

ARTICLE



# From plumbing to profitability: bringing **FinOps discipline** to enterprise messaging.



Enterprise messaging systems have quietly become foundational to modern digital businesses, evolving from simple “plumbing” to the backbone of customer experiences and real-time decision-making. However, the management of these platforms—including Apache Kafka®, RabbitMQ, and IBM MQ—has not kept pace with their elevated importance. This has led to a pervasive problem of “invisible costs,” with enterprises losing millions to wasteful configurations, redundant infrastructure, and unmonitored usage.

This article argues that applying a FinOps discipline to messaging is a fundamental shift that enables faster innovation and business value creation. By treating messaging platforms not as static infrastructure but as dynamic systems with measurable costs, companies can transition from reactive cost control to proactive value creation. This transformation can eliminate 20–40% of unnecessary messaging spend and improve team accountability, ultimately enabling the business to move faster with confidence.

## The invisible cost of enterprise messaging: a black box problem.

### The fragmented fabric of modern communication.

In large enterprises, messaging environments are often fragmented, with different platforms managed by different teams, such as data teams owning Apache Kafka®, DevOps managing RabbitMQ, and infrastructure teams overseeing IBM MQ. This siloed approach creates a profound lack of visibility and makes it difficult to answer fundamental questions about who is consuming resources and how they are being used.

Traditional messaging systems were not built with financial transparency in mind. Unlike modern cloud services that offer native tagging and usage analytics, traditional brokers lack built-in capabilities for detailed consumption views by application or team. This absence of financial intelligence transforms the messaging fabric into a black box, a critical, expensive, and unmanaged system that grows without the necessary oversight to manage its escalating costs.

## The hidden anatomy of messaging waste.

The problem of invisible costs is not hypothetical—it's pervasive and measurable. Across all major messaging platforms, organizations are hemorrhaging resources due to inefficient configurations, redundant allocations, and outdated architectures. Whether it's Apache Kafka®, RabbitMQ, IBM MQ, Apache ActiveMQ®, or Solace, the patterns of waste are both platform-specific and universally impactful.

- **Apache Kafka® Over-Provisioning:** One of the most insidious forms of waste stems from over-provisioning and replication overhead. In multi-AZ cloud deployments, network transfer—especially cross-AZ traffic—often accounts for more than 50% of infrastructure costs. Teams frequently over-engineer Apache Kafka® environments by creating topics with hundreds of partitions or applying retention policies set for days when hours would suffice. These practices drive up costs, increase operational complexity, and impair scalability.
- **RabbitMQ Inefficiencies:** Idle and unmonitored queues are silent cost drivers. Even when unused, RabbitMQ queues continue consuming disk and memory via their persistent index and storage. Moreover, persistent messaging is often applied by default—even when the data could be safely re-ingested—causing unnecessary disk I/O and throttling throughput. These performance-cost trade-offs are rarely reviewed and almost never optimized.
- **Legacy System “TCO Debt” (IBM MQ, Apache ActiveMQ®):** Legacy brokers like IBM MQ and Apache ActiveMQ® represent a substantial portion of enterprise messaging. Yet they are often running on outdated infrastructure, lack visibility, and accumulate “technical debt” in the form of high maintenance costs and mounting security risks. Without patching and modernization, these platforms can consume a disproportionate share of IT budgets while exposing the business to compliance issues and cyber vulnerabilities. The hidden cost here isn't just operational, it's existential.
- **Solace and Event Mesh Overhead:** Even in more modern event streaming environments like Solace, waste persists. Systems are frequently over-provisioned to handle theoretical peak loads that never materialize. Events are routed to consumers that no longer exist. Bandwidth is consumed, memory is reserved, and throughput is throttled, without any correlated business value. The illusion of “modern architecture” masks the same inefficiencies that plague older systems.

Every messaging platform has blind spots. But with proper observability and financial accountability, every one of them can be optimized. FinOps doesn't just reduce waste, it restores agility. It unlocks the full potential of your messaging infrastructure by shifting the focus from fire-fighting and cost-cutting to strategy and value creation.

# The FinOps solution.

FinOps is the practice of bringing financial accountability to technology usage by fostering collaboration among engineering, finance, and business teams. The FinOps Foundation defines a three-phase lifecycle:

- **Inform:** This phase is about achieving visibility into usage, performance, and cost across all messaging systems. This involves collecting data on message volume and resource consumption, then accurately attributing those costs to the correct teams using tagging and metadata.
- **Optimize:** With data from the Inform phase, this phase focuses on identifying and eliminating waste. This includes rightsizing underutilized resources, tuning retention settings, and retiring unused resources.
- **Operate:** The final phase involves embedding FinOps practices into daily workflows through governance, automation, and continuous improvement. This ensures that FinOps is an ongoing, living practice rather than a one-time project.

## Key metrics for messaging FinOps.

To operationalize FinOps, organizations must prioritize key metrics that provide both technical and financial context:

- **Message Volume per Topic or Queue:** Measures the direct consumption of resources.
- **Retention Duration and Replication Factor:** Directly impacts storage, network, and financial costs.
- **Consumer Lag and Queue Backlog:** Indicates the health and efficiency of the system and signals potential processing bottlenecks.
- **Cost per Transaction, per Service, or per Customer:** The ultimate measure of a mature FinOps practice, as it directly ties technology spend to business value.



# Platform-specific strategies and the case for modernization.

A one-size-fits-all approach will fail to capture the most significant optimization opportunities. Success depends on platform-specific strategies tailored to each broker's architecture:

- **Apache Kafka®:** The optimization strategy should focus on designing for network efficiency and rightsizing retention policies. Leveraging tiered storage is a game-changer, offloading older data to cheaper object storage and reducing costs by over 90%.
- **RabbitMQ:** Key strategies include keeping queues short to reduce overhead and using transient messages for non-critical data to significantly boost throughput and reduce disk I/O.
- **IBM MQ:** For legacy systems, the FinOps analysis reveals a compelling case for modernization. The move to a cloud-native or SaaS solution transforms a high-overhead, capital-intensive model into a flexible, operational one. This transition offers automated provisioning, maintenance, and security patching, freeing up engineers to focus on higher-value work.
- **Apache ActiveMQ®:** Many Apache ActiveMQ® environments suffer from legacy sprawl, with unknown or orphaned queues still consuming resources. FinOps efforts should focus on auditing queue usage, retiring obsolete applications, and consolidating deployments to reduce duplication. For teams with strict latency or compliance requirements, migrating to managed messaging services can also yield operational and cost efficiencies.
- **Solace:** Solace platforms are often over-provisioned for peak throughput scenarios that rarely occur. FinOps strategies should include detailed traffic analysis to right-size event mesh deployments and eliminate message delivery to inactive consumers. Monitoring cross-region event routing and implementing cost-aware routing policies can also reduce unnecessary bandwidth consumption and infrastructure spend.

# Conclusion and forward-looking recommendations.

Enterprise messaging has evolved into a strategic asset that must be managed with the same discipline as cloud infrastructure. The fragmented and opaque nature of its management has created a pervasive problem of hidden costs and mounting technical debt. The discipline of FinOps offers a clear, actionable, and culturally transformative framework for addressing this problem.

To begin this journey, a phased approach is recommended:



**Phase 1: Gain Visibility (Inform):** Implement a unified operational intelligence platform that provides a single source of truth for all messaging brokers. This platform must collect and correlate essential metrics for cost, usage, and performance across all platforms.



**Phase 2: Attribute and Analyze (Inform & Optimize):** Once the data is centralized, enforce a robust tagging and metadata strategy to enable granular cost attribution by team and application.



**Phase 3: Operationalize and Automate (Operate):** Integrate FinOps into daily workflows and toolchains for continuous governance and rapid feedback loops. Embed metrics, accountability, and controls into CI/CD pipelines and development workflows.

# meshIQ: enabling messaging FinOps at scale.

To make this vision a reality, organizations need a platform purpose-built for messaging observability and financial governance. This is where meshIQ is uniquely positioned to help.

## meshIQ provides:

- **Unified observability** across Apache Kafka®, RabbitMQ, IBM MQ, Solace, and Apache ActiveMQ®
- **Per-queue/topic usage analytics** with trends, backlogs, and volume analysis
- **Tagging and metadata support** for granular cost attribution and ownership
- **Anomaly detection** and heatmaps to surface inefficiencies in real time
- **A single API across platforms** to integrate directly into CI/CD workflows
- **Dashboards and reporting** that align message behavior with business context and cost

With meshIQ, platform teams gain visibility. Finance teams gain accountability. And business leaders gain confidence that their messaging infrastructure is aligned with strategic goals and optimized for cost, performance, and agility.

## Want to see what's really happening inside your messaging infrastructure?

**Schedule a personalized demo with meshIQ** and discover how you can turn messaging from an invisible cost center into a measurable competitive advantage.

