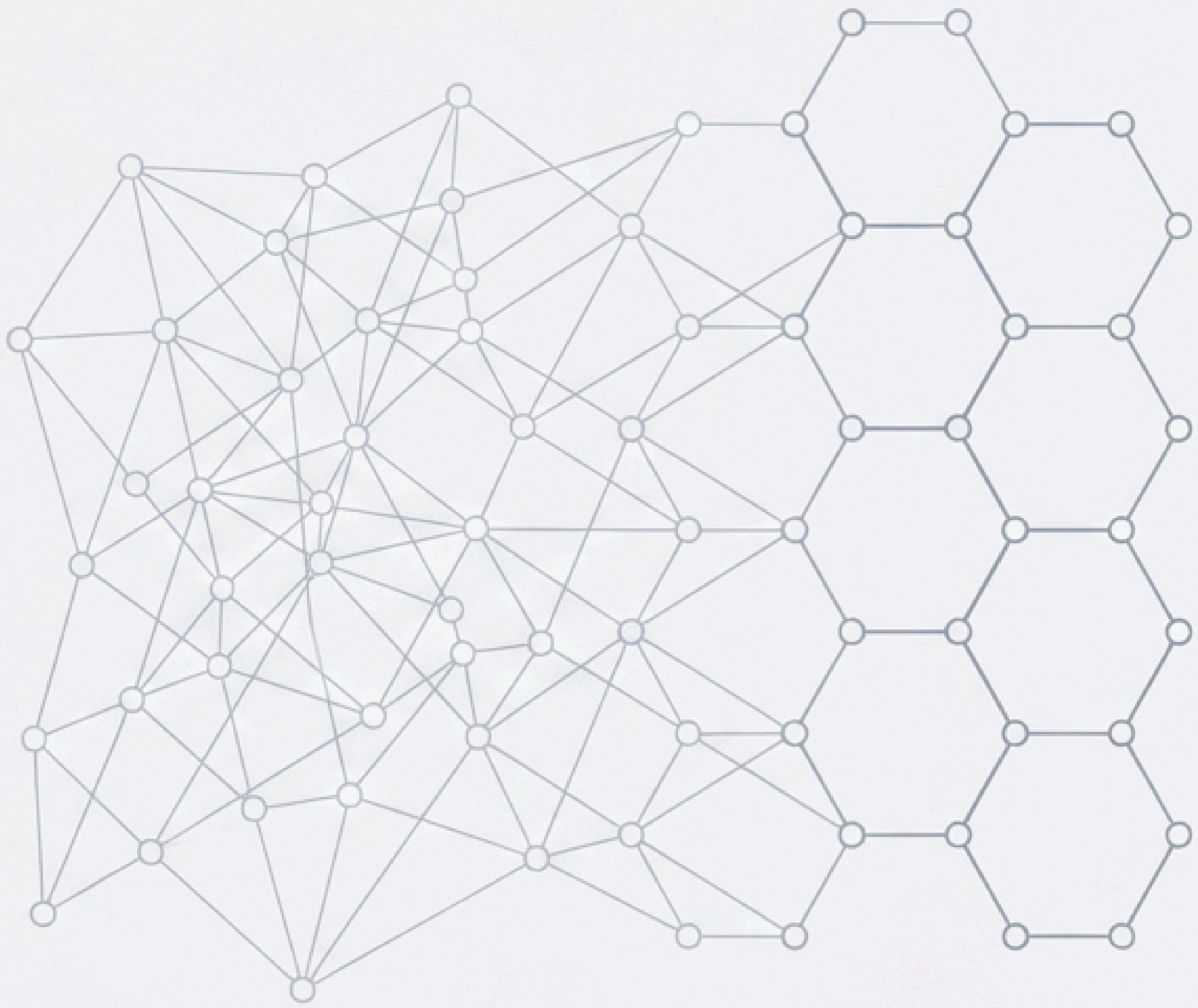
Big Data Analytics Project:

Design and Benchmarking of Distributed P2P Architectures

From Unstructured Flooding to Hybrid Structured Systems

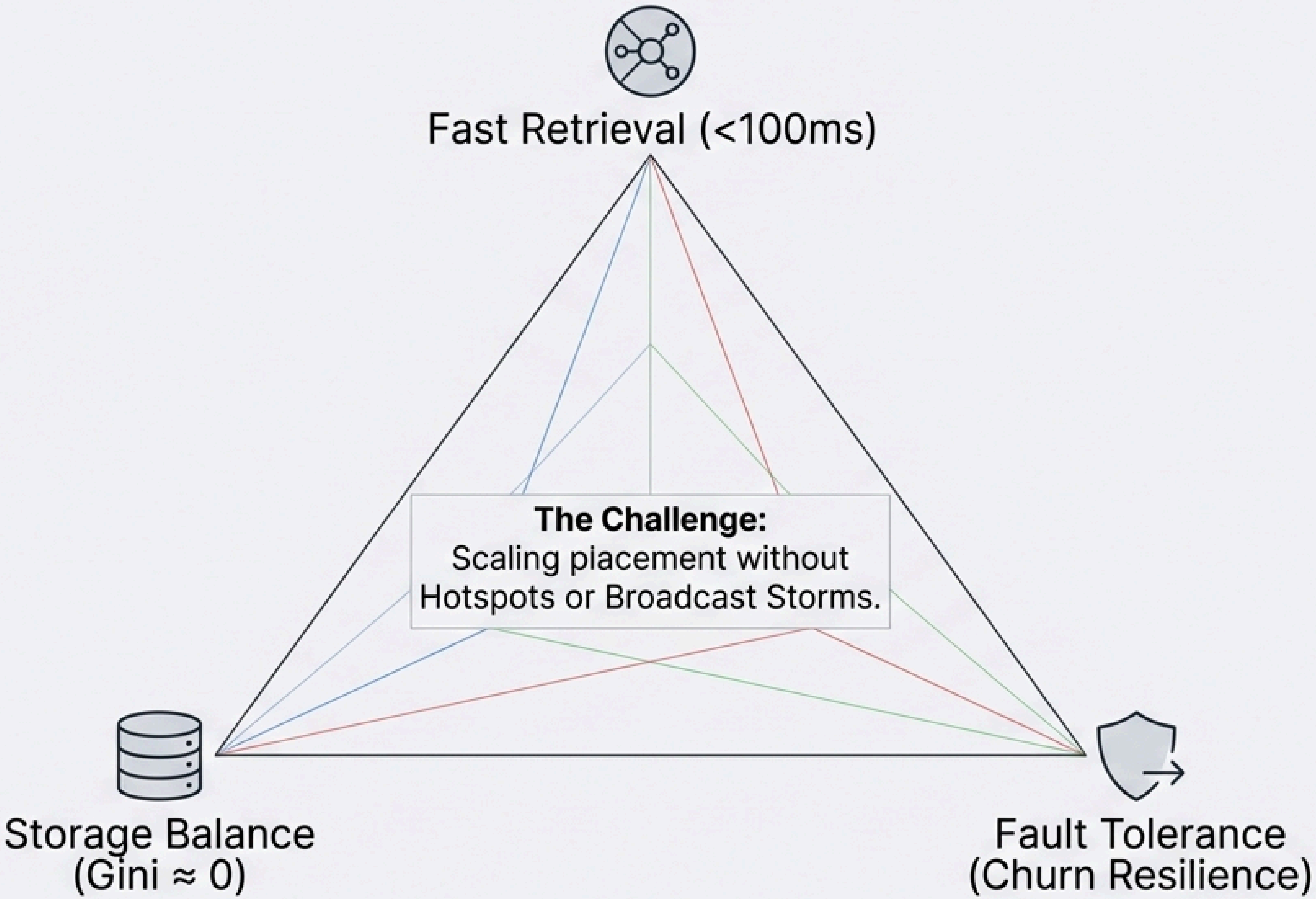
Candidate: Di Franco Federico
Date: January 20, 2026



Project Definition: A rigorous performance evaluation of three peer-to-peer strategies—Naive, Semantic, and Metadata Sharding—validated through local simulations and distributed Google Cloud deployments.

The Distributed Storage Trilemma

Balancing competing engineering goals in P2P overlays.



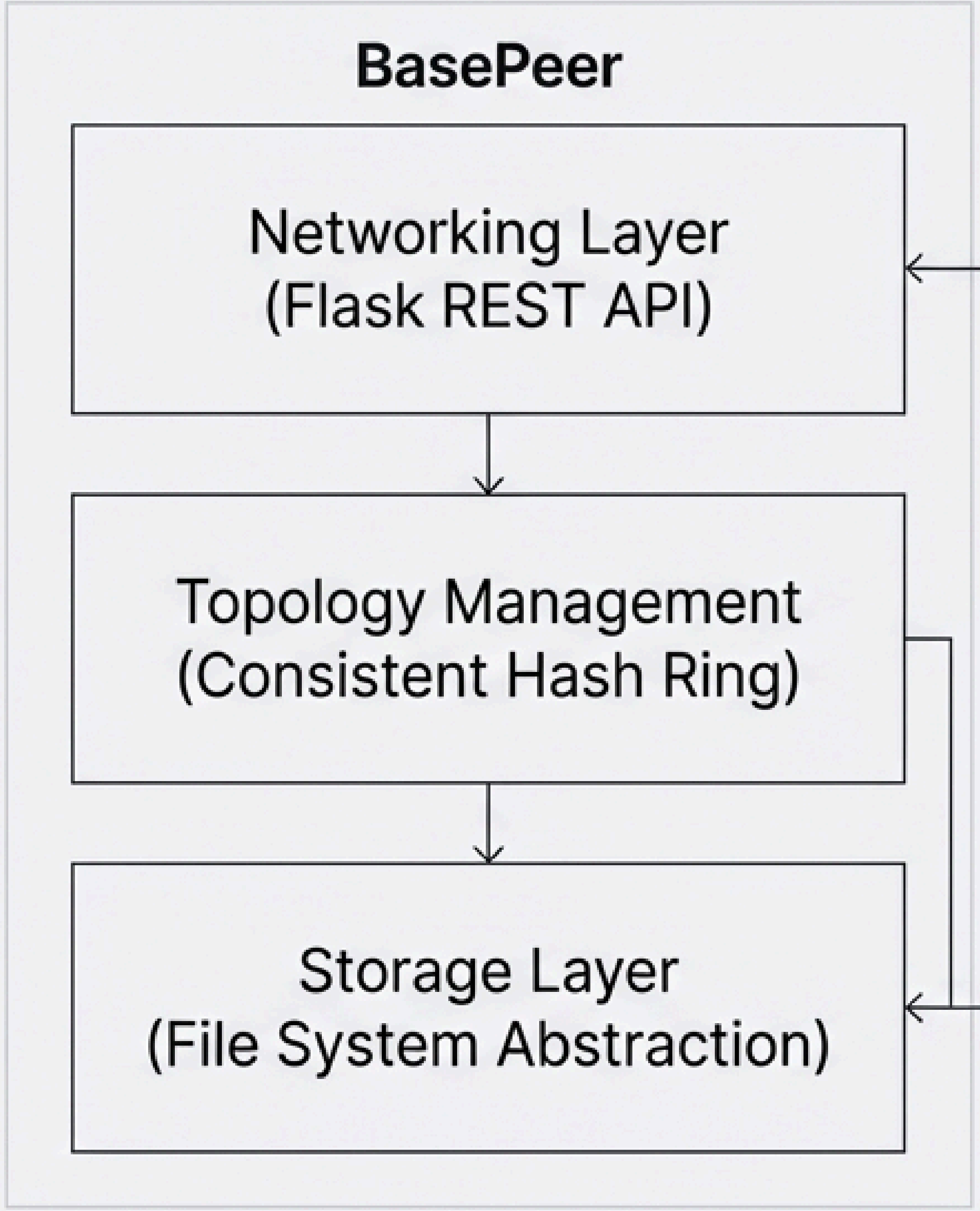
Context:

Evolution from unstructured Gnutella-style networks to structured hybrid overlays.

Approach:

Comparative analysis of three architectural strategies on shared Python/Docker infrastructure.

System Architecture: The Shared Foundation



Consistent Hashing

Algorithm: MD5 (128-bit integer space)
Distribution: k=100 Virtual Nodes per physical peer to mitigate skew.

Storage Model

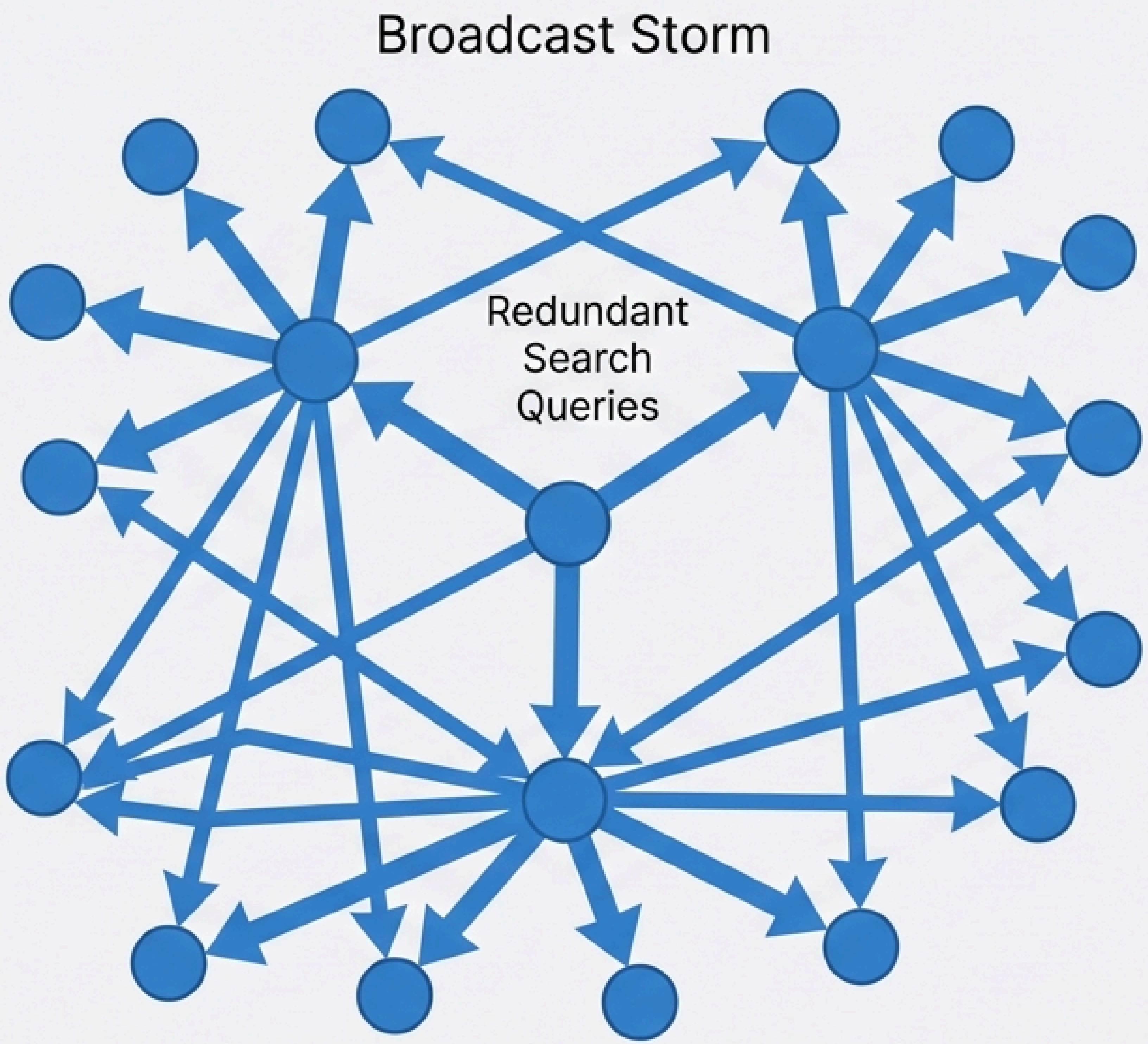
Strategy: Split-and-Merge (BitTorrent-style)
Chunking: 1MB fixed-size blocks hashed via SHA-1.

Availability

Protocol: Primary-Push Anti-Entropy
Convergence: Gossip protocol for topology updates.

Strategy 1: The Naive Approach (Flooding)

BASELINE



Mechanism

- **Data Placement**: Pure Consistent Hashing
- **Search Logic**: Recursive flooding of neighbors.

Critical Distinction

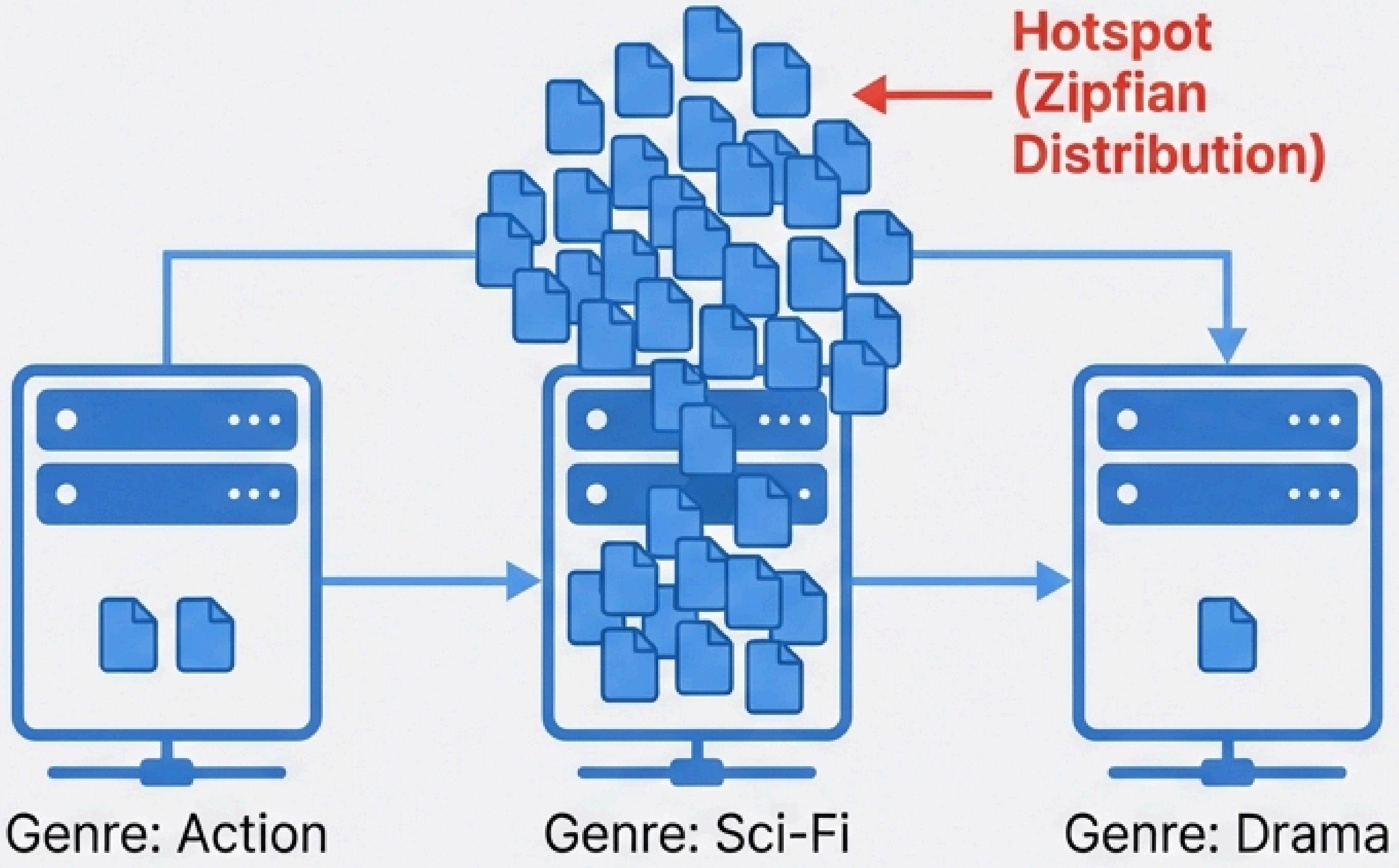
- **Search Complexity:** `O(N)` - High Network Cost.
- **Download Complexity:** `O(1)` - Direct DHT fetch.

Hypothesis

Robust against churn, but unscalable due to message volume.

Strategy 2: Semantic Partitioning (Data Locality)

OPTIMIZED FOR QUERY



Mechanism

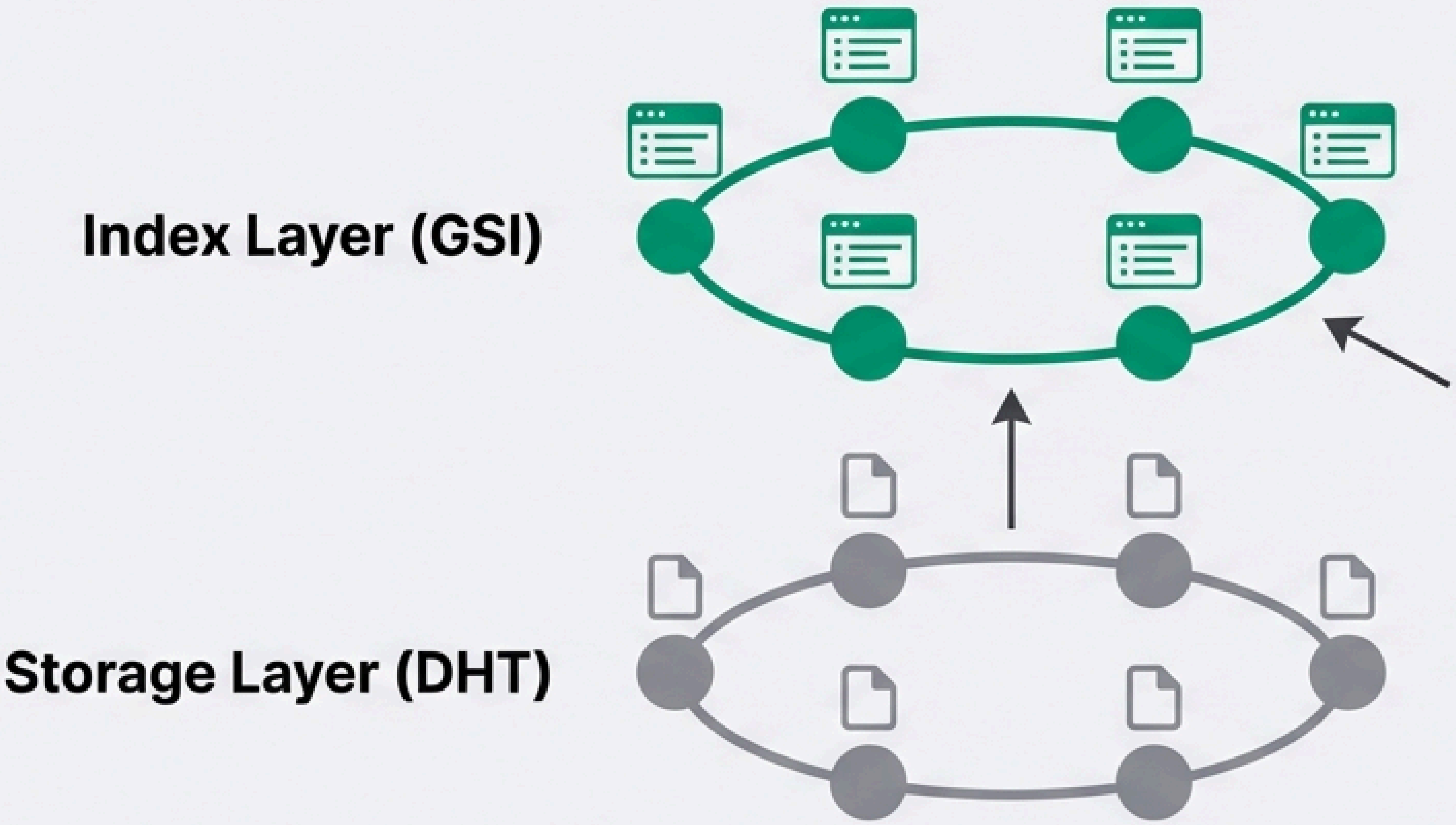
- **Placement:** Overrides DHT. Forces data co-location based on attributes.
- **Routing:** O(1) directed routing for known keys.

The Trade-off

- **Benefit:** Zero-latency lookup for specific attributes.
- **The Flaw:** Storage Hotspots overload specific nodes.
- **Fallback:** Partial queries revert to O(N) Broadcast.

Strategy 3: Metadata Sharding (GSI & Salting)

THE HYBRID SOLUTION



Salting Logic

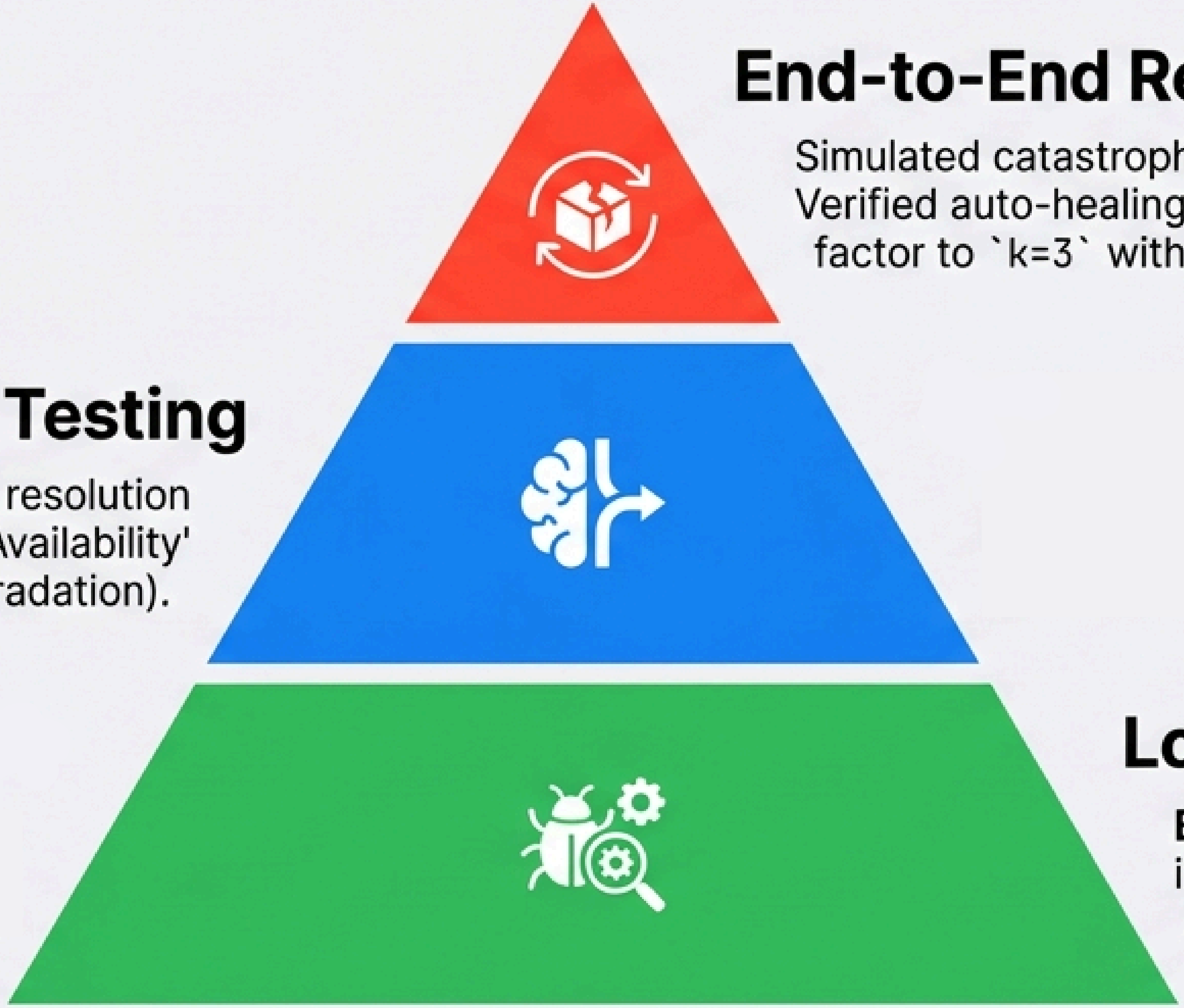
```
ShardID = random(0, INDEX_SHARDS - 1)
```

Prevents write hotspots for popular keys (e.g., 'Brad Pitt').

Search Mechanism: Scatter-Gather

- Parallel queries dispatch to fixed shard locations.
- **Complexity: O(S × A)** (Shards × Attributes).
- **Result:** Decouples search latency from cluster size N.

Verification: The Testing Pyramid



End-to-End Resilience

Simulated catastrophic failure (`docker stop`). Verified auto-healing restored replication factor to `k=3` within 40s.

Unit Testing

Validated 'Split Brain' resolution (Last-Write-Wins) and 'Partial Availability' (Graceful degradation).

Logic Simulation

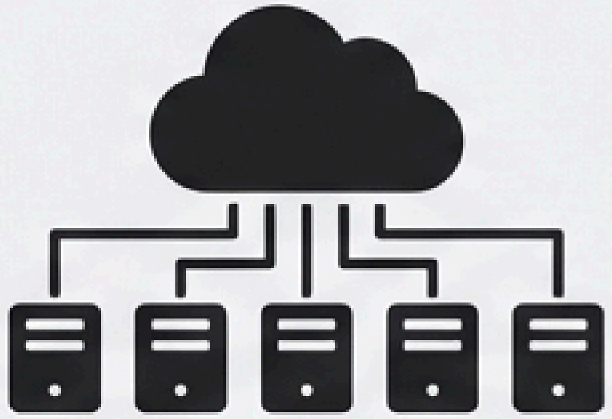
Bug Discovery: Identified critical flaw in Semantic repair logic where repair threads checked wrong nodes due to attribute hashing mismatch.

Benchmarking Setup: Local vs. Cloud

Local Baseline (Workstation)

CPU: AMD Ryzen 7 5700U
Network: Docker Bridge (<0.1ms Latency)
Constraint: CPU Bound

Google Cloud Cluster

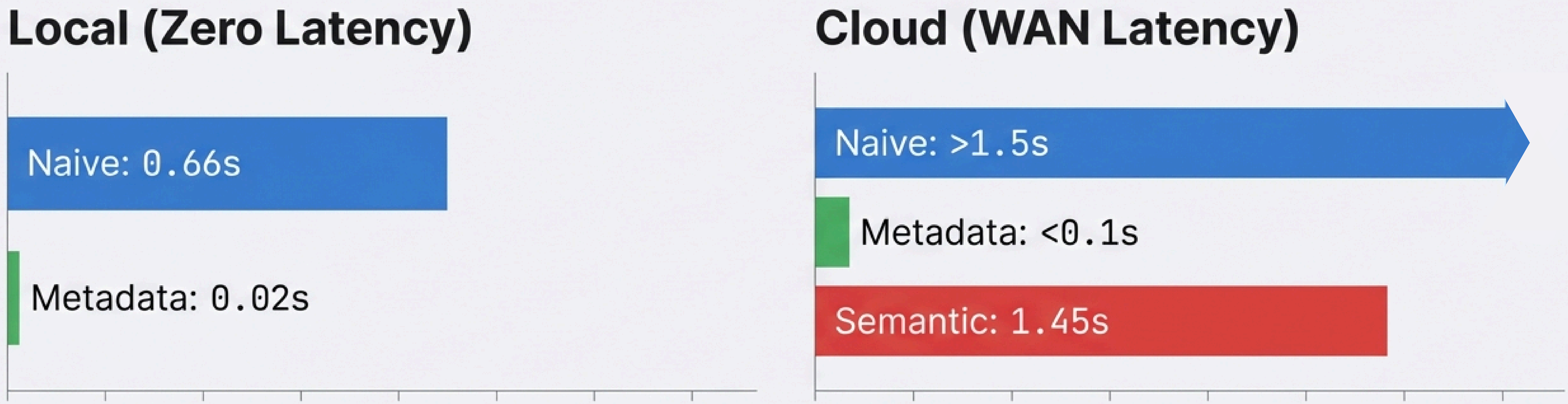


Scale: 5 VMs (e2-medium), 30 Nodes
Network: Real WAN Latency
Constraint: I/O Bound

Workload Generation

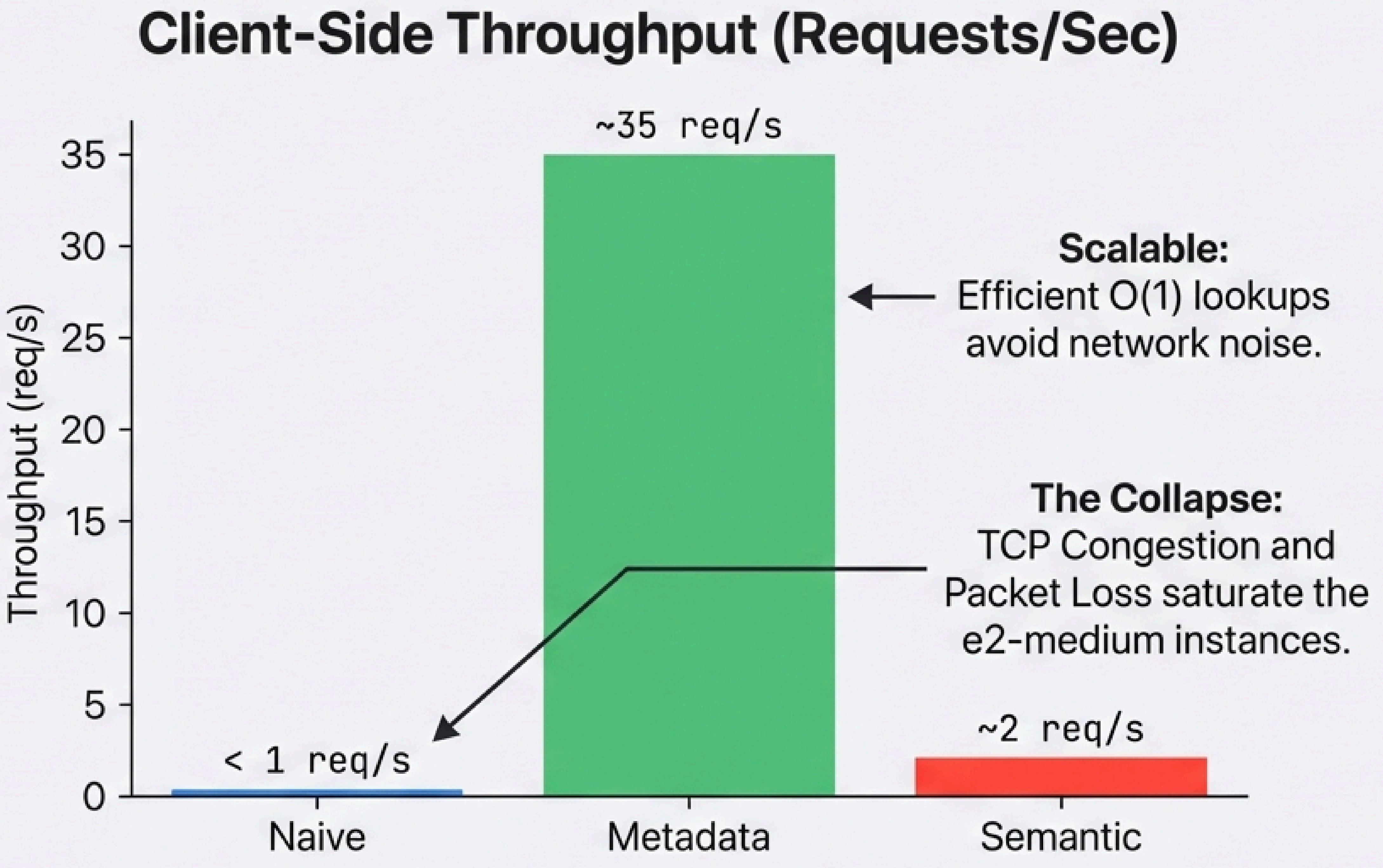
Distribution: Zipfian (Power Law). 80% traffic to popular items.
Volume: 5,000 synthetic binary files. 5,000 automated queries.

Latency Analysis: The Network Amplification Effect



Naive strategy degrades by ~2.5x in Cloud due to broadcast message volume hitting physical network limits.

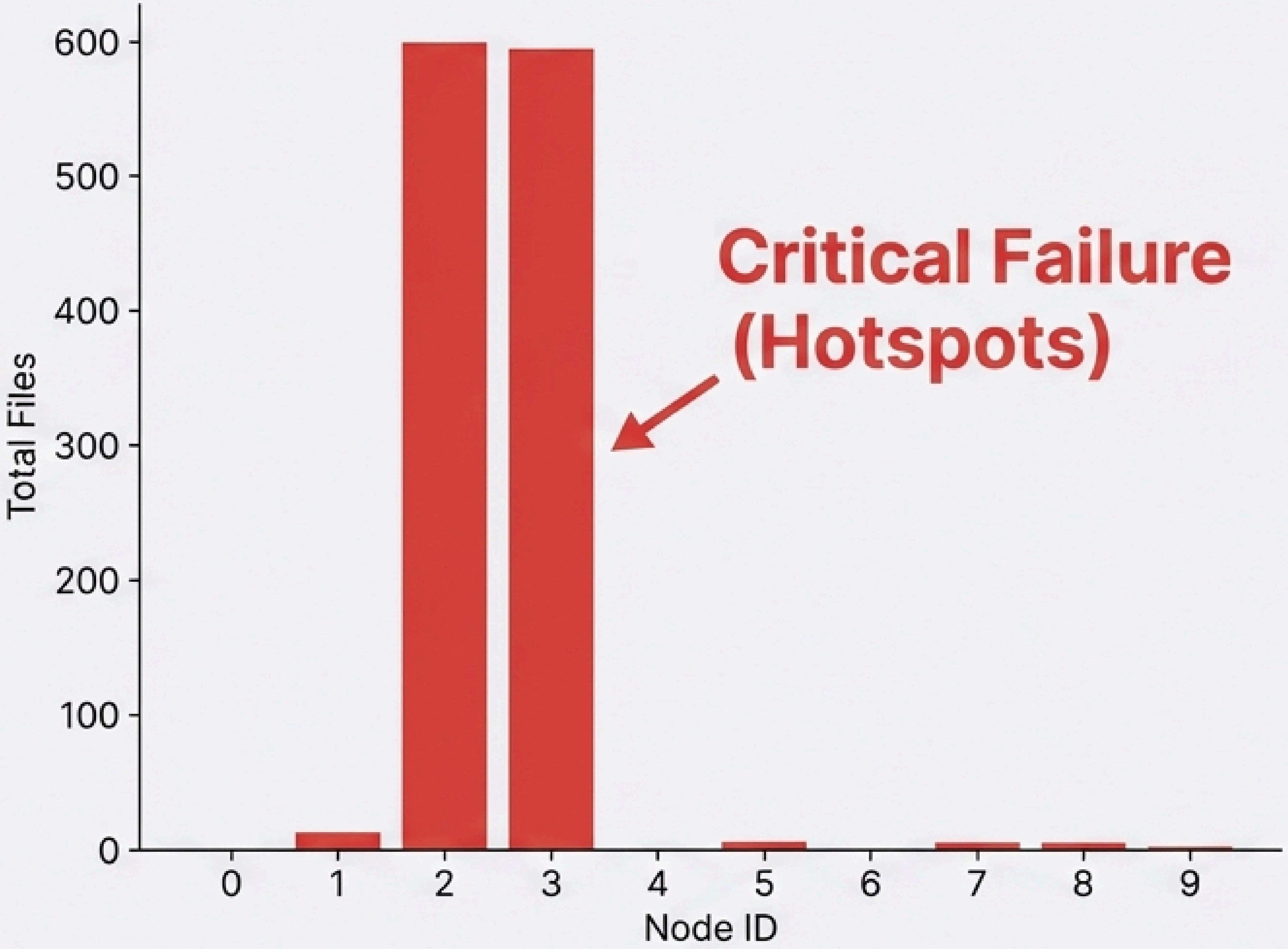
Throughput & Saturation: The Cloud Collapse



Storage Load Balancing: Algorithmic Integrity

Files per Node

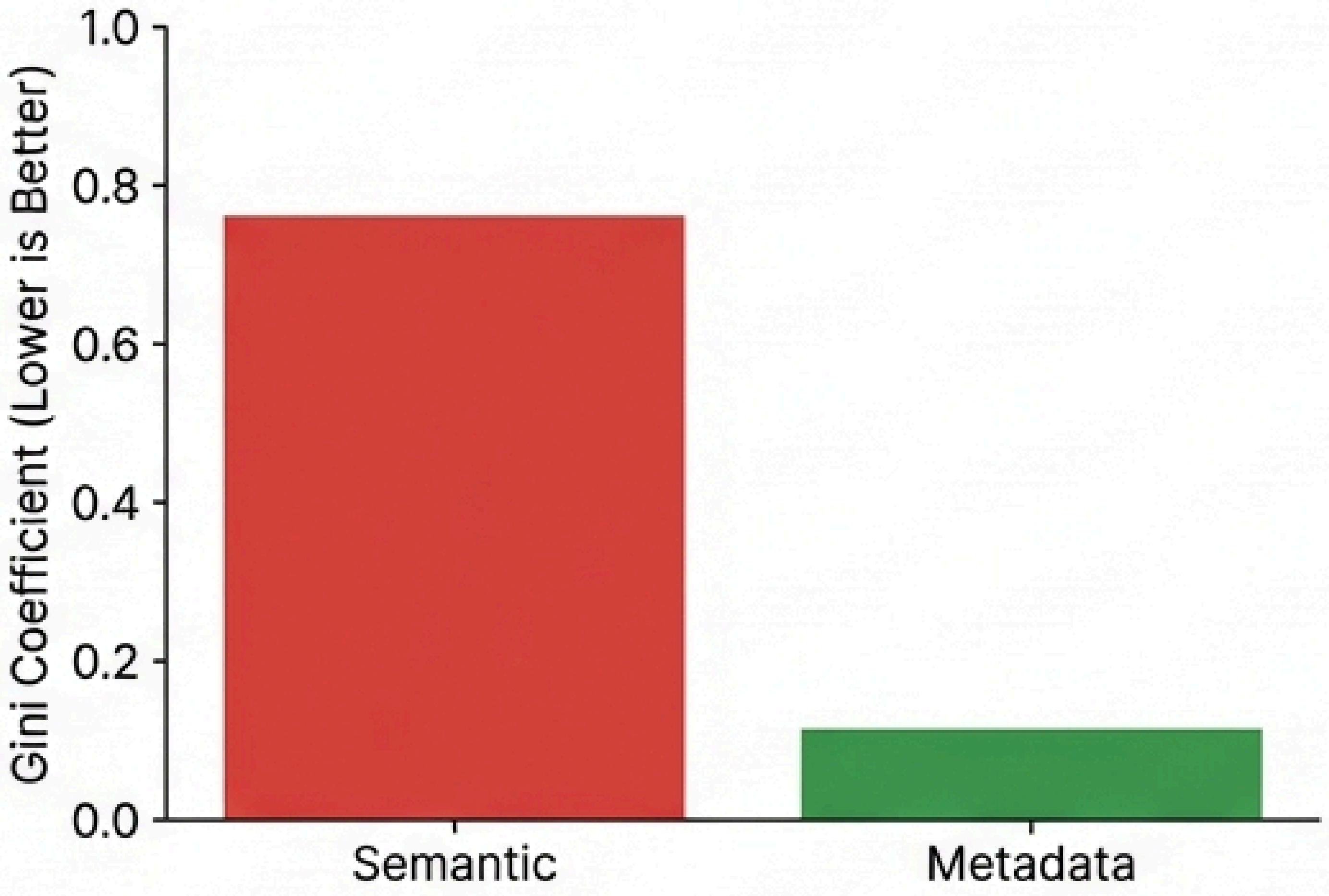
Semantic Strategy



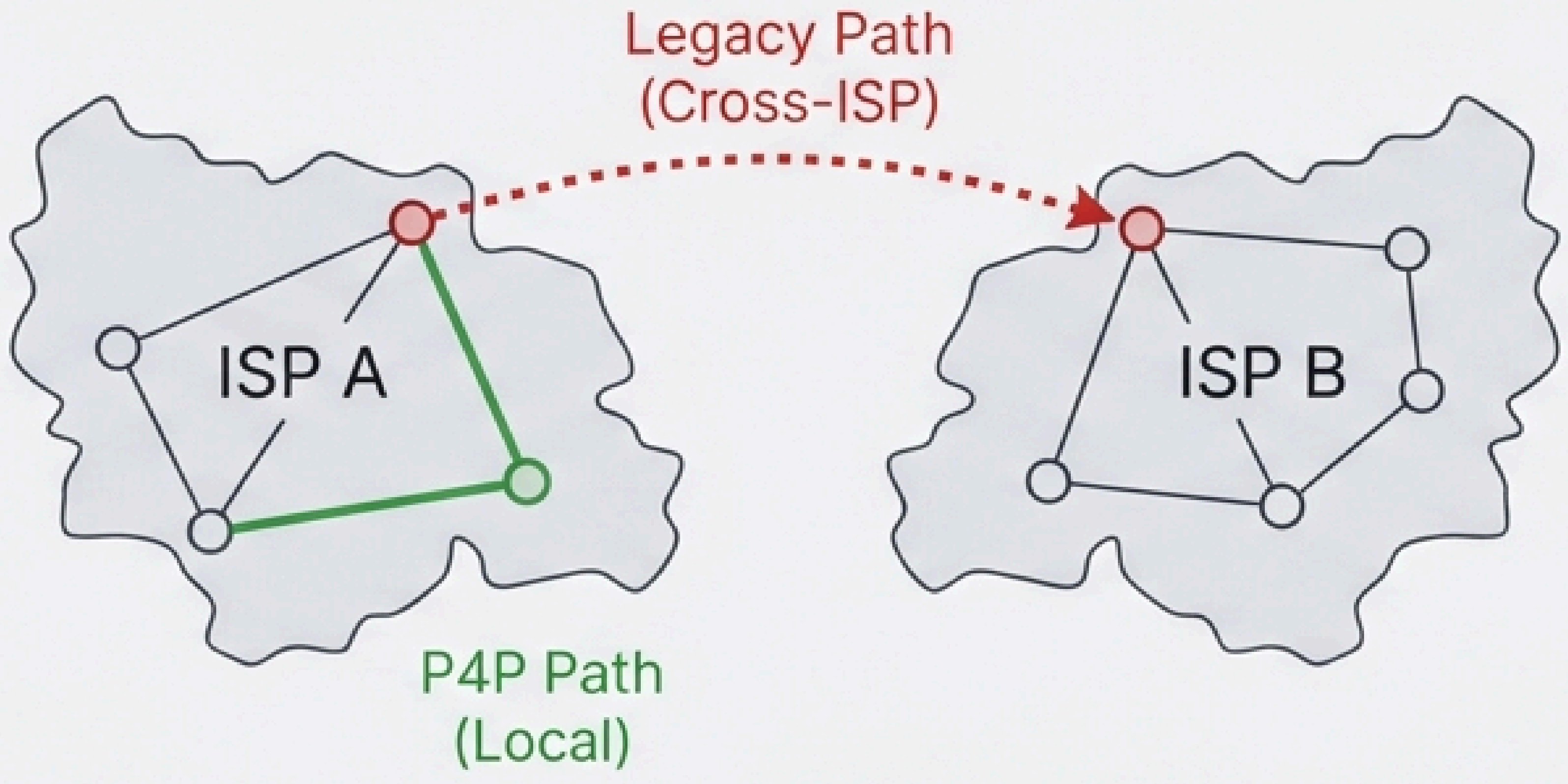
Gini Coefficient (0 = Perfect, 1 = Imbalance)

Metadata: `0.11 (Perfect Balance via Salting)`

Semantic: `> 0.75 (Critical Imbalance)`



Network Optimization: P4P & ALTO Integration



Problem: Application-Layer overlays are blind to physical topology.

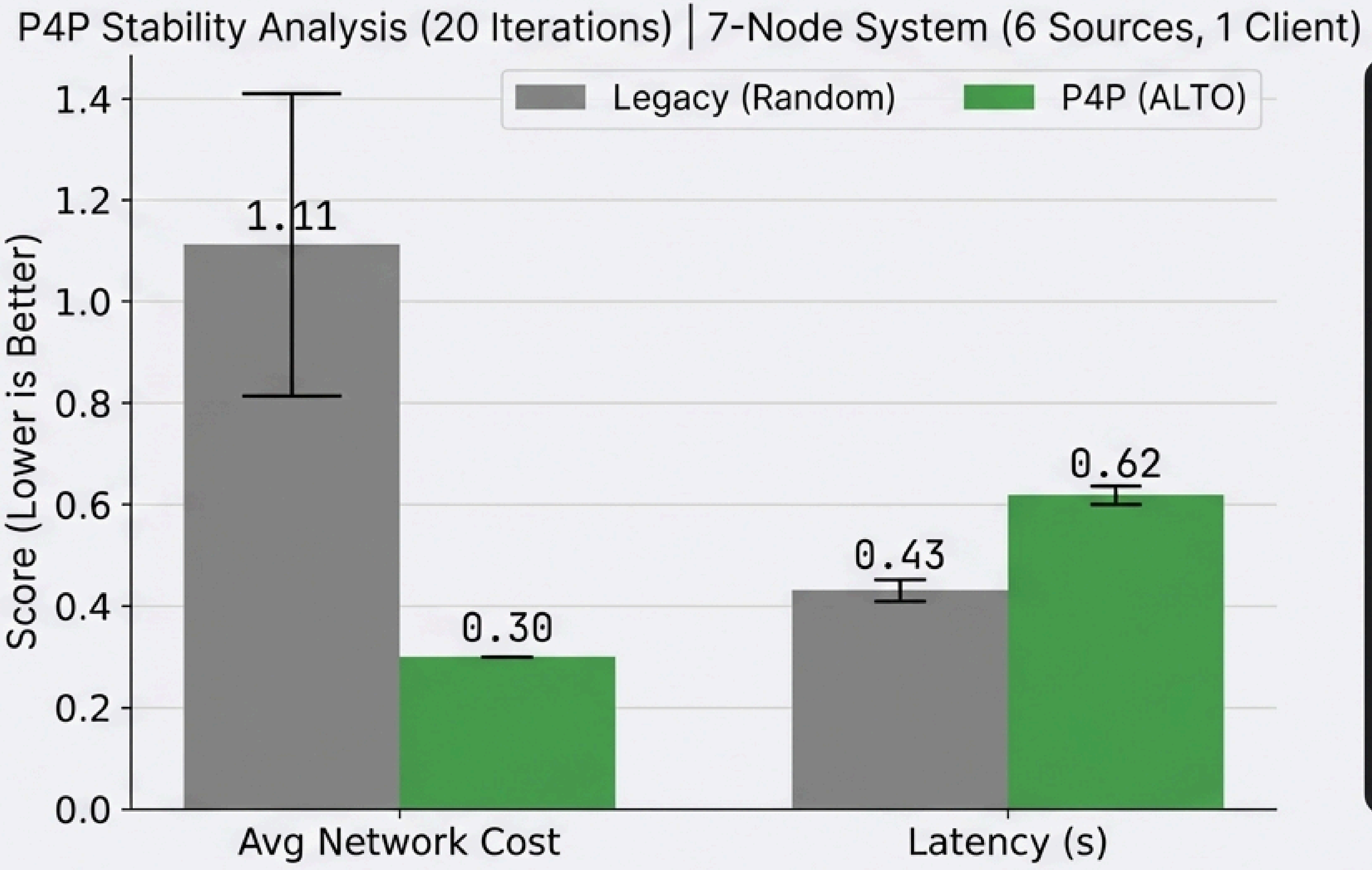
Solution: Integration with an **iTracker** using the ALTO protocol.

Cost Function Optimization:

```
Cost = α * NetworkCost + β * RTT
```

Goal: Minimize ISP transit costs without sacrificing latency.

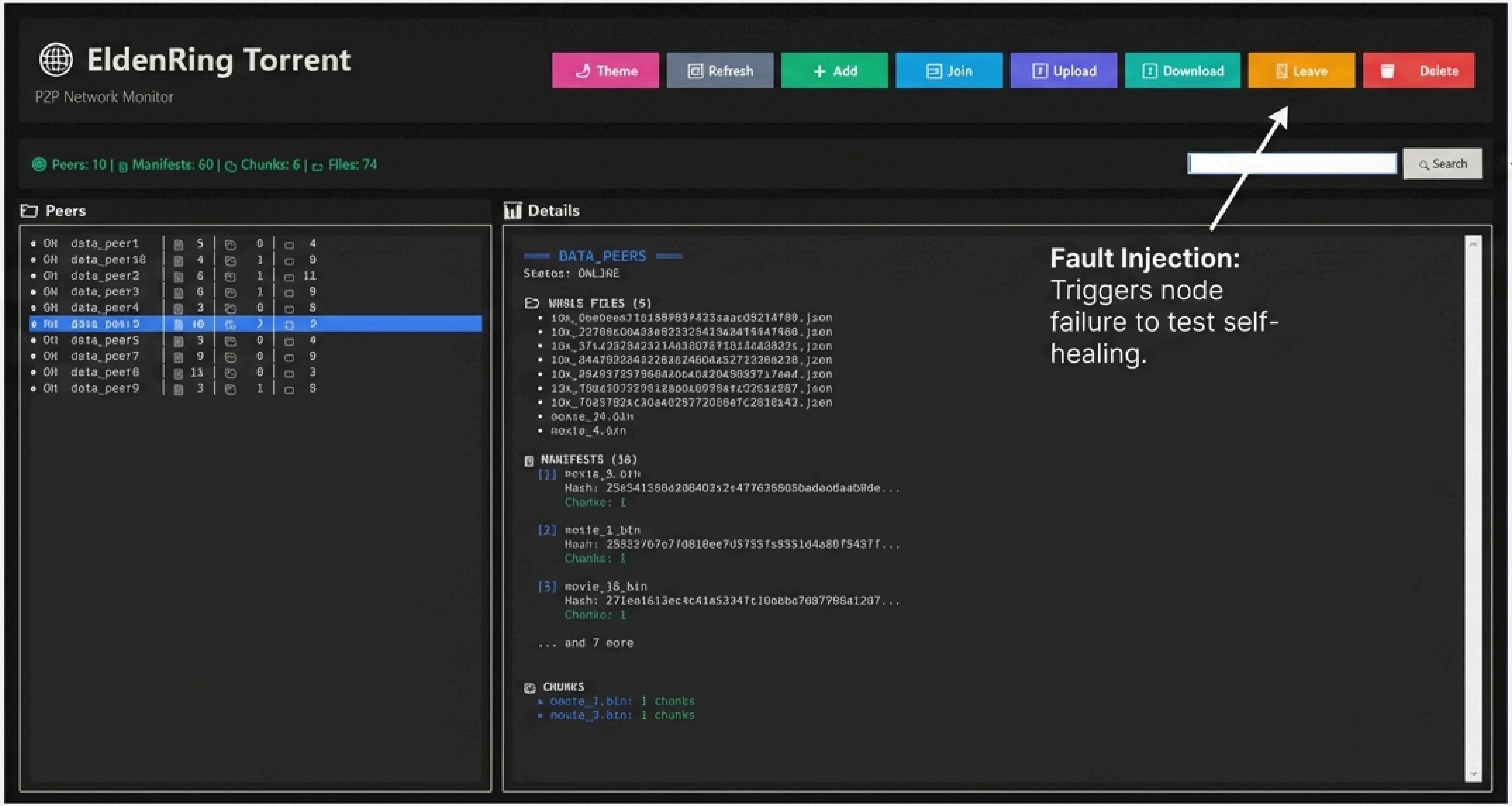
P4P Results: Deterministic Cost Reduction



Key Stats

- **Cost Reduction:** 71% drop in average network cost.
- **Cross-ISP Traffic:** Reduced from 50% → 0%.
- **Stability:** Standard Deviation drops to 0.00. System always picks local peer.

Observability: The EldenRing Torrent Monitor



Fault Injection: Triggers node failure to test self-healing.

Real-time Parsing: Reads JSON manifests from Docker volumes.

Dual Mode: Supports Metadata queries and Hash lookups.

Final Verdict: The Hybrid Architecture

	Speed	Balance	Cost
Naive	Failed (O(N))	Good	Unscalable
Semantic	Good (O(1))	Failed (Hotspots)	Risky
Metadata (GSI)	Optimal	Optimal	**Winner**

Recommendation: A robust Big Data P2P system requires a hybrid approach:

1. **Consistent Hashing** for payload (Chunks).
2. **Sharded/Salted Indexes** for discovery (Metadata).
3. **Topology Awareness (P4P)** for network efficiency.