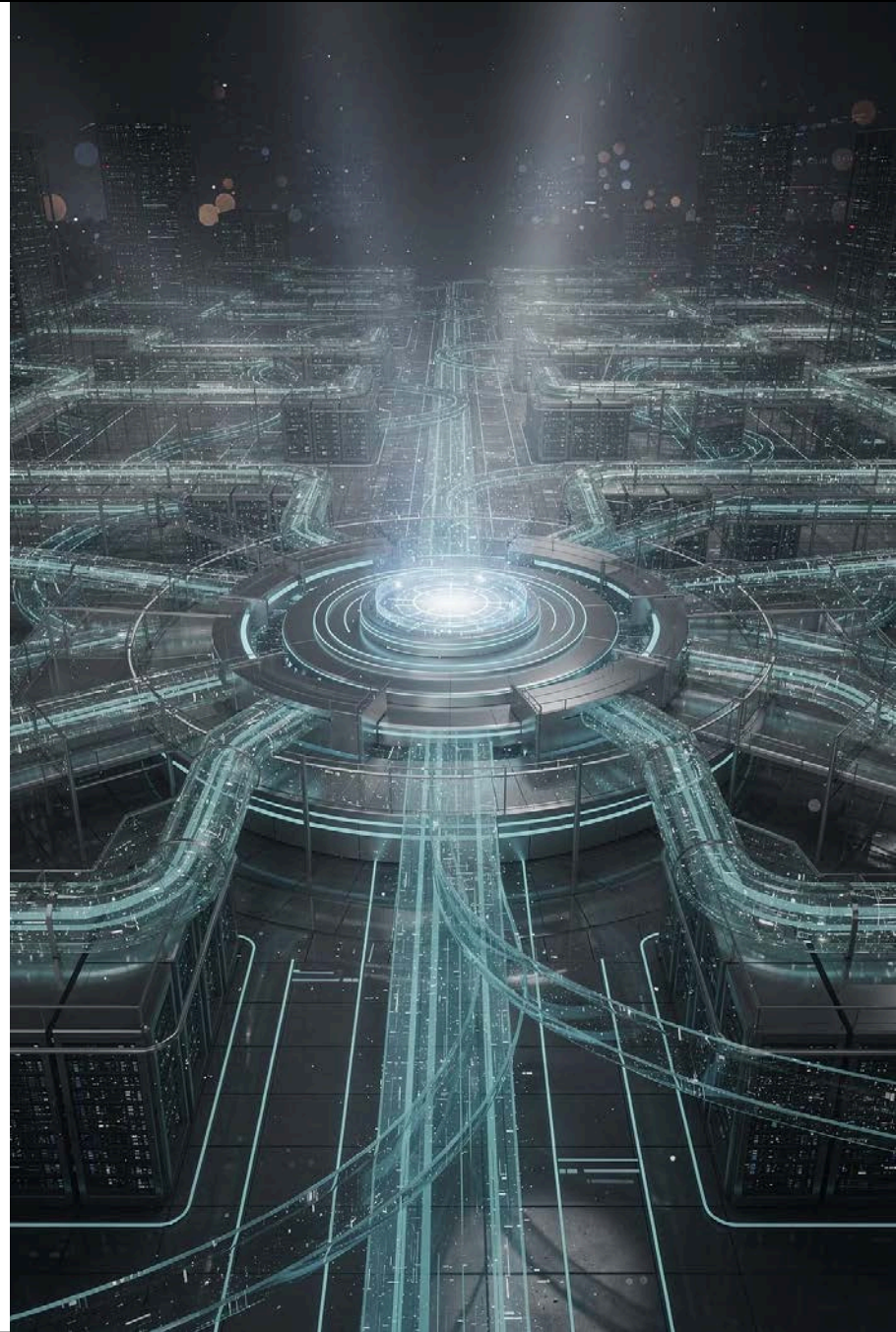
AI-Governed Lakehouse Ingestion with Flink on Kubernetes for Reliable DataOps

Conf42 Site Reliability Engineering (SRE) 2026

RELIABILITY · STREAMING · AI GOVERNANCE



SPEAKER

Jyothish Sreedharan

Vice President · Independent Researcher

Jyothish brings deep expertise at the intersection of distributed systems, stream processing, and AI-driven data platform engineering. His research focuses on building self-healing, semantically-aware ingestion pipelines that reduce operational toil at scale.

Focus Areas

Cloud Lakehouse, Apache Flink, Kubernetes, MLOps

Research Themes

Schema intelligence, semantic contracts, DataOps reliability

The Ingestion Problem at Scale

Modern lakehouse platforms consume data from **transactional databases, event streams, REST APIs, and unstructured text** each evolving independently on its own timeline.

Schema Drift

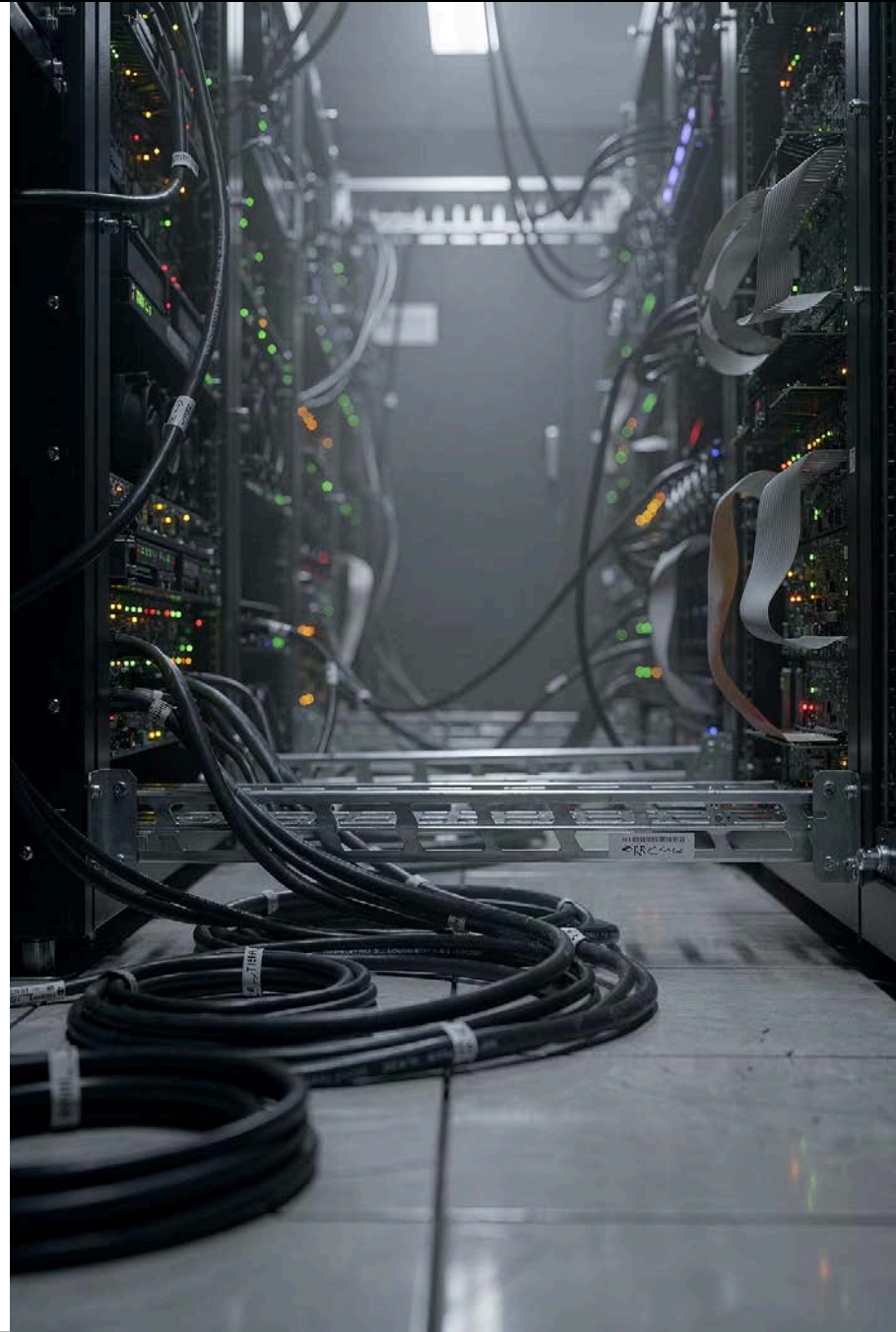
Upstream producers change column types, rename fields, or drop attributes without notice.

Semantic Gaps

Syntactic validation passes while semantic meaning silently diverges bad data lands in the lakehouse.

Manual Toil

SRE and data engineering teams spend significant cycles firefighting schema breakages rather than building.



Why Static Schema Registries Fall Short

Prior research confirms that static schema validation captures only a **limited subset of semantic data quality issues** particularly in continuously evolving distributed systems.

Binary Validation

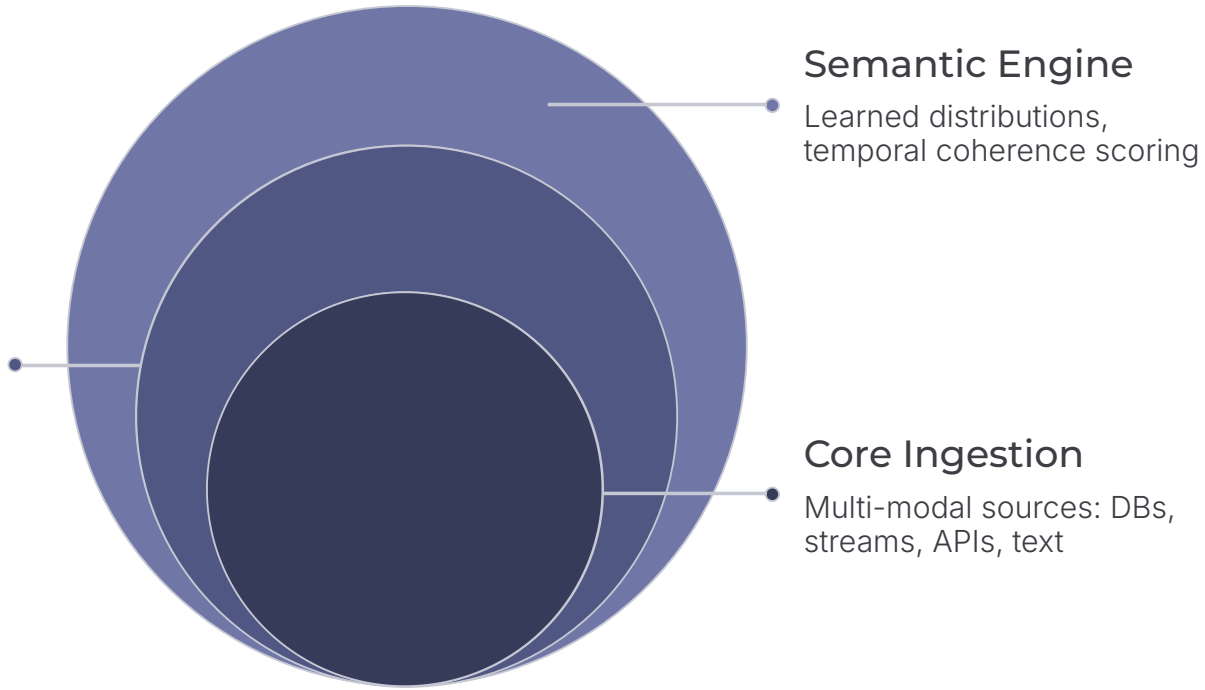
Schema registries enforce structure, not meaning. A valid schema can still carry semantically corrupt data.

No Drift Intelligence

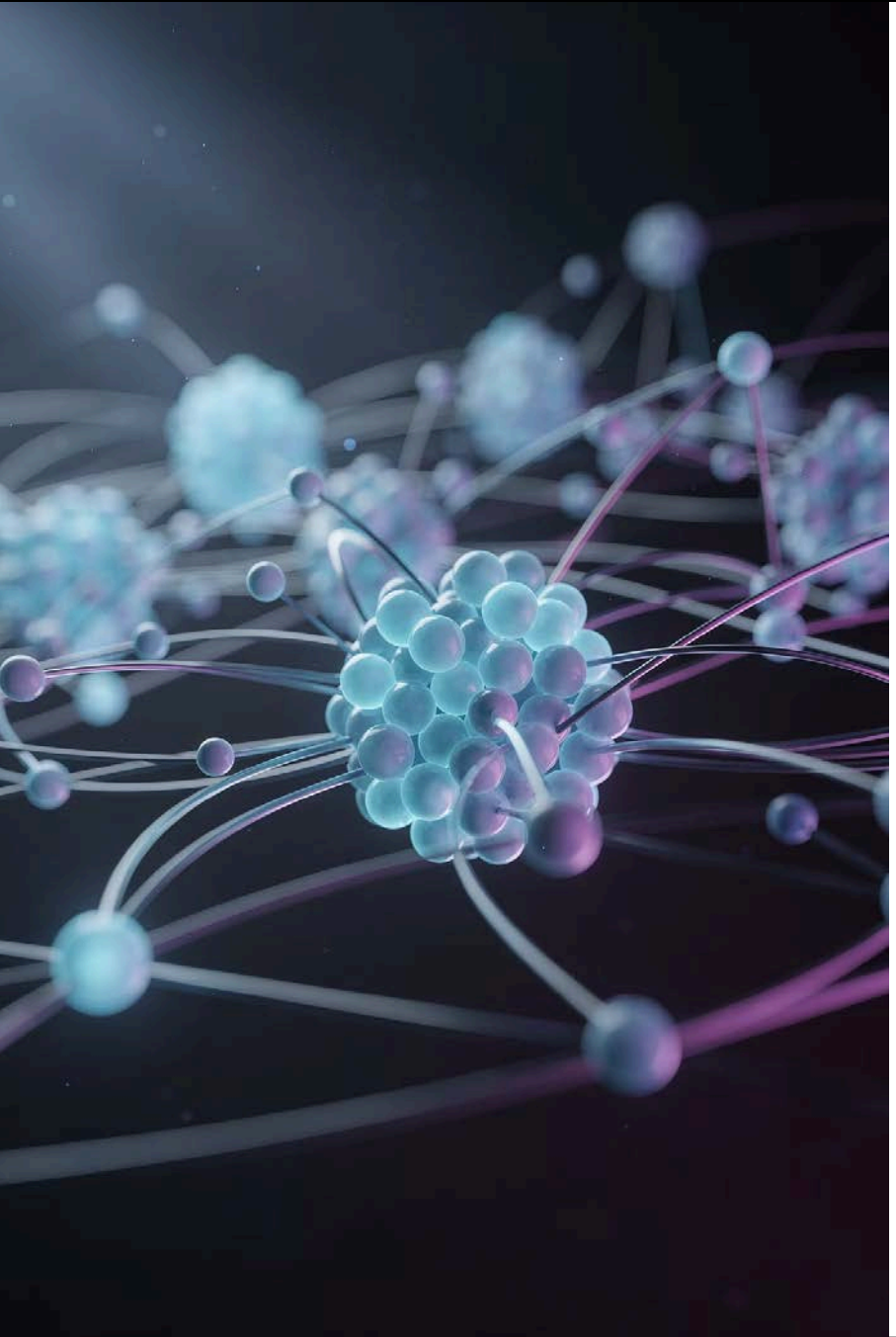
Breaking changes require manual reconciliation cycles, delaying pipelines and increasing MTTR.

ARCHITECTURE OVERVIEW

AI-Governed Multi-Modal Ingestion Architecture



The architecture replaces brittle point-to-point validation with a layered, AI-augmented governance plane that adapts to change autonomously — from raw ingestion through to lakehouse persistence.



Semantic Contracts: Beyond Syntactic Correctness

Semantic contracts encode **learned value distributions, attribute relationships, and temporal patterns** derived from historical data — extending validation far beyond schema shape.

Value Distribution Modeling

Statistical baselines per attribute detect anomalous shifts in real time, even when schema structure is unchanged.

Attribute Relationship Encoding

Cross-field correlations are learned and continuously validated, catching logical inconsistencies between related fields.

Temporal Pattern Awareness

Time-series modeling identifies seasonality violations and late-arriving events that break downstream aggregations.

Self-Evolving Schema Intelligence Layer

The schema intelligence layer uses **LLMs and embedding-based similarity scoring** to interpret schema evolution events autonomously.

01

Detect Evolution Event

Intercept schema change signals from upstream producers in the ingestion stream.

02

Infer Semantic Equivalence

Embedding similarity determines whether a renamed or restructured field carries the same semantic meaning.

03

Generate Transformation Logic

Few-shot LLM prompting produces mapping and transformation code automatically no manual intervention required.

MULTI-MODAL UNIFICATION

Unifying Structured, Semi-Structured & Unstructured Inputs

Structured Sources

CDC from transactional DBs (Postgres, MySQL). Schema-aware extraction with type coercion and null handling.

Semi-Structured

Recursive JSON/Avro/Parquet parsing with AI-assisted field inference and nested schema flattening.

Unstructured Text

LLM-based entity extraction converts free-form logs, emails, and documents into typed, queryable lakehouse records.

Event-time alignment across all three modalities follows the **Dataflow model**, ensuring consistent watermarking and deterministic window semantics regardless of source format.

Apache Flink: The Reliability Engine

Exactly-Once Semantics

Two-phase commit sinks and Flink checkpointing guarantee no duplicate writes to the lakehouse under failure.

Stateful Processing

RocksDB-backed state stores semantic contract baselines and temporal models across long-running operators.

Asynchronous Snapshots

Chandy-Lamport snapshots enable in-flight state capture without pausing data processing critical for low-latency SLAs.

Kubernetes: Elastic, Zero-Disruption Operations

Kubernetes orchestrates Flink operator lifecycle, providing the elastic infrastructure layer that makes AI-governed ingestion operationally viable in production.

KEDA-Based Autoscaling

Task manager replicas scale on Kafka consumer lag, not just CPU aligning capacity with actual backpressure.

Operator Rolling Updates

Flink job upgrades use savepoint-triggered operator restarts, preserving state across version transitions.

RELIABILITY PATTERNS

Resilience Patterns in AI-Governed Pipelines

1

Circuit Breakers

AI inference failures trigger fallback to last-known-good semantic contract, preventing cascade failures.

2

Dead Letter Queues

Records that fail semantic coherence are routed to DLQ with full lineage metadata for async remediation.

3

Backpressure Propagation

Flink's native credit-based flow control prevents memory overflow when AI scoring adds inference latency spikes.

Reliability-Driven Design Tradeoffs

Latency vs. Accuracy

Deeper semantic scoring increases detection precision but adds per-record inference latency. Async scoring with bounded SLA keeps throughput high.

Autonomy vs. Auditability

Auto-generated transformations must be logged, versioned, and human-reviewable. Every LLM-produced mapping is stored with its prompt context.

Model Drift vs. Stability

Semantic contract models must themselves be versioned and rolled back independently of Flink job state.

OPERATIONAL LESSONS

Lessons from Production: What Actually Breaks

AI Model Cold Start

New sources have no historical distribution baseline. Bootstrapping requires a supervised warm-up window before autonomous governance activates.

State Store Explosion

Storing per-attribute semantic baselines for thousands of fields grows RocksDB state rapidly. TTL-based eviction and tiered state backends are essential.

LLM Rate Limiting

Schema evolution bursts can overwhelm external LLM API quotas. Local embedding models and caching layers reduce external dependency.

KEY TAKEAWAYS

What to Take Back to Your Team

1

Extend Beyond Schema Registries

Semantic contracts with learned distributions catch data quality failures that syntactic validation fundamentally cannot.

2

LLMs as Operational Automation

Embedding-based similarity scoring and few-shot generation reduce schema migration toil from days to minutes.

3

Flink + Kubernetes is Production-Ready

Exactly-once semantics, KEDA autoscaling, and savepoint-based upgrades form a reliable operational foundation.

4

Design for Auditability

Every autonomous decision made by the AI governance layer must be versioned, logged, and reversible — reliability demands it.

Thank You!

Jyothish Sreedharan · Vice President · Independent Researcher

---

CONF42 SRE 2026