IPFS Solved Content Addressing. It Didn't Solve Naming, Persistence, or Governance.

by Nick Clark | Published March 27, 2026 | PDF

IPFS introduced content addressing as a genuine alternative to location-based retrieval, but left three structural problems unsolved. Persistence re-centralized around pinning services, naming re-centralized around gateways, and namespace governance was never addressed. All three gaps trace to a single architectural absence: the lack of scoped, anchor-governed namespace coordination for local content governance.

IPFS made a precise and correct observation about the web: content should be identified by what it is, not where it is stored. A file addressed by its cryptographic hash is verifiable anywhere, retrievable from any peer that holds it, and immune to link rot caused by servers going offline or URLs changing. Content addressing is a genuinely better model than location addressing for a distributed system.

That insight produced a protocol that works. IPFS is real infrastructure. Developers build on it. Filecoin extends it with storage incentives. ENS and other naming systems reference it. The content addressing model it introduced has influenced how the web thinks about data identity.

The problem is that content addressing solves one problem and defers three others: persistence, naming, and governance. Those three problems all have the same structural root, and IPFS's current solutions to them all introduce centralization through the back door.

The persistence problem

In IPFS, content exists on the network as long as at least one node is pinning it. Pinning means a node has explicitly committed to keeping a piece of content available rather than allowing it to be garbage collected. If no node is pinning a CID, that content becomes unreachable.

This is not a bug. It is a deliberate design choice: IPFS does not force nodes to store content they did not choose to store. The consequence is that content persistence requires an active decision by someone to maintain it.

In practice, most developers who need reliable persistence delegate that decision to a centralized pinning service: Pinata, Infura, Filebase, and others. These services maintain the infrastructure that keeps content available. They are the de facto persistence layer for most IPFS deployments. The content is addressed in a decentralized way. The guarantee that it remains available is provided by a centralized service.

The naming problem

Content addressing by hash is immutable by design. Change a file and its CID changes. This is the property that makes content addressing trustworthy. It is also the property that makes it unusable for anything that needs to be updated.

IPNS (InterPlanetary Name System) addresses this by providing mutable pointers: an IPNS name is a hash of a public key, and the owner of the corresponding private key can update what the name points to. The name stays stable while the content it references changes.

The structural problems with IPNS are specific. IPNS records published to the DHT expire after 48 hours and must be actively republished. IPFS's own specification confirms this: records must have an expiration time set within 48 hours after publication. A name that is not actively maintained disappears from the network.

Resolving an IPNS name via DHT requires collecting responses from a quorum of 16 out of 20 DHT servers before returning a result. This is slow in practice, which is why IPFS's own documentation notes that DNSLink is "currently much faster than IPNS." The workaround most developers reach for is an IPFS gateway. IPFS's documentation is explicit about the tradeoff: "IPNS resolution via an IPFS gateway is trusted, which means you delegate IPNS resolution to the gateway without any means to verify the authenticity of the response."

The naming layer has re-centralized around gateways and pinning services for the same reason the persistence layer did: the protocol provides the mechanism but not the infrastructure to operate it reliably at scale.

The governance problem

Neither content addressing nor IPNS provides any mechanism for a segment of the namespace to define its own mutation policy, coordinate cache state across the nodes responsible for it, or make governed structural decisions about how the namespace evolves.

An IPNS name is owned by whoever holds the private key. That person can update what the name points to. What they cannot do is define who else can propose updates, what conditions those proposals must meet, how updates should be validated across multiple nodes, or how the history of updates should be preserved and traversable. There is no concept of scoped governance in IPFS's naming layer. There is ownership, and there is the DHT.

The cache coordination problem is similar. Content is cached at nodes that have recently accessed it and garbage collected when it has not been accessed recently or when the node needs space. There is no mechanism for the nodes responsible for a given namespace scope to coordinate on what should be cached, when cache state should be invalidated, or how cache policy should be defined and enforced locally. Those decisions happen through individual node behavior and centralized pinning, not through governed coordination among the nodes responsible for a given scope.

What the three problems share

Persistence, naming, and governance are three symptoms of the same structural gap: IPFS has no mechanism for a scope of the namespace to be governed by the nodes responsible for it.

Content addressing works without scoped governance because it is stateless: a CID is a hash, verification is cryptographic, no coordination required. The moment you need mutability (naming), persistence (cache coordination), or policy (access control and mutation governance), you need a coordination mechanism. IPFS defers that coordination to the DHT for naming, to pinning services for persistence, and to nothing in particular for governance. The DHT and the pinning services both re-introduce centralization because they are the only available coordination mechanisms.

Scoped governance means: the nodes responsible for a given segment of the namespace hold the policy for that segment, mutations are proposed and validated through local consensus among those nodes, cache state is coordinated within the scope, and the history of changes is preserved in a lineage that remains traversable. This is what the adaptive index provides: a hierarchical namespace where each scope is anchor-governed, resolution traverses the hierarchy stepwise through local anchor groups, and structural changes preserve lineage continuity.

IPFS's content addressing model and the adaptive index's governance model are not in competition. Content addressing identifies what a resource is. Anchor-governed scopes determine how that resource can be named, found, updated, and persisted without delegating those decisions to centralized infrastructure.

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

Patent
[US 19/326,036](#) · published
Primary Technical Disclosure
[○ The Adaptive Index: A Scalable Foundation for Decentralized Systems](#)
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