

Top 6 Reasons to Modernize Aging Messaging Infrastructure

- Overview** 2
 - Challenge #1: Data coupling between systems creates inter-team dependencies and hinders innovation. 2
 - Challenge #2: Batch processes cannot enrich data in flight for real-time use cases. 3
 - Challenge #3: Low fault tolerance at scale creates performance bottlenecks and operational complexities 4
 - Challenge#4: Legacy middleware leads to increasing technical debt and soaring operational costs 6
 - Challenge #5: Ephemeral persistence and lack of data context limits use as a system of record 7
 - Challenge #6: Slow consumers can degrade performance and eventually lead to outages, requiring costly intervention. 8
- Why you need a platform for data in motion.** 9
- Why Confluent is unlike any other messaging system.** 9
- How to modernize** 10
- Want to learn more?** 11

Overview

We're in an unprecedented time and going through the largest digital transformation in history, where enterprises are moving from using software to becoming software-driven. This is a trend being manifested across all industries: financial services, retail, manufacturing, transportation and logistics, media and entertainment, healthcare, and many more. To become a software-driven business that operates in real time, enterprises are undertaking multiple modernization initiatives, including application modernization, artificial intelligence and machine learning, the cloud, and edge analytics.

But modern applications no longer live in isolation. They are built on microservices and rely on other services to move data between applications. Nowadays, modern distributed applications comprise hundreds of thousands of remote services operating in multiple tiers, all of which are loosely coupled and need to exchange data with each other. In such distributed architectures, reliable data and message distribution is key.

Message-oriented middleware or messaging middleware has been widely used to handle the needs of message distribution and interservice communication across distributed applications. Message-oriented middleware systems like enterprise service buses (ESBs), message queues (MQs), and data integration tooling for extract, transform, and load (ETL) act as the critical layer among services that need to communicate. REST APIs have also emerged recently to enable simple point-to-point communication.

This paper takes a look at six Confluent customer implementations, the challenges posed by legacy approaches to messaging middleware, and the strategies they used to overcome them.

Challenge #1: Data coupling between systems creates inter-team dependencies and hinders innovation

In 2020, a Fortune 500 company—one of the largest transportation logistics companies in North America—had to accelerate its digital efforts to meet the needs of its customers during the pandemic. With increased customer expectations for faster services and greater transparency on shipments, real-time data was becoming a cornerstone in the \$1 trillion logistics industry. To rapidly scale to meet its customer needs, the logistics company began its journey to the cloud. However, limitations with MQs, a core component in the company's data infrastructure, plus ongoing cost overruns with a supposedly real-time data ingestion service the company was already using on a cloud platform, had to be addressed first.

The company's existing MQs were monolithic and did not fit within the broader move to the cloud and microservices. Moreover, MQ point-to-point solutions simply moved messages around, which did nothing to address issues with data coupling and schema mismatches between different systems. This meant that adding new data sources (producers) and sinks (consumers) to their messaging system took weeks, if not months, due to inter-team dependencies. This significantly slowed down developer productivity and impeded innovation. Furthermore, operating MQs frequently led to outages during upgrade cycles and required costly interventions to get back up and running. Downtime and outages in the messaging infrastructure had adverse impact on the company's logistics business, where lost time meant lost money.

Confluent fully decouples and standardizes data across systems to increase developer productivity

The logistics company recognized that they needed a different way to connect systems beyond the tightly coupled methods they were using. An event-driven architecture in Confluent meant that services could be fully decoupled. Schema Registry builds a "contract" between producers and consumers to standardize data format across various systems while preserving backwards compatibility, and as producers and consumers evolve their schemas over time. Pre-built connectors made connecting to any data source or sink easy without having to build custom functionality from scratch each time. Moving data in and out of Confluent was an effortless task, giving the developer teams more time to focus on application development. This improved developer velocity and made it possible for teams to operate independently. By removing technology barriers and democratizing data access to the teams that need it faster, various teams across the organization are now able to take advantage of all their data and implement new stream processing applications within days, instead of months. With Confluent's fully managed cloud service, the company was able to move faster, reduce its overall total cost of ownership (TCO), and get its best people out of operations.

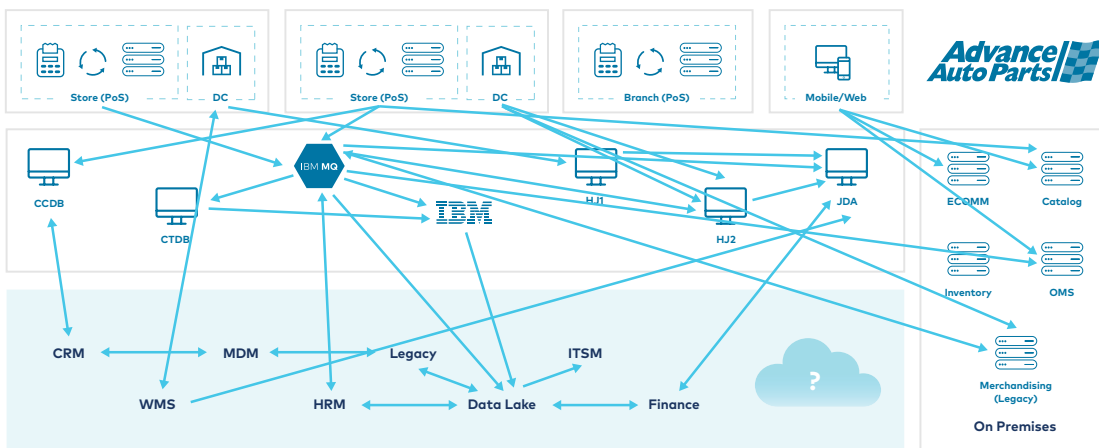


Challenge #2: Batch processes cannot enrich data in flight for real-time use cases

With more than 5,200 stores, multiple online brands, and tens of thousands of suppliers, Advance Auto Parts is the largest automotive parts provider in North America. The company's rapid growth, spurred in part by a series of acquisitions, had begun to put a strain on the IT infrastructure. Merchandising, pricing, and other core systems operated in silos and relied on batch ETL processes that ran overnight after stores had closed. As a result, business-driving reports and analytics were not available until the next morning, offering few insights into intraday events. Further, legacy messaging systems that underpinned many of the business processes lacked the reliability the company needed, and manual interventions were frequently required, costing time and money

"One of our largest customers had a critical need for us to invoice them in near real time, and the batch-oriented process we were using couldn't meet that need."

— Prabh Chandrasekhar, Director of Data and Analytics at Advance Auto Parts



Legacy infrastructure and architecture did not meet real-time performance requirements [Before]

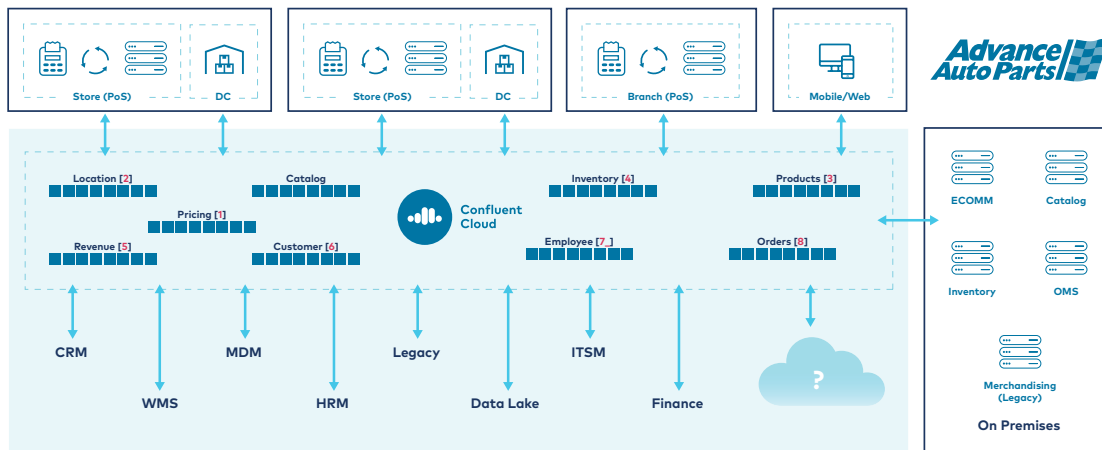
The small IT team at Advance Auto Parts had to manage a maze of different systems and on-prem data infrastructure, all of which were coupled to a large extent, hindering their ability to meet customer needs around real-time invoicing. This dated architecture was also at odds with the "cloud-first" strategy being broadly adopted across their organization.

Confluent sets data in motion to enable a real-time software-driven enterprise

To address their real-time invoicing use case, begin decoupling systems, and to modernize its business-critical merchandising system, Advance Auto Parts built an internal platform with the help of Confluent. The platform integrates and publishes data from various activity hubs set up for finance, student, research, employee, and health data, and subscribers to the platform gain real-time access to data in motion to meet the needs of business operations. As a result, the company paved the way to move from on-prem data warehouses to the cloud.

Using Confluent, Advance Auto Parts also addressed two critical merchandising use cases:

- Real-time invoicing for a large commercial customer by sending data to third-party systems
- Dynamic pricing updates by store and location



Future-proof data architecture for real-time invoicing and dynamic pricing [After]

By integrating data from more than a dozen different sources through Confluent to the company’s ERP supply chain system, and by using ksqlDB, a stream processing database used to build applications for data in motion, events for specific customers could be filtered in real time. The benefits of using ksqlDB also carried over to enriching product data, by connecting to Amazon S3 and Snowflake using Confluent Cloud sink connectors. This has set the stage for continued improvements in operational efficiency and customer experience for years to come.

“One of the first projects we completed with Confluent Cloud enabled customer transactions to be filtered and streamed in real time to support immediate invoicing.”

— Prabh Chandrasekhar, Director of Data and Analytics at Advance Auto Parts

Challenge #3: Low fault tolerance at scale creates performance bottlenecks and operational complexities

Expedia Group, a US-based online travel company, experienced business transformation challenges from batch ETL and REST API-driven processes. While building a conversational chatbot to redefine customer support and experience, Expedia Group needed to have natural language conversations with an automated agent via text, Facebook, or the channel of choice for millions of customers. This automated chatbot would allow customers to easily book trips, make changes or cancellations, create complex itineraries, and ask simple questions such as: “How long is my layover?” “Does my hotel have a pool?” This required bringing together huge volumes of customer data from every system across all of Expedia’s various brands and applying machine learning models to enable natural language processing. In doing so, Expedia Group could automate customer service, giving customers exactly what they’re looking for instantly and with minimal manual intervention.

To quickly make decisions that incorporate contextual information, Expedia Group needed data in near real time from a wide range of services and systems. Batch jobs using ETL, back-end processing, or offline APIs to feed their conversational platform for real-time interactions just didn't cut it. Expedia needed a highly available, persistent, and fault-tolerant messaging stack with blazingly-fast performance that their existing messaging and batch processes simply could not deliver.

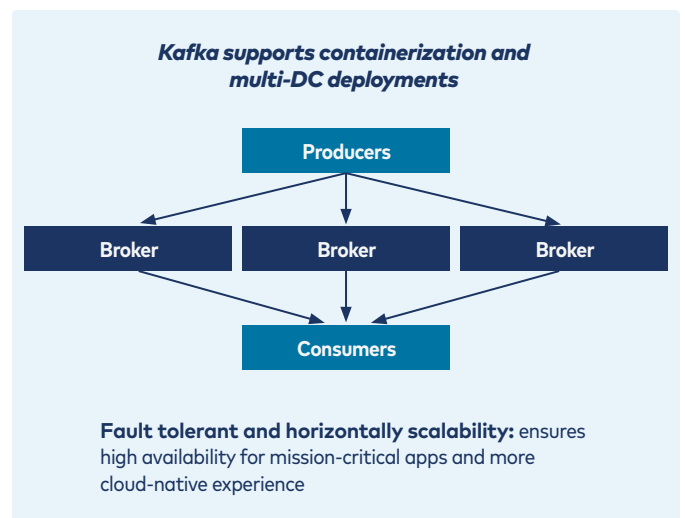
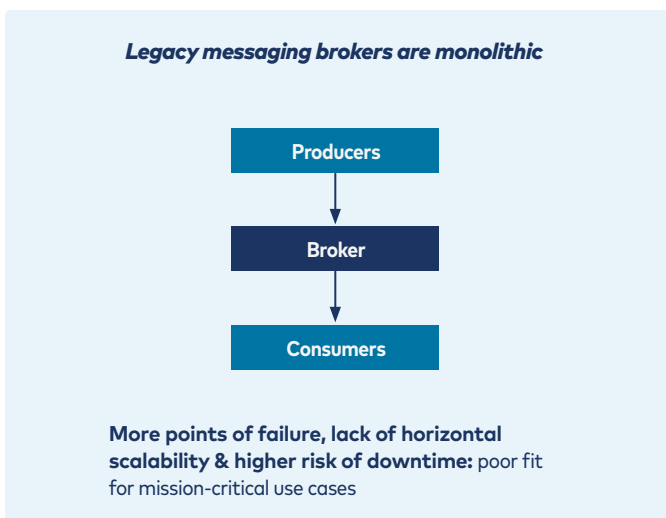
Confluent powers a fault-tolerant platform with elastic scalability and the highest levels of availability and reliability

To feed data to their Conversations Platform in real time, Expedia architected the platform around a central nervous system and event-driven architecture based on Confluent. Confluent made it possible to orchestrate huge volumes of data (millions of messages per second) from decoupled systems and enrich data streams between these systems using ksqlDB, so that data was actionable by the time it reached its destination. Aggregated data could also be surfaced for analytics and reporting. With Confluent, Expedia was able to ramp up and accelerate the pace of innovation to enable context-aware multi-party and multi-channel customer engagement in real time.

"We chose event-driven architecture as the core of our platform, for which we needed a messaging service that gave us all the guarantees... not to mention that it had to be extremely scalable, highly available, and simple to use."

— Ravi Vankamamidi, Director of Technology at Expedia

This type of elastic scalability was underpinned by a highly fault-tolerant platform. Confluent spans multiple data centers and cloud regions for a highly scalable and fault-tolerant architecture. If any of the servers failed, other servers could take over to ensure continuous operations without any data loss. Cluster Linking in Confluent separated the concerns of data storage from data processing. With this separation, it became much easier to scale each cluster independently by offloading a majority of the data to remote stores and reducing the time and cost to rebalance, expand, or shrink clusters. A highly available architecture (backed by 99.95% SLAs) made it possible for developer teams to upgrade, perform hotfixes, and restarts without the system going down.



Confluent made it possible for Expedia to meet its tight launch deadline of four months to production with limited resources. With Confluent's cloud-native solution, Expedia Group needed no costly new hires to maintain clusters and had no worries about 24x7 reliability.

At the time Expedia built the platform, no one could have foreseen a worldwide pandemic and the profound effect it would have on the travel industry. Travel-related companies were hit with a tidal wave of customer questions, cancellations, and rebookings. Throughout this unprecedented global event, the Conversations Platform, powered by Confluent, proved up to the challenge, autoscaling as necessary and taking much of the load off live agents.

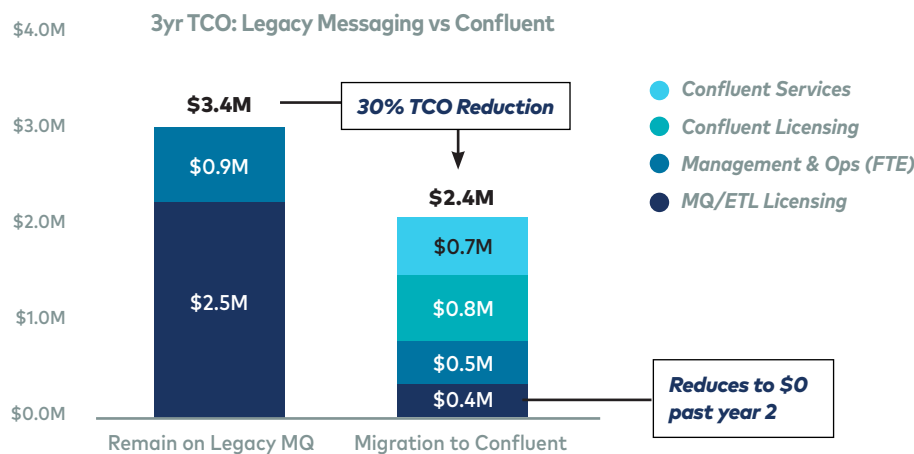
Challenge#4: Legacy middleware leads to increasing technical debt and soaring operational costs

One of the largest banks in North America, with lines of business in retail and other financial services, including mortgages, corporate banking, and insurance, was using messaging middleware systems that consisted of third-party tools including TIBCO and ETL tools such as Ab Initio for crunching data at scale. During an effort to simplify its operational infrastructure and reduce manual processes for data reconciliation, data deduplication, and multiple transformations, the bank realized that its cumulative spend on messaging middleware and data integration tools, including setup and development costs, and ongoing operational expenses, were more than \$8 million dollars cumulatively over three years.

The bank needed a more flexible licensing model with a true cloud-native data integration service that would greatly simplify or eliminate the setup costs it was incurring, and which aligned with the business objectives of building new digital initiatives to serve the bank’s customers.

Modernize workloads incrementally over time and reduce your messaging TCO

With the help of a streamlined messaging and integration platform from Confluent, the bank was able to generate a significant cost savings of up to \$10 million dollars in licenses alone by gradually removing and replacing both their middleware and integration stack with Confluent. Cost savings also came about by reducing the setup costs and operational costs, all of which together meant that the bank could break even within year one and save millions in total costs over a period of three years. When the benefits of addressing additional use cases with a future-proof stack were factored in, these savings jumped by a factor of five. Critical and expensive full-time equivalent employees (FTEs) could now turn their attention to supporting innovation and business initiatives and facilitating developer velocity as opposed to the day-to-day management of messaging infrastructure.



Illustrative TCO reduction from modernizing legacy messaging to Confluent and Kafka

The bottom line is that regardless of how you deploy Confluent—on-premises or in the cloud—you will see significant TCO savings. For most customers who have moved to Confluent Cloud, we have seen that they can [save up to 60%](#) on the cost of managing their infrastructure in comparison to running a self-managed Apache Kafka® service.

Challenge #5: Ephemeral persistence and lack of data context limits use as a system of record

The wealth management division of a top 20 global bank provides investment advice to help clients preserve and grow wealth, customizing their strategy with each client using a combination of equities, options, and mutual funds. However, the bank's proprietary trade order management system was in need of a technology refresh in order to provide greater persistence, and to streamline the way data could be loaded into new application services from a central system of record.

The bank was saddled with a legacy messaging middleware stack: IBM MQ, mainframes, and an ESB that were costly to maintain and cumbersome to access data from. If throughput rose above five MBps, the MQ would crash, occasionally lose messages, and be unable to replay those messages from a central system of record. With increased regulatory pressure, financial institutions often need to track risk, monitor for fraud, and make optimized real-time predictive decisions based on current and past data streams. Messages for a trade order also sometimes arrived out of sequence, which meant that the bank was incurring penalties and paying millions in fines to regulators for not being able to complete trades on time. This also had a significant negative impact due to costs paid to traders to cover the difference in asset price based on when the trade was placed and when it was completed. Furthermore, messages in the IBM MQ system did not persist once downstream applications had consumed these messages, meaning that these systems could not be used as a single source of truth. The bank was also bottlenecked by the ESB system, which required significant man hours to maintain.

Confluent provides a persistent system of record for designing contextual stateful applications

The global bank launched a modernization initiative, powered in part by data in motion with Confluent, enabling teams to deliver faster and with greater autonomy. For the wealth management company, the new platform built on Confluent had to meet the following requirements:

- Process a minimum of 250 orders per second with the ability to scale linearly
- Real-time replication across multiple data centers
- Recovery of messages from a specific point in time (in case of unavailability of external systems)
- Message sequence guarantees for all events related to the same trade order
- Elimination of single points of failure

A multi-region cluster was built across three states for synchronous, real-time replication in an active-active environment. As a result, the IT team at the bank was able to ensure the system could withstand a single data center failure, without message loss. Additionally, Tiered Storage provided a way to store all messages to a central system of record in Confluent. A future business benefit from having this type of central system of record is to provide data context for new stateful applications that will be built as decomposed services.

Confluent also helped ensure data was being delivered to customer-facing services and applications and that these messages were persistent in order (or so called "guaranteed" delivery). "Exactly once" message delivery also freed up client applications and services from the cognitive overhead of deduplicating data. With the ability to persist data and replay messages in order as needed, data is never lost and translates to better customer experiences, for example by letting customers search for all their transactions online instead of just the last six months.

Risk Focus, a financial-technology consultancy and one of our partners, worked closely with Confluent to deliver a high-performance, resilient, and persistent order management system for the client. As a result, the bank now has an application built on top of Confluent's platform for data in motion that can handle orders for its entire line of products in a highly available, and persistent system.

Challenge #6: Slow consumers can degrade performance and eventually lead to outages, requiring costly intervention

Following an acquisition of a digital consumer lending business, one of the largest bank-based financial services (finserv) companies in the US, with assets of approximately \$170 billion, launched a national digital bank initiative focused on doctors, dentists, and other healthcare professionals seeking to refinance student loans and buy homes. This acquisition further compounded the existing number of siloed applications, systems, and point-to-point integrations, slowing down innovation. The bank wanted to modernize this infrastructure and move their on-prem Teradata data marts to Google BigQuery to quickly launch the national digital bank initiative and reduce time to market for new products and applications. The other challenge involved downstream consumers, including on-prem data warehouses that handled periodic large loads of data, that were chronically slow in consuming data from messaging systems and caused messages to accumulate. This degraded the performance of these systems over time, and required costly and manual interventions by operations and infrastructure teams in the case of outages. The more these data warehouses fell behind, the slower the messaging system performed and as a result they lagged further and further behind. In traditional messaging queues, producers and consumers are really coupled from a performance perspective.

Confluent eliminates slowdown and outages due to slow or offline consumers

Confluent's platform to set data in motion underpins the bank's national digital bank initiative and supports its cloud migration and connection to cloud-based applications. With Confluent, the bank has created a center of excellence to help teams throughout the bank. Confluent's complete, cloud-native service is helping fuel the bank's commitment to remaining a technology innovator in banking. The results include:

- **Accelerated time to market:** Integration between systems requires significantly less effort now that data has been made available via Confluent's platform, and time-to-market estimates for developing new customer-facing apps have been significantly reduced.
- **Reduced costs:** The bank will reduce its mainframe and legacy message queue costs as it implements more use cases on Confluent.
- **Improved data pipeline for online and slow or offline consumers/applications:** Data production and consumption rates have no effect on the performance of the system.

"As we implement more use cases on Confluent, we will reduce our mainframe and legacy message queue costs. We expect to see a measurable reduction in mainframe MIPS, IBM MQ usage, and in our use of ETL process software."

— SVP and Chief Architect at the Bank

Taking a step back, all these organizations have one common hurdle: their legacy data infrastructure stack is built on top of messaging systems like ESBs, MQs, and ETL systems. While there is nothing inherently wrong with these middleware and data integration technologies, they were designed and implemented at a time when monolithic applications were the norm and the phrase "the cloud" made the average person on the street look up at the sky.

How do you ensure the investments you're pouring into these middleware systems are preparing you for your journey to the cloud? How do you begin to modernize the workloads running on these systems and reduce your technical debt while maximizing the use of your FTEs? How do you use a hybrid and multi-cloud messaging architecture to iterate at speed in the cloud while enabling communication across network boundaries?

Why you need a platform for data in motion

Having gotten this far in this paper, odds are that you have very similar challenges in your organization. We want to help you keep the investments you've made in your middleware systems while building a path to modernize your workloads over time, build an event-driven architecture on top of microservices, and move to the cloud. Confluent is pioneering a fundamentally new category of data infrastructure focused on data in motion, to be the intelligent connective tissue enabling real-time data, from multiple sources, to constantly stream across your organization.

Confluent's platform to set data in motion, built on the foundation of Apache Kafka, is a cloud-native offering designed to collect and distribute messages for distributed systems for high-throughput read/write operations at extremely low latency, processing millions of messages per second. Messages are persistent, idempotent, replayed from any point in time, and can dynamically scale to enable elastic on-demand processing on spikes of message traffic or changing application demands. It offers a flexible fault-tolerant design whereby partitions can be replicated to multiple regions or zones to provide high availability and continuous operations.

You can remove operational complexity with a decoupled event-driven infrastructure so your developers can focus on building business logic faster with a rich ecosystem of connectors and sophisticated stream processing. Lastly, you can migrate at your own pace to a cloud-first architecture.

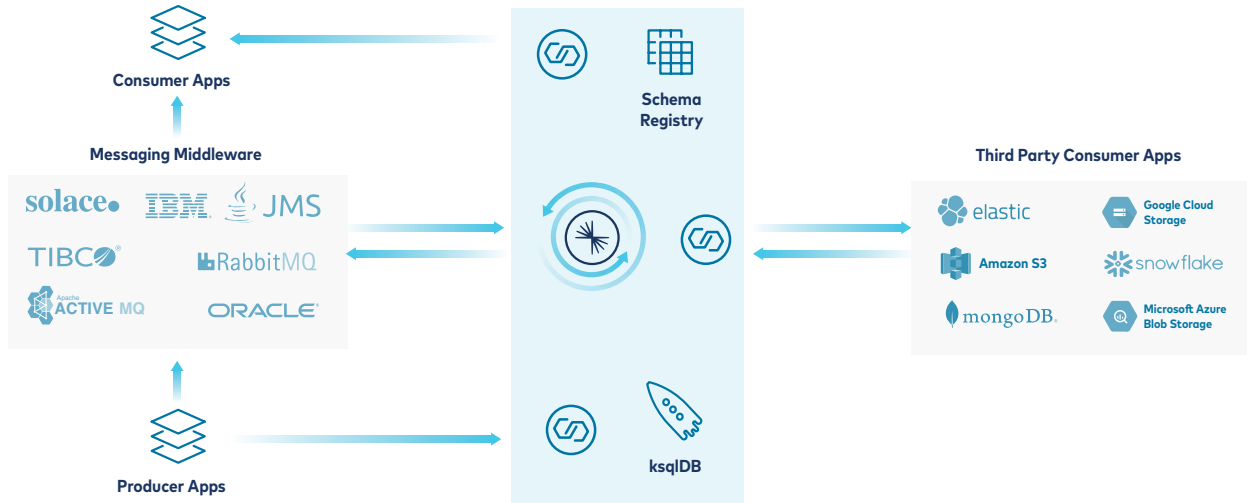
Why Confluent is unlike any other messaging system

As demonstrated by the customer examples provided above, there is clear value in investing in a platform designed to support today's data in motion. No matter the industry, augmenting or modernizing your existing messaging-oriented middleware with a platform for data in motion will help you deliver your key business initiatives. The following features make Confluent the platform of choice:

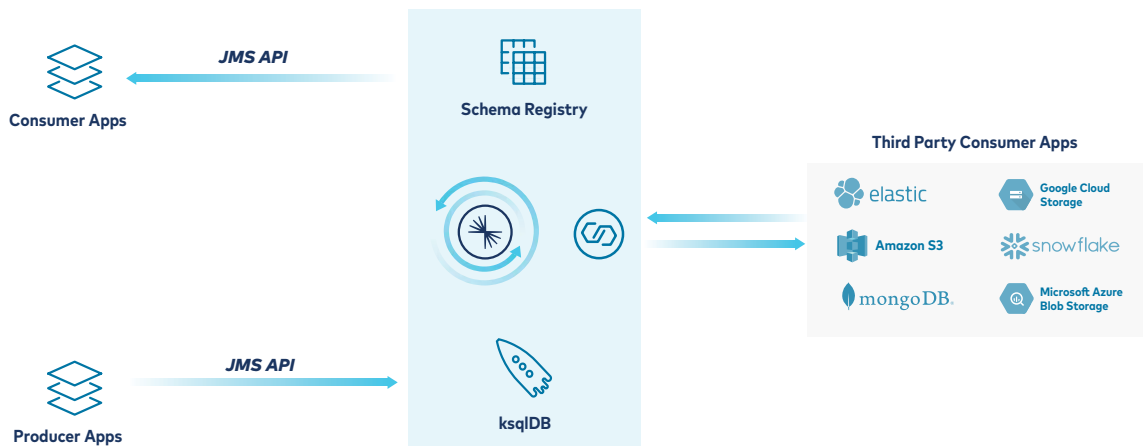
- **Cloud native:** We offer a fully managed, cloud-native service so you can provision Confluent clusters on demand that scale elastically between 0 to 100 MBps or get you to GBps+ scale with a few clicks. Instant elasticity like this means you can dynamically scale clusters on demand, on spikes of message traffic, or changing application demands. You pay for what you need, when you need it, and no more. These clusters are also fault tolerant and highly available, always updated to the latest version, and backed by a 99.95% SLA and Confluent's committer-led Kafka expertise. This is a completely different experience than what would result from taking an on-prem messaging middleware application or a Kafka service and simply offering it on cloud virtual machines. From a cost perspective, you can save \$3 million to \$8 million dollars by migrating from on-prem ETL, EMS, and MQ tooling to Confluent, as experienced by many of the customers we've helped. If you're currently self-managing a Kafka service on prem, you could save up to [60% in lowered TCO by migrating to our cloud service](#).
- **Complete:** Confluent delivers a combined infrastructure layer for data integration and messaging versus the use of multiple vendors.. You pay only for the storage you use, yet storage potential is unlimited—meaning that messages can be stored and played back from any point in time. You can continuously process real-time streams using ksqlDB to build applications for data in motion and perform high-throughput read/write operations for millions of messages per second at extremely low latency ([15x faster than RabbitMQ and 2x faster than Pulsar](#)). You have access to a rich, pre-built, open source Kafka ecosystem of over 120+ connectors, including specific connectors for your existing MQ and EMS systems such as IBM MQ and TIBCO. This allows your developer teams to quickly unlock the data from your existing on-prem messaging workloads and feed it to third-party applications on the cloud (e.g., Snowflake and MongoDB) to power hybrid cloud use cases. We also support the most widely used enterprise messaging patterns, including point-to-point and publish/subscribe routing semantics and exactly-once delivery semantics.
- **Everywhere:** You can easily build multi-data center, multi-cluster, and hybrid cloud deployments and create a highly available, durable, and distributed global data fabric by leveraging fully managed multi-region cluster linking in Confluent. Your teams can stay agile and work in the language of their choice such as Java, C/C++, Go, .NET, and Python—or use the Confluent REST Proxy. Plus, Confluent is available across all major cloud marketplaces (AWS, Azure, GCP), so your team can use your existing cloud provider credentials to leverage Kafka and quickly scale it up.

How to modernize

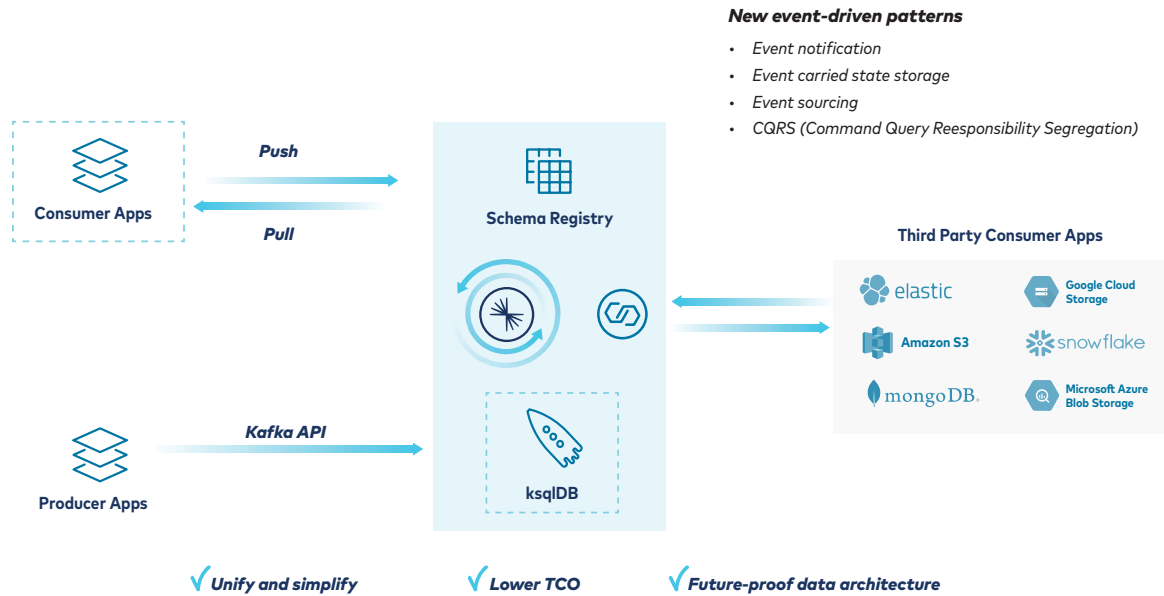
There are multiple phases in which enterprises can begin modernizing messaging workloads and unlock the value of data that has always been locked away in legacy applications.



Phase 1 – Power hybrid cloud use cases by unlocking legacy data with connectors: First, replace your data integration layer with Confluent. Keep your existing on-prem messaging middleware in place and use fully managed Confluent connectors to feed the data from IBM MQ, TIBCO, HiveMQ, ActiveMQ, and RabbitMQ to third-party applications (MongoDB, Snowflake, etc.) on the cloud to immediately unlock new use cases with a hybrid cloud messaging and integration infrastructure. tantly, set themselves up for intra-day reporting by continuously sourcing data directly from the upstream trading systems using Confluent. The scalability, durability, and resiliency of Confluent have set the bank up to add additional data sources to enable regulatory reporting for other asset classes, and new analytics use cases.



Phase 2 – Lift and shift to the cloud: Refactor your JMS-based consumer and producer applications to point to Confluent with configuration-only code changes and begin replacing messaging middleware in a phased, incremental way. All the data you write will land in Kafka topics, but as far as your applications are concerned, they are talking to JMS queues and topics. You can also combine the most common enterprise messaging patterns (such as point to point or publish/subscribe routing semantics), with exactly-once delivery semantics from Confluent.



Phase 3 – Create streaming applications by leveraging new event-driven patterns: Shift off JMS completely to a cloud-native messaging and integration cloud service. Create streaming applications with new and complex event-driven patterns using ksqlDB for event notifications (for interservice notifications or adding new decoupled services on your message bus), event-carried state storage (independent scaling of services while subscribing to same event), event sourcing (replay events and create materialized views), and CQRS (command query responsibility segregation for read-only materialized views). [This e-book](#) on designing event-driven systems goes into more detail on these patterns.

Want to learn more?

Organizations today face similar challenges with their existing messaging middleware systems. With a deeper understanding of the drawbacks of these systems and the impact they have on your business, a modern platform for data in motion is the answer you may be looking for. There are several ways you can start your journey:

1. If you want to get a sense of what working with a cloud-native data-in-motion platform looks like and what your development teams could achieve, [try Confluent Cloud for free](#).
2. Check out our related whitepapers:
 - For a deeper technical dive, download the whitepaper on "[How Kafka is different from traditional middleware systems](#)."
 - To learn how easily you can integrate with existing messaging systems (Phase 1 of the modernization journey described above), download the whitepaper and check out the demo on "[How Confluent can integrate with existing messaging systems](#)."
3. [Contact Confluent](#) to discuss your particular use case.