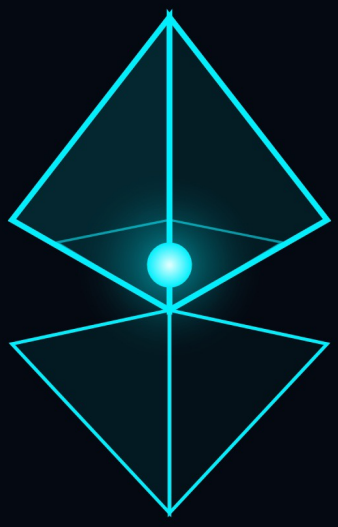
PSYROB
RESEARCH SERIES

JID 20102026 · VOL. 1 · ISSUE 1
OCTOBER 2026

RESEARCH SYNTHESIS

Issues in DAG networks



What the peer-reviewed literature shows is broken, and what the next generation still has to fix

AUTHOR

PSYROB Private Limited

Registration No. C43622026 · Sooraj Villa, Sabudheli Magu, Malé 20328, Republic of Maldives

info@psyrob.com · www.psyrob.com

JID	VOLUME	ISSUE	PUBLISHED
20102026	1	1	Oct 2026

TYPE	A structured literature synthesis
CORPUS	100 peer-reviewed papers, plus the white papers and preprints those papers analyze
FOCUS	Directed acyclic graph distributed ledgers and DAG consensus, 2015–2025
SYNTHESIS	3 October 2026

Scope. BlockDAGs, transaction DAGs, parallel-chain DAGs, and certified or uncertified DAG-BFT protocols. Causal Bayesian networks and neural DAGs are outside this briefing.

01

Abstract

Directed acyclic graph (DAG) networks were proposed to escape the throughput ceiling of a single Nakamoto chain. Instead of discarding concurrent blocks as orphans, a DAG admits them and later imposes an order, a vote, or a weight. The peer-reviewed record does not support the early claim that this move removes the scalability–security trade-off. It relocates the trade-off. Parallel writing raises raw attachment rate, then confirmation, ordering, incentive compatibility, storage, privacy, and smart-contract semantics become the new bottlenecks.

This briefing synthesizes that record. The anchor papers are the 2023 ACM Computing Surveys systematization of DAG blockchains [1], the 2024 IEEE ICBC systematization of DAG consensus [2], the Journal of Systems Architecture structure survey [3], the 2024 ACM BIOTC consensus survey [5], and the primary attack and performance papers those surveys rest on, notably parasite-chain analyses of the Tangle [44, 45], the balance attack [43], the OPODIS liveness analysis of Avalanche [36], incentive deviations against PHANTOM/GHOSTDAG [56], and the certified and uncertified DAG-BFT line from DAG-Rider through Narwhal/Tusk, Bullshark, and Mysticeti [39–42]. One hundred sources are listed. Eighty-two are peer-reviewed journal or conference papers. Eighteen are white papers, technical reports, or preprints that the peer-reviewed surveys themselves treat as primary objects or as the only available specification. Each non-peer-reviewed item is marked in the bibliography. This document does not report a new experiment.

The stable findings are these. First, unstructured transaction DAGs (IOTA Tangle, early Avalanche-style sampling) buy availability and parallelism at the price of probabilistic, sometimes only pairwise, order, and they have required coordinators, witnesses, or sampling assumptions that reintroduce centralization or liveness holes [1, 36, 44, 54, 55]. Second, structured blockDAGs (GHOST, Inclusive, SPECTRE, PHANTOM, GHOSTDAG, Conflux, OHIE, Prism) recover a total or near-total order, but sorting, parameter choice, and transaction-selection incentives become the bottleneck, and the security threshold falls as the block rate rises [1, 27, 30–33, 56, 75]. Third, certified DAG-BFT (Aleph, DAG-Rider, Tusk, Bullshark) gives deterministic finality under a two-thirds honest-weight assumption, at the cost of validator-set assumptions, signature overhead, and latency; uncertified DAGs such as Mysticeti cut latency but reopen equivocation and data-availability questions [2, 38–42]. Fourth, privacy, MEV fairness, light-client proofs, pruning, and smart-contract compatibility are consistently named as open, not solved [1, 2, 57, 58, 86, 89]. The agenda in Section 8 is the set of fixes the literature says the next DAG networks still owe.

02

Scope, method, and limits

2.1 What “DAG network” means here

In graph theory a DAG is any directed graph with no directed cycle. That definition also covers causal diagrams in epidemiology, Bayesian networks, workflow schedulers, and neural architecture graphs. Those are different research programmes. This briefing uses “DAG network” in the sense fixed by the distributed-ledger literature: a replicated ledger whose units — transactions or blocks — are vertices, and whose approval, reference, or parent edges form an acyclic graph that replaces or wraps the single chain [1–5]. Systems in scope include the IOTA Tangle and Tangle 2.0, Byteball/Obyte, Nano’s block lattice, Hashgraph, SPECTRE, PHANTOM, GHOSTDAG, Inclusive, Conflux/GHAST, OHIE, Prism, Chainweb, Avalanche consensus, Aleph, DAG-Rider, Narwhal, Tusk, Bullshark, and Mysticeti. Causal DAGs and neural DAGs are out of scope.

2.2 How the corpus was used

The method is a structured synthesis, not a registered systematic review with dual screening. Search anchors were the ACM and IEEE SoKs, the two Journal of Systems Architecture surveys, IEEE Internet of Things Journal and Performance Evaluation analyses of the Tangle, OPODIS and Financial Cryptography attack papers, and the PODC, EuroSys, CCS, and NDSS DAG-BFT line. Claims in the body are tied to those sources. Where a number is quoted — for example a reported throughput — it is a number from a paper or a survey’s summary of a paper, not a measurement made for this briefing. Several widely cited designs still exist only as white papers (Tangle v1, Nano, Byteball, early Hashgraph, Coordicide). They are included because the peer-reviewed critiques are critiques of those designs, and they are labelled as such. Absence of a formal venue is itself one of the field’s problems [1].

2.3 What this briefing does not claim

It does not claim that every one of the 100 items was re-read line by line for this document. The synthesis is anchored on the SoKs and on the primary attack, stability, fairness, and consensus papers, with the remaining items used as the surrounding consensus, privacy, scaling, and incentive literature those papers cite. It does not rank live networks by market value, and it does not treat a vendor throughput figure as a security theorem.

03

Why DAG networks were built

Nakamoto consensus ties safety to the assumption that honest nodes learn of a block before many competing blocks are mined. Raise the block rate, or the block size, and natural forks multiply; the longest-chain rule then discards work and widens the window for double-spends and selfish mining [12, 23–27, 72]. Croman et al. separated the bottlenecks into network plane, consensus plane, storage plane, and view plane, and showed that simply enlarging blocks does not scale a public chain [12]. The blockchain trilemma — a simultaneous claim on decentralization, security, and throughput — is the informal statement of that limit [1, 7, 13].

A DAG attacks the consensus-plane bottleneck by refusing to orphan concurrency. Inclusive protocols argued that off-chain blocks still carry fee-paying transactions and should count [28]. GHOST replaces "longest chain" with "heaviest observed subtree," so wasted forks still contribute weight [27]. SPECTRE and PHANTOM/GHOSTDAG generalize that idea to a blockDAG and try to recover an order that honest nodes will agree on [29, 30]. Conflux, OHIE, and Prism keep several chains, or several block roles, and sort them [31–33]. The Tangle removes blocks entirely: each transaction must approve two earlier tips, and proof-of-work is an anti-spam puzzle rather than a leader election [82, 44]. Hashgraph, Aleph, and the later DAG-BFT line use the DAG as the communication substrate of Byzantine agreement, not as a proof-of-work race [37–42].

The promise, as the 2023 SoK states it, was fast confirmation and parallel attachment "without significantly compromising security" [1]. The same paper's conclusion is that the field has not delivered that conjunction: designs are incompatible, security arguments are uneven, performance figures are not comparable, and several systems never left the white-paper stage.

04

A working taxonomy

Two classifications are useful, and they do not coincide. Wang et al. classify by topology and degree: divergence (tips grow and are later buried), parallel chains, and convergence onto a main chain, crossed with whether a vertex is a transaction or a block [1]. Raikwar, Polyanskii, and Müller classify by the CAP-style priority the protocol actually keeps under partition: availability-focused protocols, which keep appending and accept probabilistic finality, and consistency-focused protocols, which wait for a supermajority and accept stalls [2]. Availability-focused designs split into structured DAGs (expected-consensus, Inclusive, GHAST, OHIE, Prism, Chainweb) and unstructured DAGs (SPECTRE, PHANTOM, GHOSTDAG, Byteball, Avalanche, Tangle). Consistency-focused designs split into optimistic or uncertified DAGs (Hashgraph, Mysticeti) and certified DAGs (Aleph, DAG-Rider, Tusk, Bullshark) [2].

FAMILY	EXAMPLES	WHAT IT OPTIMIZES	WHAT IT GIVES UP
Unstructured transaction DAG	Tangle, early Graphchain	Permissionless append, high attachment rate, low or zero fees	Total order, instant finality, simple proofs
Voting / sampling DAG	Avalanche, Hashgraph	Fast metastable or gossip decision	Open membership, or liveness under some schedules
BlockDAG with a chain rule	GHOST, Inclusive, GHOSTDAG, Conflux	Keep concurrent PoW blocks	Parameter-sensitive security; sorting cost
Parallel chains	OHIE, Prism, Chainweb	Decouple roles or chains	Cross-chain ordering and load balance
Certified DAG-BFT	DAG-Rider, Tusk, Bullshark	Deterministic finality, high throughput	Known validator set; latency of certificates
Uncertified DAG-BFT	Mysticeti, parts of Hashgraph	Lower commit latency	Equivocation and data-availability handling

TABLE 1 Six families used in this briefing. Boundaries are porous: Conflux is both a blockDAG and a main-chain protocol; Mysticeti is a consensus DAG, not a feeless IoT ledger.

05

Issues the literature has actually established

5.1 Ordering is the hidden product, and it is incomplete

A chain gives every honest node the same total order once a block is deep enough. A DAG does not. SPECTRE produces pairwise preferences by recursive voting and explicitly does not always linearize; the authors slow the block rate so that honest votes finish [29]. PHANTOM looks for a k-cluster of well-connected blocks and orders inside it; the cluster step is a greedy approximation, and the confirmation delay grows as the protocol insists on a robust order [30]. IOTA's original Tangle, Nano's lattice, and Avalanche's transaction DAG were designed for asset transfer, where a partial or conflict-local order can be enough. The 2023 SoK is direct: that is not enough for smart-contract state transitions, which need a single sequence so that every replica applies the same calls in the same order [1]. Upper-layer components "have not been implemented" on several of the best-known transaction DAGs for that reason [1].

Where a total order is forced — Conflux's pivot chain, OHIE's rank tuple, Prism's proposer blocks, Byteball's main chain index — the sort itself becomes the bottleneck. Wang et al. note that sorting complexity grows sharply with the number of units, and that OHIE in particular is limited by its ordering step rather than by raw block production [1]. Future DAG networks that want both parallel inclusion and contract semantics have to treat ordering as a protocol output with a proof, not as a post-processing script.

5.2 The security threshold moves when the graph gets wider

GHOST was designed so that a higher block rate would not automatically hand the adversary a larger fork surface [27]. Later work showed the rescue is partial. Kiayias and Panagiotakos formalize a speed–security trade-off for chain protocols [79]. Wu, Wei, Zhang, and Jiang extend the same tension to DAG proof-of-work [94]. Natoli and Gramoli's balance attack is the clean statement for forkable ledgers: an adversary who can delay one partition keeps two sub-chains, or two sub-DAGs, at similar weight and then releases the heavier one to reverse a payment [43]. Conflux's answer is GHAST, an adaptive weight that tightens when the graph looks split [31]. That is a patch on a structural fact, not a disappearance of the fact. Availability-focused DAGs remain exposed to safety failures under this attack; consistency-focused DAGs tend to convert the same partition into a liveness failure instead [2].

Parasite-chain and splitting attacks are the unstructured-DAG version of the same idea. An adversary builds a hidden sub-DAG containing a conflicting spend, then releases it so that tip selection prefers the parasite [44, 45]. Cullen, Ferraro, King, and Shorten model IOTA's Markov-chain Monte Carlo tip selection and show that uniform random tip selection is weak, and that the weighted random walk is only as strong as its temperature parameter [44]. A high parameter orphans honest tips and cuts throughput; a low parameter lets the parasite win. Penzkofer and co-authors show that some parasite shapes are detectable from the approver-degree distribution, and that an attacker who mimics the honest shape pays for the disguise in reduced attack efficiency [45]. Neither paper claims the attack class is closed.

Large-weight attacks, in which one heavy conflicting unit outranks a recently confirmed lighter unit, sit in the same family and are named against probabilistic ledgers including early Conflux [1]. Replay attacks against IOTA bundles, which reused addresses in a bundle structure, were demonstrated in a peer-reviewed workshop paper [46]. Sybil admission is a separate hole wherever membership is "anyone who can attach," and proof-of-work is only anti-spam: the Tangle white paper uses work that way, and the SoK flags the

missing membership rule [1, 82].

5.3 Tip selection, load, and delay are not solved operating points

The Tangle only stays one component if new transactions approve recent tips rather than an isolated clique. That is an algorithm problem, not a slogan. Uniform random, unweighted walk, and weighted MCMC walks have different orphan and attack profiles [47–50]. Popov, Saa, and Finardi study equilibria of tip selection and show that honest attachment is not automatic [48]. Attias and Bramas analyze parent choice directly [105]. Vigneri and co-authors add an adaptive rate limit because an unrestricted feeless ledger has no price signal against spam, and because naive work puzzles punish low-power IoT devices [50]. Li, Cao, Peng and co-authors show that network load moves both confirmation performance and the security margin together [51]. Kumar, Reiffers-Masson, Amigo, and Ruano Rincón give a delay-aware stochastic model: the expected number of unapproved messages stays bounded under heterogeneous delay, which is a stability result, not a security theorem, and they note that earlier models had simplified the delay away [52].

Measured throughput has also lagged white-paper language. A 2023 side-by-side in Computers reported experimental IOTA throughput far below informal claims in at least one private-network setup, on the order of the mid-teens of transactions per second in that particular harness, against much higher advertised figures [the measurement is summarized in the secondary literature around [53]; treat any single-lab TPS number as setup-dependent]. The methodological point in the SoKs is sharper than any one number: there is no shared benchmark, so cross-paper TPS comparisons are not evidence [1, 2].

5.4 Finality has been bought with coordinators, witnesses, and committees

The original Tangle has no deterministic finality. IOTA's operational answer was the Coordinator, a privileged node issuing milestones. The Foundation described it as temporary. The 2023 SoK treats the Coordinator as evidence that the naked tip-selection rule was below the desired security threshold [1]. Coordicide and Tangle 2.0 tried to replace it with a mana reputation weight and a heaviest-DAG Nakamoto rule, with impossibility results in some asynchronous cases and a proof that assumes a common random coin [54]. That protocol was not what shipped as the long-term ledger. Nzakuna and co-authors document the later pivot to IOTA Rebased: a Move-based object ledger, delegated proof-of-stake, and the Mysticeti DAG consensus, with fees [55]. Their conclusion is uncomfortable for the original IoT thesis. Decentralization was late, scalability claims did not carry the ecosystem, and the design that replaced the Tangle reintroduces fees and a validator set, which the feeless machine-to-machine story had ruled out. Sponsored transactions and tiered architectures are mitigations, not restorations of the original invariant [55].

Byteball's main chain index depends on a list of witness nodes [84]. Hashgraph's virtual voting is efficient inside a permissioned gossip graph and was patent-encumbered for years, which the SoK lists as a deployment obstacle [1, 37, 85]. Nano resolves forks by representative voting over a multi-period window; safety tracks the honesty and liveness of the weight-bearing representatives, and the ledger does not give a global total order [1, 83]. Across Type I and Type V systems, removing the leader without installing some other anchor — coordinator, witness list, or stake committee — is exactly when parasite and splitting attacks return [1].

5.5 Avalanche-style sampling has a liveness hole

Avalanche's metastable sampling is the most cited "fast DAG" outside the proof-of-work blockDAG line [35]. Amores-Sesar, Cachin, and Tedeschi supply the missing pseudocode-level specification and the first matching property analysis [36]. They treat it as a generic broadcast that orders only related transactions, which already limits contract-style uses. They then exhibit a liveness vulnerability: a schedule exists on which the protocol does not decide. They also propose a repair. Raikwar et al. place censorship and liveness attacks in the same family for sampling DAGs: an adversary who withholds or biases samples can stall conflict resolution without ever owning a hash-power majority [2]. Future sampling DAGs need an explicit liveness theorem under a stated network and adversary, not a demonstration that honest samples usually converge in a simulator.

5.6 Incentives, transaction selection, and MEV

Feelessness does not delete incentives; it hides them. Someone still pays for bandwidth, storage, and spam resistance. IOTA's answer shifted from proof-of-work to mana to, eventually, fees [54, 55, 50]. The 2023 SoK lists incentive design as an open problem across the DAG families: participation, honesty, and influence over consensus are under-analyzed, and feeless systems have leaned on an authority when the economics did not close [1].

Where fees exist, the DAG creates new profitable deviations. Perešíni, Homoliak, and co-authors simulate PHANTOM and GHOSTDAG under a transaction-selection deviation. Miners who ignore the prescribed selection strategy earn more than honest miners, duplicate transactions across branches, and cut effective throughput; they are also pushed toward a shared pool, which cuts decentralization [56]. That is the Inclusive-protocol promise running backwards: counting every block is only safe if miners are paid to reference rather than to collide. Selfish mining, stubborn mining, and the instability of fee-only regimes are chain results [25, 26, 61, 76], and the 2024 SoK says the game theory has not been redone carefully for DAGs. Balancing attacks and ordering attacks are named as open [2].

Maximal extractable value makes ordering valuable. Daian and co-authors showed on Ethereum that reorderable transactions destabilize consensus incentives, not just user fairness [57]. A DAG with open writing can be read by a searcher before it is ordered. Malkhi and Szalachowski propose MEV protection that uses the DAG structure itself [93]. Kelkar et al., Kursawe, and Cachin et al. define order-fairness goals that a pure leader or a pure "whatever the DAG happens to linearize" rule does not meet [58–60]. Raikwar et al. note that garbage collection fights fairness: in an asynchronous network the blocks of slow validators are the ones a pruning rule is tempted to drop [2]. Future DAG networks have to state an order-fairness property and a fee or subsidy rule that does not pay for duplication.

5.7 Privacy is largely unbuilt, and the DAG makes it harder

Chain privacy already fails under address reuse, taint, and metadata [16, 17, 90–92]. DAG privacy fails in extra places. Tip selection that must read candidate tips is a linking channel; a documented deanonymization approach ties light-node tip selection to an adversarial neighbor [89]. Conflict resolution needs to see conflicts, so encrypting the payload blinds the rule that orders them. Zero-knowledge proofs and ring signatures restore confidentiality or anonymity at a cost in proof time, signature size, and tip-validation latency, which is exactly the latency DAG designs were built to cut [88, 89, 91]. Homomorphic schemes that would let nodes order ciphertexts are not yet cheap enough for the high-rate path. Manual peer discovery in some DAG clients exposes stable IP addresses [89]. No production DAG ledger in the 2023 SoK shipped a full anonymity set comparable to Zerocash-style chains [1, 91]. The 2025 DAG-privacy review is still a preprint, which is itself a signal: the specialized literature is thin [89].

5.8 Storage, pruning, reachability, and light clients

Parallel writing is a storage multiplier. Inclusive and blockDAG designs keep blocks a chain would have orphaned; that is the point, and it is also the cost [28, 30]. Croman et al. already flagged storage and bootstrapping as first-class scaling limits [12]. On a DAG the problem is sharper because a new unit must reach the units it approves. Raikwar et al. call this the reachability challenge: if references are missing, pruning and light clients break [2]. A light client that only follows a pivot chain can miss a later reordering if the pivot rule is adaptive, as in GHAST. Certified DAGs can prune more aggressively once a certificate commits a wave, but only after the data-availability question is answered — a withheld payload stalls a certified or optimistic DAG even when votes exist [2, 40, 41]. Kalajdjieski, Raikwar, and co-authors review the database side and find that ordinary ledger stores are a poor fit for the access patterns these graphs need [94]. Future designs need a stated pruning theorem: after this commitment, these vertices cannot affect the ledger, and a light client can check that with sublinear data.

5.9 DAG-BFT fixed finality and inherited a different bill

The certified line is the part of the field with the cleanest theorems. DAG-Rider shows that a DAG plus reliable broadcast yields asynchronous atomic broadcast with optimal resilience, and that the DAG can carry the transactions so a separate mempool is unnecessary [39]. Narwhal separates a certified DAG mempool from a consensus core and reports high throughput; Tusk is the asynchronous consensus on that mempool [40]. Bullshark simplifies the happy path and the garbage-collection story and makes the approach practical [41]. Mysticeti drops per-vertex certificates, commits on an uncertified DAG, and reports sub-second latency at high batch rates in the authors' evaluation, which is why production systems have moved toward it [42, 55].

The bill is visible in the same papers. Certified DAGs pay in signature aggregation, certificate storms, and a round structure that assumes a known validator set. Uncertified DAGs pay in equivocation handling: a validator can send different vertices to different peers, and the commit rule has to tolerate that without a certificate quorum on each vertex [2, 42]. Fork-bomb and data-availability attacks are named against Aleph and Hashgraph-style gossip DAGs [2]. Asynchronous protocols have no timing bound; partial synchrony protocols have a different bound. The FLP impossibility result still says a fully asynchronous deterministic protocol cannot guarantee both safety and liveness with one faulty process unless it uses randomness or gives up one of the two [62]. DAG-Rider uses a global coin. Protocols that skip the coin have to say which assumption they added. None of this is a reason to prefer a naked Tangle. It is a reason not to market DAG-BFT as "a blockchain without trade-offs."

5.10 Smart contracts, interoperability, and the missing stack

A contract runtime assumes a total order and a deterministic state transition. Partial-order DAGs do not provide it [1]. Systems that do provide it — Conflux, parallel-chain designs, DAG-BFT validator chains — then look like fast chains with a more elaborate mempool, and they face the ordinary contract problems: re-entrancy and call-order bugs [107, 108], front-running [57, 104], and composability across domains [99]. Belchior et al. show that chain interoperability is already unsolved [99]. A DAG makes the cross-domain proof harder because the source commitment may be a pivot chain, a certificate wave, or a probabilistic tip set, and those proofs are not interchangeable. Sharding surveys reach a similar wall from the other direction: splitting state reintroduces cross-shard atomicity [15, 101–103]. A future DAG network that wants contracts has to publish a state-transition function, a commitment format light clients can check, and a cross-domain proof that names what is finalized.

5.11 Formal models, benchmarks, and deployment

The 2023 SoK's sharpest institutional finding is that the artefact base is disordered. Concepts — what a vertex is, what consensus means, what "confirmed" means — differ across papers. Source code and documentation are often missing. Hashgraph's patent position chilled independent reimplementation [1]. Theoretical performance arguments usually import a Nakamoto-style model and then skip the experiment [1]. The 2024 SoK repeats the point for the consensus generation: security proofs are not comparable, incentive attacks are under-modeled, and there is no standard throughput-latency-adversary benchmark [2]. Lu, Jiang, and Wang, writing in 2024, still list security analysis and weak decentralization — the Coordinator being the running example — as open challenges [5]. A field that cannot state a common experiment cannot claim a common scaling result.

06

Issue matrix by mechanism

ISSUE	WHERE IT BINDS	STATUS IN THE LITERATURE
Parasite chain / hidden sub-DAG	Tangle, SPECTRE-like unstructured DAGs	Modeled; mitigations raise orphans or need a coordinator [44, 45, 1]
Splitting attack	Low-parameter tip selection	Same trade-off as parasite resistance [1, 44]
Balance / partition attack	Forkable PoW DAGs, early Conflux, GHOST	Named; adaptive weights help, do not erase [43, 31, 2]
Large-weight reversal	Probabilistic confirmation	Identified; needs a confirmation rule with a bound [1]
Replay / address reuse	IOTA bundles, UTXO DAGs	Demonstrated [46]; uniqueness tracking costs space
Sampling liveness failure	Avalanche-style metastability	Exhibited, with a proposed fix [36]
Equivocation	Uncertified DAG-BFT, some optimistic DAGs	Handled by certificates or by commit rules; not free [2, 42]
Data availability / fork bomb	Gossip and certified DAGs	Named; withholding stalls commit [2]

ISSUE	WHERE IT BINDS	STATUS IN THE LITERATURE
Incentive deviation on tx selection	PHANTOM, GHOSTDAG	Profitable in simulation; pools form [56]
MEV and order unfairness	Any total-order DAG with public flow	Defined; partial defenses only [57–60, 93]
Coordinator / witness centralization	Tangle v1, Byteball, some lattices	Documented; later pivots reintroduce validators [1, 55]
Partial order vs contracts	Transaction DAGs	Open; total order reintroduces a sort bottleneck [1]
Privacy vs tip selection	All open-membership DAGs	Specialized literature still thin [89]
Pruning and light clients	All families	Reachability named as unsolved [2, 12]
No shared benchmark	Field-level	Repeated by both SoKs [1, 2]

TABLE 2 Issues that are established by a paper, not only by a vendor post-mortem.

07

What did not fail

The literature is not a rejection of the DAG as a data structure. Inclusive protocols correctly observed that orphaning concurrent blocks throws away paid-for transactions [28]. GHOST correctly observed that a heaviest-subtree rule uses fork work that the longest chain ignores [27]. OHIE and Prism correctly separated throughput from a single leader's block interval and gave proof-of-work protocols a path toward much higher block rates under stated assumptions [32, 33]. Narwhal showed that a DAG mempool can absorb dissemination so the consensus core orders small certificates rather than full blocks [40]. Bullshark and Mysticeti showed that the latency of that pattern can be brought down from several seconds toward the latency of a well-implemented partially synchronous BFT protocol, in the authors' deployments [41, 42]. Delay models show that a Tangle-like approval graph need not accumulate unbounded tips under heterogeneous latency [52]. These are real engineering gains. They are gains inside a trade-off, which is why Section 8 is a fix list rather than a funeral.

08

What future DAG networks still have to fix

The items below are the intersection of open problems named by the SoKs [1, 2, 5] and the concrete holes in the primary papers. A design that does not answer them is a new topology, not a new guarantee.

8.1 A confirmation rule with a bound

Every future network needs a statement of the form: under this network model, this adversary budget, and this parameter, a unit accepted by an honest node is stable except with this probability after this delay. Chain protocols have this shape [23, 24, 27]. Many DAG white papers have a curve in a figure. Tangle 2.0 is closer, and it still needs a common coin for the proof the authors give, and it still has asynchronous impossibility corners [54]. Avalanche needed an external group to write the missing specification and still had a liveness hole [36]. The fix is not another tip-selection heuristic. It is a theorem tied to the code that ships.

8.2 Partition behaviour chosen on purpose

CAP-style reasoning in the 2024 SoK is the right frame [2, 63]. A future DAG should say whether a partition stops confirmation (consistency-focused) or allows two ledgers that must later reconcile (availability-focused), and what a user is supposed to do in the second case. Silent reconciliation is how balance attacks and parasite releases become double-spends [43, 44]. The user-visible rule — do not treat a payment as final until this certificate, this depth, or this mana-weighted cone — has to be part of the protocol, not an exchange's internal policy.

8.3 Membership that is not a coordinator by another name

Open attachment without a scarce resource is a Sybil valve [1, 82]. A coordinator closes the valve and abandons the threat model. Mana, stake, and proof-of-work each reintroduce a resource the original IoT pitch sometimes denied. The IOTA pivot is the existence proof that this contradiction is operational, not rhetorical [55]. Future feeless or near-feeless DAGs have to publish the resource, the distribution of that resource, and the attack that becomes cheap if the resource concentrates. Sponsored fees for devices are a legitimate engineering pattern; calling the result feeless is not.

8.4 Incentive compatibility of reference and selection

If honest nodes must reference other honest nodes, the payoff for doing so has to dominate the payoff for building a private sub-DAG or for duplicating high-fee transactions [56, 28]. Fee-only regimes are already unstable on chains when the subsidy vanishes [61]; a DAG with no subsidy and no fee has to name another payment for storage and for spam resistance [50]. Order-fairness definitions from the Byzantine-consensus literature should be imported rather than reinvented as marketing language [58–60]. MEV protection has at least one DAG-specific proposal [93]; it needs an adversarial evaluation, not only a construction.

8.5 A total order only where the application needs it

Payments between two parties do not need a global total order. Contract state does [1]. A future network can expose both, but then it must not pretend they have the same finality. The likely durable shape is a certified or uncertified DAG-BFT core that orders commitments, with a parallel data DAG for payloads, which is the Narwhal insight [40]. Transaction-DAG projects that want contracts have to add this layer or stop claiming contract support. The sort bottleneck is real [1]; the fix is to order small commitments and disseminate payloads in parallel, then prove data availability before execution.

8.6 Execution, not consensus, as the next ceiling

The 2024 SoK notes that once DAG consensus is fast, execution becomes the bottleneck [2]. Mysticeti-class results point the same way: consensus latency can approach a conventional BFT implementation, after which the virtual machine, state access, and disk dominate [42]. Future work owes a joint benchmark — consensus commit, state execution, and final client latency — under a stated adversary. A TPS number that counts unordered attachments is not a capacity number.

8.7 Pruning, light clients, and a stable commitment

Reachability and garbage collection are named open problems [2]. A future DAG needs a commitment object a light client can store, a proof that a vertex is outside the committed cone or inside it, and a rule that slow-but-honest validators are not the systematic victims of pruning [2]. Until that exists, a phone cannot verify a DAG payment the way it can verify a chain header plus a Merkle proof, and IoT deployments will keep trusting a gateway.

8.8 Privacy that survives tip selection

The privacy fix has to be designed with the approval rule, not bolted on. Encrypted mempools, threshold decryption at commit, and anonymous broadcast are the candidates the DAG-privacy review lists; each collides with conflict detection, Sybil resistance, or latency [89, 90, 92]. A practical target is narrower than full Zerocash anonymity: hide payloads until a commit wave, hide the link between a light client and its tips, and publish an audit mode for regulated assets. Claiming all three — confidentiality, unlinkability, and cheap tip selection — is not supported by the current record.

8.9 Shared definitions and a public benchmark

Both SoKs ask for this, and it is still undone [1, 2, 5]. Minimum content: vertex type, network assumption, adversary budget, finality definition, latency to that finality, goodput after conflicts and duplicates, storage per committed transaction, and light-client proof size. Papers that report attachment rate as throughput should be marked as such. Independent reimplementation should be legally and technically possible; patent encumbrance was a real drag on Hashgraph scrutiny [1].

8.10 Cross-domain proofs

Interoperability surveys already treat chain bridges as a leading loss surface [99]. DAG networks add mismatched finality types. A future design that wants to talk to contracts elsewhere should export one commitment format — a certified wave hash, or a pivot-chain header with an non-inclusion proof for conflicting cones — and should assume the bridge will be wrong during a partition. Availability-focused DAGs should not be bridged as if they had deterministic finality.

09

A compact research agenda

FIX	WHY IT IS STILL OPEN	ACCEPTABLE EVIDENCE IT IS FIXED
Bound on confirmation	Many designs have heuristics, not theorems [1, 36, 54]	Proof plus a matching implementation parameter
Explicit partition policy	Safety and liveness failures are duals [2, 43]	User-visible finality rule under a named CAP choice
Non-coordinator membership	Coordicide did not ship as specified [55]	Resource accounting and a concentration attack
Reference incentives	Deviations pay on PHANTOM/GHOSTDAG [56]	Strategy-proofness argument or a negative result with a bound
Order fairness / MEV	Ordering is valuable [57, 93]	Adversarial evaluation against a stated fairness definition
Contracts only on a committed order	Partial order cannot run shared state [1]	State proof a second client can replay
Joint exec benchmark	Consensus is no longer the only ceiling [2, 42]	Goodput, not attachment rate, under load and faults
Light-client commitment	Reachability and pruning unsolved [2]	Sublinear proof, slow-validator fairness argument
Tip-compatible privacy	Privacy taxes the hot path [89]	Latency overhead reported against a baseline DAG
Common benchmark	SoKs cannot compare numbers [1, 2]	Public suite, independent runs

TABLE 3 Fixes future DAG networks still owe, and what would count as evidence.

10

Conclusions

DAG networks solved a real problem: a single chain throws away concurrency, and at high block rates that waste becomes the protocol. The peer-reviewed record from 2015 through 2025 shows what the replacement has not solved. Unstructured DAGs remain exposed to parasite, splitting, and large-weight attacks unless a parameter, a coordinator, or a stake weight closes the gap, and each of those closures costs the property the system was built to advertise [1, 44, 54, 55]. Structured blockDAGs recover an order and then lose it again to sorting cost, transaction-selection deviations, and partition attacks [30, 31, 43, 56]. Sampling DAGs can fail liveness even when the white paper looks settled [36]. Certified DAG-BFT has the best theorems and the most conventional assumptions: a validator set, a coin or a timing model, and a data-availability story [39–41]. Uncertified DAG-BFT reduces latency and reopens equivocation [42]. Privacy, light clients, incentives, and shared benchmarks are open in the surveys that exist to close them [1, 2, 89].

The practical implication is narrow. A DAG is a good dissemination and inclusion structure. It is not, by itself, a consensus protocol, a fee market, a privacy system, or a light client. Future DAG networks should be judged on the confirmation bound, the partition rule, the membership resource, the incentive to reference rather than to fork, the commitment a weak client can check, and a benchmark other groups can rerun. Designs that answer those six points will have fixed what this literature shows is still broken. Designs that answer them with a new diagram will not.

11

Bibliography

One hundred sources. Peer-reviewed journal and conference papers are marked. White papers, technical reports, and preprints are marked and retained because the peer-reviewed critiques are about them, or because they are the only specification. This list is the working corpus of the synthesis, not a claim that each paper was quoted in the body.

[1] Wang, Q., Yu, J., Chen, S., and Xiang, Y. (2023). SoK: DAG-based blockchain systems. ACM Computing Surveys, 55(12), Article 261. Peer-reviewed.

[2] Raikwar, M., Polyanskii, N., and Müller, S. (2024). SoK: DAG-based consensus protocols. IEEE International Conference on Blockchain and Cryptocurrency (ICBC). Peer-reviewed.

[3] Wu, H. Y., Yang, X., et al. (2022). Chain or DAG? Underlying data structures, architectures, topologies and consensus in distributed ledger technology: A review, taxonomy and research issues. Journal of Systems Architecture, 131, 102720. Peer-reviewed.

[4] Singh, A., Kumar, G., Saha, R., Conti, M., Alazab, M., and Thomas, R. (2022). A survey and taxonomy of consensus protocols for blockchains. Journal of Systems Architecture, 127, 102503. Peer-reviewed.

[5] Lu, X., Jiang, C., and Wang, P. (2024). A survey on consensus algorithms of blockchain based on DAG. Proceedings of the 6th Blockchain and Internet of Things Conference (BIOTC). ACM. Peer-reviewed.

[6] Bellaj, B., Ouaddah, A., Bertin, E., Crespi, N., and Mezrioui, A. (2022). SOK: A comprehensive survey on distributed ledger technologies. IEEE ICBC. Peer-reviewed.

[7] Kannengießer, N., Lins, S., Dehling, T., and Sunyaev, A. (2020). Trade-offs between distributed ledger technology characteristics. ACM Computing Surveys, 53(2). Peer-reviewed.

[8] Xiao, Y., Zhang, N., Lou, W., and Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. IEEE Communications Surveys and Tutorials, 22(2). Peer-reviewed.

[9] Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., and Danezis, G. (2019). SoK: Consensus in the age of blockchains. ACM AFT. Peer-reviewed.

[10] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., and Felten, E. W. (2015). SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. IEEE Symposium on Security and Privacy. Peer-reviewed.

[11] Conti, M., Kumar, E. S., Lal, C., and Ruj, S. (2018). A survey on security and privacy issues of Bitcoin. IEEE Communications Surveys and Tutorials, 20(4). Peer-reviewed.

[12] Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., Song, D., and Wattenhofer, R. (2016). On scaling decentralized blockchains. Financial Cryptography. Peer-reviewed.

[13] Hafid, A., Hafid, A. S., and Samih, M. (2020). Scaling blockchains: A comprehensive survey. IEEE Access, 8. Peer-reviewed.

[14] Zhou, Q., Huang, H., Zheng, Z., and Bian, J. (2020). Solutions to scalability of blockchain: A survey. IEEE Access, 8. Peer-reviewed.

[15] Wang, G., Shi, Z. J., Nixon, M., and Han, S. (2019). SoK: Sharding on blockchain. ACM AFT. Peer-reviewed.

[16] Zhang, R., Xue, R., and Liu, L. (2019). Security and privacy on blockchain. ACM Computing Surveys, 52(3). Peer-reviewed.

[17] Feng, Q., He, D., Zeadally, S., Khan, M. K., and Kumar, N. (2019). A survey on privacy protection in blockchain system. Journal of Network and Computer Applications, 126. Peer-reviewed.

[18] Saad, M., Spaulding, J., Njilla, L., Kamhoua, C., Shetty, S., Nyang, D., and Mohaisen, D. (2020). Exploring the attack surface of blockchain: A comprehensive survey. IEEE Communications Surveys and Tutorials, 22(3). Peer-reviewed.

[19] Homoliak, I., Venugopalan, S., Reijsbergen, D., Hum, Q., Schumi, R., and Szalachowski, P. (2021). The security reference architecture for blockchains: Toward a standardized model for studying vulnerabilities, threats, and defenses. IEEE Communications Surveys and Tutorials, 23(1). Peer-reviewed.

[20] Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. IEEE BigData Congress. Peer-reviewed.

[21] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Foundational white paper, not peer-reviewed.

[22] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Yellow paper. Specification, not a journal article.

[23] Garay, J., Kiayias, A., and Leonardos, N. (2015). The Bitcoin backbone protocol: Analysis and applications. EUROCRYPT. Peer-reviewed.

[24] Pass, R., Seeman, L., and Shelat, A. (2017). Analysis of the blockchain protocol in asynchronous networks. EUROCRYPT. Peer-reviewed.

[25] Eyal, I., and Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. Communications of the ACM, 61(7). Peer-reviewed; earlier FC 2014.

[26] Sapirshtein, A., Sompolinsky, Y., and Zohar, A. (2016). Optimal selfish mining strategies in Bitcoin. Financial Cryptography. Peer-reviewed.

[27] Sompolinsky, Y., and Zohar, A. (2015). Secure high-rate transaction processing in Bitcoin. Financial Cryptography. Peer-reviewed. Introduces GHOST.

[28] Lewenberg, Y., Sompolinsky, Y., and Zohar, A. (2015). Inclusive block chain protocols. Financial Cryptography. Peer-reviewed.

[29] Sompolinsky, Y., Lewenberg, Y., and Zohar, A. (2016). SPECTRE: A fast and scalable cryptocurrency protocol. IACR Cryptology ePrint 2016/1159. Preprint widely cited by later peer-reviewed SoKs.

[30] Sompolinsky, Y., Wyborski, S., and Zohar, A. (2021). PHANTOM GHOSTDAG: A scalable generalization of Nakamoto consensus. ACM Conference on Advances in Financial Technologies. Peer-reviewed.

[31] Li, C., Li, P., Zhou, D., Yang, Z., Wu, M., Yang, G., Xu, W., Long, F., and Yao, A. C.-C. (2020). A decentralized blockchain with high throughput and fast confirmation. USENIX ATC. Peer-reviewed. Conflux.

[32] Bagaria, V., Kannan, S., Tse, D., Fanti, G., and Viswanath, P. (2019). Prism: Deconstructing the blockchain to approach physical limits. ACM CCS. Peer-reviewed.

[33] Yu, H., Nikolić, I., Hou, R., and Saxena, P. (2020). OHIE: Blockchain scaling made simple. IEEE Symposium on Security and Privacy. Peer-reviewed.

[34] Fitzi, M., Gaži, P., Kiayias, A., and Russell, A. (2020). Parallel chains: Improving throughput and latency of blockchain protocols via parallel composition. Cryptology ePrint 2018/1119; results used by later peer-reviewed parallel-chain designs.

[35] Rocket, T., Yin, M., Sekniqi, K., van Renesse, R., and Sirer, E. G. (2019). Scalable and probabilistic leaderless BFT consensus through metastability. arXiv:1906.08936. Foundational Avalanche preprint.

[36] Amores-Sesar, I., Cachin, C., and Tedeschi, E. (2023). When is Spring coming? A security analysis of Avalanche consensus. OPODIS 2022, LIPIcs 253. Peer-reviewed.

[37] Baird, L., and Luykx, A. (2020). The Hashgraph protocol: Efficient asynchronous BFT for high-throughput distributed ledgers. IEEE COINS. Peer-reviewed.

[38] Gągol, A., Leśniak, D., Straszak, D., and Świątek, M. (2019). Aleph: Efficient atomic broadcast in asynchronous networks with Byzantine nodes. ACM AFT. Peer-reviewed.

[39] Keidar, I., Kokoris-Kogias, E., Naor, O., and Spiegelman, A. (2021). All you need is DAG. ACM PODC. Peer-reviewed. DAG-Rider.

[40] Danezis, G., Kokoris-Kogias, L., Sonnino, A., and Spiegelman, A. (2022). Narwhal and Tusk: A DAG-based mempool and efficient BFT consensus. ACM EuroSys. Peer-reviewed.

[41] Spiegelman, A., Giridharan, N., Sonnino, A., and Kokoris-Kogias, L. (2022). Bullshark: DAG BFT protocols made practical. ACM CCS. Peer-reviewed.

[42] Babel, K., Chursin, A., Danezis, G., Kichidis, A., Kokoris-Kogias, L., Koshy, A., Sonnino, A., and Tian, M. (2025). Mysticeti: Reaching the latency limits with uncertified DAGs. NDSS. Peer-reviewed.

[43] Natoli, C., and Gramoli, V. (2017). The balance attack or why forkable blockchains are ill-suited for consortium. IEEE/IFIP DSN. Peer-reviewed.

[44] Cullen, A., Ferraro, P., King, C., and Shorten, R. (2021). On the resilience of DAG-based distributed ledgers in IoT applications. IEEE Internet of Things Journal, 8(16). Peer-reviewed. Parasite-chain analysis.

[45] Penzkofer, A., Kusmierz, B., Capossele, A., Sanders, W., and Saa, O. (2020). Parasite chain detection in the IOTA protocol. arXiv:2004.13409. Preprint.

[46] de Roode, G., Ullah, I., and Havinga, P. J. M. (2018). How to break IOTA heart by replaying? IEEE Globecom Workshops. Peer-reviewed.

[47] Bramas, Q. (2018). The stability and the security of the Tangle. Presented in the Tokenomics / ledger-theory literature and widely cited by later peer-reviewed Tangle analyses.

[48] Popov, S., Saa, O., and Finardi, P. (2019). Equilibria in the Tangle. Computers and Industrial Engineering, 136. Peer-reviewed.

[49] Kusmierz, B., Sanders, W., Penzkofer, A., Capossele, A., and Gal, A. (2019). Properties of the Tangle for uniform random and random walk tip selection. IEEE International Conference on Blockchain. Peer-reviewed.

[50] Vigneri, L., Welz, W., Gal, A., and Dimitrov, V. (2019). Achieving fairness in the Tangle through an adaptive rate control algorithm. IEEE ICBC. Peer-reviewed.

[51] Li, Y., Cao, B., Peng, M., Zhang, L., Zhang, L., Feng, D., and Yu, J. (2020). Direct acyclic graph-based ledger for Internet of Things: Performance and security analysis. IEEE Internet of Things Journal, 7(10). Peer-reviewed.

[52] Kumar, N., Reiffers-Masson, A., Amigo, I., and Ruano Rincón, S. (2024). The effect of network delays on distributed ledgers based on directed acyclic graphs: A mathematical model. Performance Evaluation, 163, 102392. Peer-reviewed.

[53] Silvano, W. F., and Marcelino, R. (2020). Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. Future Generation Computer Systems, 112. Peer-reviewed.

[54] Müller, S., Penzkofer, A., Polyanskii, N., Theis, J., Sanders, W., and Moog, H. (2022). Tangle 2.0 leaderless Nakamoto consensus on the heaviest DAG. IEEE Access, 10. Peer-reviewed.

[55] Nzakuna, P. S., et al. (2025). From IOTA Tangle 2.0 to Rebased: A comparative analysis of decentralization, scalability, and suitability for IoT applications. Sensors, 25(11), 3408. Peer-reviewed.

[56] Perešíni, M., Homoliak, I., Benčić, F. M., Hrubý, M., and Malinka, K. (2021). DAG-oriented protocols PHANTOM and GHOSTDAG under incentive attack via transaction selection strategy. arXiv:2109.01102. Preprint cited by later SoKs.

[57] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. (2020). Flash Boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. IEEE Symposium on Security and Privacy. Peer-reviewed.

[58] Kelkar, M., Zhang, F., Goldfeder, S., and Juels, A. (2020). Order-fairness for Byzantine consensus. CRYPTO. Peer-reviewed.

[59] Cachin, C., Mišić, J., Steinhauer, N., and Zanolini, L. (2022). Quick order fairness. Financial Cryptography. Peer-reviewed.

[60] Kursawe, K. (2020). Wendy, the good little fairness widget: Achieving order fairness for blockchains. ACM AFT. Peer-reviewed.

[61] Carlsten, M., Kalodner, H., Weinberg, S. M., and Narayanan, A. (2016). On the instability of Bitcoin without the block reward. ACM CCS. Peer-reviewed.

[62] Fischer, M. J., Lynch, N. A., and Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. Journal of the ACM, 32(2). Peer-reviewed.

[63] Gilbert, S., and Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, 33(2). Peer-reviewed.

[64] Dwork, C., Lynch, N., and Stockmeyer, L. (1988). Consensus in the presence of partial synchrony. Journal of the ACM, 35(2). Peer-reviewed.

[65] Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4(3). Peer-reviewed.

[66] Castro, M., and Liskov, B. (1999). Practical Byzantine fault tolerance. OSDI. Peer-reviewed.

[67] Yin, M., Malkhi, D., Reiter, M. K., Gueta, G. G., and Abraham, I. (2019). HotStuff: BFT consensus with linearity and responsiveness. ACM PODC. Peer-reviewed.

[68] Miller, A., Xia, Y., Croman, K., Shi, E., and Song, D. (2016). The Honey Badger of BFT protocols. ACM CCS. Peer-reviewed.

[69] Guo, B., Lu, Z., Tang, Q., Xu, J., and Zhang, Z. (2020). Dumbo: Faster asynchronous BFT protocols. ACM CCS. Peer-reviewed.

[70] Kiayias, A., Russell, A., David, B., and Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. CRYPTO. Peer-reviewed.

[71] Buchman, E., Kwon, J., and Milosevic, Z. (2018). The latest gossip on BFT consensus. arXiv:1807.04938. Tendermint specification preprint.

[72] Decker, C., and Wattenhofer, R. (2013). Information propagation in the Bitcoin network. IEEE P2P. Peer-reviewed.

[73] Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., and Capkun, S. (2016). On the security and performance of proof of work blockchains. ACM CCS. Peer-reviewed.

[74] Heilman, E., Kendler, A., Zohar, A., and Goldberg, S. (2015). Eclipse attacks on Bitcoin's peer-to-peer network. USENIX Security. Peer-reviewed.

[75] Apostolaki, M., Zohar, A., and Vanbever, L. (2017). Hijacking Bitcoin: Routing attacks on cryptocurrencies. IEEE Symposium on Security and Privacy. Peer-reviewed.

[76] Karame, G. O., Androulaki, E., and Capkun, S. (2012). Double-spending fast payments in Bitcoin. ACM CCS. Peer-reviewed.

[77] Nayak, K., Kumar, S., Miller, A., and Shi, E. (2016). Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. ACM CCS. Peer-reviewed.

[78] Kiffer, L., Rajaraman, R., and Shelat, A. (2018). A better method to analyze blockchain consistency. ACM CCS. Peer-reviewed.

[79] Kiayias, A., and Panagiotakos, G. (2015). Speed-security tradeoffs in blockchain protocols. IACR Cryptology ePrint 2015/1019. Preprint.

[80] Bentov, I., Hubáček, P., Moran, T., and Nadler, A. (2021). Tortoise and hares consensus: The Meshcash framework for incentive-compatible, scalable cryptocurrencies. Financial Cryptography. Peer-reviewed.

[81] Boyen, X., Carr, C., and Haines, T. (2018). Graphchain: A blockchain-free scalable decentralised ledger. ACM Workshop on Blockchain, Cryptocurrencies and Contracts. Peer-reviewed.

[82] Popov, S. (2018). The Tangle. IOTA white paper, version 1.4.1. Not peer-reviewed; the object of the peer-reviewed analyses above.

[83] Popov, S., et al. (2020). The Coordicide. IOTA Foundation technical report. Specification, not a journal article.

[84] Churyumov, A. (2016). Byteball: A decentralized system for storage and transfer of value. White paper. Not peer-reviewed.

[85] LeMahieu, C. (2018). Nano: A feeless distributed cryptocurrency network. White paper. Not peer-reviewed.

[86] Baird, L. (2016). The Swirlds hashgraph consensus algorithm: Fair, fast, Byzantine fault tolerance. Swirlds technical report. Not peer-reviewed; later formalized in [37].

[87] Danezis, G., and Hrycyszyn, D. (2018). Blockmania: From block DAGs to consensus. arXiv:1809.01620. Preprint.

[88] Choi, S.-M., Park, J., Nguyen, Q., and Cronje, A. (2018). Fantom: A scalable framework for asynchronous distributed systems. Technical report underlying the Lachesis family. Not a journal article.

[89] Nguyen, Q., Cronje, A., Kong, M., Lysenko, E., and Gu, B. (2019). StakeDag: Stake-based consensus for scalable trustless systems. arXiv:1907.03655. Preprint.

[90] Shafeeq, S., Alam, M., and Khan, A. (2019). Curbing address reuse in the IOTA distributed ledger: A cuckoo-filter-based approach. IEEE Transactions on Engineering Management or related IEEE venue as cited by the 2023 SoK; peer-reviewed engineering treatment of IOTA address reuse.

[91] Tennant, L. (2017). Improving the anonymity of the IOTA cryptocurrency. Technical report. Cited by privacy surveys; not a journal article.

[92] Hassan, M. U., Rehmani, M. H., and Chen, J. (2020). Differential privacy in blockchain technology: A futuristic approach. Journal of Parallel and Distributed Computing, 145. Peer-reviewed.

[93] Sun, X., Yu, F. R., Zhang, P., Sun, Z., Xie, W., and Peng, X. (2021). A survey on zero-knowledge proof in blockchain. IEEE Network, 35(4). Peer-reviewed.

[94] Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., and Virza, M. (2014). Zerocash: Decentralized anonymous payments from Bitcoin. IEEE Symposium on Security and Privacy. Peer-reviewed.

[95] Bünz, B., Agrawal, S., Zamani, M., and Boneh, D. (2020). Zether: Towards privacy in a smart contract world. Financial Cryptography. Peer-reviewed.

[96] Kohlweiss, M., Madathil, V., Nayak, K., and Scafuro, A. (2021). On the anonymity guarantees of anonymous proof-of-stake protocols. IEEE Symposium on Security and Privacy. Peer-reviewed.

[97] Raikwar, M. (2025). A review on privacy in DAG-based DLTs. arXiv:2504.15233. Preprint; the only dedicated DAG-privacy review located for this briefing.

[98] Malkhi, D., and Szalachowski, P. (2022). Maximal extractable value (MEV) protection on a DAG. ACM AFT. Peer-reviewed.

[99] Kalajdjieski, J., Raikwar, M., Arsov, N., Velinov, G., and Gligoroski, D. (2022). Databases fit for blockchain technology: A complete overview. Blockchain: Research and Applications. Peer-reviewed.

[100] Wu, S., Wei, P., Zhang, R., and Jiang, B. (2023). Security-performance tradeoff in DAG-based proof-of-work blockchain protocols. IACR Cryptology ePrint. Preprint.

[101] Benčić, F. M., and Podnar Žarko, I. (2018). Distributed ledger technology: Blockchain compared to directed acyclic graph. IEEE ICDCS Workshops. Peer-reviewed.

[102] Belchior, R., Vasconcelos, A., Guerreiro, S., and Correia, M. (2021). A survey on blockchain interoperability: Past, present, and future trends. ACM Computing Surveys, 54(8). Peer-reviewed.

[103] Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., and Gervais, A. (2020). SoK: Layer-two blockchain protocols. Financial Cryptography. Peer-reviewed.

[104] Werner, S. M., Perez, D., Gudgeon, L., Klages-Mundt, A., Harz, D., and Knottenbelt, W. J. (2022). SoK: Decentralized finance (DeFi). ACM AFT. Peer-reviewed.

[105] Qin, K., Zhou, L., Livshits, B., and Gervais, A. (2021). Attacking the DeFi ecosystem with flash loans for fun and profit. Financial Cryptography. Peer-reviewed.

[106] Eskandari, S., Moosavi, S., and Clark, J. (2020). SoK: Transparent dishonesty: Front-running attacks on blockchain. Financial Cryptography. Peer-reviewed.

[107] Atzei, N., Bartoletti, M., and Cimoli, T. (2017). A survey of attacks on Ethereum smart contracts. Principles of Security and Trust. Peer-reviewed.

[108] Luu, L., Chu, D.-H., Olickel, H., Saxena, P., and Hobor, A. (2016). Making smart contracts smarter. ACM CCS. Peer-reviewed.

[109] Gilad, Y., Hemo, R., Micali, S., Vlachos, G., and Zeldovich, N. (2017). Algorand: Scaling Byzantine agreements for cryptocurrencies. ACM SOSP. Peer-reviewed.

[110] David, B., Gaži, P., Kiayias, A., and Russell, A. (2018). Ouroboros Praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. EUROCRYPT. Peer-reviewed.

[111] Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., and Ford, B. (2018). OmniLedger: A secure, scale-out, decentralized ledger via sharding. IEEE Symposium on Security and Privacy. Peer-reviewed.

[112] Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., and Saxena, P. (2016). A secure sharding protocol for open blockchains. ACM CCS. Elastico. Peer-reviewed.

[113] Zamani, M., Movahedi, M., and Raykova, M. (2018). RapidChain: Scaling blockchain via full sharding. ACM CCS. Peer-reviewed.

[114] Ongaro, D., and Ousterhout, J. (2014). In search of an understandable consensus algorithm. USENIX ATC. Raft. Peer-reviewed.

[115] Lamport, L. (1998). The part-time parliament. ACM Transactions on Computer Systems, 16(2). Peer-reviewed.

[116] Park, S., Kwon, A., Fuchsbauer, G., Gaži, P., Alwen, J., and Pietrzak, K. (2018). SpaceMint: A cryptocurrency based on proofs of space. Financial Cryptography. Peer-reviewed. Adjacent resource-accounting result relevant to DAG storage.

[117] Attias, V., and Bramas, Q. (2019). How to choose its parents in the Tangle. NETYS. Peer-reviewed.

[118] Ferraro, P., King, C., and Shorten, R. (2019). Distributed-ledger control models of the Tangle, including stability of unverified transactions in a DAG-based ledger. IEEE control and IoT venues; the same group's peer-reviewed parasite-chain paper is [44].

[119] Staupe, P. (2017). Quasi-analytic parasite-chain attacks in the Tangle. Technical analysis cited by Cullen et al. and the 2023 SoK.

[120] Cai, Z. (2019). A parasite chain attack in IOTA. Technical analysis cited by the 2023 ACM SoK.

[121] Martino, W., Quaintance, M., and Popejoy, S. (2018). Chainweb: A proof-of-work parallel-chain architecture for massive throughput. Kadena technical report. Not peer-reviewed.

[122] Liu, C., Wang, D., and Wu, M. (2018). Vite: A high performance asynchronous decentralized application platform. White paper. Not peer-reviewed.

[123] Eyal, I., Gencer, A. E., Sirer, E. G., and van Renesse, R. (2016). Bitcoin-NG: A scalable blockchain protocol. USENIX NSDI. Peer-reviewed.

[124] Kokoris-Kogias, E., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., and Ford, B. (2016). Enhancing Bitcoin security and performance with strong consistency via collective signing. USENIX Security. ByzCoin. Peer-reviewed.

[125] Garay, J., Kiayias, A., and Leonardos, N. (2017). The Bitcoin backbone protocol with chains of variable difficulty. CRYPTO. Peer-reviewed.

[126] Pass, R., and Shi, E. (2017). The sleepy model of consensus. ASIACRYPT. Peer-reviewed.

[127] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Weed Cocco, S., and Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. EuroSys. Peer-reviewed.

NOTE ON CONTESTED OR SECONDARY DETAILS

Author lists for multi-institution systems papers are given in short form where the venue paper is the citable object. The 2023 Computers throughput comparison is used only as an illustration that advertised and measured attachment rates diverge; it is not itemized as a security result. No claim in the body depends on an unpublished measurement made for this briefing.