
BEAM
SUMMIT

Build Seamless Data Ecosystems: Real-World Integrations with Apache Beam, Kafka, and Iceberg

BEAM SUMMIT NYC 2025

Rajesh Vayyala
Principal Data Architect

Intro

Rajesh Vayyala is a seasoned Principal Data Architect and strategic leader in cloud data platforms, with over a decade of experience designing scalable, secure, and high-performance solutions across cloud and enterprise environments.

He brings deep expertise in data governance, modeling, and modern cloud architectures, helping organizations unlock business value, drive innovation, and streamline operations. Rajesh has successfully led complex data modernization initiatives, guiding enterprises in transitioning from legacy systems to cloud-native ecosystems that support real-time analytics, AI integration, and compliance.

A strong advocate for data democratization and AI-driven insights, he collaborates across business and engineering teams to align data strategy with enterprise goals. His work spans cloud migration, governance frameworks, and building AI-ready data platforms that support both structured and unstructured data—with a focus on performance, cost-efficiency, and trust.

He holds a Master's degree in Computer Science and a Bachelor's in Electrical and Electronics Engineering, providing a strong foundation in systems design, data engineering, and analytical thinking. His academic background supports his strategic approach to building modern, AI-ready data platforms.

Agenda

- Why modern data ecosystems need seamless integration
- How Apache Beam connects Kafka and Iceberg for unified batch & streaming pipelines
- Real-world use cases: Fraud Detection and IoT Analytics
- Integration patterns, best practices, and community-driven innovations
- Key takeaways to future-proof your data architecture

Why Seamless Data Ecosystems?

- Enterprises manage diverse, disconnected data sources
- Fragmentation causes inefficiencies, silos, and governance risks
- Unified ecosystems improve trust, accessibility, and speed-to-insight

The Power Trio

Tool	Role in Ecosystem	Key Capabilities
Apache Beam	Unified data processing engine for batch and streaming	Portability across Flink, Spark, Dataflow- Windowing, event-time, UDFs
Apache Kafka	High-throughput event ingestion and distribution	Durable, real-time streaming- Fault-tolerant and horizontally scalable
Apache Iceberg	Transactional storage layer for the data lake	ACID compliance, schema evolution- Time-travel, partition pruning

Architecture Pattern: Kafka → Beam → Iceberg

Component	Role
Kafka	Real-time event ingestion & streaming buffer
Beam	Unified stream and batch processing layer
Iceberg	Scalable, ACID-compliant analytics storage

Pipeline Flow:

- Data producers (transactions, IoT, CDC) write to Kafka topics.
- Beam pipelines ingest from Kafka, perform transformations, windowing, DQ checks, and enrich data.
- Processed output lands in Iceberg tables with partitioning and time-travel support, ready for querying.

Open-Source Apache Data Stack Reference Architecture

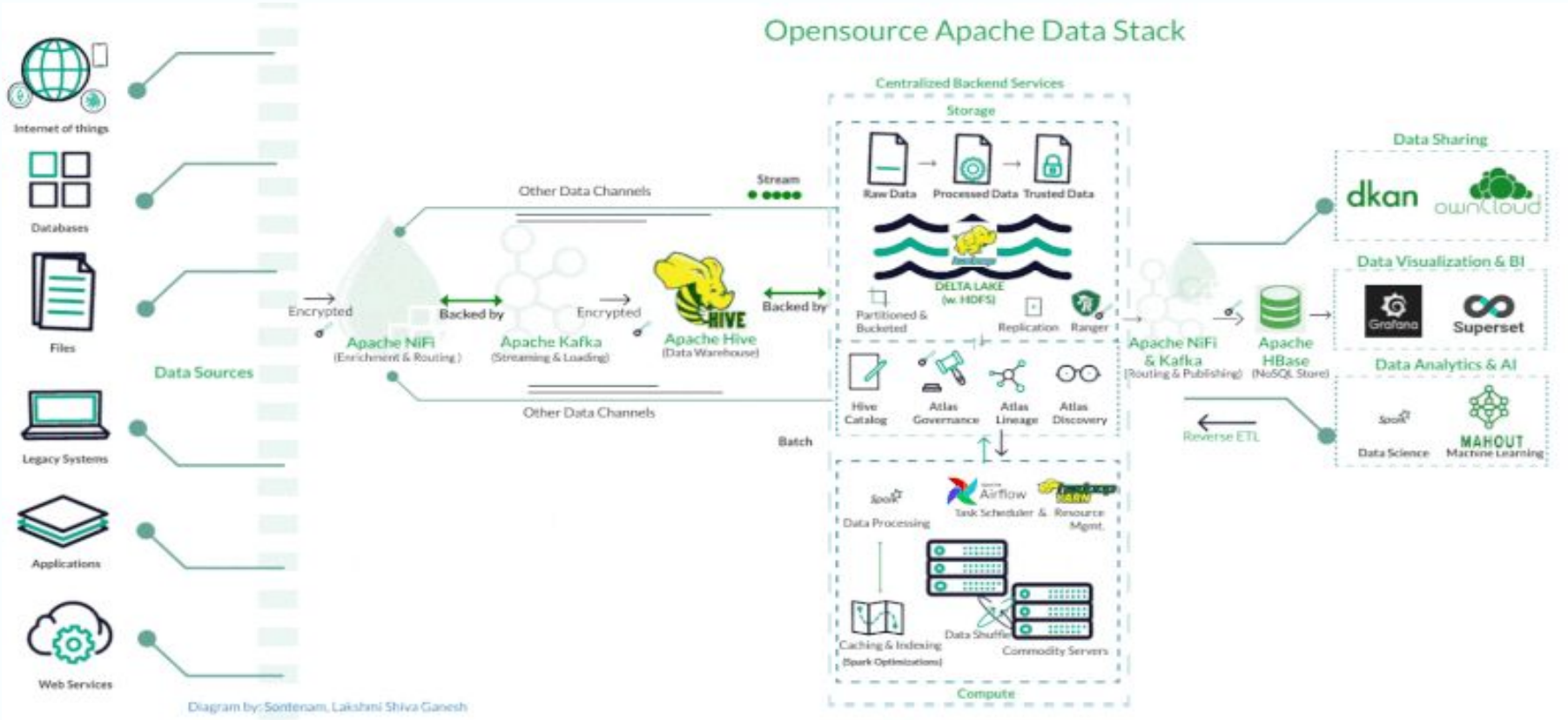


Diagram by: Sontenam, Lakshmi Shiva Ganesh

Real-Time Streaming Patterns (Kafka + Spark)

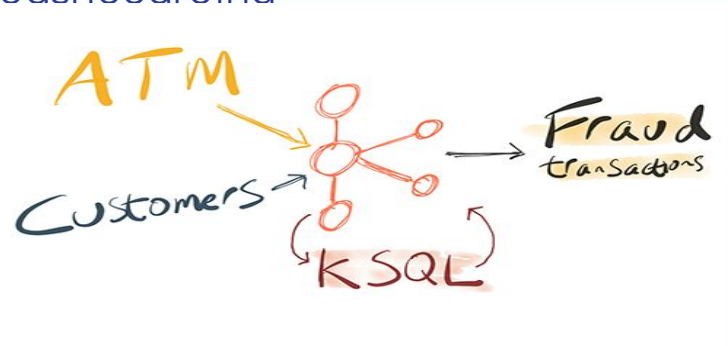
- Point-to-Point (Direct): Kafka → Spark → Sink (HDFS, RDBMS)
- Publish-Subscribe: Kafka → Multiple Spark consumers (parallel ETL, enrichment)
- Lambda Pattern: Real-time layer (Spark Streaming) + batch layer (Spark batch)
- Kappa Pattern: Streaming-only design using Kafka + Spark

Architecture Patterns: Apache Beam, Spark & Iceberg

Pattern Name	Use Case	Processing Tool	Target Format	Strength
Stream-to-Lakehouse	Real-time ETL + analytics	Beam	Iceberg	Low latency + ACID
Batch ETL to Iceberg	Historical or periodic ETL	Spark	Iceberg	Scalable, cost-effective
Real-time + Batch Unification	Merged historical + real-time	Beam + Spark	Iceberg	Comprehensive and flexible
Open Lakehouse Pattern	Multi-engine interoperability	Beam / Spark	Iceberg + Catalog	Tool-agnostic access
CDC Ingestion	OLTP → Lakehouse sync	Beam	Iceberg	Real-time snapshots and change history
Compaction via Spark	Performance optimization	Spark	Iceberg	Faster queries, smaller files

Use Case 1 – Fraud Detection

- Kafka receives financial transaction streams
- Beam applies windowing, joins with customer/account data, and flags anomalies
- Iceberg stores region-wise fraud risk metrics for auditing and dashboarding

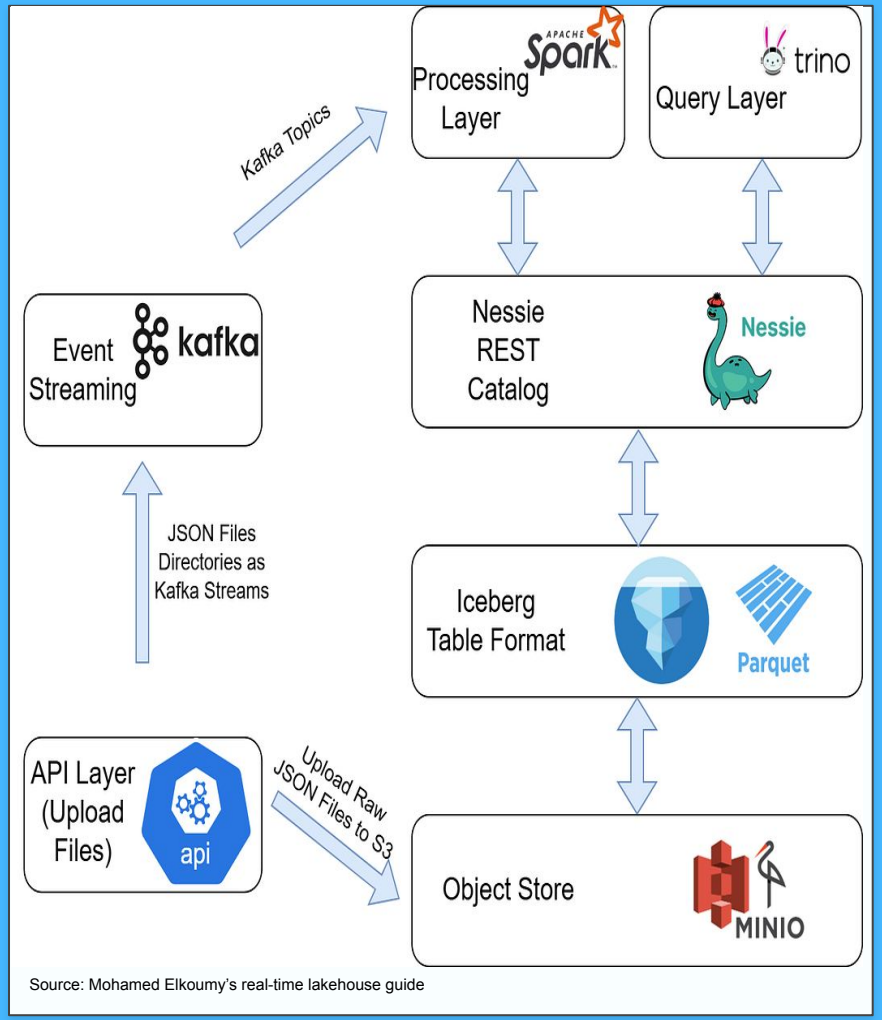


Use Case 2 – IoT Analytics

- Some Devices emit telemetry (temperature, pressure, etc.)
- Kafka ingests sensor data at high velocity
- Beam normalizes data by device type and time
- Iceberg stores time-partitioned data for reporting and ML

Unlocking Analytics with a Lakehouse

- Traditional data warehouses struggle with scalability and real-time data
- A Lakehouse powered by Spark, Kafka, and Iceberg bridges operational + analytical worlds
- Combines low-latency ingestion with long-term analytical storage
- Ideal for building BI dashboards, self-service analytics, ML features



Source: Mohamed Elkoumy's real-time lakehouse guide

Patterns & Best Practices

- Use schema registry for consistency across Kafka → Beam → Iceberg
- Design Beam pipelines with modular, reusable transforms
- Monitor latency, lag, and data quality using Beam and Kafka metrics

Open Source Innovation

- Beam: SQL support, evolving cross-language SDKs, new connectors
- Kafka: ksqlDB, tiered storage, schema evolution
- Iceberg: branching, REST catalog, integrations with Spark, Flink, Trino

Conclusion & Future Outlook

- Beam unifies real-time and batch pipelines
- Kafka handles fast, fault-tolerant event ingestion
- Iceberg ensures scalable, reliable analytics with open format
- Together, they reduce friction and future-proof your stack
- Open data ecosystems are here to stay
- Beam, Kafka, and Iceberg empower real-time + batch analytics at scale
- Community-led innovation is accelerating integration and capabilities
- Future trends: lakehouse standardization, cross-engine interoperability, catalog evolution

Rajesh Vayyala

QUESTIONS?

LinkedIn: www.linkedin.com/in/rajeshvayyala