Ceramic Built Mutable Data Streams for Web3. The Governance of Those Streams Is Still Not Local.

by [Nick Clark](#) | Published March 27, 2026 | [PDF](#)

Ceramic's stream-based data model provides mutable, verifiable data objects on decentralized infrastructure, with ComposeDB adding structured schemas and StreamID providing content-addressable identifiers. This article examines why stream organization, namespace governance, and structural adaptation are not governed locally by stream participants but instead defer to the same pattern of global authority that DNS established. The governance gap follows the same trajectory as every other distributed system that moves data to the edge while leaving namespace authority at the center.

Ceramic addressed a real gap in Web3 infrastructure. Blockchains are good at immutable state transitions but poor at mutable, application-level data. User profiles, social graphs, preferences, documents, and other data objects that change over time do not belong on-chain. They are too large, too frequently updated, and too varied to fit the transaction model that blockchains optimize for.

Ceramic built a protocol for mutable data streams that are verifiable, decentralized, and composable. Each stream has a StreamID that persists across updates. Updates are signed by the stream's controller (typically a DID, a decentralized identifier). The stream's history is a directed acyclic graph of commits, each linking to its predecessor. Commits are anchored to a blockchain for timestamp proofs, providing an ordering guarantee without requiring blockchain consensus for every update.

ComposeDB extended this foundation with structured data models: GraphQL schemas that define the shape of data within streams, enabling queries across streams that share a common model. This is a genuine capability. Developers can define a data model once and query across all streams that conform to it, creating composable data layers that multiple applications can share.

The structural problem is not in the stream model or the data layer. It is in where governance lives.

What the stream model governs and what it does not

A Ceramic stream has clear ownership semantics. The stream's controller — identified by a DID — is the only entity authorized to commit updates. This ownership is enforced cryptographically. No intermediary can modify a stream without the controller's signature. Stream-level governance is real: the controller governs their stream.

What the stream model does not govern is anything above the individual stream level.

Data model governance. ComposeDB data models define the schema that streams conform to. A data model is created by a developer and published to the Ceramic network. Other developers can use the same model, creating interoperable data across applications. But the governance of data models — how a model can evolve, who can propose changes, how breaking changes are handled, how the relationship between models is structured — is not governed by the protocol. A model author publishes a model. If it needs to change, they publish a new version. Consumers discover the change when their queries return different results or fail. There is no mechanism for the participants using a model to collectively govern its evolution.

Namespace organization. Streams are identified by StreamIDs, which are derived from the stream's genesis commit. StreamIDs exist in a flat namespace. There is no hierarchical organization, no scoping mechanism, no way to define that a set of streams constitutes a governed namespace with shared policy. An application's streams coexist in the same global namespace as every other application's streams. Organization is imposed by applications querying for streams that match specific data models, not by namespace structure.

Network governance. Ceramic nodes discover each other and synchronize streams through libp2p. The network topology is determined by which nodes are running, which streams they are interested in, and how libp2p's pubsub and DHT mechanisms route messages. There is no mechanism for a scope of the network to define its own synchronization policy, its own caching rules, or its own governance for how streams within that scope are organized and maintained. Network-level coordination follows the same pattern as any libp2p-based system: global protocols, local execution, no scoped governance.

The anchoring layer is verification, not governance

Ceramic anchors stream commits to a blockchain (typically Ethereum) to provide timestamp proofs and ordering guarantees. The anchoring layer is a verification mechanism: it proves that a commit existed at a certain time and establishes a canonical ordering of commits within a stream.

Anchoring is not governance. The blockchain confirms when a commit happened. It does not validate whether the commit should have happened, whether it conforms to namespace policy, or whether it is consistent with the structural expectations of the scope it belongs to. Anchoring is timestamping. Governance is the layer that decides what mutations are valid within a scope and how the scope itself can evolve.

The anchoring service itself introduces a centralization point. Ceramic's architecture relies on anchor services that batch commits and submit them to the blockchain. The availability and behavior of the anchor service affects the entire network's ability to produce timestamp proofs. This is an operational dependency, and Ceramic has worked to decentralize it. But even a fully decentralized anchoring layer provides verification, not namespace governance.

What local stream governance requires

The gap is between stream-level ownership (which Ceramic solves) and namespace-level governance (which remains unaddressed). An anchor-governed adaptive index provides the missing layer.

In an anchor-governed model, each scope of the namespace is maintained by anchor nodes that hold governance authority for that scope. A set of related streams — all streams conforming to a particular data model within a particular application context, for example — becomes a governed scope. Data model evolution within the scope is proposed by participants and validated through local anchor consensus. Namespace organization is maintained by the scope's anchors: what streams belong to the scope, how they relate to each other, and how the scope's structure can change.

Cross-scope discovery works through alias delegation in the hierarchy rather than through global DHT queries. A query for streams matching a particular model in a particular context is resolved by the anchors governing that scope, not by scanning the global namespace. The scope's anchors know what belongs to the scope because they govern it.

Ceramic's stream model and the adaptive index's governance layer are complementary. Streams continue to provide mutable, verifiable data objects controlled by their owners. ComposeDB continues to provide structured schemas for composable data. What changes is the organizational layer: how streams are grouped, how data models evolve, how cross-application discovery works, and who holds authority over namespace structure. That authority distributes into the namespace itself, governed locally by the anchors responsible for each scope.

Adaptive Indexing All 21 steps →

Resolution without global consensus. Anchor-governed self-organization.

Secondary Technical

○ Anchor-Governed Hierarchical Nesting: Recursive Semantic Containers at Unlimited Depth○ Entropy-Triggered Index Splitting: Deterministic Partitioning Under Mutation Load○ Dormant Index Merging: Recursive Consolidation of Low-Entropy Subindices○ Elastic Anchor Group Management: Governance That Scales With Criticality○ Trust-Weighted Quorum Voting: Consensus Where Weight Reflects Earned Trust○ Asynchronous Consensus Coordination: Offline Vote Completion With Reconciliation○ Best-Match Alias Querying: Longest-Match Resolution With Stepwise Delegation○ Action-Typed Aliases: Behavioral Intent Embedded in the Namespace○ UID Persistence Through Alias Mutation: Stable Identity Across Structural Change○ Lineage-Preserving Structural Mutation: Cryptographic History Through Every Change○ Proximity-Based Routing With Trust Scoring: Dynamic Path Selection in Decentralized Networks○ Dynamic Device Hash for Pseudonymous Authentication: Volatile Identity Without Stored Credentials○ On-Demand Adaptive Caching: Cache Instances That Follow Usage, Not Configuration○ Predictive Cache Prefetching: Forecasting Models That Proactively Instantiate Caches○ Contextual Access Enforcement: Policy Graphs Evaluated With Real-Time Telemetry○ Mutation Router With Contextual Signals: Policy-Aware Propagation Path Selection○ Impact Simulation During Mutation Staging: Pre-Execution Analysis of Proposed Changes○ DNS Bidirectional Fallback: Hybrid Resolution With Legacy DNS Compatibility○ Asset Versioning as First-Class Metadata: Version Entries Under UIDs With Lineage Tracking○ Telemetry-Driven Topology Mutation: Autonomous Network Reconfiguration From Operational Data

Applications (General)

○ Applying Adaptive Indexes to Legacy Decentralized Systems○ Why Edge Platforms Still Depend on a Central Authority○ Supply Chain Tracking Through Governed Namespace Resolution○ Social Media Platforms Without Central Namespace Authority○ Healthcare Data Federation Through Scoped Governance○ Government Identity Infrastructure at Scale○ Financial Market Data With Governed Resolution○ Gaming and Metaverse Namespace Governance

Applications (Specific)

○ Cloudflare's Edge Has a Namespace Problem○ DNS Is 40 Years Old and Still Running the Internet○ ENS Solved the Wrong Half of the Naming Problem○ Handshake Decentralized the Root. Everything Below It Is Still Ungoverned.○ IPFS Solved Content Addressing. It Didn't Solve Naming, Persistence, or Governance.○ Fastly Built the Fastest Cache Invalidation in the Industry. The Authority to Invalidate Still Lives in One Place.○ Akamai Built the Internet's Delivery Infrastructure. It Was Designed for a World That Needed Central Control.○ Bluesky Identified the Right Problem. The Architecture That Solves It Is the Adaptive Index.○ Consul's Service Catalog Is Brilliant Infrastructure. It Is Still a Central Registry.○ Istio Solved Programmable Traffic Policy. The Namespace That Routes Traffic Is Still Central.○ Unstoppable Domains Proved NFT Ownership Works. The Namespace Governance Model Is Still Unresolved.○ The Graph Built the Index Layer for Web3. The Index Itself Still Has a Governance Problem.○ Filecoin Proved Verifiable Storage. Discovery and Namespace Governance Are Still Unsolved.○ Arweave Made Data Permanent. It Has No Governance Model for What Permanent Data Means Over Time.● Ceramic Built Mutable Data Streams for Web3. The Governance of Those Streams Is Still Not Local.○ Kubernetes Service Discovery Resolves Within Clusters. Cross-Cluster Namespace Is Central.○ Amazon Route 53 Is the Most Reliable DNS on Earth. It Is Still DNS Architecture.○ HashiCorp Nomad Distributes Scheduling. The Namespace That Organizes It Is Still Central.○ ZooKeeper Coordinates Distributed Systems. The Coordinator Is a Single Point of Authority.○ etcd Stores the State of Kubernetes. The State Store Has No Scoped Governance.○ Consul KV Distributes Configuration. The Distribution Authority Is Still Central.○ Raft Made Consensus Understandable. It Did Not Make Consensus Scope-Aware.○ Paxos Proved Consensus Is Possible. It Did Not Address Namespace Governance.○ Cosmos Tendermint Enabled Sovereign Blockchains. The Namespace Between Them Is Ungoverned.○ AWS Cloud Map Discovers Services. The Discovery Authority Lives in One Region's Control Plane.○ Azure Traffic Manager Routes Globally. The Routing Authority Is Centrally Defined.○ GCP Service Directory Centralizes Service Registration. Registration Is Not Governance.○ Netlify DNS Simplifies Deployment Routing. The Namespace Authority Is Still Netlify's.○ Vercel's Edge Network Executes at the Boundary. Routing Authority Does Not.○ Bunny CDN Delivers Content Globally. Cache Governance Is Still Central.○ KeyCDN Optimized Content Delivery. The Delivery Namespace Is Centrally Controlled.○ Limelight Networks Built Private Infrastructure for Delivery. The Namespace Governance Is Still Central.○ StackPath Combined CDN With Edge Computing. Namespace Authority Remained Central.○ Envoy Proxy Made Service Mesh Data Planes Programmable. The Control Plane Still Governs.○ NGINX Powers the Web's Reverse Proxy Layer. Its Configuration Is Statically Defined.○ Traefik Discovers Services Automatically. The Discovery Namespace Is Still External.○ Linkerd Simplified the Service Mesh. The Namespace It Meshes Is Still Kubernetes.○ Namecheap Made Domain Registration Accessible. Domain Governance Remains the Registrar Model.○ GoDaddy Registered More Domains Than Anyone. The Namespace Model Has Not Changed.○ DNSimple Made DNS Management Developer-Friendly. The Governance Model Is Still DNS.○ Datadog Observes Everything. The Namespace It Observes Has No Governed Structure.○ Grafana Unified Observability Visualization. The Data Namespace It Queries Has No Governed Structure.○ Prometheus Defined Cloud-Native Monitoring. Its Metric Namespace Has No Governance Layer.○ New Relic Pioneered APM. The Telemetry Namespace It Built Is Centrally Indexed.○ Splunk Indexes Machine Data at Scale. The Index Namespace Is Centrally Administered.

Adaptive Indexing overview →

AQ
deterministic
autonomy

Legal

Last updated: 2026-03-03

-
- nick@qu3ry.net
- 72 28 14 36 01

[Invented by Nick Clark](#) | Founding Investors: Devin Wilkie