in each consensus round, but some resource kinds allow for

[3] For example, Bridges and cross chain solutions sometimes lack independent validators that jeopardizing 2-3 keys may allow severe attacks (https://limechain.tech/blog/biggest-blockchain-bridge-hacks/), while Ethereum POS (https://notes.ethereum.org/@vbuterin/single_slot_finality#What-are-the-issues-with-validator-economics) now deploy hierarchal aggregator-committee structure and keep improving its protocols to cope with its ~ 440K validators set.

several usage attempts that could be taken advantage of by malicious actors. For example, a malicious miner in Bitcoin cannot use the same computation power to create a valid block for the main chain, and also attach the same block (or another block with the same header) to a secret fork to start a selfish mining attack (or a double spend) because the puzzle solution (nonce) depends on the hash of the previous block and the included transactions. On the other hand, the block creation process itself in POS or even proof of space blockchains doesn't require extensive computation power; hence opens the door for malicious actors to try attaching the same block (or another block with the same header) secretly wherever it is possible without extra cost allowing forks to keep growing (will be discussed further in section 3.7).

2.3 Time Synchronization

One may view blockchains as an asynchronous environment since there is no unified system clock, where deterministic consensus is impossible in the asynchronous setting[4]. Fortunately, blockchains are described as Δ-***synchronous*** (synchronization is achieved within time alpha) [7], since all Blockchains and their applications define a *finality time* after which most nodes have received (confirmed) the final block. The family of ***Byzantine Fault Tolerance (BFT)*** style protocols (from 1999 [19]) operate in *rounds* in the ***partially synchronous model***, by *electing a round leader* to start the broadcast of a block which should be validated by all nodes, and the protocol can tolerate (*guaranteed to terminate achieving a consensus*) within a maximum threshold of ***33%*** malicious or faulty (so called *Byzantine*) nodes; *Tendermint* [16] as depicted in Fig.3 is a classical blockchain era example, while *Honeybadger* [20] and [1] are consensus protocols that could work in an asynchronous setting. In fact, only a subset of blockchains (*QP*) can theoretically work in the partial synchronous setting; it is the large enough number of available nodes in famous blockchains that keep them practically operating at network speed (*live*) and hence the *alpha synchronous* assumption remains valid[5]. Appendix A provides the formal definitions of all time-related terms.

---

[4] M. J. **F**ischer, N. A. **L**ynch, and M. S. **P**aterson, Journal of the ACM, 1985, https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf; ***GRANDPA, G****host-based* ***R****ecursive* ***AN****cestor* ***D****eriving* ***P****refix* ***A****greement*, paper [22] proved in 2020 that the ***FLP*** impossibility result remains valid (still impossible) even with finality gadgets.

[5] Tendermint can achieve consistency and liveness in the partial synchronous model because nodes keep their memory (1st quorum certificate, or *Precommit* in Fig.3) from previous unfinished rounds [70/9], this gets little complicated when applied to a *DA* setting where participating nodes change in the next round like in Algorand for example [70/12.18].

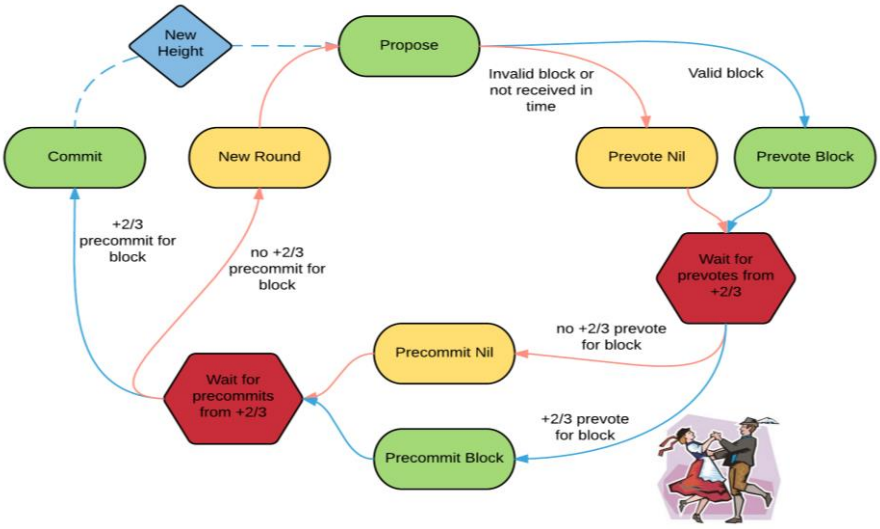

Fig3. The Tendermint protocol rounds (taken from [16])

2.4 Time to Finality (TTF)

In Fully permissionless blockchains, typically POW, the term ***time-to-finality*** defines roughly the time needed for a transaction to be finalized with overwhelming probability; i.e., if there were any forks, the probability is negligible (exponentially decreases with each block) that the block containing the transaction will be reverted and the other branch in the fork will be selected[6]. Finality time for Bitcoin is ~ 1 hour (6 blocks), for Solana it is ~ 12 seconds (very faster block production rate of 300-800 ms), and for Ethereum it is ~ 13 minutes (2 epochs); Algorand provides instant finality because blocks are never reverted.

Seeking more time efficiency and faster block production rate, blockchains deploy the ***Friendly Finality Gadget (FFG)***[7] [21,22,23] concept where block production is decoupled from voting (leader election); i.e., nodes do not wait for the voting results, they just keep producing blocks over the last finalized (voted upon) block according to some chain selection rule between possibly resulting forks. Ethereum is an example of applying FFG, where blocks are produced in parallel and a fork choice rule,

[6] Sometimes referred to as escrow time for wallets and/or large payments where the required action after payment is withheld until enough confirmation is received to the block containing the payment transaction.

[7] The term *"finality gadget"* first appeared in the Casper FFG paper [21] 2017 and Ethereum POS uses it with LMD **G**HOST (*Latest Message Driven - Greedy Heaviest Observed Subtree*) fork choice of the heaviest chain, so called ***Gasper FFG***; (https://inevitableeth.com/home/ethereum/network/consensus/pos) provides a simple illustration with block diagrams explanation, while GRANDPA in [22] (used by Polkadot) elaborates more on the theoretical bases.

confirmation rule, of heaviest chain is added [23]. Note that this is different from Byzcoin [13] and alike blockchains which also decouple leader election from block production, but only the elected leader produces blocks

2.5 Status-quo Summary & Ongoing Research

Table 1, inspired by the tables and videos in [8] and [3], summarizes different blockchain kinds with their byzantine thresholds on different modes and performance metrics according to proven possibility and impossibility results.

Less restrictions and hence more complexity and impossibility results ←

	Fully Permissionless	**Dynamically Available**	**Quazi Permissionless**	**Permissioned**
Description	Pure anonymity; the protocol has absolutely no knowledge of participating nodes	Registration before participation; the protocol has a list of "*possibly participating" (active)* IDs (assumes at least 1 honest is always active)	A committee is selected and assumed to be active from registered IDs	A fixed set of known IDs that are assumed to be always active
Examples	Bitcoin	Ouroboros POS, Snow White	Algorand, Byzcoin	IoT blockchains
Deterministic	Probabilistic	Can be deterministic		
EAAC (Expensive to Attack in the Absence of Collapse [9])	F < 1/2 N	F < 1/2 N	F < 2/3 N Consistent & live 5/9 N≤F< 2/3N trade consistency for liveness	
Synchronous	All above rows assume synchronous setting			
Partial Synchronization Latency ≤ a maximum delay GST (Global Stabilization Time)	Can't operate in partial synchrony		Can operate in partial synchrony for N ≥ 3F+1 (i.e., F < 1/3N)	
Accountable (can find malicious nodes, Proof of	Can't be		Can be Accountable	

Guilt [3]), called *PISA*[8] in [24]		
Optimistically Responsive Latency [8] is function of message delay (can operated at network speed)	Can't be	Can be Optimistically Responsive Hotstuff [25], Vena [26] and Simplex consensus in the partial synchronous mode

Table 1: a table summarizing blockchain kinds and properties. Note that the authors in [3] prove larger values by dividing malicious nodes into different kinds (see section 4.1)

Section 4 will go through different research directions on blockchain consensus, but it should be clear by the end of this section that challenges [27] on the steps highlighted in this section include, but not limited to, *time* and/or *bandwidth* especially with interoperability and cross chain applications when there are more than one blockchain involved [section2 in 28,29,30], the number of *compromised (Byzantine) nodes* the network could tolerate, the degree of *fairness* and *decentralization* [31] in the committee selection, or handling *large committees (validator sets)* [23,26]; the economic incentives to maintain the participating nodes and byzantine thresholds was discussed from different aspects in [3,9,23] and will be further discussed in section 4. Also, the idea of *decoupling* some of the consensus steps was tried several times; Casper FFG [21] is the most deployed practically, we have shed some light on Byzcoin [12,13], and there were other attempts like [32]. On the other hand, although the idea of *coupling* or *mixing sharding with consensus* [33] was dropped from application on Ethereum, it remains a useful read and was discussed again in another paper [34]. Finally, theoretical results can be found in [3,8,9,35,36,37] and textbook style material [7,38].

3 Main Types of Consensus Protocols Used in Blockchains

There are endless variants of consensus protocols out there; in this section we will try to cover the main consensus mechanisms in chronological order of their appearance along with their important subvariants. Previous efforts include peer reviewed papers like [27,39-44], and an encyclopedia site [45].

[8] Although the paper in [24] is concerned about broadcasting details and do not address blockchain consensus specifically, it defines ***Provable Identifiable Selective Abort*** and "*certificate of cheating*" to achieve the same requirements as *Accountability* and *Proof of Guilt*; it also defines ***Guaranteed Output*** in analogy to *Optimistic Responsiveness* and introduces "*certificate of non-responsiveness*" to function as *dummy blocks* in Vena and Simplex consensus [26].

3.1 Byzantine Fault Tolerant (BFT)

The traditional *BFT* [6], or a variant of it, remains suitable for private permissioned and consortium blockchains. For example, Estonia governmental *KSI* blockchain uses a proprietary *BFT* protocol [46]; the default choice for sustainability applications blockchains is usually a variant of ***Practical BFT (PBFT)*** [47], as will be further detailed in 3.12. ***Federated Byzantine Agreement (BFA)*** [48] variants are little different since they allow different nodes to have different trust zones resulting in overlapping *quorum* sets (detailed in 3.13); FBA is often used in payment-protocol-based blockchain platforms such as *Stellar* [49]. Also, as mentioned above, *BFT* type protocols are usually the second phase after committee selection in many blockchains, with ***Tendermint*** [7] being the most celebrated example.

3.2 POW

The basic idea originated in the 90's as a protection from spam and DoS attacks by performing some non-trivial amount of computation that outweighs the expected attack revenue; POW failed in spam email protection [50] but became a breakthrough in cryptocurrency consensus since Bitcoin 2008 [51] as the heavy computation can prevent (with overwhelming probability) different attacks (*double spending*, *selfish mining*, *sybil attacks*)[9] unless attackers possess 50% of all existing computational power. Basically, miners compete to solve a cryptographic puzzle with a predefined difficulty level, combine it with their proposed block and try to append it to the chain; from the different possible future chains (new system state) nodes elect the longest chain (more accurately, the one with the heaviest computation). Despite all its merits, the high energy consumption along with its carbon footprint remains a significant argument against POW; attempts to decrease it by performing the computation only on certain key blocks include *Bitcoin-NG* [52] and *Byzcoin* [13] discussed previously.
3.3 P

3.3 Proof of Space/Capacity

In 2013, [53] suggested dedicating a significant amount of memory or disk space as a greener alternative to POW computation. First crypto applications of proof of Storage include, [54], *Signum* (2014) and *Spacemint* (2015), while *Chia* (2018) deploys a variant called ***Proof of SpaceTime*** which necessitates reserving the storage for an

[9] ***Double spending*** is trying to spend the same coins more than once, ***selfish mining*** is trying to mine several blocks secretly to produce a fork with a longer chain than the existing one (thus invalidates all TXs and mining rewards after the fork), and ***Sybil attack*** is for a single attacker to create a vision of multiple nodes creating a false majority (recall the black stars in Fig.2); however, the POW mechanism protects from all such attacks unless the attacker controls at least 50% of all available hash power worldwide (recall resource restriction from 2.4 and the bars in Fig.2).

amount of time[10]. Naturally, a variant of the latter is most suitable for distributed storage systems like ***Filecoin***[11] which has a 2-phase protocol (*Expected Consensus EC* [55]); a probabilistic Byzantine fault-tolerant consensus protocol runs a leader election among a set of storage providers to submit a block every time epoch, where the likelihood of being elected depends on how much provable storage a miner contributes to the network. As we clarified in section 2.2 space as resource is vulnerable to possible "resource reusability".

3.4 Proof of Activity

First suggested in 2014 paper with *Litecoin* Creator, Charlie Lee, as one of its authors and could be viewed as an improvement over POW. Miners solve an easier puzzle for empty blocks, then a set of validators from coin holders (an early variant of POS) verify the transactions part of the block and reward is split between miners and validators [56]; such an arrangement also harden the 51% hash power attack in POW to necessitate 51% malicious validators (coins holders) in addition. Proof of Activity is used in *Decred* (*https://decred.org*) and *Espers* (*https://espers.io*); a different variant that shares the same name, but targets incentivizing participation, appeared recently in *Fastex* [57] uses a smart contract deployed by validators to evaluate a user's activity level before granting the chance of being a validator or a block producer. *Proof of Contribution* introduced in 2021, [58], is a similar idea.

3.5 Proof of Burn (POB)

Instead of spending the money on energy consumption and mining devices, just remove it from circulation and increase coin scarcity (and hence its price) instead of increasing carbon emissions. The idea so called "burning coins" appeared in 2019 [59], and was first used in *Slimcoin* (*https://slimcoin.info*). In fact, burning crypto got more popularity in burning tokens of different exchange tokens [60], cross chain asset transfer, and of course burning ratio of the fees like in the *Near* protocol and Ethereum *EIP-1559*. Also, a recent 2024 paper [61] suggests upgrading cryptocurrencies with new tokens using *POB* via multi-currency auction.

[10] Check appendix A, page 72, in [9] for a more theoretical enlightenment.

[11] Wikipedia includes *arweave* too (a distributed storage that follows a structure called *blockweave* similar to blockchains but not one); however, we have found that *arweave* used *Proof of Access* for some time (https://www.reddit.com/r/a:t5_67b622/comments/u37ldb/arweave_consensus_protocol_poa/) and seems to switched to another protocol inspired by *Perma coin* 7 months ago as shown in (https://github.com/ArweaveTeam/arweave-standards/blob/master/ans/ANS-103.md)

3.6 Proof of Elapsed Time (PoAT)

Was originally developed by Intel engineers and contributed to *Hyperledger Sawtooth* [62] to replace the POW heavy computation by generating a random waiting time using a *trusted execution environment (TEE)* and select the least waiting time as the leader; ***Proof of Luck (PoL)*** [63] on the other hand chooses the one with highest random number as the leader. In 2017, [64] studied *PoET* under a theoretical framework and found that it can be attacked by θ(log log n/log n) fraction of the nodes[12], then [65] introduced a more practical simplified version (***ET***) of both *PoET* and *PoL* in 2021. *PoET*'s GitHub is now archived, and a following experimental *PoET2* was also archived [66]; *PoL* may capture the attention [67] from time to time since its first appearance 2016, but we also have not encountered any real implementation. However, the idea of leader selection based on randomly generated numbers, *Verifiable Delay Functions* ***VDF*** (repeat a certain function sequentially to consume time + a way to quickly verify the correctness of the result and hence the time consumed, which is basically *PoET*) and *Verifiable Random functions* ***VRF*** (a function of candidate's private key and some randomness must fall beyond a certain threshold+ could be verified using the public key; the inverse of *PoL*), is deployed in the consensus mechanisms of a number of layer-1 blockchains [68] including *Chia*, *Algorand*, *Cardano*, *Internet Computer*, and *Polkadot* to randomly select block producers; only those generate their random numbers in software code and except ***Chia***[13] are all considered *POS* variants.

3.7 Proof of Stake (POS)

Although, *Peercoin* [69] claims to be the pioneer cryptocurrency using POS in 2012, the massive use of POS and its many variants started in 2017. The basic idea is that selection is done probabilistically according to staked tokens; some POS blockchains deploy a variant of BFT consensus to the selected committee (ex. Algorand), others apply a variant of the longest chain confirmation rule (ex. Cardano) after selecting a leader [70]. In general, POS consensus is more efficient (higher TX throughput) and consumes less energy; also, it has introduced the possibility of ***slashing*** bad behavior (since stakes are registered, identifying the guilty nodes that signed contradicting messages is possible) although not deployed in all POS blockchains [70/12.20]. Many researchers studied the accountability problem, finding the proof of guilt, and also handling the slashing process such that the adversary cannot affect the recovery phase; [3,9] proved that recovery through slashing at least ***1/3*** of the consistency violation stake can be possible with up to **5/9N** Byzantine nodes in the *Quasi-Permissionless* settings (section 2.1 and Table.1), among other results.

However, due to the change in the economic incentive game model, more strategic manipulations are possible [71]; hence, POS attacks are wider leading to more

[12] That's less than 1/3 the nodes for n>10; i.e., worse than *BFT*

[13] Chia *proof of time* could be considered a variant of *PoET*, and does use VDFs (https://docs.chia.net/proof-of-time/)

complicated and varying designs that some can only tolerate only **27.8%**[14] Byzantine ratio [72,70/12.19] in some cases due to a possible ***grinding attack***. The attack make use of *resource reusability* feature (as discussed in section 2.4) and generally fall (with some of its mitigations as well) in the larger threat of ***predicting or simulating the randomization*** process in selecting the leader who proposes the block (or the BFT committee members) which may magnifies MEV threats and leads to different kinds of manipulation like reorgs, delaying finality, or even DoS attacks targeting a certain elected leader. The usual solution is to use *Verifiable Random Functions (**VRF**)*s or *Verifiable Delay Functions (**VDF**)*s as just mentioned above [68]; see Appendix B for more details. If *slashing* is the only defense, ***nothing at stake*** attacks can wait enough to build a reputation and then withdraw their staked tokens before revealing the attack; Cardano *Ouroboros*[15] protocol [73], and most followed it, uses a ***warm up epoch*** (to fix the key before calculating the VRF) and a ***withdrawal (cooldown) epoch*** where tokens remain locked without their owner being a candidate for selection, and the withdrawal time must be long enough for the last block the staker participated in to be finalized. Another threat comes from the fact of possible *stake reusability*; unlike *selfish mining* in POW where hash power can only be used once, forking and creating an alternative longer chain (or even different possible forks) at any desired block in history costs nothing but simulating the randomization selection process. ***Long range attacks*** may buy or steal old withdrawn keys with high credibility to conduct the simulation and create an alternative chain to deceive nodes about the chain history; the authors of [8] proved that every POS protocol that uses only *time-malleable* cryptographic primitives is vulnerable to long range attacks and hence *Verifiable Delay Functions (**VDF**)*s can be used, among other possible options, to defend them [70/12.21-22]. Ethereum *Casper FFG* [21], as an example, publishes the last checkpoint (finalized block) in the official website for new nodes; in addition it assumes that nodes will "log on" and gain a complete up-to-date view of the chain at some regular frequency ~ 1-2 months (*out-of-bound communication time* as called by [9] is roughly proportional to the *cooldown* delay), and then never revert a finalized block given so. The literature holds other ideas to hold and confirm verified

[14] In longest chain POS, the attacker chance in satisfying the VRF condition increases exponentially with each new block if tried to feed all possible predecessor blocks into the VRF calculations and can succeed through grinding with an initial Stake > 1/(1+e) where e=2.78; see [72] for the full details, and also (https://eprint.iacr.org/2021/660.pdf) asserts the ratio as an impossibility result and overcomes it by allowing honest nodes to grind too (attach the same block to several virtual chains in a d-distance greedy strategy) to achieve 57% honest threshold (the threshold of 42.7% Byzantine is explained in chia documents too since it shares resource reusability feature and follow longest chain consensus too).

[15] Cardano was earlier in deploying POS than Ethereum and many others, before the fine lines was drawn between POS and delegated POS; specifically, ouroboros design (2017) was followed in many details including having time epochs that are divided to slots, and the warmup & cooldown times. We will also see in section 5 more variants of ouroboros deployed in interoperability solutions.

checkpoints[16] including a public trusted blockchain (Bitcoin) and a group of decentralized servers; deploying AI techniques were also suggested by a few papers.
Ethereum is the most famous POS blockchain that deploys slashing, and its documentations provide a summary of attacks & defense [74]. Tezos also deploys adaptive slashing, while it is still in the future roadmap for Solana as can be found in the table of Appendix C.
Other security aspects of POS like minimizing the risk effect on honest nodes were discussed in [75]; while [76] take the same attitude as [72] and provides a formal comparison of different security aspects of POW versus POS consensus protocols favoring POW at the end.
Another comparison aspect is that some criticize POS as more centralized and less democratized, since selection according to stakes could be viewed as "*making the rich richer*", we believe that hash power or mining devices in *POW*, space in *PoSpace*, internet coverage in *PoCoverage*,...etc. also costs money so it is nearly the same; however, there are mathematical formulas (exponential distribution) to uniformly distribute the leader/committee election in case of unequal stakes [70/12.16-17], while in practice some like *Axeler* [77] avoid vote monopoly by using a *quadratic voting* mechanism to slow down (by the square root) the growth of a validator voting power with the increase of his stake.
Finally, POS has many variants that contains almost all what follows, ***Multi Token POS (MPOS***) [78] could be viewed as the multi-chain version; in nearly all versions, there is a group (committee) selection phase then there is a leader selection phase among the group members.

3.8 Delegated Proof of Stake (DPOS)

In *DPOS* stake tokens are not physically transferred to another wallet, but instead are utilized through a staking service provider in a staking pool [79]. Most sites refer to Cardano *Ouroboros* protocol [73] as a textbook example, also Aptos [80], and TRON [81]; EOS [82] consensus protocol, *EOSIO*, involves a *DPoS* phase to elect the active producers who will be authorized to sign valid blocks in the network, then the actual process of confirming each block until it becomes final (irreversible) is performed in an *asynchronous BFT* manner. Introducing staking pools in *DPOS*, like mining pools in POW, make things easier and increases the number of *Transaction Per Second (TPS)* compared to pure *POS*; could be viewed by some as allowing participation with less than the minimum stake, and by others as concentrating power into staking service providers[17]. The authors in [83] suggest what they describe as a tweak to *DPOS*, ***Preferential delegated proof of stake (PDPoS),*** where block creators have to

[16](https://www.researchgate.net/publication/365185617_Pikachu_Securing_PoS_Blockchains_from_Long-Range_Attacks_by_Checkpointing_into_Bitcoin_PoW_using_Taproot) suggested using Bitcoin in 2022, and (https://eprint.iacr.org/2024/684) in 2024.

[17] According to [83], EOS has only 21 validators and TRON has 27. Also, staking pools in Ethereum allow participation with less than 32 ETH; the largest staking pool *Lido* holds as of today 14/1/2025 ~27.94% of total staked ETH (https://dunecom/hildobby/eth2-stakimg).

stake more tokens in order to validate or assemble TXs sent directly to mainnet, while TXs sent to L2 costs less to users, rewards less to block creators, and maybe delayed to 24hrs; an arrangement that seems like a city for the rich and a city for the poor is claimed to give much higher TPS. It's also worth mentioning that we found a 2024 paper [84] that deploys *DPOS* in an intelligent task scheduling system using blockchains.

3.9 Proof of Authority (PoAu)

Suggested by Ethereum founder Gavin in 2017 [85], a variant of POS, where [85-87] instead of choosing block miners based on their stakes in cryptocurrency tokens, a small group of authorities are selected as transaction validators by their identity or reputation[18] staked in the network. When it started, *Ronin* [88] used Proof of Authority consensus between limited validators to relay *Axie Infinity players* TXs through *Ronin-chains*; now it first selects a set of validators using *DPOS*, then validators take turns producing blocks in *PoAu* manner.

3.10 AI-based Consensus

In 2018 the authors of [89] proposed a conceptual framework, theory, and research methodology for using Artificial Intelligence (AI) techniques in selecting super nodes and random nodes to reach consensus. The introduced motivation was to save the POW computation and avoid monopoly in any form of wealth-based selection. Recent research papers that pursued this thread include *AICon* [90] in 2023 which utilizes the local ML models trained by all nodes to generate a global ML model for selecting winners and distribute rewards fairly between nodes, and *Hybrid Consensus* [91] in 2024 exploring the integration of various Machine Learning (ML) techniques with various known consensus protocols. The literature also holds another innovation that utilizes the computation done in POW alike protocols as a ML training for management tasks in IoT networks deploying blockchains[19] like *Outlier-Aware* [92] and *Proof of Learning* [93]. The authors in [94/section 3.5] present a comprehensive, and recent (Nov 2024), survey of different AI suggested uses.
It's also interesting to note that it works both ways; consensus protocols can be used to perform distributed machine learning efficiently like in [95]; the Flare network team used the *Slush* consensus protocol which is a randomized sampling based consensus protocol.

---

[18] The theme is similar whether it is called Proof of *Reputation/Activity/Authority*, it is some form of evaluation function coded in a smart contract.

[19] The same intuition existed in older research that did not necessarily use AI techniques; examples include *Grid coin* in 2014 (https://bravenewcoin.com/insights/crypto-currency-using-berkeley-open-infrastructure-network-computing-grid-as-a-proof-of-work), *Proof of Useful Work* in 2017 (https://eprint.iacr.org/2017/203), and *Proof of Search* in 2019 (https://ieeexplore.ieee.org/document/8917609)

3.11 Sampling-based Leaderless Consensus

The Avalanche whitepaper [96,97] introduced a family of consensus protocols (*Slush*, *Snowman*, *Snowball*, and *Snowflake*) with a main new feature of leaderless consensus. Consensus in avalanche work in rounds where at each round every node randomly samples ($k < n$) other nodes (less message exchange than regular BFT style protocols) and decides on a transaction with a majority threshold $\alpha \geq k/2$. Favoring safety over liveness, a node stops only when the same decision is repeated β consecutive rounds. For n nodes, this happens in O(log n) when everything is perfect making the approach optimistically responsive and faster than other protocols. However, with a Byzantine ratio O($\sqrt{n}$) the protocols could suffer liveness attacks. Hence, in Sep 2024, AVA Labs introduced *Frosty*, [98], which can tolerate ($f < n/5$) byzantine node by triggering a liveness module to forgo the low communication benefit of sampling in case of attacks. Another research group, [99,100], proposed *Blizzard* which maintains 2 decision counters and simply halts when the difference between them reaches β (not necessarily in consecutive rounds).

3.12 Application Specific Kinds

There are an enormous number of application-optimized consensus innovations. We have mentioned Helium *Proof of internet Coverage* [14], this paper [101] defines *Proof of Absence* for its IoT blockchain system consisting of 10 devices at maximum, the survey in [27] mentions *Proof of Movement* used in healthcare blockchain systems, the Proof of Learning mentioned earlier [93] was developed for a water management system, and there will always be a lot more. A good example is **Hyperledger Fabric** [102/case studies], part of Linux Foundation *(LF)Decentralizedtrust*, which is an open-source enterprise-grade permissioned Distributed Ledger Technology (*DLT*) platform [103] that is widely used in sustainability applications. The deterministic nature of the permissioned case (recall Table.1) allows Hyperledger protocols to assume a trusted environment and handle transactions differently in an *execute-order-validate* manner; i.e., according to the specific application execute the transaction (only once by only a subset of nodes) and check its correctness before endorsing it. Then, there is the *pluggable consensus* feature which allows for the integration of a variety of consensus protocols to do the ordering phase [104]. Older versions (till v2.5) of Hyperledger Fabric mostly use Crash Fault Tolerance (*CFT*) consensus protocols to faster handle only crash/offline failures like *Raft, Kafka* (replaced by *KRaft* now)[20]. However, malicious attack threats necessitate the availability of BFT protocols to handle byzantine failures (recall Fig.1 and footnote 1). Hence, the new version 3 comes with *Smart-BFT* [105] which is a variant of a 2012 simplified version of *PBFT*; there is also a recent paper, [106], that

[20] *Kafka* is now deprecated by Hyperledger Fabric which recommends *Raft*; also, Apachi provides *KRaft* as the new *Kafka* after replacing the *Zookeeper* ordering by Raft ordering (https://docs.confluent.io/platform/current/kafka-metadata/kraft.html)

provides a successful integration of *BDLS-BFT* (a finality gadget old multi value BFT protocol that is based on the Seminal *DLS* protocol [107]). Note that some ledgers that run on Trusted Execution Environment (TEE) still use the CFT protocol like the *Nimble* cloud service project [108].
Another kind of applications that use their specific consensus protocols are payment systems. **Ripple** acts as a real-time settlement and payment system to connect large financial institutions like banks and payment providers [109]. In Ripple Protocol Consensus Algorithm (*RPCA*), [110], each node maintains a Unique Node List (*UNL*) of its trusted subset of nodes. The protocol can tolerate up to $f < n/5$ byzantine nodes; [111] connects the degree of decentralization to the number of malicious nodes, while [112] analyzes and criticizes its security. In fact, one can view Ripple as a variant of avalanche consensus where samples are not random and k could be different for each node; this is reflected in having the same tolerance threshold as *frosty* [98].

3.13 Proof of Agreement POA

Stellar forked from Ripple, target individuals and businesses underserved by the financial system, to have its own consensus protocol. The Stellar Consensus Protocol (*SPC*) [113] is based on FBA and hence that can tolerate $f < n/3$. In *SPC* Proof of Agreement consensus, each node defines its set of trusted nodes called a quorum set and when quorum sets overlap trust is transmitted through the overlapping quorum slices. Formally, the protocol can guarantee agreement, and hence consensus is reached, only if the quorum slices satisfy a validity property called quorum intersection. The security of Stellar has been analyzed in several papers; [114] shows that all nodes cannot run *SPC* (liveness issue) if only two specific nodes (controlled by the foundation) fail. As for applicability, although *SPC* was meant for financial systems [113/use cases], [115] suggested its usage in Vehicular Ad-hoc Networks *VANET*s where vehicles' trust quorum is constructed from nearby vehicles and/or Road Side Units (*RSU*).

3.14 AI-generated Consensus

Finally, it might be worth mentioning that Wikipedia contains an "AI-generated" consensus protocol, [116], *Proof of Identity*. One may expect AI participation in innovating future consensus criteria whether mentioned explicitly or done implicitly at the design phase, especially for specific-application projects.

4 Other Research Directions on Blockchain Consensus

This section tries to summarize the most important research areas in blockchain consensus and the milestone achievements that generated more research threads to continue for the future. Section 4.1 discusses theoretical research that concentrates on

deriving resilient threshold, possibility and impossibility results. Section 4.2 goes through improvements on the BFT family consensus and discusses the DAG approach as a major milestone. Then section 4.3 shed some light on unauthenticated consensus protocols that do not assume a Public Key Infrastructure (PKI). We will dedicate section 4.4 to research getting Bitcoin into the land of on-chain resources (recall section 2.2) with *Bitcoin Staking* and *Stubborn Nakamoto consensus* as spotted examples. Finally, section 4.5 will deduce some continuously needed research areas by going through the protocol development cycle highlighting Ethereum as a case example and its new Beam Chain.

4.1 Theoretical Research

We have already discussed in section 2 theoretical research in [7,8,9] concerning establishing thresholds and/or proving possibility/impossibility results, and summarized its finding in Table 1. Here, we highlight two recent papers which managed to find some way around those results. The first one, ZLB [3] 2024, by classifying byzantine nodes into different kinds of behavior. The second, [117] 2025, studies the impact of observer clients' behavior (whether they are communicating, whether they are always awake) on those thresholds.

The Zero Loss Blockchain (ZLB) Protocol

The authors in [3] argue that in most cases nodes are incentivized to participate (to gain rewards). Hence, they introduce the *Basilic* class of protocols that can support different solutions as long as $N > 3T+d+2q$, where "*T*" for *Byzantine* "*d*" for *Deceitful* that violates safety and "*q*" for *Benign* that violates termination; the so called *BDB* model necessitates some conditions on the voting threshold, *h*, to solve the eventual consensus problem: $ho \in \left(\frac{N}{2}, N\right]$, $(h > d + T)$, $(q + T) \leq (N - h)$. Then, they introduce the Zero Loss Blockchain *ZLB* protocol as a representative of this class and experimentally test its performance through a geo-distributed set of nodes (as in Fig.1). One of their important thresholds is supporting $F < 5/9\ N$ in partial synchrony if "*F*" can be divided into $T < 1/3N$ byzantine and $d=F-T$ alive but corrupt. This is why [9] considered [3]'s 5/9 bound for permissioned case when deriving the same bound for *Post*; since it does not include the "*q*" kind. Table.3 in the *ZLB* paper holds more interesting bounds they were able to prove through this categorization. It is worth mentioning that the paper, [3], was one of three winners of the "*best paper award*" from DSN'24 (Dependable Systems and Networks) conference.

Achievable Security Thresholds Considering Clients' Role

The authors in [117] take a different angle and discuss the achievable security thresholds if regular users that do not participate in consensus (and are not authenticated through PKI infrastructure) were allowed to communicate with each other and broadcast the consensus messages they observe on the chain. They start from an earlier, 2018, Vitalik Buterin blog [118] about how 99% fault tolerance could

be achieved, and then prove different thresholds and impossibility results[21] for 16 possible cases according to the status of both clients and validators; always on or *DA* (they use the term *sleepy*), and silent or communicating off-chain. However, we notice that although the introduced model assumes an adversary can impersonate any number of clients (since clients are not authenticated), the authors never include the ratio of malicious clients or discuss their communication capabilities; regarding Vitalik note, he was considering the larger set of validators that are not selected in the current 4096 committee as the observers and hence assumes honest majority[22].

4.2 Improvements and Alternatives of BFT

Older famous POS blockchains, like Cardano, kept the POW convention of heaviest chain selection; Ethereum also deployed a variant of the *GHOST* (*Greedy Heaviest Observed Subtree*) along with provable finality [119]. However, we believe the majority of newer POS deployments will go for a variant of BFT; after choosing the set of validators to construct a committee, usually a round leader is elected to start the broadcast in some variant of the traditional BFT protocol. Hence, almost every new application based blockchain introduces an innovation or at least a modification on one of the BFT protocols family as its new consensus protocol. Examples include [105, 106] for Hyperledger Fabric, *SPC* being a variant of FBA, and *RBFT* in [120] is adding *PBFT* to *RCPA* for applicability on supply chains; this kind of papers, whether peer reviewed or white, is expected to keep appearing.

Then there are different kinds of performance enhancement research:

-Researchers could go deep down into the communication details; for example ***BBCA (Byzantine Broadcast with Commit Adopt)*** from Chainlink labs [121], is an active probing API over the traditional *BRB* (Byzantine Reliable Broadcast) that stops a node from further participation in the broadcast if it has already committed to a value (i.e. removed the voting phase using *lock before finalize*; recall [5]).

-Then comes a joint research area between theory and performance enhancement, where researchers try to approach the proved theoretical lower bounds or reduce the gap between performance in good (no faults) and bad (faults) case. ***Hybrid-BFT*** [122] (2020) achieved latency of 3δ (message delay) in the optimistic case, where 2δ is the lower bound, and (1.5-3)*δ + O(Δ network latency) in the presence of faults.

-Recent examples include ***Autobahn*** [123] (June 2024) which addresses impracticality issues in the theoretical partial synchronous model and the Global

[21] They assume a fixed set of validators (say in an epoch), and use a few known consensus protocols, like Hotstuff [25] in some case, as the backbone internal consensus protocol.

[22] Recall from section 3.7 how Ethereum uses its official website as a public Bulletin Board, and how it assumes an *out of bound communication time* of ~ 2 months [9]; i.e, we could say it assumes clients communicate off-chain within 2 months.

Stabilization time *GST*. The authors find real partial synchrony to be piece-wise; periods where timeouts are met separated by '*blips*', during which progress is stalled. Blips are frequent in reality due to DDoS attacks and/or route hijacking on leader elections; moreover, blips cause performance degradation (*hangovers*) that can last beyond the return of a 'good' interval. By deploying an asynchronous highly parallel data dissemination layer, *Autobahn* can instantaneously commit the entire data backlog (independent from its size), thus minimizing the effect of *hangovers*. As the authors state, *Autobahn* architecture is inspired by *DAG* consensus protocols.

DAG Consensus Protocols

This innovation [124] does not elect a round leader; instead, a *Directed Acyclic Graph (DAG)* is constructed first to reflect the data dependency between Transactions (to guarantee *safety*) then all validators start to build and broadcast blocks in parallel to speed up the process. Round after round a *safe* total ordering is achieved (on several blocks instead of just one, and while still keeping the 33% byzantine threshold) either by excluding *equivocating* (contradicting) blocks or preventing their construction from the beginning [125,126]. DAG-based consensus protocols include *DAG-Rider* and *BullShark*, and are currently deployed by many blockchain companies like Aptos [127] and Chainlink [128]; Aptos introduced the *Proof of Availability* idea in *Narwhal* where only meta data about the transactions are broadcasted till a consensus is reached to save bandwidth, Finally, as it might cross one's mind, on whether *DAG*-based total ordering schemes affect *MEV* (*Maximal Extracted Value*), the Aptos lab discussed it [129] in light of their original paper [127], while Chainlink also introduced their DAG-based protocol called *Fino* [130] that integrates MEV-resistance features into DAG-based BFT, before [128].

4.3 Unauthenticated Consensus Protocols

From the beginning of this paper, in section 2 and Fig. 2, we assumed the existence of a public key infrastructure (PKI) where each node is authenticated through its public-private key pair. Although not our main scope, there are other kinds of consensus protocols that allow for unauthenticated nodes. In each BFT view, and in each view change, each node follows the leader in deciding the safety of a message and echoes it to every other node; so called *primary backup view-based paradigm*. The need for such protocols was driven by nodes with limited resources (less storage and computation requirements than cryptographic functions) and also by different nodes having different trust assumptions for each other (heterogeneous trust systems).

Example research contributions include *Information Theoretic Hotstuff (IT-HS)* [131] (2020) which is the unauthenticated version of Hotstuff [25], while *Tetra BFT* [132] is a recent research article (Jun 2024) that is partially sponsored by the Stellar Foundation. *Tetra* means 4, since the protocol is 1) optimistically responsive (latency 5δ) in the good case), works for nodes with 2)constant local storage & 3)linear communication ($O(n^2)$ bits), 4) and can tolerate byzantine ratio $f < n/3$. The authors

extend to propose a possible pipelining (State Machine Replication) which has not been explored enough in the unauthenticated case.

Related to mention here is a similar line of research on a model called *Consensus with Unknown Participants* (*CUP*) that studies the knowledge required to solve consensus in settings in which each participant joins the network knowing only a subset of other participants and the fault threshold of the system. However, we did not encounter any research on blockchain consensus under the *CUP* model, except for [133] (Jun 2023) that studied whether Stellar can solve consensus under the *CUP* model and proposed an oracle to do so. One may suspect this work could be further investigated for the vehicles system discussed in [115] since there are many cases where strange unauthenticated cars may happen to hit the road. As for more generality, the authors of [133] left the question of whether the *BFT-CUP* [134] approach can be used for implementing a permissionless blockchain as a future work.

4.4 Bitcoin Related Research

Since the appearance of smart contracts and the POS idea, there were many research attempts to achieve some mixed version that limits the classical POW consensus to a pre-phase (only key blocks) and build upon it; we have already mentioned Byzcoin [13] and Bitcoin-NG [52]. We will defer research & development solutions to include Bitcoin in the smart contract based DeFi world to section 5. A related research thread is protocols that reach consensus through chains of platform-dependent messages embedded in the Bitcoin blockchain[23].The authors of [24] discuss the security of the idea of using an extra blockchain in an abstract manner, while [135] is an 2017 example on Bitcoin.

Here, we will shed some light on *Bitcoin Staking*; a research thread that suggests using locked Bitcoins as a staking asset to secure a POS blockchain. Then, we also recognize a research paper targeting financial rails with a protocol they call *Stubborn Nakamoto consensus*.

Bitcoin Staking

The idea is based on an old 2015 paper [136] which introduced a novel cryptographic primitive "*accountable assertion"* that reveals the party's Bitcoin credentials (private key)[24] if it equivocates (signed contradicting messages); i.e., it made *slashing* possible

[23] Bitcoin transaction format allows for nonstandard transactions that holds metadata for many use cases one of them is "*sidechains*", where the metadata attaches a subchain to the Bitcoin main chain; sidechains applications include lightening networks where people with repeated small payment transactions use them without burdening the Bitcoin blockchain, another important use is blockchain interoperability as explained in [28] and will be discussed further in section 5 (specifically [179] is the classical reference).

[24] The primitive is based on *chameleon hash functions* which although collision resistance allows a trapdoor with an auxiliary secret to efficiently compute collisions; however, the Babylon team in the references that follow used *double-authentication-preventing signatures (DAPS*) instead which is a stronger primitive that does not require an auxiliary secret, and the authors of [136] discussed it on their appendix. The interested reader may

in Bitcoin using the bitcoin script language. The first paper in 2021 to one of the authors [137] started suggesting an energy efficient slashing in the same sidechain framework as in [135], then introduced the first version of their Bitcoin-backed POS consensus mechanism; the project was made to existence by Babylon labs and Cosmos partnership [138] where the validator locks Bitcoin assets inside the bitcoin blockchain and if cheated 1/3 of his/her asset is slashed by the lock contract (Bitcoin timestamps were used to adjust the finality gadget of their protocol). In their preprint released Dec 2024 [139], the team generalized the idea to provide *remote staking* of assets in a *provider* chain to secure (ensure *optimal economic safety* of) a *consumer* chain with Bitcoin as a case example of the provider chain.

Stubborn Nakamoto Consensus

The authors in [140] start by modeling and analyzing the attacks and vulnerabilities against the regular Bitcoin POW Nakamoto consensus, then introduce their protocol as incorporating the community response in restoring the correct ledger after detecting an attack (via a recovery oracle); they acknowledge the fact that reliance on external inputs should be minimized, but prove that some external dependence is necessary to recover from attacks. Their protocol assumes the same Fully Permissionless settings as Bitcoin, but introduces a confirmation depth parameter *k*; i.e., a node certifies a block not just for being in the longest viewable chain but must also be at least *k-deep* in that chain. Then, the protocol follows what we could say a similar approach to finality gadget by allowing nodes to append the block to its local view only after two time units (2Δ) passing without seeing a conflicting block; otherwise the node halts. This arrangement favors consistency over liveness by making nodes halt during attacks; however, the protocol can regain liveness after the recovery oracle.

Since the protocol relies on community response and nodes external communication[25], it was analyzed by the authors in [117] and they presented a concrete attack on it (their appendix E) where the adversary can create 2 disjoint worlds with 2 disjoint chains (since the adversary in [140] could have $1-\varepsilon$ ratio of the hash power). Assuming the adversary can produce a block to each of the 2 chains every $\Delta/2$ time, an honest node will always ignore blocks gossiped to it from the other world after more than Δ time as not being in the longest chain; hence the two chains will keep growing and the protocol is not safe under this attack even with always on and fully communicating clients. This is expected, as the authors of [140] also stated, they cannot overcome the known impossibility results [9] (recall section 2.1 and Table 1) and be safe in a fully permissionless setting (*FP*) without honest majority.

---

refer to the paper and references (30,43) on it for chameleon hashes, and their (38) for DAPs along with the detailed comparison with DAPs in their full version (http://crypsys.mmci.uni-saarland.de/projects/PenalizingEquivocation/penalizing.pdf); however, beware that this paper was published in 2015 before Bitcoin deployed *Schnor signatures*.

[25] They stated in their footnote 47 that they also assume, like [117], that observer nodes whom do not participate in consensus (do not mine in Nakamoto consensus) will echo messages.

4.5 Research Through Protocols Development Cycle

There is an endless continuous flow of justified research in the blockchain development cycle, and we expect it to continue; recent (2024) examples include [141] suggesting a modified confirmation rule and [142] introducing *Goldfish* protocol as a suggested replacement of *LMD-GHOST* for Ethereum. The two papers happen to be proposed for Ethereum since it is a case example of how a blockchain consensus design is continuously evolving; Fig.4 is imitated from a presentation by the Ethereum Foundation in DEVCON24 [143] showing evolving forks since the Beacon Chain in 2020. The last part of this section is dedicated to the Beam Chean [143-145], a drastic new changes Ethereum L2 is heading to.

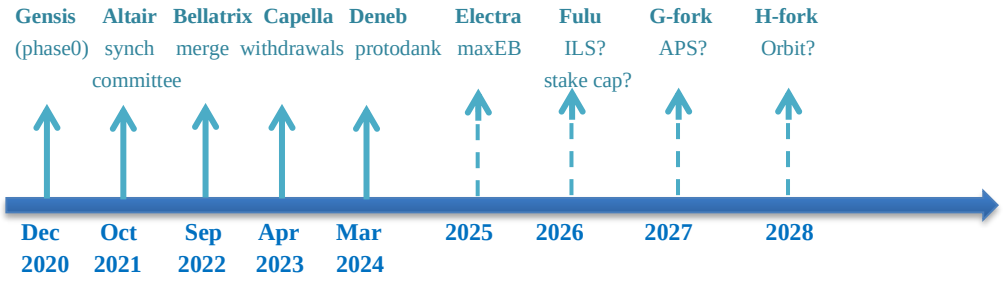

Fig.4: Ethereum evolving forks through the years (~ 1 fork/year) since the Beacon chain in 2020 (the pre-step for the transition from POW to POS in Sep 2022); adopted from [143]. Details of each fork can be found at (https://ethereum.github.io/consensus-specs/)

-Fair Ordering

Fair ordering of transactions in a block is important especially for DeFi applications to prevent attacks like frontrunning, backrunning, sandwiching of profitable transactions by miners or validators; i.e., even if the transaction paid a high transaction fee, the block builder is still in control. The profit of a block builder gains from transaction ordering is called MEV (Miner Extracted Value previously, Maximal Extracted Value now). Ethereum implemented *Proposer-Builder Separation* as protect from MEV [146]; hence, fair ordering of transactions is not part of Ethereum Consensus anymore. Validators who can be elected as block proposers are not responsible of gathering transactions into a block. This worked as a performance enhancement as well [147]; validators now only participate in ~1% of block building and spend their time validating while others are building and ordering blocks concurrently.

However, there is still ongoing research on fair ordering consensus for many other applicable cases. In 2020, [148] introduced the *Fair Ordering* property to consensus in the permissioned setting, [149] (2022) extended the property to the permissionless case, and we have already mentioned [129,130] that integrate it in DAG consensus.

Finally, [150] is a recent Systemization of knowledge (Nov 2024) about Fair Ordering consensus.

-Economic incentives

Mechanisms of deploying economic incentive mechanisms (rewarding & slashing) will always remain an important part of blockchain consensus research. The economic model behind different *rewarding strategies* has been a well-studied research topic from the very beginning of Nakamoto consensus [51] to Ethereum EIP-1559 burning ratio of the transaction fees, then POS consensus opened the door for more strategic manipulations especially in the block choice step (leader selection and/or fork choice). This research thread, although overlaps with cryptography that we will discuss next, is very rich in mathematical proofs that depend on probability distributions, Markov Decision Processes (MDP), random walks, and is usually backed up by simulations; examples include [72] which was preceded by [151], and [71] in 2021 followed by [152] in 2022 then pursued further in 2024 [153]. Research on the BFT committee size like [154] is of the same type, where sampling from a population with variable stakes is better described by a binomial distribution with bias τ [70/12.]; also research discussing the optimal stake size for a generic user like [155] about Algorand and [156] about Tezos.

Slashing on the other hand, although a strong defense mechanism, has its pros & cons [70/12.20] and is not deployed on all POS blockchains; Ethereum is an example that deploys penalties not just for signing contradicting blocks but also *inactivity leaks* [9,157] to guarantee the QP (Quasi-Permissionless setting)[26], while other blockchains like Algorand suffice with rewards as an incentive. One can point out to [3,7,8,9] as presenters of this research area; on the other hand, slashing with rewarding, is also common in the form of a *reputation* in many crypto projects including games and metaverses which is reflected in their white papers [77,88] and in some research papers [30].

Naturally, with all those carefully thought of incentive models out there, there will always be ongoing research finding exploits and/or defending them. An example is *discouragement attacks* [158] by Vitalik Buterin himself in 2018 with a defend strategy, then researchers in [159] found a built-up on it they called *staircase attack* is still possible in 2023 with only **29.6%** malicious nodes and Ethereum fixed it on 2024; like [71,72], the authors of [159] notified the significance of the economic incentive model in blockchain security and discussed other attacks in previous literature.

Finally, although POW economic incentives research seems to have reached a settlement since 2016-2017, for example *bribery attacks* [160], the detailed analysis of Nakamoto consensus vulnerabilities and the bribery model in [140/section 3] points out to a lot of ongoing research. From which we feature an interesting 2024 paper [161] that demonstrates how majority attacks could become *zero net cost* if block

[26] An incident example mentioned in [70/12.20] is stakers who deployed a replicated server to avoid inactivity leak, for some technical reason each of the 2 servers signed a different block (with the same signature) causing a certificate of guilt for them.

rewards were included, and adversaries were able to strategically affect the difficulty adjustment. However, [140/appendix D] discusses how factors, other than economic incentives, deter attacks introduced in those papers and protect major public blockchains like community response, the practical difficulties in getting miners to participate (even if it is rationally profitable), regulations and the practical difficulties of cashing the stolen money in fiat currency.

-The underlying cryptography

It's natural to see consensus research appearing in security related publications [40]. For a start, communication involves cyber security threats like DDoS and/or route hijacking [162, sections 6-8 of 163]; those threats are usually handled through some form of a heartbeat system like sending dummy blocks in Vena and Simplex Consensus [26]. Naturally, heartbeats as well as all validators' messages are cryptographically signed.

Towards scalable signatures Bitcoin moved to *Schnor* signatures [164]; while Ethereum Zcash, and Chia used *BLS* aggregating signatures [165] and the suggested *Beam Chain* plans to use post quantum Zero Knowledge *SNARK*s[27] (*Succinct Non-Interactive Argument of Knowledge*) and not just in place of *BLS*, as they call it "*Snarkification"* of the consensus layer [143,144]. The approaching threat of quantum computing has also introduced innovative solutions for POW consensus; [166] is an interesting paper that proposes a consensus protocol that is based on quantum sampling techniques. Although the protocol requires staking an equal amount of tokens for all participants, rewards are based on a required computation task (implementing quantum sampling); the Nash equilibrium is found based on rewarding miners committing to honest samples and penalties for miners committing dishonest samples. The authors point out to an alarming fact that the required quantum hardware has already been experimentally demonstrated at a sufficient scale and is becoming commercially available.

Research also includes the use of *Trusted Execution Environments (TEEs)* for faster cryptography, especially with DAG consensus like *TEE-Graph* for IoT blockchains [167] (2022), and *Fides* [168] (Jan 2025) which was experimentally evaluated for both local and geo-distributed networks. Recall that some projects still suffice with CFT (Crash Fault Tolerance) consensus in the presence of TEEs [108].

Finally, another ongoing research on providing a strong, and efficient, source of *randomization* to make leader (or committee) election process unpredictable and un-

[27] Although basic primitives in blockchains like *Merkle Trees* and *Verkle Trees* are theoretically viewed as *SNARK*s (they provide a short non-interactive proof of knowledge), the *Beacon chain* was designed in 2020 before many important advancements in the Zero Knowledge field have appeared; things like ***ZK-rollups*** are a layer-2 scaling solution but are not part of the Beacon chain. Note also, that the term ***zkEVM*** refers to the class of virtual machines that can execute smart contracts that involves Zero Knowledge proof computations (like ZK-rollups) and still be compatible with the existing Ethereum infrastructure and its original EVM (https://chain.link/education-hub/zkevm); put it another way make it possible for developers to use ZK functions and applications as libraries that extends the EVM without getting involved in the underlying cryptography.

manipulatable [169] falls in the area of cryptography; *VRF*s [68,70/12.9-10,170,171], *VDF*s [70/12.13,172], while *SSLE* [173,174] is still experimental.

-Applying AI

The huge advancement of Artificial Intelligent (AI) techniques made plenty of room to apply it in every step of way. There are AI security defense strategies like [175] applying deep learning to protect from POS long range attacks; the authors also demonstrated that AI models in general can be used as a mitigating checkpoint for long-range attacks. There are protocol modeling and security assessments like this agent-based modelling of Stellar in [176]. Finally, [177] provides a very recent review (Dec 2024) of the technological convergence between blockchain and AI.

-The Beam Chain

Although the main promoted theme of the project is to deploy post quantum *SNARK*s in every validation/proof step and provide stronger VDF[28] randomness through which will enhance both the security and the performance in many aspects. Including also Vitalik published future plans [145], we feature the following consensus related points; [178] is the dedicated site to follow the continuous updates and open research problems:

- Attestor-Proposer separation where anyone can be a proposer according to auctions; this could be viewed as a follow-up step after proposer-builder separation and is expected to further fasten the validators (attestors) work after removing the small burden they now have of proposing a block. Also, most of the *SNARK*s are planned to be calculated off chain.
- Only Honest minority will keep the whole chain history; i.e., allowing validator nodes to get rid of part of the old history.
- Reducing slot time to 4 seconds; due to the previous enhancements, it is expected to be able to reduce the block latency which is currently 3 slots. The old persisting idea of reaching a single slot finality (hence epochs will not be needed, and blocks will become finalized instantly as in Algorand) is also in the plan.
- Allow Staking to be possible from 1 or 2 ETH; although Vitalik blog in 2018 [23] considered raising the stake above 32 ETH, the plan now is to lower it for the same reversed reasons (making becoming a validator possible with less ETH); could be also to decrease the centralization in staking pools as mentioned in footnote 35.
- Pivoting the Verkle Tree in a binary tree; although not mentioned in the original presentation [143], it was mentioned in [144] that since Verkle Trees are not post-quantum and hence there are future plans to pivot the verkle tree

[28] According to [144], there are available built-in *ASIC*s (*Application Specific Integrated Circuit*) for Ethereum VDFs right now; naturally it is expected to be based on MinRoot as stated in [143] and also mentioned with more cryptographic details in the suggestions of (https://orochi.network/blog/Origami-Simplifying-Ethereum-VDF-with-Customized-Plonk).

in a binary tree design with post-quantum hash functions; this was discussed in Vitalik blog "*the verge*" too [145, part 4].

5 Consensus Across Chains in Interoperability Solutions

Reaching consensus in a multi-chain environment is much more coplicated; for a start, safety involves more than one ledger like the simple combined (train-hotel) and (plane-hotel) reservation example in [6]; then selecting validators set or relay nodes is more complicated and involves more threats as depicted by the long history of cross chain attacks in their short lifetime [28].

5.1 Interoperability solutions classification

Most of the literature follows the classification presented in [179] that first divides them into ***heterogeneous*** solutions between different kinds of blockchains with Hyperledger as the most dominating example and we have just mentioned how it offers flexibility by supporting pluggable consensus protocols. Then, there are ***Blockchain of Blockchains (BOB)*** solutions that provide a Cross Chain Communication extra layer blockchain to handle transactions between EVM like blockchains; famous contributors to the consensus literature like Chainlink [25,128,130], COSMOS [16], Zetachain [180], Polkadot [181], and Horizen [182] fall in this category, where innovations may include efficiency improvements and/or guarantees, and are most concentrated in choosing the validators committee in ways that provide enough rewarding incentives [183] and guarantee decentralization [68] and attack defense [77]. The last category, ***public connectors*** that include all notary systems like side chains and bridges have the same challenges of key leakage and possible collusion more magnified since they can only provide smaller number of validators [184]. The authors in [185] examine in detail the security of many interoperability solutions including some that deploy POW instead of BFT consensus and some that use ***distributed private key control***; other articles that discuss challenges in interoperability solutions include [28], while [30] is about Metaverse specialized solutions, and we have mentioned Axe-infinity game in Ronin [88].

5.2 Case Studies

We will go through some case examples in this subsection; the reader interested in more may check the consensus mechanism of interoperability solutions listed in [28,180,182].

- ***Polkadot***
  For a start Polkadot [181] uses *Nominated POS*, *NPOS*, where stakers can vote to nominate validators who are willing to dedicate the time and resources to run a validator node. Then a hybrid consensus (2 protocols) is

applied to split block production protocol, BABE, from handling forks and guaranteeing finality protocol, GRANDPA. *Blind Assignment for Blockchain Extension (BABE)* is comparable to a recent *ouroboros* variant than Latus called *Praos* [186]; epochs are divided to slots (~6 secs) and validators are assigned to slots via a Lottery algorithm based on VRF[29], if more than one validator are selected to one slot they race to finish the block and if no validator is selected a backup round-robin validator selection is used[30]. *GRANDPA (GHOST-based Recursive ANcestor Deriving Prefix Agreement)*, [22], where *GHOST* stands for *Greedy Heaviest Observed Subtree*, is run by validator nodes in parallel to reach provable finality through consecutive rounds of validators voting (on chains instead of single blocks); so *BABE* builds on the chain finalized by *GRANDPA* and if there are forks afterwards, it favors the chain with more primary blocks (generated thru VRF selection not round-robin). Finally, Polkadot supports a bridge, *BEEFY* (*Bridge Efficiency Enabling Finality Yielder*), to remote, segregated blockchains like Ethereum; *BEEFY* operates on top of *GRANDPA*, utilizing a consensus extension and a light client protocol[31].

- ***Bool Network***
  The original paper in 2022 [187] proposed a relay chain scheme that is based on secure multi-party computation and distributed private key management over an evolving hidden committee; the committee is elected using a *Ring verifiable random function (Ring VRF) protocol*, where the real public key of a VRF instance can be hidden among a ring. Furthermore, all the *key management procedures are executed in the TEE*, such as Intel SGX, to ensure the privacy and integrity of partial key components; although identities are hidden, committee members who behave maliciously can be detected and disqualified, and the cost of launching DoS or double-spending attacks is high. The (https://Bool.network/) site promotes itself as *a Bitcoin verification layer that turns all blockchains into Bitcoin layer 2*.

- ***Subsidy Bridge***
  A general and decentralized relay scheme with special incentive design similar to Bitcoin mining. The main idea, [188], is to keep utility of honest relayers (basic subsidy from target chain + transaction fee from cross-chain users) always positive even when users are temporarily inactive; it's worth

[29] Note that each validator has an equal chance in this slots' lottery; i.e. randomization is not based on stakes as in Cosmos [181/Polkadot comparisons] and many other POS variants.

[30] Recall that VRFs can lead to 0 or more than 1 winner (footnote in section 2/7-POS), on the other hand round robin (the backup approach here) is described as the first thought approach in [70/12.].

[31] BEEFY uses *Merkle Mountain Ranges (MMR)* as an efficient data structure for storing and transmitting block headers and signatures, which is almost the same Utreexo forest data structure introduced in Bitcoin as a stateless server providing worst case O(*log n*) Merkle proofs to stateless clients.

mentioning that the designers of Latus [183] too, although a POS variant, emphasize that it is very important for the consensus security to continue issuing blocks, even if empty to keep the rewarding incentive. The authors calculated the Nash equilibrium conditions of the *subsidy bridge* game and proved security under honest majority of relay developers with at least one honest relayer from the source chain; they also claim their design is flexible to support any source chain of any secure consensus, so can support Bitcoin efficiently in contrast with Polkadot and Cosmos and still support other blockchains if compared to BTC-Relay.

- ***Deterministic Cross-Blockchain Token Transfer (DEXTT)***
  All observer candidates use their private keys to sign the cross-chain transactions as soon as they discover it, *the observer the minimum signature value* [189] is considered the contest winner and thus broadcast the transaction and wins the witness reward; *VETO transaction*s reporting double spending and penalizing bad behavior are subject to the same contest with a reward incentive. Finally, in their evaluation implementation the authors used three geth nodes in *Proof of Authority (PoA)* mode, creating three private blockchains, to ensure a reproducible and uniform ecosystem of blockchains.

- ***Practical AgentChain***
  Agentchain appeared in 2019 [190] and could be viewed as a *Proof of Reputation* consensus side chain that uses multi-signature schemes; trading operators can be combined as several decentralized trading groups by locking tokens to ensure credibility. Users choose a "*reputable*" trading group and deposit assets to the trading group's multi-signature address on the existing blockchain. Then the assets will be mapped to *AgentChain* by the trading group, on which token fair exchange is supported. However, the design was in the conceptual stage with poor implementation; a follow-up paper introduced *Practical AgentChain* [191] in 2022 that introduced a complete system with more functionalities.

- ***Identity-Based Encryption (IBE-BCIOT)***
  Proposed electing proxy nodes according to a clustering algorithm based on density peaks [192]; aiming to elect nodes with efficient computing power, the algorithm assumes that if the cluster centers are surrounded by neighbor nodes with lower local density and the distance between any nodes is relatively large, then the clustering center will be defined as the local maximum of the data point density. Then, the elected proxy nodes are authenticated through a trusted cross-chain notary deploying an *Identity Based Encryption* (*IBE)* algorithm; this shares some resemblance with the AI generated *Proof of Identity* in [116]. Finally, selected nodes reach consensus using *PBFT*.

6 Summary & Conclusions

New research innovations, also literature surveys and comparative studies, on blockchain consensus will continue to appear as we write [193,194]. We hope the effort in this paper provides a condensed consolidation of the topic; we presented an enumeration of the main blockchains consensus protocols and mainly tried to cover all design aspects pointing out to all significant research directions. The paper also shed some extra light on consensus schemes used in interoperability solutions to inform the reader with their added challenges and demonstrate with real application projects; we then extend with appendices containing some formal definitions, further depth on the significant leader/committee selection problem, and summarizing tables. The paper can serve as a starting point that provides the necessary reading material for researchers and designers of blockchain consensus protocols, with enough guidance on how to go deeper. Possible future work could be to present a special study that summarizes and organizes reported and/or discovered attacks on blockchain consensus protocols.

Acknowledgment

Although it has been almost 25 years post my PhD, I am still grateful to all those contributed into my undergraduate & postgraduate education especially my old supervisors; those are the ones who built my research skills and taught me to be persistent and always search for the new from most sophisticated top ranked resources. Then, I cannot possibly be less grateful to all those in the academia who have made their sincere efforts freely available online and sometimes replied to enquiries; although this work has not received any funding, it has benefited from all the freely available (probably funded) material cited in the paper references.

Declaration, Data Availability & Conflict of Interest statement

-This paper is a single author paper, so there cannot be any conflict of interest between authors.
- The single author, me, is temporarily out of work and has not received any funding for this work; only the freely available online material cited in the reference. Hence, this is completely independent work of a single person.
-The study did not contain any experiments, so data availability is not applicable

References

1. https://decentralizedthoughts.github.io/2019-10-15-consensus-for-state-machine-replication/; last accessed 13/4/2024.

2. Sisi Duan et al., "*BEAT: Asynchronous BFT Made Practical*", ACM 2018, doi:10.1145/3243734.3243812, https://youtu.be/u0nypF5AIF4
3. Vincent Gramoli, "*Strengthening Blockchain Security with Accountability*", SBC'24, https://youtu.be/9NueWaGeZ8g; a re-presentation of an awarded best paper in DSN'24: V. Gramoli and A. Ranchal-Pedrosa , "*ZLB: a Blockchain to Tolerate Colluding Majorities*"; https://arxiv.org/pdf/2305.02498v1
4. Alysson Bessani, Eduardo Alchieri, Jo ao Sousa, Andre Oliveira, and Fer nando Pedone. *"From byzantine replication to blockchain: Consensus is only the beginning"*. In IEEE/IFIP DSN, pages 424436, 2020. (haven't read yet, but the title is what is needed and can't find the one I earlier meant now, could add mine about the effect of geo)
5. https://decentralizedthoughts.github.io/2020-11-29-the-lock-commit-paradigm/; last accessed 13/4/2024.
6. https://anoma.net/blog/heterogeneous-paxos-and-multi-chain-atomic-commits; last accessed 12/10/2023.
7. Tim Roughgarden, "*Foundations of Blockchains*", lecture notes, Columbia University, 2021, https://github.com/timroughgarden/fob21/blob/main/l/l2-7-overview.pdf, https://timroughgarden.github.io/fob21/; last accessed 11/4/2024.
8. Anrew Lewis-Pye and Tim Roughgarden, "*Permissionless Consensus*", https://arxiv.org/pdf/2304.14701; tutorial videos
9. Eric Budish, Anrew Lewis-Pye and Tim Roughgarden, "*The Economic Limits of Permissionless Consensus*", FC'24 Keynote, https://arxiv.org/pdf/ 2405.09173v1
10. Vitalik Buterin, "*Properties of Consensus in Theory and in Practice*", SBC'24; https://youtu.be/1hb1X45JRJs
11. J. Wu et al., "*SEGBFT: A Scalable Consensus Protocol for Consortium Blockchain*", ICBCT'22: The 2022 4th International Conference on Blockchain Technology, March 2022, Pages 15–2, https://doi.org/10.1145/3532640.3532643, https://dl.acm.org/doi/abs/10.1145/3532640.3532643
12. https://salemal.medium.com/paper-review-of-byzcoin-9ce7676d1c32; last accessed 15/11/2024.
13. Byzoin, https://arxiv.org/abs/1602.06997, to add presentation video
14. The Helium blockchain and its whitepaper, https://status.helium.com/; last accessed 11/4/2024.
15. https://www.telefonica.com/en/communication-room/blog/wi-fi-and-mobile-to-improve-coverage-enable-mobile-data-traffic-offloading/; last accessed 11/4/2024.
16. https://cosmos-network.gitbooks.io/cosmos-academy/content/introduction-to-the-cosmos-ecosystem/tendermint-bft-consensus-algorithm.html; last accessed 12/4/2024.
17. Rafael Pass and Elaine Shi. 2016. "*Hybrid consensus: Efficient consensus in the permissionless model"*, Cryptology ePrint Archive (2016). (cross reference from 8)
18. Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren ,and Alexander Spiegelman. 2016. "*Solida: A blockchain protocol based on reconfigurable byzantine consensus*", https://arxiv.org/abs/1612.02916, (cross reference from 8)
19. Miguel Castro and Barbara Liskov, "*Practical Byzantine Fault Tolerance*", In Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, USA, February 1999, https://pmg.csail.mit.edu/papers/osdi99.pdf
20. Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song, "*The Honey Badger of BFT Protocols*", 2016, https://youtu.be/Qone4j1hCt8
21. V. Buterin and V. Griffith, "*Casper the Friendly Finality Gadget*", 2017, https://arxiv.org/abs/1710.09437

22. A. Stewart and E. Kokoris-Kogia, *"A Byzantine Finality Gadget, **GRANDPA, G**host-based **R**ecursive **AN**cestor **D**eriving a**P**refix **A**greement*, 19 June 2020, https://github.com/w3f/consensus/blob/master/pdf/grandpa.pdf
23. Vitalik Buterin, https://notes.ethereum.org/@vbuterin/single_slot_finality#How-and-why-Ethereum-staking-works-today; last accessed 25/11/2024.
24. Yashvanth Kondi et al., *"Practical MPC for Re-staking: Separating Broadcast from Cheater Identification"*, DeCompute'24, https://youtu.be/MOyTlB1tjsU
25. M. Yin, D. Malkhi, M.K. Reiter, G.G. Gueta, and I. Abraham, "*Hotstuff: BFT Consensus with Linearity and Responsiveness*", PODC'19, p. 347-357,Toronto, Canada, ACM; https://doi.org/10.1145/3293611.3331591
26. Vena, https://hackmd.io/@patrickogrady/vena; last accessed 25/11/2024.
27. Y. Merrad, M. H. Habaebi, E. A. A. Elsheikh, F. E. M. Suliman, M. R. Islam, T. S. Gunawan, and M. Mesri, ''*Blockchain: Consensus algorithm key performance indicators, trade-offs, current trends, common drawbacks, and novel solution proposals*'', Mathematics, vol. 10, no. 15, p. 2754, Aug. 2022.
28. https://medium.com/@shymaa.arafat/blockchain-interoperability-part1-9d2e29da691b; last accessed 15/4/2024.
29. https://www.researchgate.net/publication/343730274_A_New_Consensus_Protocol_for_Blockchain_Interoperability_Architecture
30. Tianxiu Xie et al., "*RAC-Chain: An Asynchronous Consensus-based Cross-chain Approach to Scalable Blockchain for Metaverse*", Mar 2024, https://dl.acm.org/doi/10.1145/3586011
31. https://www.semanticscholar.org/paper/A-Truly-Decentralized-Blockchain-Consensus-Protocol-Wimal-Liyanage/200b1c126b3cccf88a92c6c4ed2d7139148dba40
32. https://www.researchgate.net/publication/335788453_Flow_Separating_Consensus_and_Compute
33. Sharding, https://medium.com/nearprotocol/so-what-exactly-is-vlads-sharding-poc-doing-37e538177ed9; last accessed 25/11/2024.
34. Sharding, https://www.mdpi.com/2079-9292/11/16/2597
35. Andrew Lewis-Pye and Tim Roughgarden, "*Byzantine Generals in the Permissionless Setting*", Jan 2023, https://arxiv.org/pdf/2101.07095.pdf
36. L. Nuzzi, K. Waters, and M. Andrade, "*Breaking BFT: Quantifying the Cost to Attack Bitcoin and Ethereum*", Mar 2024, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4727999
37. Silvia Bonomi et al., "*Sok Achieving State Machine Replication in Blockchains Based on Repeated Consensus*", Jan 2022, https://arxiv.org/pdf/2105.13732v2.pdf
38. Elaine Shi, "*Foundations of Distributed Consensus and Blockchains*", http://elaineshi.com/docs/blockchain-book.pdf (cross reference from [7])
39. S. Kaur, S. Chaturvedi, A. Sharma, and J. Kar, ''*A research survey on applications of consensus protocols in blockchain*'', Secur. Commun. Netw., vol. 2021, pp. 1–22, Jan. 2021.
40. Z. Hussein et al., "*Evolution of Blockchain Consensus Algorithms: a Review on the Latest Milestones of Blockchain Consensus Algorithms*", Cybersecurity 6, 30 (2023), https://doi.org/10.1186/s42400-023-00163-y
41. Yang Xiao, Ning Zhang, Wenjing Lou, and Y. Thomas Hou, "*A Survey of Distributed Consensus Protocols for Blockchain Networks*", IEEE Comm. Surveys & Tutorials, Vol. 22 Issue 2, 2020, https://ieeexplore.ieee.org/abstract/document/8972381/
42. "*Blockchain consensus mechanisms and their applications in IoT: A literature survey*", in Proc. Int. Conf. Algorithms Archit. Parallel Process. Cham, Switzerland: Springer, 2020, pp. 564–579, https://slogix.in/blockchain-technology/blockchain-consensus-mechanisms-and-their-applications-in-iot-a-literature-survey/

43. M. Salimitari, M. Chatterjee, and Y. P. Fallah, "*A survey on consensus methods in blockchain for resource-constrained IoT networks*", Internet of Things, vol. 11, 2020; https://www.sciencedirect.com/science/article/abs/pii/S2542660520300482
44. A. Menon, T. Saranya, S. Sureshbabu, and A. Mahesh, ''*A comparative analysis on three consensus algorithms: Proof of burn, proof of elapsed time, proof of authority*'', in Computer Networks and Inventive Communication Technologies. Singapore: Springer, Jan. 2022, pp. 369–383.
45. https://tokens-economy.gitbook.io/consensus/blockchain-consensus-encyclopedia-infographic; last accessed 12/4/2024.
46. "*Blockchains for Governmental Services: Design Principles, Applications, and Case Studies*", Dec 2017, https://www.ctga.ox.ac.uk/201712-CTGA-Martinovic-I-Kello-L-blockchainsforgovernmentalservices.pdf; last downloaded 9/4/2024.
47. Y. Liu and C. Shang, "*Application of Blockchain Technology in Agricultural Water Rights Trade Management*", Sustainability 2022, 14(12), 7017; https://doi.org/10.3390/su14127017; https://www.mdpi.com/2071-1050/14/12/7017
48. https://medium.com/data-science/fedrated-byzantine-agreement-24ec57bf36e0, TDS 2021; last accessed 28/3/2025.
49. Marta Lokhava, Giuliano Losa, David Mazières, Graydon Hoare, Nicolas Barry, Eli Gafni, Jonathan Jove, Rafa Malinowsky, and Jed McCaleb, 2019, "*Fast and Secure Global Payments with Stellar*" In Proceedings of the 27th ACM Symposium on Operating Systems Principles. 80–96.
50. Dwork, C., Naor, M. (1993), "*Pricing via Processing or Combatting Junk Mail*", In: Brickell, E.F. (eds) Advances in Cryptology — CRYPTO' 92. CRYPTO 1992. Lecture Notes in Computer Science, vol 740. Springer, Berlin, Heidelberg, https://link.springer.com/chapter/10.1007/3-540-48071-4_10
51. Satoshi Nakamoto, 2008, "*Bitcoin: a Peer-To-Peer Electronic Cash System*"; https://bitcoin.org/bitcoin.pdf
52. Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse, "*Bitcoin-ng: A scalable blockchain protocol*", In 13th USENIX symposium on networked systems design and implementation, NSDI'16, 45–59
53. Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak, "*Proofs of Space*", 2013, https://eprint.iacr.org/2013/796.pdf
54. https://en.wikipedia.org/wiki/Proof_of_space; last accessed 10/4/2024.
55. https://docs.filecoin.io/basics/what-is-filecoin/blockchain; last accessed 10/4/2024.
56. https://www.coinbureau.com/blockchain/proof-of-activity-explained-hybrid-consensus-algorithm/; last accessed in 9/4/2024.
57. https://cointelegraph.com/news/proof-of-stake-and-activity-posa-a-consensus-mechanism-for-the-new-era-in-web3; last accessed in 9/4/2024.
58. Hongyu Song, Nafei Zhu, Ruixin Xue, Jingsha He, Kun Zhang, and Jianyu Wang, "*Proof-of-Contribution consensus mechanism for blockchain and its application in intellectual property protection*", Information Processing & Management, Volume 58, Issue 3, May 2021, 102507, https://www.sciencedirect.com/science/article/abs/pii/S0306457321000170
59. Kostis Karantias, Aggelos Kiayias, and Dionysis Zindros, "*Proof of Burn*", September/2019, Financial Cryptography, https://doi.org/10.1007/978-3-030-51280-4_28, https://eprint.iacr.org/2019/1096.pdf
60. https://21shares.com/research/tokens-burns; last accessed 12/4/2024.
61. M. Rodinko et al, "*Decentralized Proof-of-Burn Auction for Secure Cryptocurrency Upgrade*", Vol. 5, Issue 1, March 2024, 100170, https://www.sciencedirect.com/science/article/pii/S2096720923000453

62. https://github.com/hyperledger-archives/sawtooth-poet; last accessed 9/4/2024.
63. https://mitar.tnode.com/post/proof-of-luck-consensus-protocol-and-luckychain/; last accessed 8/4/2024.
64. https://www.researchgate.net/publication/320246838_On_Security_Analysis_of_Proof-of-Elapsed-Time_PoET
65. Mic Bowman, Debajyoti Das, Avradip Mandal, and Hart Montgomery, "*On Elapsed Time Consensus Protocols*", 2021, https://eprint.iacr.org/2021/086
66. https://labs.hyperledger.org/labs/archived/sawtooth-poet2.html; last accessed 8/4/2024.
67. https://medium.com/@saimmoin64/exploring-the-pros-and-cons-of-proof-of-luck-pol-consensus-mechanism-in-blockchain-2cef9ada98c1; last accessed 8/4/2024.
68. https://chain.link/education-hub/verifiable-random-function-vrf; last accessed 9/4/2024.
69. *"Peercoin The Pioneer POS",* Whitepaper, 2012, https://www.peercoin.net/read/papers/peercoin-paper.pdf; last downloaded 7/4/2024.
70. Tim Roughgarden, "Lecture 12: *Proof of Stake*", 24 videos, Jun 2023, https://youtu.be/ZSqxGdsmLHs; last accessed 13/4/2024.
71. https://www.researchgate.net/publication/353326391_Proof-of-Stake_Mining_Games_with_Perfect_Randomness
72. V. Baagaria et al., "*Proof of Stake Longest Chain Protocols: Security vs Predictability*", 23 Feb 2020, https://arxiv.org/1910.02218v3.pdf
73. Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov, "*Ouroboros: A provably secure proof-of-stake blockchain protocol*", CRYPTO 2017, Part I, volume 10401of LNCS, pages 357–388. Springer, Heidelberg, 2017.
74. https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/attack-and-defense/; last accessed 11/12/2024.
75. Tim Roughgarden and Naveen Durvasula, "*How secure is your Restaking Network*", SBC'24, https://youtu.be/81jXwzNFFwY
76. Iván Abellán Álvarez, Vincent Gramlich, and Johannes Sedlmeir. 2024. *"Unsealing the secrets of blockchain consensus: A systematic comparison of the formal security of proof-of-work and proof-of-stake"*. In The 39th ACM/SIGAPP Symposium on Applied Computing (SAC'24), April 8–12, 2024, Avila, Spain. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/ 3605098.3635970
77. https://docs.axelar.dev/learn/security; last accessed 14/4/2024.
78. https://www.researchgate.net/publication/343730274_A_New_Consensus_Protocol_for_Blockchain_Interoperability_Architecture
79. https://www.gemini.com/cryptopedia/proof-of-stake-delegated-pos-dpos; last accessed 7/4/2024.
80. https://aptos.dev/concepts/delegated-staking/; last accessed 7/4/2024.
81. https://developers.tron.network/docs/concensus; last accessed 7/4/2024.
82. https://developers.eos.io/welcome/v2.2/protocol-guides/consensus_protocol; last accessed 7/4/2024.
83. V. Bachani, and A. Bhattacharjya, "*Preferential delegated proof of stake (PDPoS)-modified DPoS with two layers towards scalability and higher TPS*", 2022, Symmetry 15(1):4, https://www.mdpi.com/2073-8994/15/1/4
84. O. Younis etc al., "*A Proposal for a Tokenized Intelligent System: A Prediction for an AI-Based Scheduling, Secured Using Blockchain*", March 2024Systems 12(3):84, DOI:10.3390/systems12030084, https://www.researchgate.net/publication/378790195_A_Proposal_for_a_Tokenized_Intelligent_System_A_Prediction_for_an_AI-Based_Scheduling_Secured_Using_Blockchain
85. https://en.m.wikipedia.org/wiki/Proof_of_authority; last accessed 7/4/2024.

86. https://www.sciencedirect.com/topics/computer-science/proof-of-authority; last accessed 7/4/2024.
87. https://www.lcx.com/proof-of-authority-explained/; last accessed 7/4/2024.
88. https://docs.roninchain.com/basics/white-paper; last accessed 7/4/2024.
89. J. Chain et al., "*An AI-based Super Nodes Selection Algorithm in Blockchain Networks*", 2018, https://arxiv.org/abs/1808.00216
90. Qi Xiong, Nasrin Sohrabi, Hai Dong, Chenhao Xu, and Zahir Tari, "*AICons: An AI-Enabled Consensus Algorithm Driven by Energy Preservation and Fairness*", April 2023, https://arxiv.org/abs/2304.08128
91. K. Venkatesan and Syarifah Bahiyah Rahayu, "*Blockchain Security Enhancement: An Approach Towards Hybrid Consensus Algorithms and Lachine Learning Techniques*", Jan 2024, Scientific Reports 14, Article number: 1149 (2024)
92. M. Salimitari et al., "*AI-Enabled Blockchain: An Outlier-Aware Consensus Protocol for Blockchain-Based IoT Networks*", IEEE 2019, https://ieeexplore.ieee.org/document/9013824
93. Mahmoud, H. H. M., Wu, W., and Wang, Y., "*Proof of Learning: Two Novel Consensus Mechanisms for Data Validation using Blockchain Technology in Water Distribution System*" in 27th International Conference on Automation and Computing (ICAC) (IEEE), University of the West of England, Bristol, UK, 1-3 September 2022.
94. M. Abdelhamid et al., "*A Review on Blockchain Technology, current Challenges, and AI-Driven Solutions*", ACM Computing Surveys, Vol. 57, Article: 73, 22 Nov 2024, https://dl.acm.org/doi/full/10.1145/3700641
95. H. Magureanu and N.Usher, "*Consensus Learning: A Novel Decentralized Ensemble Learning Paradigm*", https://arXiv.org/pdf/2402.16157; https://flare.network/consensus-learning-harnessing-blockchain-for-better-ai/; last accessed 14/4/2024.
96. Team Rocket, Maofan Yin, Kevin Sekniqi, Robbert van Renesse, and Emin Gün Sirer, "*Scalable and Probabilistic Leaderless BFT Consensus through metastability*", https://arxiv.org/pdf/1906.08936.pdf
97. https://build.avax.network/docs/quick-start/avalanche-consensus; last accessed 23/3/2025
98. Aaron Buchwald, Stephen Buttolph, Andrew Lewis-Pye, Patrick O'Grady, and Kevin Sekniqi, "*Frosty: Bringing strong liveness guarantees to the Snow family of consensus protocols*", https://arxiv.org/abs/2404.14250
99. https://cryptobern.github.io/snow_part3/; last accessed 23/3/2025
100. Ignacio Amores-Sesar, Christian Cachin, and Philipp Schneider, "*An Analysis of Avalanche Consensus*", International Colloquium on Structural Information and Communication Complexity SIROCCO 2024, https://link.springer.com/chapter/10.1007/978-3-031-60603-8_2
101. Alex Shafarenko, "*Indexing structures for the PLS blockchain*", 2021, https://doi.org/10.48550/arXiv.2107.08970
102. https://www.ibm.com/products/blockchain-platform-hyperledger-fabric; last accessed 24/3/2025.
103. https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatis.html; last accessed 24/3/2025.
104. https://hyperledger-fabric.readthedocs.io/en/release-2.5/orderer/ordering_service.html; last accessed 24/3/2025.
105. https://lfdecentralizedtrust.org/blog/hyperledger-fabric-v3-delivering-smart-byzantine-fault-tolerant-consensus; last accessed 24/3/2025.
106. A. Al-Salih et al., "*Pluggable Consensus in Hyperledger Fabric*", Oct 2024 , Proceedings of the 6th Blockchain and Internet of Things Conference, BIOTC'24, https://dl.acm.or/doi/10.1145/3688225.3688237

107. Y. Wang, "*Byzantine Fault Tolerance For Distributed Ledgers Revisited*", Sep 2022, Distributed Ledger Technologies: Research and Practice, Vol. 1, Issue 1, Article: 2, pages 1-26, https://dl.acm.or/doi/10.1145/3538227
108. S. Angel et al., "*Nimble Rollback Protection for Confidential Cloud Services*", July 2023, OSDI'23: Proceedings of the 17th USENIX Symposium on Operating Systems Design and Implementation, 978-1-939133-34-2; https://usenix.org/confrerence/osdi23/presentation/angel
109. Chuanwang Ma, Yu Zhang, Binxing Fang, Hongli Zhang, Yidong Jin, Dasheng Zhou, "*Ripple+: An Improved Scheme of Ripple Consensus Protocol in Deployability, Liveness and Timing Assumption*", 24 Nov 2021, CMES: Computer Modeling in Engineering and Sciences, Vol. 130, Issue 1, Pages: 463-481; https://www.sciencedirect.com/org/science/article/pii/S1526149221000874
110. https://how.dev/answers/how-does-the-ripple-protocol-consensus-algorithm-rpca-work; last accessed 28/3/2025.
111. K. Christodoulou et al., "*Consensus Crash Testing: Exploring Ripple's Decentralization Degree in Adversarial Environments*", 2020, https://www.mdpi.com/1999-593/12/3/53
112. Ignacio Amores-Sesar, Christian Cachin, and Jovana Mićić, "*Security Analysis of Ripple Consensus*",2020, https://arXiv.org/abs/2011.14816
113. https://stellar.org/learn/stellar-consensus-protocol; last accessed 29/3/2025.
114. M. Kim, Y. Kwon and Y. Kim, "*Is Stellar As Secure As You Think?*", 2019 IEEE European Symposium on Security and Privacy Workshops (EuroSPW'19); https://ieeexplore.ieee.org/document/8802516
115. H.C. Jang and C.W. Chang, "*Using Stellar Consensus Protocol to Ensure the Security of Message Transmission in VANETs*", 2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom'23); https://ieeexplore.ieee.org/document/10271912
116. https://en.m.wikipedia.org/wiki/Proof_of_identity_(blockchain_consensus); last accessed 14/4/2024.
117. S. Sridhar, E. N. Tas, J. Neu, D. Zindros, and D. Tse, D, "*Consensus under adversary majority done right*", to be presented in Financial Cryptography April 2025 , http://arxiv.org/abs/2411.01689; simplified article https://a16zcrypto.com/posts/article/beyond-51-attacks/, last accessed 28/2/2025.
118. Vitalik Buterin, "*A guide to 99% fault tolerant consensus*", https://vitalik.eth.limo/general/2018/08/07/99_fault_tolerant.html; last accessed 28/2/2025.
119. Vitalik Buterin et al., "*Combining GHOST and Casper*", May 2020, https://arxiv.org/abs/2003.03052
120. C. Stathakopoulou, M. Wei, M. Yin, H. Zhang, and D. Malkhi, "*BBCA-LEDGER: High Throughput Consensus meets Low Latency*", Jun 2023, https://arxiv.org/abs/2306.14757
121. H. Chen and J. Ma, "*RBFT: An Improved Blockchain Consensus Algorithm Based on RPCA and PBFT*", Sep 2024, 4th International Conference on Computer Science and Blockchain (CCSB'24); https://ieeexplore.ieee.org/document/10735652
122. Atsuki Momose et al., "*Hybrid-BFT: Optimistically Responsive Synchronous Consensus with Optimal Latency or Resilience*", 2020, https://eprint.iacr.org/2020/406.pdf
123. Neil Giridharan, Florian Suri-Payer, Ittai Abraham, Lorenzo Alvisi, Natacha Crooks, "*Autobahn: Seamless high speed BFT*", Nov 2024, ACM SIGOPS 30th Symposium on Operating Systems Principles (SOSP'24), DOI: 10.1145/3694715.3695942; https://arxiv.org/abs/2401.10369 (v4 in Jan 2025)
124. K. Gai, Z. Hu, L. Zhu, R. Wang, and Z. Zhang, "*Blockchain meets DAG: a BlockDAG consensus mechanism*", In: Algorithms and architectures for parallel processing: 20th international conference, ICA3PP 2020, New York City, NY, USA, October 2–4, 2020,

Proceedings, Part III, vol 20. Springer, pp 110–125, https://dl.acm.org/doi/abs/10.1007/978-3-030-60248-2_8
125. Andrew Lewis-Pye, "*Directed Acyclic Graph (DAG)-based Consensus*", a16z group, Oct 2022, https://youtu.be/v7h2rXNtrV0
126. https://blog.chain.link/bft-on-a-dag/; last accessed 14/4/2024.
127. Alexander Spiegelman, Aptos Labs, "*DAG Meets BFT- The Next Generation of BFT Consensus*", Science and Engineering of Consensus Workshop An Affiliated Workshop of SBC'22, https://youtu.be/_lKfdHT6ZFU
128. Dahlia Malkhi, "*Break throughs in consensus research from chainlink labs*", SmartCon 2023, https://youtu.be/R8K48CgoCHs
129. https://hackmd.io/@0xtrojan/mev_meets_dag; last accessed 15/4/2024
130. Dahlia Malkhi and Pawel Szalachowski, "*Maximal Extractable Value (MEV) Protection on a DAG*", Dec 2022, https://arxiv.org/abs/2208.00940
131. Ittai Abraham and Gilad Stern, "*Information Theoretic HotStuff*", 2020, https://arxiv.org/abs/2009.12828
132. Qianyu Yu et al., "*TetraBFT: Reducing Latency of Unauthenticated, Responsive BFT Consensus*", June 2024, PODC '24: Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing, p. 257-267, DOI:10.1145/3662158.3662783; https://arxiv.org/html/2405.02615v1
133. Robin Vassantlal, Hasan Heydari, and Alysson Bessani, "*On the Minimal Knowledge Required for Solving Stellar Consensus*", June 2023, the 43rd IEEE International Conference on Distributed Computing Systems (ICDCS'23); https://arxiv.org/abs/2305.17989 (v2)
134. E. A. P. Alchieri, A. Bessani, F. Greve, and J. da Silva Fraga, "*Knowledge Connectivity Requirements for Solving Byzantine Consensus with Unknown Participants*," 2018, IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 2.
135. Massimo Bartoletti, Stefano Lande, and Alessandro Sebastian Podda, "*A Proof-of-Stake Protocol for Consensus on Bitcoin Subchains*"; https://eprint.iacr.org/2017/417.pdf
136. Tim Ruffing, Aniket Kate, and Dominique Schroder, "*Liar, Liar, Coins on Fire! Penalizing Equivocation by Loss of Bitcoins*", 2015, https://www.cs.purdue.edu/homes/akate/publications/Non-equivocationWithBitcoinPenalties.pdf
137. Robin Linus, "*Stakechain: A Bitcoin-backed Proof-of-Stake*", Dec 2021; https://coins.github.io/stakechains.pdf
138. David Tse, "*Bitcoin Staking*"; SBC'24, academic talk: https://youtu.be/6WVPH0oz7iM; more practical talk: https://youtu.be/PgUy7PavpuM
139. X. Dong et al., "*Remote Staking with Optimal Economic Safety*", Dec 2024; https://arxiv.org/pdf/2408.01896
140. J. Leshno, E. Shi, and R. Pass, "*On the viability of open-source financial rails: Economic security of permissionless consensus*", March 2025 version, http://arxiv.org/abs/2409.08951v2
141. Aditya Asgaonkar et al., "*A Confirmation Rule for the Ethereum Consensus Protocol*", May 2024, https://arxiv.org/abs/2405.00549v1
142. F. D'Amato, J. Neu, E. N. Tas, and D. Tse, "*Goldfish: No more attacks on Ethereum?!*", In: Financial Cryptography, 2024, https://eprint.iacr.org/2022/1171
143. Justine Drake, "*The Beam Chain*", DEVCon'24, https://www.youtube.com/watch?v=rGE_RDumZGg&t=7200s; last accessed 2/12/2024.
144. The Hyped Beam chain idea by Justine Drake, contains video, add spotify direct link too, https://x.com/therollupco/status/1858628995392147641; last accessed 29/11/2024.

145. Vitalik Buterin, "*Possible Futures of the Ethereum Protocol*", https://vitalik.eth.limo/general/2024/10/14/futures1.html, https://vitalik.eth.limo/general/2024/10/17/futures4.html; last accessed 8/12/2024.
146. Dan Boneh, "*MEV and Fair Ordering*", https://youtu.be/T1bD7_OTD1o; last accessed 14/4/2024
147. K. Mu, B. Yin, A. Asheralieva, and X. Wei, "*Separation is Good: A Faster Order-Fairness Byzantine Consensus*," in Network and Distributed System Security Symposium, 2024.
148. Mahimna Kelkar, Fan Zhang, Steven Goldfeder and Ari Juels, "*Order-Fairness for Byzantine Consensus*", CRYPTO 2020, pages: 451-480, https://eprint.iacr.org/2020/269, https://link.springer.com/chapter/10.1007/978-3-030-56877-1_16
149. Mahimna Kelkar, Soubhik Deb and Sreeram Kannan, "*Order-Fair Consensus in the Permissionless Setting*", May 2022, APKC '22: Proceedings of the 9th ACM on ASIA Public-Key Cryptography Workshop, p. 3-14, https://dl.acm.org/doi/10.1145/3494105.3526239
150. Zhuolun Li and Evangelos Pournaras, "*SoK: Consensus for Fair Message Ordering*", Nov 2024, https://arxiv.org/html/2411.09981v1
151. Amir Dembo, Sreeram Kannan, Ertem Nusret Tas, David Tse, Pramod Viswanath, Xuechao Wang, Ofer Zeitouni, "*Everything is a Race and Nakamoto Always Wins*", 2020, https://arxiv.org/abs/2005.10484
152. M. V.X. Ferreira et al., "*Optimal Strategic Mining against Cryptographic Self-Selection in Proof-of-Stake*", 2022, the 23rd ACM conference on Economics & Computation EC'22, pages 89-114.
153. Matheus V. X. Ferreira et al., "*Computing Optimal Manipulations in Cryptographic Self-Selection Proof-of-Stake Protocols*", June 2024, https://arxiv.org/abs/2406.15282
154. P. Gaži, A. Kiayias and A. Russell, "*Fait Accompli Committee Selection: Improving the Size-Security Tradeoff of Stake-Based Committees*", https://eprint.iacr.org/2023/1273
155. Nicola Dimitri, "*The Economics of Consensus in Algorand*", 2022, https://www.mdpi.com/2674-1032/1/2/13.
156. Tezos Nomadic Labs, "*On Adaptive Maximum vs. Adjusting The Staking Target*", Oct 2024, https://research-development.nomadic-labs.com/Adaptive-Maximum.html; last accessed 25/12/2024.
157. https://eth2book.info/capella/part2/incentives/inactivity/
158. Vitalik Buterin, "*Discouragement Attacks*", Dec 2018, https://github.com/ethereum/EIPs/blob/master/assets/eip-2982/ef-Discouragement-Attacks.pdf
159. M. Zhang, R. Li, and S. Duan, "*Max Attestation Matters: Making Honest Parties Lose Their Incentives in Ethereum PoS*", 2023, https://eprint.iacr.org/2023/1622.pdf; fixed in : https://github.com/ethereum/consensus-specs/pull/3339#issuecomment-1637117341
160. J. Bonneau, "*Why buy when you can rent? bribery attacks on bitcoin-style consensus*", In International Conference on Financial Cryptography and Data Security, 2016, pages 19–26.
161. J. S. Gans and H. Halaburda, "*Zero cost majority attacks on permissionless proof of work blockchains*", 2024, Management Science; https://dl.acm.org/doi/abs/10.1287/mnsc.2023.02426
162. Giacomo Giuliari, Alberto Sonnino, Marc Frei, Fabio Streun, Lefteris Kokoris-Kogias, and Adrian Perrig, Jun 2024, "*An Empirical Study of Consensus Protocols' DoS Resilience*", In Proceedings of the 19th ACM Asia Conference on Computer and Communications Security. 1345–1360; https://dl.acm.org/doi/10.1145/3634737.3656997
163. Abhishek Guru et al., "*A Survey on Consensus Protocols and Attacks on Blockchain Technology*", Feb 2023, https://www.mdpi.com/2076-3417/13/4/2604

164. https://en.bitcoin.it/wiki/Schnorr, https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki; last accessed 4/12/2024.
165. Boneh-Lynn-Shacham "*BLS digital signatures"*, https://eprint.iacr.org/2018/483.pdf; https://eth2book.info/capella/part2/building_blocks/signatures/; last accessed 9/1/2025.
166. Deepesh Singh, Gopikrishnan Muraleedharan, Boxiang Fu, Chen-Mou Cheng, Nicolas Roussy Newton, Peter P Rohde and Gavin K Brennen, "*Proof-of-Work Consensus by Quantum Sampling*", Deepesh Singh et al 2025 Quantum Sci. Technol. 10 025020; https://iopscience.iop.org/article/10.1088/2058-9565/adae2b
167. X. Fu, H. wang, P. shi, and X. Zhang, "*Teegraph: A Blockchain Consensus Algorithm Based on TEE and DAG for Data Sharing in IoT*", Jan 2022, Journal of Systems Architecture, Vol. 122-102344; https://www.sciencedirect.com/science/article/abs/pii/S1383762121002356
168. Shaokang Xie, Dakai Kang, Hanzheng Lyu, Jianyu Niu, Mohammad Sadoghi, "*Fides: Scalable Censorship-Resistant DAG Consensus via Trusted Components*", Jan 2025, https://arxiv.org/abs/2501.01062
169. Kevin Choi, Aathira Manoj and Joseph Bonneau,"*SoK: Distributed Randomness Beacons*", IEEE symposium on Security and Privacy Jun 2023, https://eprint.iacr.org/2023/728.pdf.
170. Yang Shi et al., "*Obfuscating Verifiable Random Functions for Proof-of-Stake Blockchains*", 2023, https://ieeexplore.ieee.org/document/10268576
171. Bong Gon Kim et al., "*Quantum-Secure Hybrid Blockchain System for DID-based Verifiable Random Function with NTRU Linkable Ring Signature*", Jan 2024, https://arxiv.org/abs/2401.16906
172. Suhyeon Lee et al., "*Implementation Study of Cost-Effective Verification for Pietrzak's Verifiable Delay Function in Ethereum Smart Contracts*", Aug 2024, https://arxiv.org/html/2405.06498v4
173. https://a16zcrypto.com/posts/article/leader-election-from-randomness-beacons-and-other-strategies/; last accessed 17/12/2024.
174. Dan Boneh et al., "*Post-Quantum Single Secret Leader Election (SSLE) From Publicly Re-randomizable Commitments*"; https://eprint.iacr.org/2023/1241
175. Olanrewaju Sanda, Michalis Pavlidis, Saeed Seraj, and Nikolaos Polatidis, "*Long-Range attack detection on permissionless blockchains using Deep Learning*", May 2023, Expert Systems with Applications, Vol.218, 119606 ; https://www.sciencedirect.com/science/article/pii/S0957417423001070
176. P.K. Makode, "*Agent-based Modelling of Stellar Consensus Protocol*", Master-Level, Zurich; https://www.ifi.uzh.ch/dam/jcr:9aa9b5bb-566b-4565-8828-b9a95633dba9/Parminder_ABM%20of%20Stellar.pdf
177. Nakhoon Choi and Heeyoul Kim, "*Technological Convergence of Blockchain and Artificial Intelligence: A Review and Challenges*", Electronics 2025, 14(1), 84; https://doi.org/10.3390/electronics14010084
178. https://beamroadmap.org/; last accessed 28/3/2025.
179. R. Belchior et al., "*Survey on Blockchain Interoperability: past, present and future*", 2021, https://dl.acm.org/doi/10.1145/3471140
180. https://www.chainalysis.com/blog/zetachain-security-halborn/, last accessed 8/9/2023.
181. https://wiki.polkadot.network/docs/learn-consensus; last accessed 15/4/2024.
182. https://www.horizen.io/research/; last accessed 27/10/2023.
183. A. Garoffolo, D. Kaidalov, and R. Oliynykov, "*Latus Incentive Scheme: Enabling Decentralization in Blockchains based on Recursive SNARKs*", Mar 2021, https://arxiv.org/pdf/2103.13754.pdf
184. Dawn Song et al., "*zkBridge: Trustless Cross-chain Bridges Made Practical*",2022, https://arxiv.org/abs/2210.00264

185. H. Yuan, S. Fei, and Z. Yan, "*Technologies of blockchain interoperability: a survey*", Aug 2023, https://www.sciencedirect.com/science/article/pii/S2352864823001335
186. Bernardo David, Peter Ga˘zi, Aggelos Kiayias, and Alexander Russell, "*Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain*", April 2023, https://eprint.iacr.org/2017/573.pdf
187. Z. Yin, B. Zhang, J. Xu, K. Lu, K. Ren, "*Bool network: An open, distributed, secure cross-chain notary platform*", IEEE Transactions on Information Forensics and Security 17(2022), 3465–3478, https://ieeexplore.ieee.org/abstract/document/9903072; https://bool.network/
188. Y. Geng et al., "*Subsidy Bridge: Rewarding Cross-Blockchain Relayers with Subsidy*", Information and Communications Security (pp.571-589), Oct 2023, DOI:10.1007/978-981-99-7356-9_34
189. M. Borkowski, M. Sigwart, P. Frauenthaler, T. Hukkinen, and S. Schulte, "*Dextt: Deterministic cross-blockchain token transfers*", IEEE access 7 (2019) 111030–111042, https://ieeexplore.ieee.org/abstract/document/8794500
190. D. Li, J. Liu, Z. Tang, Q. Wu, and Z. Guan, "*Agentchain: A decentralized cross-chain exchange system*", in: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (Trustcom/BigdataSE), IEEE, 2019, pp. 491–498, https://ieeexplore.ieee.org/abstract/document/8887310
191. Y. Hei, D. Li, Chi Zhang, J. Liu, Y. Liu, and Q. Wu, "*Practical AgentChain: A compatible cross-chain exchange system*", May 2022, https://doi.org/10.1016/j.future.2021.11.029
192. S. Shao, F. Chen, X. Xiao, W. Gu, Y. Lu, S. Wang, W. Tang, et al., "*IBE-BCIOT: An ibe based cross-chain communication mechanism of blockchain in IoT*", World Wide Web 24 (5) (2021) 1665–1690, https://link.springer.com/article/10.1007/s11280-021-00864-9
193. Aniruddha Bhattacharjya, Remigiusz Wisniewski and Venkatram Nidumolu, "*Holistic Research on Blockchain's Consensus Protocol Mechanisms with Security and Concurrency Analysis Aspects of CPS*", MDPI Electronics 2022, 11(17), 2760; https://doi.org/10.3390/electronics11172760
194. Boyuan Yang, "*Review of blockchain's consensus algorithms Comparative Analysis and Future Directions of Blockchain Consensus Mechanisms*", Dec 2024, Journal of Computing and Electronic Information Management 15(2):41-45, DOI: 10.54097/3jxvn691
195. I. Ibraham et al., **"***On the Optimality of Optimistic Responsiveness*", arXiv:2020.458
196. D. Constantinescu et al., **"***Unifying Partial Synchrony*", arXiv:2405.10249v1
197. A. Nagy et al., "*Forking the RANDAO: Manipulating Ethereum's Distributed Randomness Beacon*", preprint 2025, https://eprint.iacr.org/2025/037.pdf
198. D. Boneh, J. Bonneau, B. B¨ unz, and B. Fisch, "*Verifiable delay functions*," CRYPTO 2018.
199. Wesolowski, "*Efficient Verifiable Delay functions*", 2018, https://eprint.iacr.org/2018/623
200. D. Khovratovich and M. Maller, "*MinRoot: Candidate Sequential function for Ethereum VDF*",2022, https://eprint.iacr.org/2022/1626; https://crypto.ethereum.org/events/minrootanalysis2023.pdf
201. https://research-development.nomadic-labs.com/verifiable-delay-functions.html; last accessed 25/12/2024.
202. https://research.protocol.ai/publications/single-secret-leader-election/; last accessed 27/12/2024.

Appendix A: Formal Definitions of Important Term

Byzantine Fault-tolerant State Machine Replication [195]:

A Byzantine fault-tolerant state machine replication protocol commits client requests as a linearizable log to provide a consistent view of the log akin to a single non-faulty server, providing the following two guarantees.

- ***Safety***. Honest replicas do not commit different values at the same log position.
- ***Liveness***. Each client request is eventually committed by all honest replicas.

Partial Synchrony [196]

Synchronous communication assumes that messages get delivered within a publicly known timeframe and that parties' clocks are synchronized, while *Asynchronous* communication only assumes that messages get delivered eventually. A middle ground between the two extremes, is given by the *partially synchronous model*, which is arguably the most realistic option. This model comes in two commonly considered flavors:

1. The ***Global Stabilization Time (GST)*** model: after an (unknown) amount of time, the network becomes synchronous. This captures scenarios where network issues are transient.
2. The ***Unknown Latency (UL)*** model: the network is, in fact, synchronous, but the message delay bound is unknown.

The second case is what we called earlier in section 2 ***Δ-synchronous***; i.e., there is a value Δ such that every message sent by time t is delivered by time t+Δ but this value Δ is unknown to the protocol. The cited paper proves that the 2 cases, 1&2, can be treated equally by distributed computing protocols; i.e., any time agnostic property that can be achieved in the *Δ-synchronous* case, can as well be achieved in the GST case.

Consistency [9]

Consistency requires that, with probability at least $1 - \epsilon$, the following two conditions always hold:

1. ***No roll-backs***: If a transaction *tr* is confirmed for honest *p* at time *t*, then tr is confirmed for at all $t' \geq t$.
2. ***Confirmed transactions never conflict***: where if Tr and Tr′ are the sets of transactions confirmed for honest *p* and *p′* at *t* and *t′*, respectively, then Tr ∪ Tr′ is a valid set of transactions relative to S_0

(meaning that either Tr = Tr′ exactly or one of them is a subset of the other, and S_0 here is the initial stake distribution at time 0).

Liveness [9]

Blockchain protocols are run relative to a determined input $\epsilon \in [0,1)$, which is called *the security parameter* (In the deterministic model, $\epsilon = 0$). Liveness requires the existence of a value ℓ usually called *ℓatency*, which can depend on the determined inputs (such as ϵ and *d*) but must be sublinear in the network delay *d* (i.e., with $\ell =$

O(d) as $d \to \infty$), such that with probability at least $1-\epsilon$, the following condition is satisfied for every t. Suppose that:

- t^* :=max{GST, t}+ℓ is at most d
- The transaction tr is received by an honest player p at some timeslot ≤ t, and tr is valid for all honest players through timeslot t^*. (Formally, for every honest p and every $t' \in [t, t^*]$, if Tr is the set of transactions confirmed for at t', then Tr ∪ {tr} is a valid set of transactions relative to S_0).

Then, tr is confirmed for all honest players active at any timeslot after t^* and is confirmed for those players at the first timeslot ≥ t^* at which they are active.

Optimistic Responsiveness [8,122,195]:

Optimistic responsiveness [122] was introduced to shorten the latency ℓ of a synchronous Byzantine consensus protocol (or allow it to commit instantaneously when some optimistic conditions are met [195]), where ℓ is inherently lower bounded by the pessimistic bound on the network delay d (recall liveness definition).

A protocol is *optimistically responsive* [8] (with security parameter ϵ) if there exists a *liveness* parameter ℓ = O(δ) and a "*grace period*" parameter Δ^* = O(Δ) such that, in every instance consistent with the setting, with probability at least $1-\epsilon$, the following condition is satisfied for every t. Suppose that:

- t^* :=max{GST, t}+ℓ is at most d
- The transaction tr is received by an honest player p at some timeslot ≤ t, and tr is valid for all honest players through timeslot t^*.

Then, tr is confirmed for honest players at the first timeslot ≥ t^* at which they are active, and for all honest players active at any timeslot after t^*.

Put it simpler, an *optimistically responsive* protocol [122] can work faster with better scenarios, it can make a shortcut decision (with ℓ satisfying the above conditions) if the number of actual byzantine nodes is significantly smaller than the worst-case threshold f. The interested reader may follow the comparisons in both [122,195] between latency and Δ^* for different state of the art optimistic responsive protocols.

Accountability [3][32]:

For n nodes where f of them are Byzantine, the problem of *accountable consensus* is:

1- To solve consensus if the number of Byzantine faults is $f < n/3$
2- For every honest process to eventually output at least $f_d \geq n/3$ faulty processes if two honest processes output distinct decisions.

(meaning that it is always possible to find the guilty nodes with more than 1/3 of the nodes agreeing they are guilty, and guilty here means provably signed contradicting messages. Recall footnote 38 the protocol, especially when slashing is programmed inside a smart contract like in Ethereum, is neither aware nor responsible of their good intentions; however, putting penalties on nodes for going offline is not part of the

[32] Reference [3] cites *Polygraph* for this definition (Pierre Civit, Seth Gilbert, and Vincent Gramoli. Brief announcement: *Polygraph: Accountable byzantine agreement*. In DISC, pages 45:145:3, 2020; and In Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), pages 403413, 2021.)

accountability problem, it is assumed to be *impossible* in the Dynamically Available *DA* setting).

Confirmation Rule [141]:

A Confirmation Rule, within blockchain networks, refers to an algorithm implemented by network nodes that determines (either probabilistically or deterministically) the permanence of certain blocks on the blockchain. In other words, it is an algorithm that allows determining whether a block is confirmed, meaning that that will forever stay in the canonical chain of any honest validator under certain assumptions. Formally,

A Confirmation Rule for the fork-choice function FC_B is a tuple **$(CONF^v, sg)$** where

• $CONF^v$ is an algorithm that has access to the view of validator *v* and provides a function $CONF.isConfirmed_v$ which takes in input a block and a time, and outputs a boolean value

• *sg*, called *security guard*, is a function that takes in input a block, a time and the value of GST, and outputs a boolean value ensuring the following properties hold for any *block b* and *time t* such that **sg(b,t,GST) = True**

1. ***Safety***: $CONF.isConfirmed_v(b,t)$ implies that there exists a time t_0 such that for any $v' \in J$ and $t' \geq t_0$, b $FC_B^{v'}(t')$. Specifically, if a block *b* is confirmed at time *t*, there exists a finite time t_0 such that, at time t_0 and thereafter, *b* is part of the canonical chain of any validator $v' \in J$.

2.***Monotonicity***: $CONF.isConfirmed_v(b,t)$ implies that for any time $t' \geq t$, $CONF.isConfirmed_v(b,t')$. Specifically, once a block b is confirmed at time t, it remains confirmed for all future times $t' \geq t$.

(note that for both safety definitions here and above long range attacks, or all reorg attacks in general, are considered attacks against safety)

Checkpoints

(used in the Friendly Finality Gadget [21,141] and in Bitcoin Staking too [135,139]**):**

A check point *C* is a tuple of a block and an epoch; ***C=(b,e)*** where ***b*=block(*C*)** and ***e*=epoch(*C*)**. Checkpoints are periodical blocks that are used to guarantee finality gadget (recall Fig.4); i.e, blocks before the last finalized checkpoint are never reverted or rearranged (reorg). This achieved by applying a kind of a recursive rule that a block is finalized if it is directly connected to a thread of *finalized* checkpoints (checkpoints that are attested by a super majority, >2/3, of the validators); the recursive rules can be written formally as follows:

Gensis Checkpoint= $C(b_{gensis},0)$

If *C=(b,e)*, for every epoch *e' < e* if there exists a checkpoint *(b_c,e')*

Then b_c -» *b* , and *$slot(b_c)$* is the largest slot satisfying $slot(b_c) \leq e=epoch(b)$

{b_c is the last block in the previous epoch and belongs to the same chain as *b*}

Latest Checkpoint of a block b is C(b):=(b, epoch(b))

{the ones before it are, by previous rule, checkpoints in the same chain *b* is part of}

Time Malleability [8]

We call an oracle O *time malleable* if it satisfies the following conditions for each response function *f* in the support of O and for every pair (*q*, *t*):

- *f(q , t)= (r , t)* for some *r*, and;
- If *f(q , t)= (r , t)* then *f(q, t')= (r, t')* for all *t'*.

These conditions assert that every oracle response is delivered instantaneously and is independent of the time slot in which the query is made[33].

Appendix B: Leader/Committee Election and Randomization

Leader/Committee Election:

In POW blockchains the miner who wins the competition of solving the puzzle becomes the block proposer and gets to add a new block to the blockchain, while in POS family of blockchains block proposers are supposed to be selected probabilistically according to stake ratios. Hence, in POS, at each time unit (slot/epoch) a fresh subset of nodes is randomly elected to either be the leaders or form a BFT style committee that will next select a leader [154,169]; since stakes change dynamically with transactions, the probabilistic selection is done either according to previously locked stakes or the stake distribution at a previous finalized state (recall the warm-up and cool-down intervals mentioned in section 3).

Since the leader is the one who have the right to generate new blocks, the randomness used in the selection process is of great importance to avoid attacks and manipulations; researchers [72,169] have discussed known randomness metrics such as *unbiasability*, *liveness*, and an extra metric that has a lot of merits in blockchain consensus is *predictability*.

Predictability:

If a malicious node can predict when it will be elected as leader it can start many strategic attacks on the blockchain; moreover, if any adversary knew the selected leader prior proposing the block, it could bribe/penetrate or launch a DoS against that leader [162]. Informally, predictability refers to measuring the expected time interval a leader, and other nodes, can know earlier about the election results; this depends on the network delay (Δ in the partial synchronous definition, appendix A) since it defines who can know before who and when a node is considered to be offline (ex.: Fig.3 in [153]), and more manageably the predictability of the used source of randomness (so called *Distributed Random Beacon* ***DRB***).

-The authors of [72], 2020, measured predictability as a look ahead window; formally:

W-predictable: Given a POS protocol *P*, let C be a valid blockchain ending with block B with a time stamp t. We say a block B enables *w-length prediction*, if there exists a time $t_1 >$ t and a block B_1 with a time stamp t_1 such that:

[33] Note that most current random sources in blockchains depend on the predecessor block or chain of blocks; i.e., **not** time malleable.

(i) B_1 can be mined by miner (using its private state and the common public state) at time t

(ii) B_1 can be appended to *C'* to form a valid blockchain for any valid chain *C'* that extends C by appending *w*-1 valid blocks with time stamps within the interval (t, t_1).

By taking the maximum over the prediction length over all blocks in *P*, we say *P* is ***W-predictable***.

In short, *W* is the size of the prediction window measured in units of number of blocks; ex.: for the *Cardano* Ouroboros protocols *w=epoch length* (in slots) since random seed is refreshed every epoch and a block is appended (a new leader) every slot.

-Then the authors of [169], 2023, differentiated between 2 kinds of predictability to clarify between best (enough honest nodes are online so the random output appears at the end of epoch) and worst case (the epoch is stalled and no online nodes to compute the random seed, although rare in most blockchains) scenarios. Formally, suppose a DRB's epoch τ starts at time $T_{\tau,0}$ and finalizes (the random output Ω_τ becomes publicly available) at $T_{\tau,1}$ in the optimistic case (if every node is honest and online) and at $T_{\tau,2}$ in the worst case; we say:

- DRB is ***α-intra-unpredictable*** ($\alpha > 0$) if an adversary *A* participating in DRB_τ cannot predict any property of Ω_τ at time $T_{\tau,2-\alpha}$ with non-negligible advantage.
- DRB is ***β-inter-unpredictable*** ($\beta \geq 1$ if we are not on the optimistic case) if an adversary *A* cannot predict any property of $\Omega_{\tau+\beta'}$ (as defined above) for any $\beta' \geq \beta$ before $T_{\tau,2}$ with non-negligible advantage.

For most known DRBs (as in table1 of [159]), $\beta = 1$ and $\alpha = O(\Delta)$ for Δ-synchronous protocols; hence, we only include the predictability window in our table-2 in this appendix.

Verifiable Random Functions (VRF) [68,70/12.9-10,72,139,143]:

A verifiable random function is a cryptographic function that takes a series of inputs including a secret key and produces a pseudo random number that can be cryptographically (and efficiently) verified using the corresponding public key. VRFs are used as a robust random number generator by a wide variety of blockchain applications through oracles [68]; however, we are here interested in their usage for *randomized leader/committe election* in consensus protocols where a predictable or a biased leader election could lead to an endless possibilities of strategic game manipulations by different players [71,72]. Hence, in light of our main interest, we can characterize VRFs as cryptographic functions satisfying the following 4 properties:

1-Easy to evaluate using the secret key.

2-No one can predict the result without knowing the secret key.

3-Everyone can verify the result efficiently using the public key.

4-Result is expected to be uniformly distributed over the possible output space (random).
A simple example is signing a random message using a participant's private key.

Algorand (https://algorand.co) was the pioneer blockchain to use this VRF formula both in BFT committee selection and in leader election inside it. The rather complicated formula is to acquire stakers nearly equal chances (that is proportional to their stake) whether put all stakes in one or more *ephemeral VRF key pairs* (sk_i, pk_i); more on the effect of distributing the stake can be found in [70/12], Ferreira et al. work [152,153][34], and [155] is specifically about Algorand.

Let Ω_0 be a pure random seed for used forthe genesis block, at each round *n* each user "*i*" (wallet) calculates VRF_i using sk_i and checks if selected according to the following inequality:

$VRF_i(n) = \text{hash}[\text{time} \| VRF_{leader}(n-j)] < Q_i \tau$

(exactly $< 1 - e^{-\mu Q_i}$ where $\mu = \ln(1/(1-\tau))$ which leads to nearly the same for small values of τ)

time is the timestamp (for that if there is an empty slot for any reason, we still get new randomness next slot)
Qi is the number stake units of user i
τ is a tunable parameter that is supposed to be near to zero and presents the probability of selection per unit stake (the larger τ the higher the probability of getting more than 1 leader satisfying the threshold, while the smaller τ the higher the probability of no node satisfying the threshold and 0 leader for this round).
$VRF_{leader}(n-j)$ is the output randomness (the VRF of the elected leader) at step n-j; in general, this have the same security effect as using any function of it, i.e. $fn(VRF_{leader}(n-j))$ as opposed to using a function of the block itself.
j is another tunable parameter that presents how many blocks back in time the random source depends on so that we do not get a cascaded collapse if used a VRF of an "*unfinalized*" block (for Algorand *j*=1, for Tezos they use "n-5-2" because finality is after 2 blocks). However, since "*j*" is a known parameter, the result can be predictable unless a commit-then-reveal approach is used to hide the VRF value (*j* reflects *the prediction window*) and/or manipulatable (if the leader of round "n-*j*" was not obligated to use a predefined PKI in calculating its VRF they could try as many as possible values); Table2.

[34] They conjecture [152] that adversaries cannot increase their probability of being selected by splitting their stake into different wallets (due to the presence of Q_i in the inequality. However, they assume in their analysis of the possible gain from looking ahead T future rounds that an honest node puts all stakes in one wallet and an adversary distributes stake in a set A of say *k* wallets; their Lemma 10 in [153] (Lemma 3.1 in [152]) that for each winning strategy π that divides stake into *k* wallets, there exists a strategy π' where the adversary divides the stake into *2k* wallets and Reward (π') $\geq$ Reward (π).

Manipulatable/biasable:
If the formula used to evaluate the output randomness depends on transactions inside a previous block (or any external sources in general [24, 158]) then it becomes an economic incentives problem to define the threshold at which adversaries can manipulate the used value to get elected in a certain round; in fact even if it depends on the previous credentials as in Algorand, the references above shows exhaustive analysis that estimates (through simulation) the possible gain from studying T advanced rounds for stakes ratio as 29% and 38% .

Ethereum (https://eth2book.info/)
Uses an aggregated BLS signature (Aggregate($VRF_i(n)$= $BLS_i(n-2)$)) as an input to their robust DRB known as RANDAO.
The aggregation is used to hide previous inputs and also make the result less manipulatable by being a function of all committee member randomness; still, the last validator can foresee the effect on the selection and choose whether to aggregate or to not aggregate (pretend to be offline) its BLS signature on the resulting selected leader which is estimated to have the effect of 1-bit bias ability (called *the last revealer advantage* and could be extended to a number of last *k* colluding committee members), and that's why VDFs are needed as a final unpredictable step so that the last revealer cannot predict the effect of revealing.
It is worth mentioning that a very new cryptology preprint, Jan 2025, [197] introduced another *manipulation attack* on RANDAO that could be described as a form of grinding that depends on the adversary being elected as a leader on epoch boundary slots (called *tail slots* in the paper). The authors found it allows deliberate forking out honest leader blocks to maximize fee revenue with stake ≥ 20% and examined the times major stakers like Lido, Coinbase, Binance, …etc. were leaders of tail slots but did not find actual occurrence of such manipulation; however, the paper studied RANDAO leader election without the deployment of VDFs.

Dfinity (https://dfinity.org) & ***Drand*** (https://drand.love/)
To avoid dependency on a last revealer, Dfinity blockchain aggregates *BLS* signatures in a threshold multi-party computation *MPC* manner. To understand how this is possible, think of having *n* points on a polynomial of degree *t* (only needs *t*+1 points to solve) where $t+1 < n$; this way, *t*+1 aggregated signatures are enough to calculate the targeted aggregation which enables the system to tolerate *t* byzantine nodes. However, this involves extra communication and computation to prior exchange shares of a group secret key and combine partial signatures to the limit that could jeopardize protocol liveness if used in large scale blockchains.

Verifiable Delay Functions (VDF):
The references starting from [62] for *PoET* could be the first intuitions of the idea; i.e., a cryptographic proof that the evaluator had spent some amount of time in evaluating the function. The main theme is through repeated execution where the output of a step is the input of the next, and the challenge that necessitates a careful

primitive function choice is the use of more parallel computation power or mathematical shortcuts by malicious actors (recall $\theta(log\ log\ n/log\ n)$ threshold from [64] or even well-known shortcuts for repeated squaring).

VDFs were defined formally in 2018 [198] as a function f(.) satisfying:
1-given input x, anyone can compute f(x) and a "certificate of computing" π in T time steps.
2-No one, with any computing power and/or number of parallel machines, can compute f(x) in time << T.
3-given an alleged output y & a certificate π, anyone can quickly check if y=f(x).
Examples:
$(g^2)^T$ for some values of $g \in G$ (repeated squaring over groups of unknown order) [199] used in Chia and Tezos.
Ethereum ASIC VDF uses MinRoot (where an adversary cannot compute it in faster than 10x in expectation) [200]

VDFs role in blockchain consensus:
VDFs were first used in Chia as part of Proof of Time in conjunction with Proof of Space which could be viewed as part of selecting the block proposer; i.e., leader. Then, VDFs became widely used (or suggested) in POS[35] blockchains as part of the leader election process to mitigate the vulnerabilities of VRFs. If ***the VRF output*** was ***fed as an input to the VDF***, no one can predict the resulting output and/or manipulate it to their best interest. Fig.5 is an illustrative diagram from Tezos documents, while Table2 summarizes the vulnerabilities of all possibilities discussed above.

[35] Some, like (https://eprint.iacr.org/2021/660.pdf), consider VDF as an energy consuming computation process that deviates from the original POS concept and hence came the term "*Pure POS*" for POS blockchains that do not use them (or plan to in the near future).

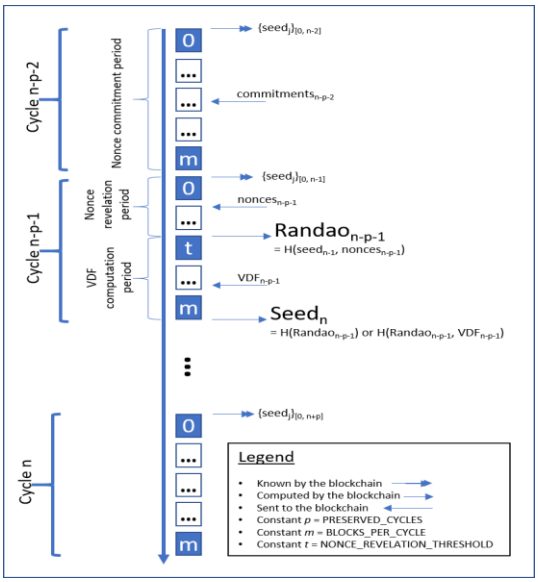

Fig.5: Creating a random seed in Tezos using both VDFs, and commit-then-reveal VRFs, adopted from [201]

Table 2 Randomness used in leader and Committee selection

	Predictable	**Manipulatable**	**Examples**	**Risks**
Hash (time)	Yes	No		Everyone can know in advance who is the selected leader
Hash(time|| fn(previous state n-*j*))	Yes W= *j*	Yes By jth leader		The block producer at state "j" can adjust the state (the block) to get a desired leader selection; to become the leader again for example.
Hash(time || fn(previous *i states* starting at n-j))	Yes W= *j-i*	Yes By the collusion of the *i leaders*	Cardano (n-j is the start of the previous epoch)	Same argument as above, but needs more colluding leaders
Hash(time || VRF(state n-j))	Yes W= *j*	Yes	Algorand	A staker with more than 1 staking unit can choose how many VRFs to

				create and whether to submit them or not [152,153,169] (***withholding attack***)
Hash(time|| VRF(state n-j)) + commit-reveal (commit to VRFs by publishing their hashes first then reveal (and use) them at next state	No	Yes	Previous Ethereum and Tezos	A staker is committed to VRFs, but still the *last k revealers* can choose whether it is best for them to reveal/not reveal (act as offline) their VRFs (***last revealer advantage***)
Threshold Multi Party Computation (MPC)	Yes	No	Dfinity and Drand	1-As any *t*+1 nodes can construct the VRF, also any *t*+1 can predict it. 2-Since secret shares are exchanged in an earlier step, no one can choose not to reveal. However, needs extensive communication cost of $O(n^2)$, which makes it impractical.
Hash(time|| VRF(state n-j)) + commit-reveal + VDF	No	No	Current Tezos and Ethereum	The effect of not adding your credential cannot be predicted unless able to compute the VDF fast than the protocol expected a powerful adversary is able to (thresholded by machine power)

Single Secret Leader Election (SSLE)

Conceptually, the term describes the optimal target of any leader selection process; to randomly and fairly select exactly 1 leader (single) in a way that is unpredictable and un-manipulatable by the participating players (secret; i.e., only the elected node discovers it is the leader till it reveals that later with a proof). Cryptographically, the term was defined in AFT'20 paper [202] and is still under research [174]; i.e., not currently deployed in any blockchain (the interested reader may find earlier dropped suggestions discussing possible implementations for Ethereum [173]).

Appendix C Table of famous blockchains with main consensus features like protocols, links to VRF/VDF used, …?

Blockchain	Consensus Criteria	Consensus Protocols	Slashing	Leader Election	Finality
Bitcoin	POW+ heaviest chain	Nakamoto Consensus [51]	No	A miner's block is selected based on its SHA256 satisfying a periodically adjusted number of leading zeros	*Probabilistic finality* 6 blocks~1hour
Ethereum	POS Weight fork choice= *accumulated sum of validator votes weighted by their staked balance*	Casper FFG [21] + LMD GHOST [103]	Yes	*VRF(BLS)+RANDAO* [128] Secret leader (https://ethresear.ch/t/secret-non-single-leader-election/11789) plans to add to ASIC *MinRoot VDF*	*Provable finality gadget* 64 blocks~2 epochs =13 mins (time is divided to epochs and slots within an epoch)
Tezos	POS+BFT	Tenderbake	Yes (Adaptive since 7/2024)	*RANDAO+VDF* Since 11/2022 https://research-development.nomadic-labs.com/verifiable-delay-functions.html	2 blocks ~ 30 secs
Algorand https://developer.algorand.org/docs/get-details/algorand_consensus/	Pure POS + BFT	https://eprint.iacr.org/2017/454	No	*VRF* https://developer.algorand.org/solutions/avm-evm-randomness/ https://github.com/algorand/go-algorand/blob/6d6f028446b96b42805f5e3b516d902117dcdc30/data/committee/credential.go#L77	Instant finality
Cardano https://docs.cardano.org/about-cardano/learn/c	DPOS + Longest Chain	Ouroboros Praos [143]	No	*VRF* -Fresh seed every epoch (dependent on the previous	1 day (epochs ~5days and slots ~1sec

onsensus-explained				epoch) to prevent grinding attacks [73] -Secret leader election	within an epoch)
Solana https://solana.com/developers/evm-to-svm/consensus	POS based variant of PBFT	*Proof of History* (acts as a global system clock) + POS-*TowerBFT* (a variant of PBFT)	No, but on the future roadmap https://solana.com/docs/economics/staking	*VRF (ORAO)* -an iterative SHA256 hash fn. -public leader -Fresh seed every epoch https://orao.network/solana-vrf https://docs.anza.xyz/consensus/leader-rotation/	32 blocks ~12 secs (epoch~2 days, Slot~400m sec)
COSMOS https://docs.cosmos.network/main/build/modules/consensus	DPOS+ Tendermint	*Bonded POS* (a variant of DPOS) *CometBFT* (a fork from Tendermint) https://docs.cometbft.com/v0.38/introduction/#what-is-cometbft	Yes	*Round Robin* leader selection (public) https://medium.com/@notional-ventures/cometbft-consensus-and-security-in-cosmos-part-2-8895525a2231#f2b3 (Still, there are VRF providers for applications)	Instant Finality
Polkadot https://wiki.polkadot.network/docs/learn-consensus	Nominated POS Hybrid consensus	BABE [181] + GRANDPA [22]	Yes	*VRF* *(Round Robin* at empty slots)	*Deterministic finality* of 10 blocks depth (still, time is divided to epochs and slots ~6secs within an epoch)
Chia https://docs.chia.net/consensus-analysis/	Proof of Space+ Proof of Time		No	*VDF* (repeated squaring) + infusion (for proof of time) https://docs.chia.net/consensus/Proo	*Probabilistic finality* of 6 blocks ~ 2 mins (like Bitcoin, but with

				f%20of%20Time%20(VDFs)%20_%20Chia%20Documentation.mhtml	byzantine threshold < 42.7% due to VDFs) -For rare attacks, 32 block ~ 10 mins