



[Home](#) [Licensing](#) [Patents](#) [Articles](#)

## **On-Demand Adaptive Caching: Cache Instances That Follow Usage, Not Configuration**

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

Caches in the adaptive index are not statically provisioned. They are instantiated when usage metrics indicate demand and expired when demand subsides. Each cache instance is scoped to the index segment it serves and governed by the anchors responsible for that segment. This produces a caching layer that tracks actual access patterns in real time rather than requiring administrators to predict and provision cache topology in advance.

---

### **What It Is**

On-demand adaptive caching creates cache instances dynamically in response to measured query load within specific index scopes. When a scope begins receiving resolution requests above a configured threshold, the governing anchors instantiate a cache that serves subsequent queries from locally stored state. When query volume drops below the expiration threshold, the cache is dissolved and its resources are reclaimed.

The cache scope matches the governance scope: a cache can only serve data from the index segment that authorized its creation, and it is invalidated when that segment mutates. This ensures cache consistency is structurally enforced rather than externally managed.

## Why It Matters

Static cache provisioning requires predicting access patterns, pre-allocating resources, and configuring invalidation rules manually. Over-provisioning wastes resources. Under-provisioning creates hot spots. Stale caches serve incorrect data. Each of these failures is a consequence of configuring caches at design time rather than responding to demand at runtime.

Adaptive caching removes the prediction requirement. Caches appear where and when they are needed and disappear when they are not. The system is never over-provisioned or under-provisioned for caching because provisioning is continuous and automatic.

## How It Works Structurally

Each index scope maintains query rate metrics. When the query rate exceeds the instantiation threshold defined in the scope's caching policy, the governing anchors authorize cache creation. The cache is populated from the scope's current state and begins serving resolution requests.

Cache invalidation is event-driven: when the governing anchors commit a mutation to the scope's data, all caches serving that scope are invalidated. The invalidation is scoped to the specific data that changed, allowing the rest of the cache to continue serving.

When query rate falls below the expiration threshold for a sustained period, the cache is dissolved. The dissolution is governed: anchors verify that no pending queries depend on the cache before reclaiming its resources.

## What It Enables

Adaptive caching enables the index to handle burst traffic without pre-provisioned infrastructure. Flash crowds, viral content, and sudden demand spikes trigger automatic cache creation at the affected scopes. When demand normalizes, the caching overhead disappears. The system maintains consistent performance under variable load without manual scaling intervention.

This is particularly valuable for edge deployments where infrastructure resources are constrained: caches consume resources only when the access pattern justifies them, and release resources immediately when demand subsides.

[Adaptive Indexing All 21 steps →](#)

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

Subject to one or more pending U.S. and international patent applications, see [Patents](#) for the current list and status. No license, express or implied, is granted. Any use requires a separate written agreement—see [Licensing](#). Patent applications referenced on this site are pending. Claim scope, if any, is subject to examination and may issue in altered form or not at all. See [Legal](#) for terms and conditions.

Adaptive Query™ is a trademark of Nicholas Clark. U.S. federal registration is pending. federal registration. AQ™, AQ Inside™, Adaptive Index™, Adaptive Network™, Semantic Agent™, @AQ™, AQID™, and Adaptive Coin™ are used as trademarks in connection with the Adaptive Query platform and brand. Other names may be trademarks of their respective owners.

Platform operated by Adaptive Query LLC, which provides patent and trademark licensing services. Copyright © 2025-2026 Nicholas Clark. All rights reserved.

Last updated: 2026-03-03



- [Inventive Steps](#)
- [Licensing](#)
- [Patents](#)
- [Articles](#)
- [Legal](#)
- [Opportunities](#)
- [Sitemap](#)



- 
- [nick@qu3ry.net](mailto:nick@qu3ry.net)
- 72 28 14 36 01



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