

IBM MQ / Kafka



Hervé Grange
hgrange@fr.ibm.com



IBM MQ | Best-in-class Enterprise Messaging



The world depends on reliable, secure messaging and [85% of the fortune 100 depend on IBM MQ](#):

- 98 of the top 100 global Banks using IBM MQ
- 8 of the top 10 global Manufacturers
- 9 of the top 10 global Healthcare providers
- 6 of the top 10 global Retailers
- 9 of the world's top 10 global Airlines
- 9 of the top 10 global Insurers
- 9 of the top 10 global IT Services companies

“Financial institutions use MQ heavily, it’s very good at what it does; MQ is by far the biggest product from the messaging space”

- [The Bank of New York Mellon](#)

Case Study: BNY Mellon uses IBM MQ to ensure that highly-sensitive, transactional data is delivered every time.

Business Challenge

As a large bank, BNY Mellon has a significant number of transactional messages that need to be sent every day. They need to be sure that every message will be delivered, and they want to be able to use a single system across all their platforms.

Transformation

BNY Mellon uses IBM MQ because it's good at what it does. They know that their transactional data won't get lost, and that IBM MQ will work across whatever platforms they use. BNY Mellon also takes advantage of the IBM experts available and uses IBM as a sounding board.

“Financial institutions use MQ heavily, it’s very good at what it does; MQ is by far the biggest product from the messaging space”

- Paul Hatcher, Head of Enterprise Messaging, [The Bank of New York Mellon](https://www.ibm.com/case-studies/the-bank-of-new-york-mellon-).



<https://www.ibm.com/case-studies/the-bank-of-new-york-mellon->



Promising a high level of service at a low cost, this North American airline launched its service in 1996 with just three planes. Now, it fills more than 100 planes, carrying 50,000 guests a day on more than 450 flights to 85 destinations throughout North America, Central America and the Caribbean.

Challenges

- A North American airline wanted to extend its thoughtful customer service, especially during flight cancellations or delays, by sending customers personalized, rules-generated e-mail or text messages.

Solutions

- A rules-based travel notification system uses IBM Operational Decision Manager, IBM Integration Bus, IBM MQ, IBM WebSphere DataPower and IBM Tivoli Directory Integrator.

Benefits

- With millions of messages sent, the travel experience is improved, passenger traffic is up 7.8%, 60% of passengers click for more information and just 0.5% have opted out.

Transportation | Customer Care

Customer Care: Respond to events in real time



Required: Events

Need:

Improve overall customer experience when unexpected delays or cancellations occur.

Solution:

Leverage event streams and messaging integration patterns to automatically rebook customers, extend travel vouchers, book/rebook hotels and cars, etc.

Integrations:

- Streaming and aggregation of weather events
- Connectivity to airline reservation systems (backend)
- Third party integration with hybrid multi-cloud vendors & systems

Business Outcome:

- Reduced cost of re-accommodating customers by 10%
- Reduced the time of re-routing disrupted passengers by 50%
- Increased airline Net Promoter Score

Retail | Personalized in-store experience

Mobile: Deliver responsive in-store customer experiences



Required: API management + App Integration + Events

Need:

Provide exceptional in-store experience leveraging personalized offers and VIP treatment.

Solution:

Use in-store activity (events) and geo-position in combination with existing customer data to help them find products, offer customized promotions and complete purchases.

Integrations:

- Secured integration from mobile device
- Data aggregation from a variety of systems (order management, CRM, inventory)

Business Outcome:

- Increase in customer loyalty and repeat purchasers, leading to overall revenue increases
- Improvement in inventory management enabled by better buyer interactions

Messages or Events?

Messages† are “work that needs to be done”

Events are “things that have happened”

Generalised solution



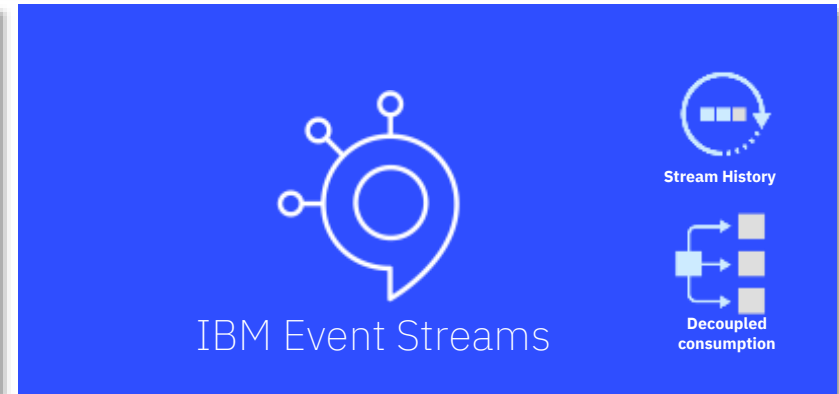
This panel illustrates a generalised solution for messages. It features the IBM MQ logo (a stylized 'Q' with a dot) on the left and a white Swiss Army knife icon on the right, symbolizing a multi-purpose, generalist tool. The text 'IBM MQ' is centered between the two icons.

Specialised technology



This panel shows specialised technology for messages. It includes the IBM MQ logo on the left. On the right, there are two icons: a 'Request / Reply' icon showing two boxes with arrows between them, and an 'Assured Delivery' icon showing a checkmark inside a square box.

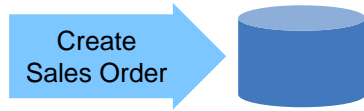
Specialised for **message exchange** and **transactions**



This panel shows specialised technology for events. It features the IBM Event Streams logo (a stylized 'E' with a dot) on the left. On the right, there are two icons: a 'Stream History' icon showing a circular arrow with a bar chart inside, and a 'Decoupled consumption' icon showing a square box with arrows pointing to three other square boxes.

Specialised for **streaming** of **events**

Commands vs Events



“**Create** Sales Order” is a *command* to do an action in the *future*

Commands are **directed to known components** (although their location may be unknown)

They typically use a **point to point pattern**

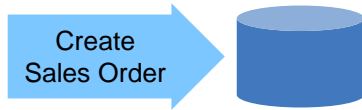


“Sales Order **Created**” is an *event* describing an action that has *occurred*

Events are created with little or **no knowledge of who will consume** them

Events typically use a **publish/subscribe pattern**

Commands vs Events



“**Create Sales Order**” is a *command* to do an action in the future

- We can guarantee that a command will be delivered, but there is no guaranteed that it will happen
- Commands are **directed to known components** (although their location may be unknown)
- Commands *sometimes* require a reply, or at least an acknowledgement
- Commands are often *very* sensitive to lost or duplicated data
- The timestamp is not fundamental, but it can be necessary for example for request expiration
- Commands are typically transmitted using messaging such as IBM MQ. Kafka is not well suited to transmission of commands.

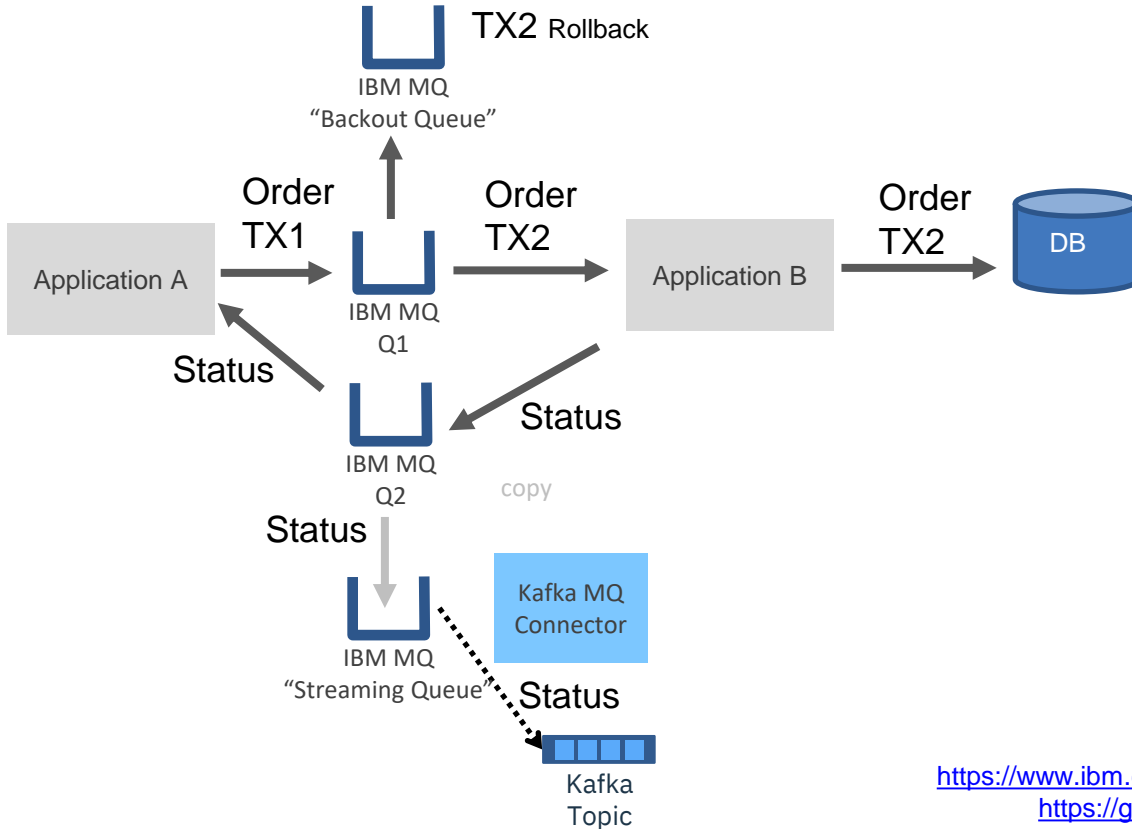


“**Sales Order Created**” is an *event* describing an action that has occurred

- Events record something happening, so they are immutable – the event can’t “unhappen”
- Events are created with little or no knowledge of who will consume them
- Events are typically propagated using a **publish/subscribe pattern**
- Events are *sometimes* less sensitive to lost or duplicated data
- An event occurs at a specific time – its timestamp is a fundamental part of its data
- Events can be transmitted over messaging (e.g. IBM MQ pub/sub) or via an event log (e.g. Apache Kafka based implementations)

Using Kafka and MQ together

Queues and event logs are complementary rather than competing



IBM MQ “Backout Queue”

- created via simple configuration
- automatically receives the rollbacked message after n retries
- Can be moved back, when stable again

IBM MQ “Streaming Queue”

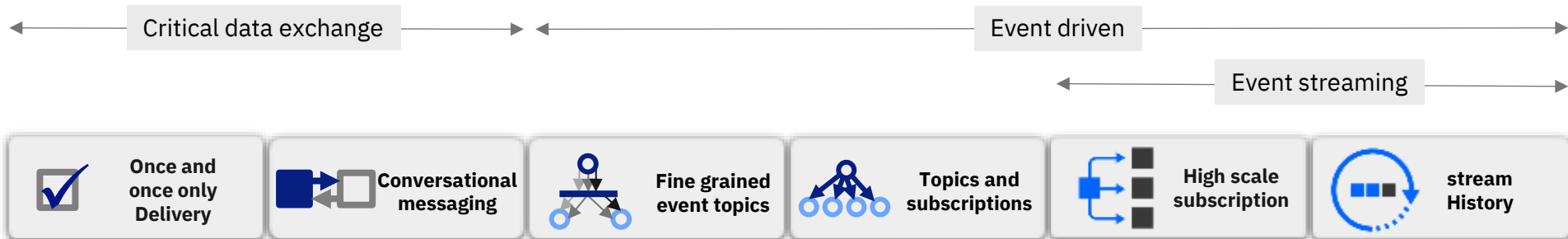
- automatically receives a copy of every message
- does not require required to the applications
- consumed just like any other queue
- created via simple configuration

MQ Source Connector

- Durably passes MQ messages from MQ to Kafka
- Based on the open-source Kafka Connect Framework
- Available with IBM MQ advanced or Event Automation

<https://www.ibm.com/docs/en/ibm-mq/9.3?topic=scenarios-streaming-queues>
<https://github.com/ibm-messaging/kafka-connect-mq-source>

Comparing messaging and event streams



Messaging (queue-based)

Focused on message exchange and transactions

Supported
connectivity

Event Streams (e.g. Kafka)

Focused on streaming of events

The deciding factor between the two is rarely about non-functionals. Both are highly available, performant, scalable and secure etc. The relevant characteristics are much more likely to be about their differences in behavior.

<https://developer.ibm.com/messaging/2018/05/18/comparing-messaging-event-streaming-use-cases>

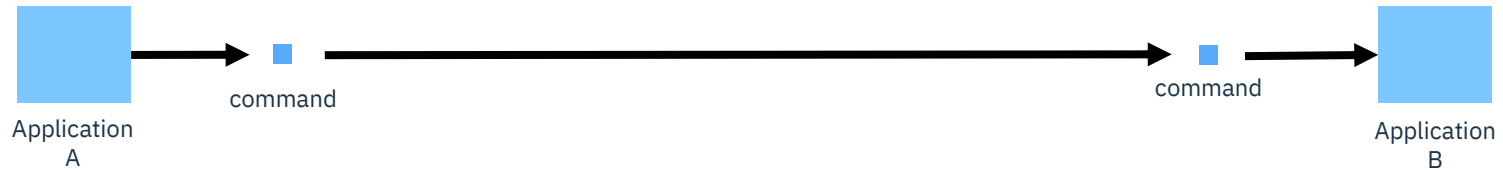
<https://techcon2021.ibm.com/agenda/session/578147>

<http://ibm.biz/eda-resurgence-blog>

Point to point messaging pattern for **commands**

Application A requests an action (a command) to be performed.

The messaging network ensures the message is transported securely and durably to the right place



Delivery assurance of **once and once only** is often imperative
Commands are typically **read and processed independently**
Sometimes a **response message** is also transmitted back from B to A

Commands are suited to queue-based transport, and this is the **most common use case for IBM MQ**

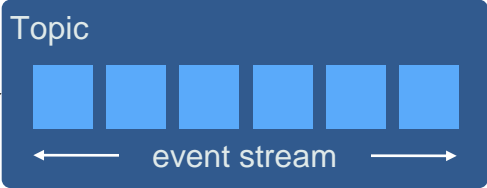
This is **NOT** a good use case for Kafka

“Publish/subscribe” pattern for events

A “producer” application **publishes** related events to a *topic* when there are changes to something it owns (e.g. data, or a sensor etc.)



publish



An “**event stream**” is sometimes described as an “**event log**”, especially when it is persisted to disk.

“consumer” applications **subscribe** to a *topic* in order to receive events

subscribe

consume



consume

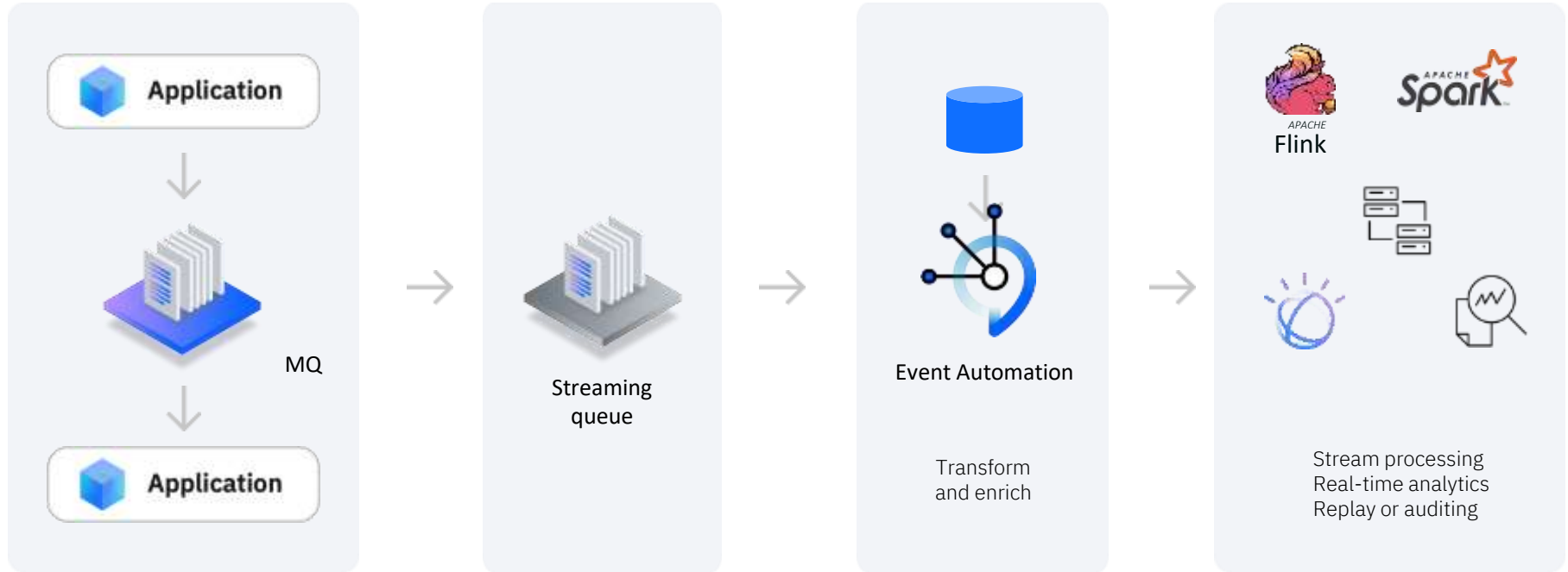


consume

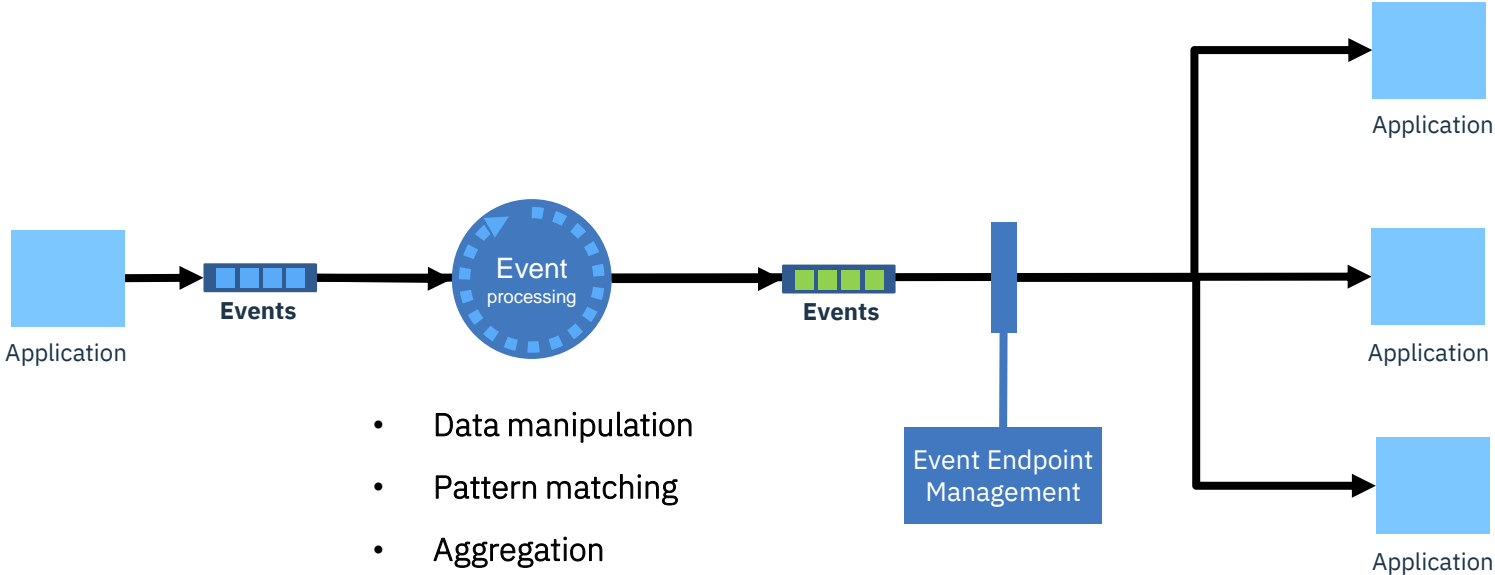


Provide high quality data to event processing (Transform & Enrich data)

The data format on a streaming queue be very application specific. Use an integration runtime such as IBM App Connect to transform the data format and enrich the data from other sources to make it more accessible and valuable for stream processing.

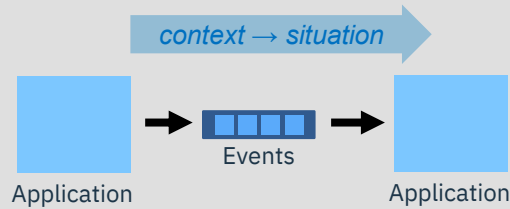


Example: Making the events more valuable with event processing



Next weeks session: EDA usage patterns

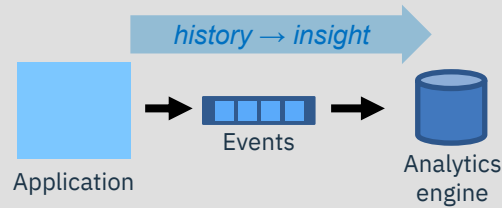
Event Notification



*“I want to be **informed immediately** when something happens”*

*Advanced: Relate events to their **business context***

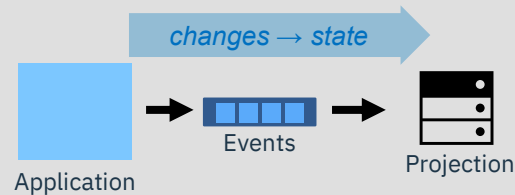
Event Analytics



*“I want to perform **continuous analysis** as changes occur”*

*Advanced: Perform **stream-based analytics***

Event Projections



*“I want a **local copy** of backend data that is kept current”*

*Advanced: Enable **distributed event projections***

Questions?

