Unified Event Streams: A Program Management Lens on Architectural Transformation[1]

Sejal Harishchandra Shah

Modern organizations increasingly struggle with fragmented event pipelines with each team building isolated systems for logging, telemetry, analytics, and transactions. This fragmentation creates duplicated engineering effort, inconsistent governance, and significant integration complexity.

Unified Eventing (Event Streams) initiative demonstrates how a well-managed program can overcome these issues, delivering a single, abstracted platform that improves reliability, governance, and developer experience.

This article explores the program management approach that enabled this transformation, highlighting how stakeholder alignment, technical governance, phased change management, and value measurement came together.

1. Setting the Stage: What are Events and Why Unification Was Needed

What Are Events?

In technology systems, an *event* simply marks that something happened at a specific time. Think of a shopper clicking the "buy" button on an e-commerce site, a user logging into their account, a temperature sensor capturing a reading, or an error message being written to a system log.

Events are the building blocks of how modern digital platforms capture behavior, diagnose issues, and make decisions. They carry information like **who did what, when, and under what conditions**, and they are stored or streamed so that other systems analytics dashboards, fraud detection engines, personalization models, or monitoring tools can react appropriately.

Why Do Events Matter?

Events matter because they create the **data fabric** for an organization.

[1] How to cite this article: Shah, S. H. (2025). Unified Event Streams: A Program Management Lens on Architectural Transformation, PM World Journal, Vol. XIV, Issue IX, September

Events form the connective tissue of modern organizations. They show how people interact with products and services, which helps teams measure adoption, engagement, and satisfaction. They also provide the training data that powers machine learning models whether predicting customer churn, tailoring recommendations, or detecting unusual activity. Without events, both analytics and intelligence systems would have little to act upon, decisions are slower, personalization is clumsy, and system reliability is harder to maintain.

The Problem with Fragmented Events

In many organizations, each team builds its own way of capturing and processing events. For example:

- Marketing captures events for campaigns.
- Engineers log service metrics.
- Product teams track user behavior for analytics.

Because these pipelines grow independently, they often **duplicate effort**, define **different meanings for the same event**, and use **incompatible formats**. This makes it expensive to maintain systems, hard to trust the data, and frustrating for developers who need to integrate across silos.

What Is Unified Eventing?

Unified Eventing is the effort to consolidate all these fragmented pipelines into a single, standardized platform. Instead of every team reinventing the wheel, the organization creates a **shared backbone** with clear interfaces and schema validation. This allows:

- One definition of each event across the company (improving consistency).
- A single ingestion point, reducing duplicated infrastructure.
- Standard governance (schema registries, contracts, versioning).
- Better developer experience and faster innovation.

In practice, Unified Eventing is about treating event infrastructure as a **platform product**, with tools, SDKs, and documentation that make it easy for every team to publish and consume events reliably.

2. Defining the Core Strategy

The Unified Eventing program addressed these challenges through three key architectural abstractions:

i. Customer Data Platform (CDP) Client - For analytics and marketing event ingestion, offering schema validation and routing to analytical destinations (Amplitude, Databricks, Snowflake).
ii. Kafka SDK - For low-latency transactional streams, enabling service-to-service communication and machine learning feature feeds.
iii. DataDog SDK - For observability data, integrating directly with monitoring and incident-response systems.

By standardizing on Confluent Kafka as the shared transport backbone, these abstractions offered consistent ingestion paths while allowing teams to choose the interface suited to their use case.

3. Managing the Rollout: Phased and Low-Disruption

Large-scale infrastructure changes can stall if they are perceived as disruptive. We mitigated this risk through a phased adoption strategy:

- Phase 1: Low-risk pilots with CDP and analytics teams to validate technical feasibility and developer experience.
- Phase 2: Early-adopter expansion to high-value transactional workloads with measurable latency and stability improvements.
- Phase 3: Broad enablement with migration playbooks, SDK adoption guides, and automated schema validation tools.

The principle was simple: deliver early wins to build credibility before mandating change.

4. Governance Through Contracts and Schema Enforcement

Technical consolidation requires governance that empowers rather than constrains. Unified Eventing achieved this by:

- Establishing a schema registry with version control, CI/CD enforcement, and automated validation.
- Defining data contracts to ensure consistent meaning and quality across teams.
- Documenting onboarding and publishing standards for all event classes.

This shifted governance from a manual review process to an embedded quality system that caught issues before deployment.

5. Program Governance and Change Management

From a program management perspective, success hinged on transparent governance and clear accountability:

- A steering committee comprising architecture, platform, and product leads oversaw direction and decisions.
- OKRs and KPIs such as the percentage of event traffic flowing through new abstractions were tracked publicly.
- Bi-weekly checkpoints helped alignment across engineering, data, and business stakeholders.

Importantly, adoption benefits were tied to real incentives like reduced integration friction, measurable latency improvements, and fewer production errors rather than compliance for its own sake.

6. Implications for Project and Program Managers

This highlights lessons relevant to the Project Management audience:

- Beyond the triple constraint - Infrastructure programs demand adaptability, stakeholder engagement, and governance discipline, not just time/cost/scope control.
- Platform as product - Treating infrastructure as a consumable product increases adoption and satisfaction.
- Iterative rollout beats big bang - Incremental delivery builds trust and reduces change fatigue.
- Embedded governance works - Automating standards enforcement scales better than manual oversight.

7. Recommendations

For leaders planning similar transformations:

i. Prioritize minimal-friction migration paths to encourage voluntary adoption.
ii. Use schema enforcement and versioning as proactive governance tools.
iii. Lead with metrics and visibility to sustain momentum and demonstrate value.
iv. Frame infrastructure as a service to internal teams, with clear onboarding, documentation, and support.

Conclusion

Unified Eventing was more than a technical consolidation, it became a culture-shaping initiative that raised efficiency, improved quality, and made life easier for developers. What made the effort successful wasn't only the architecture itself, but the way the program was framed, governed, and executed. For program managers, it offers a practical example of how to steer large, cross-enterprise initiatives from concept to adoption.

About the Author

Sejal Harishchandra Shah

Sunnyvale, California, USA

Sejal Harishchandra Shah is a Senior Technical Program Manager and a seasoned expert in data platform architecture, big data systems, and marketing technology (MarTech) execution at scale. With over 20 years of leadership in technical program management, he has delivered enterprise-grade solutions in customer data platforms, experimentation infrastructure, and AI-powered engagement systems. Sejal has driven critical initiatives at Apple and Block Inc., including schema-governed eventing, OpenAI-integrated marketing tools, and self-serve experimentation frameworks. His deep understanding of data reliability, large-scale analytics pipelines, and MarTech orchestration uniquely positions him at the intersection of engineering, product strategy, and scalable system delivery in modern digital ecosystems. He can be contacted at sejal.s.tpm@gmail.com