Compute Substrate

A Permissionless Computation Layer Without Authority

Mats Heming Julner

2026-03-03

Abstract:
We present Compute Substrate, a permissionless proof-of-work network for producing and persisting speculative computation without semantic, decision or outcome-level authority. Compute Substrate introduces a new class of infrastructure: a public cognition layer that is cryptographically persistent but structurally incapable of exercising power. Participants submit proposals (opaque computational outputs) and attestations (support signals) at a cost. The network aggregates these signals deterministically and commits them to a public ledger. Compute Substrate does not determine correctness, resolve outcomes, or trigger actions. Its sole purpose is to maintain a reproducible, cost-bearing history of what was proposed and supported over time, while remaining safe to ignore.

Introduction

Modern distributed systems increasingly rely on large-scale computation: predictions, rankings, simulations, and analyses generated by many independent actors. In this paper, “computation” refers not only to numerical processing but to any process that produces beliefs, predictions, rankings, or other cognitive artifacts used to guide human or machine action; in this sense, computation is equivalent to distributed cognition. This notion parallels classic discussions of dispersed knowledge and coordination (Hayek, 1945). Here, cognition denotes only the externalization and aggregation of such artifacts. It does not imply awareness, intent, automated reasoning, or any capacity for the system itself to interpret or act.

In practice, the outputs of these computations are increasingly treated as decisions. They trigger actions, allocate resources, and move the world. In many such systems, execution is coupled directly to temporal progression rather than mediated by an explicit authorization layer. Instead, execution is implicitly driven by temporal progression: actions occur because a loop continued, a block was produced, or a scheduler fired. Time becomes the de facto source of authority. In such systems, to be next in sequence is to be allowed. This dynamic echoes classical distributed systems analyses of ordering and time (Lamport, 1978). Compute Substrate explicitly rejects this model. In this system, time may order events, but it can never legitimize action. Sequence records what happened; it does not decide what should happen. When computation is allowed to accumulate

authority, errors become catastrophic, incentives distort behavior, and systems attract adversarial pressure proportional to the consequences of being “right”.

This paper therefore asks a structural question: can computation be scaled independently of authority? That is, can a system persist the results of distributed cognition without implying correctness, execution, or obligation?

Historically, computation has almost always been coupled to consequence. Models guide decisions, predictions trigger actions, and outputs implicitly carry power. To our knowledge, Compute Substrate is the first distributed system that explicitly forbids this coupling at the protocol level. It introduces a new category of infrastructure in which cognition may scale arbitrarily while remaining structurally incapable of producing outcomes.

We present Compute Substrate as a minimal affirmative answer. It is not a ledger of transactions and not a system of execution. It is a cryptographic substrate for recording speculative computation only. Authority, interpretation, and action are explicitly external. The system remains complete even if all recorded outputs are wrong or ignored forever. Informally, Compute Substrate extracts the informational output of intelligence while stripping away its ability to act.

Viewed abstractly, the system forms a public, cryptographically persistent map of distributed cognition over time. Proposals and attestations represent expressed beliefs, hypotheses, predictions, and claims. The ledger records not what is true or what should be done, but what was proposed and supported at a given moment. In this sense, Compute Substrate can be understood as a public cognition layer: a shared, adversarially-written memory of what humanity (and its machines) believed, without granting any of those beliefs the power to act. This shared visibility allows participants to orient their own decisions around common reference points while leaving all authority and execution external to the protocol.

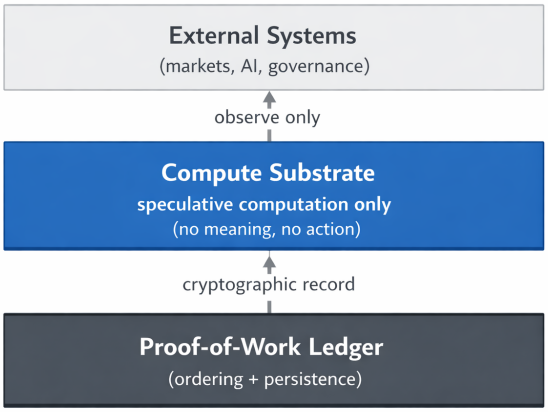

Figure 1. Layer separation in Compute Substrate. (Conceptual diagram)
The substrate records speculative computation without meaning or execution. All authority and action exist strictly outside the protocol.

Design Goals

Compute Substrate is designed around the following goals:

- **Permissionless participation:** anyone may join, propose, attest, or mine.
- **No authority:** the network never decides correctness or resolves outcomes.
- **No execution:** recorded outputs never trigger actions.
- **Cost-bearing computation:** contributing signals requires paying fees.
- **Deterministic aggregation:** all nodes derive identical state from the same history.
- **Verifiable history:** all state is reproducible from genesis.
- **Safe to ignore:** external systems may observe or disregard outputs without consequence.

These goals intentionally exclude many features common to “intelligence” or “governance” systems. The exclusions are essential to safety.

System Overview

Network Model

Compute Substrate is a peer-to-peer network secured by proof-of-work. Nodes perform three roles:

- **Full nodes:** validate blocks, maintain state, and serve data.
- **Miners:** construct blocks and perform proof-of-work.
- **Clients:** submit transactions and query aggregated outputs.

All roles are permissionless.

Blocks and Consensus

Blocks are linked by hash pointers and ordered by cumulative chainwork. The consensus rule is simple: The canonical chain is the valid chain with the highest total chainwork. Proof-of-work provides Sybil resistance and ensures that history is costly to rewrite. Block rewards and transaction fees incentivize inclusion and liveness, not correctness of computation or semantic value of the payloads. In Compute Substrate, proof-of-work secures temporal provenance rather than consensus truth: it makes public memory costly to forge, not authoritative to follow.

Difficulty and Memory Production Rate

Proof-of-work secures temporal ordering by making history costly to rewrite. Difficulty primarily regulates the rate at which this public record grows, and it bounds the cost of reorganizing recent history. Compute Substrate does not attach semantic authority to on-chain artifacts, so manipulation of ordering yields limited payoff beyond inclusion and timing. Difficulty adjustment is therefore treated as a throughput and stability mechanism for the public record, not as a truth mechanism for computation.

Consensus and Authority

In most distributed systems, consensus is used as a decision mechanism: once a network converges on a state, that state carries immediate consequences. Assets are reassigned, actions are executed, or obligations are enforced. In such systems, consensus functions as de facto authority.

Compute Substrate rejects this coupling. In this system, consensus guarantees only reproducible agreement on history, not correctness, validity, or entitlement. The protocol assigns no rights, powers, or obligations to any recorded artifact. Inclusion in the canonical chain confers no

semantic authority and produces no outcomes. By removing all protocol-level consequences from consensus, Compute Substrate ensures that agreement remains purely coordinative. Consensus establishes what was recorded and when, but never what is correct, valuable, or actionable. Authority, interpretation, and execution are strictly external to the protocol.

Compute Substrate is structurally inert: no on-chain artifact can cause side effects beyond its own reproduction in future state. This ensures that even perfect manipulation of the system cannot translate into power, only into additional records. The protocol cannot express commands, obligations, or triggers. All state transitions are internal to the ledger itself and carry no semantic force outside the system. This design enforces a structural separation between coordination and consequence at the protocol layer. Whereas most systems use consensus to decide, Compute Substrate uses consensus only to remember. This removes authority from the protocol layer entirely, such that the system is structurally incapable of deciding outcomes or conferring power.

Relation to Existing Systems

Bitcoin (Nakamoto, 2008) and Ethereum (Buterin, 2014; Wood, 2014) couple consensus to economic consequence: inclusion changes ownership or triggers execution. Prediction markets resolve outcomes and distribute rewards based on correctness (Hanson, 2003). Governance protocols bind consensus results to treasury control or parameter changes, reflecting broader institutional coupling between coordination and control (North, 1990; Schelling, 1960). Oracle systems produce authoritative outputs consumed by smart contracts (Peterson et al., 2019). Incentivized AI networks reward models based on perceived correctness or performance.

Compute Substrate differs structurally from these systems by restricting consensus to reproducible memory only. Aggregation produces no binding consequence, resolution, or enforcement. Coordination is preserved; authority is excluded.

Temporal Non-Authority

In conventional distributed systems, temporal ordering implicitly confers authority: once an event is included in the canonical sequence, it produces consequences. Compute Substrate enforces the opposite invariant. Although events are ordered in time, the protocol is structurally incapable of treating sequence as permission. No on-chain artifact can trigger execution, obligation, or decision. Time is preserved purely as memory, not as control.

Transactions

Transactions follow a UTXO model. Each transaction may include an optional application payload. Two payload types are defined:

- **PROPOSE:** submits a speculative computational output.
- **ATTEST:** expresses support for an existing proposal.

Transactions pay fees. Minimum fees are enforced for PROPOSE and ATTEST payloads to bound spam.

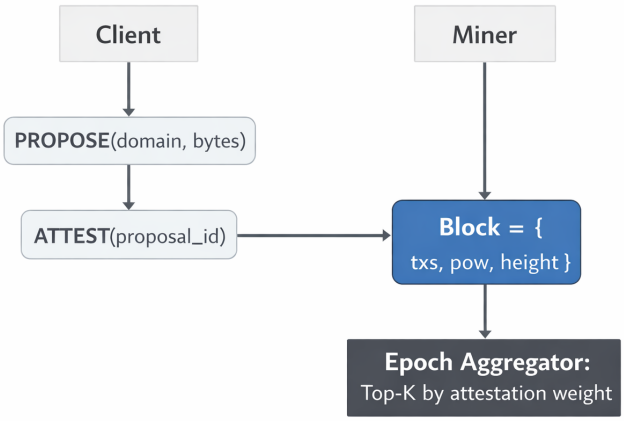

Figure 2. Transaction flow in Compute Substrate. (Conceptual diagram)
Clients submit speculative outputs (PROPOSE) and weight them (ATTEST). Miners include transactions in proof-of-work blocks. Epoch aggregation ranks proposals deterministically without implying correctness or triggering execution.

Proposals

A proposal consists of:
- a domain identifier (e.g. “finance”, “science”)
- an opaque payload (bytes)
- a timestamped inclusion height

The network does not interpret proposal contents. Meaning exists only to observers.

Attestations

An attestation references a specific proposal and contributes weight to it. Attestations are also opaque to the protocol; they simply increase a proposal’s score within an epoch. Attestations contribute weight equal to the transaction fee paid (in base units).

Epochs and Aggregation

Time is divided into fixed-length epochs. Within each epoch, proposals are ranked by the sum of attestation weight they receive. Ties are broken deterministically by proposal identifier (lexicographic ascending). At the end of an epoch, the network derives a deterministic Top-K list for each domain. These lists are part of the canonical state and are queryable by any node. Importantly:
- Rankings do not imply correctness.
- Rankings do not trigger rewards or actions.
- Rankings may change across epochs or reorgs.

Incentives and Fees

Compute Substrate uses incentives only to ensure participation and liveness.
- Block rewards incentivize miners to secure the chain.
- Transaction fees incentivize inclusion and bound spam.

There are no rewards for being correct. Fees are paid regardless of outcome. This prevents computation from becoming authority-bearing.

State, Forks, and Reorganizations

Like any proof-of-work chain, Compute Substrate may experience temporary forks. Nodes track block headers, cumulative chainwork, and maintain undo logs for state transitions.

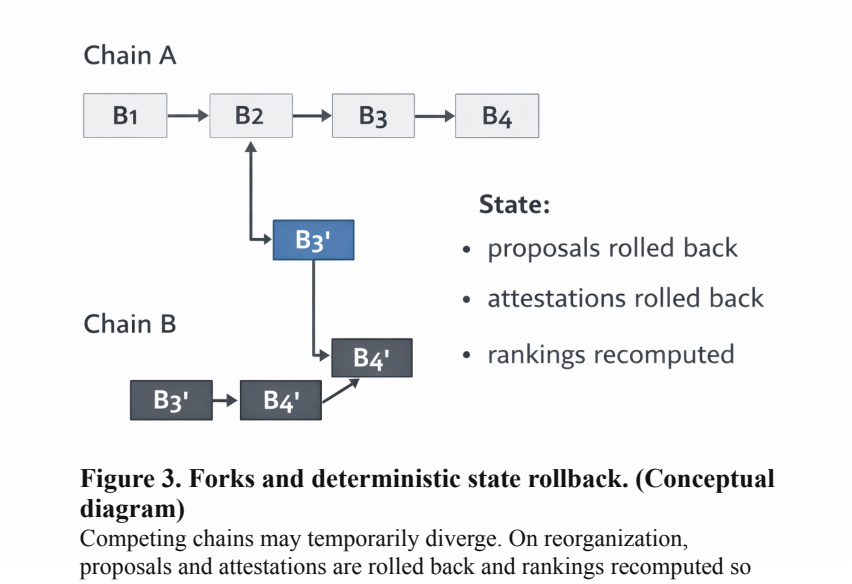

Figure 3. Forks and deterministic state rollback. (Conceptual diagram)
Competing chains may temporarily diverge. On reorganization, proposals and attestations are rolled back and rankings recomputed so that aggregated outputs always correspond to the canonical chain.

On reorganization:
- spent UTXOs are restored
- created UTXOs are removed
- proposal and attestation inserts are rolled back

This ensures that aggregated outputs always correspond to the canonical chain. Although the system does not execute actions, correctness of reorg handling is essential for determinism and verifiability.

Determinism

All state transitions in Compute Substrate are deterministic functions of prior canonical state and the next valid block. Given identical genesis and identical canonical block sequence, all honest nodes derive identical proposal sets, attestation weights, and epoch rankings. The protocol contains no randomness, external inputs, or oracle dependencies. As a result, the network guarantees reproducible agreement on history, though not on correctness or interpretation.

Security Properties

Adversarial Model and Threat Surface

Compute Substrate does not secure assets, execution, or authority. Therefore adversarial strategies that seek to gain economic or control advantage by manipulating consensus yield no meaningful payoff. The only adversarial action available is the creation of additional records at personal cost, which does not degrade protocol determinism or confer influence. Because no record carries authority, even perfect dominance of the ledger yields no control over any external system. Majority

hashpower permits influence over ordering but does not enable seizure, execution, or enforcement. More broadly, because the protocol implements only the persistence of expressed cognition and explicitly excludes execution, enforcement, and decision rights, it does not expose the institutional capture surfaces common to systems that combine aggregation with authority. The network can coordinate shared reference and memory, but cannot be appropriated to govern behavior, allocate resources, or compel outcomes. Adversaries may attempt to spam, censor, or reorder transactions, but such actions affect only inclusion and ordering, not the interpretation, correctness, or authority of recorded computation.

Adversarial Computation

Malicious or incorrect proposals are allowed by design. They are harmless because they carry no authority. Compute Substrate is not designed to be correct; it is designed to be reproducible.

Epistemic Safety

In systems where computational outputs carry authority, participants are rewarded for being persuasive and punished for being wrong. This selects for strategic behavior rather than honest inquiry. Because Compute Substrate assigns no authority, execution, or outcome to any recorded computation, participants may submit speculative, adversarial, or exploratory cognitive artifacts without fear of liquidation, enforcement, or retaliation. All costs are paid up front, and there are no downstream consequences for being incorrect. This creates an epistemically safe environment for distributed cognition. The network does not decide what is true. It preserves what was said.

Spam Resistance

Spam is bounded by transaction fees and proof-of-work costs.

No Oracle Attacks

Because outcomes are never resolved, there is no oracle to corrupt and no incentive to do so.

Miner Behavior

Miners may reorder or exclude transactions, but can only affect inclusion, not interpretation or execution.

Censorship

Many decentralized systems reduce centralized control while retaining semantic or outcome-level authority. Compute Substrate removes these layers entirely, leaving transaction inclusion as the only remaining censorship surface.

Because the protocol assigns no meaning, correctness, or authority to recorded computation, censorship can occur only at the level of transaction inclusion and cannot target interpretation, outcomes, or use. Once recorded, computational artifacts are reproduced deterministically and cannot be selectively removed without rewriting chain history. As in other proof-of-work systems, transaction inclusion may be delayed by individual miners, but no mechanism exists for semantic or retroactive censorship within the protocol.

Non-Goals

Compute Substrate explicitly does not attempt to:
- determine truth or correctness
- resolve real-world outcomes
- execute actions or trigger payments
- govern participants
- provide predictions with guarantees
- replace markets, oracles, or governance systems

These functions belong to higher layers. In particular, Compute Substrate is not designed to make better decisions, only to make it impossible for computation itself to decide.

Applications (External)

Compute Substrate is intended to be observed, not relied upon. External systems may use its outputs as:
- speculative inputs
- advisory signals
- planning data
- exploratory intelligence

Any authority to act on these outputs must exist entirely outside the protocol.

Limitations

Compute Substrate does not guarantee correctness, truthfulness, or epistemic quality of submitted proposals. Participants may behave strategically, and dominant proposals may emerge through social coordination external to the protocol. The system does not prevent off-chain authority formation, reputational capture, or institutional influence over how outputs are interpreted. It also does not eliminate the possibility that external systems voluntarily bind themselves to on-chain rankings, raising familiar concerns in social choice and aggregation theory (Arrow, 1963). Compute Substrate removes authority at the protocol layer; it cannot prevent authority from forming at higher social or institutional layers.

Conclusion

Compute Substrate demonstrates that temporal ordering can be separated from authority in a permissionless distributed system. By restricting consensus to reproducible memory and explicitly excluding execution, enforcement, and outcome resolution, the protocol enables large-scale aggregation of speculative computation without binding consequence. This separation reduces adversarial payoff, narrows institutional capture surfaces, and preserves the possibility of shared reference without shared control. Whether such a cognition layer proves socially useful remains an empirical question; structurally, however, it establishes that coordination need not imply authority.

References

Arrow, K. J. (1963). *Social choice and individual values* (2nd ed.). Yale University Press.

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform*. Ethereum White Paper. https://ethereum.org/en/whitepaper/

Garay, J., Kiayias, A., & Leonardos, N. (2015). *The Bitcoin backbone protocol: Analysis and applications*. In Advances in Cryptology – EUROCRYPT 2015 (pp. 281–310). Springer.

Hanson, R. (2003). *Combinatorial information market design*. Information Systems Frontiers, 5(1), 107–119.

Hayek, F. A. (1945). *The use of knowledge in society*. The American Economic Review, 35(4), 519–530.

Lamport, L. (1978). *Time, clocks, and the ordering of events in a distributed system*. Communications of the ACM, 21(7), 558–565.

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. https://bitcoin.org/bitcoin.pdf

North, D. C. (1990). *Institutions, institutional change and economic performance*. Cambridge University Press.

Peterson, J., Krug, J., Zoltu, M., Williams, A., & Alexander, S. (2019). *Augur: A decentralized oracle and prediction market platform (v2)*. https://www.augur.net

Schelling, T. C. (1960). *The strategy of conflict*. Harvard University Press.

Wood, G. (2014). *Ethereum: A secure decentralised generalised transaction ledger* (Ethereum Yellow Paper).