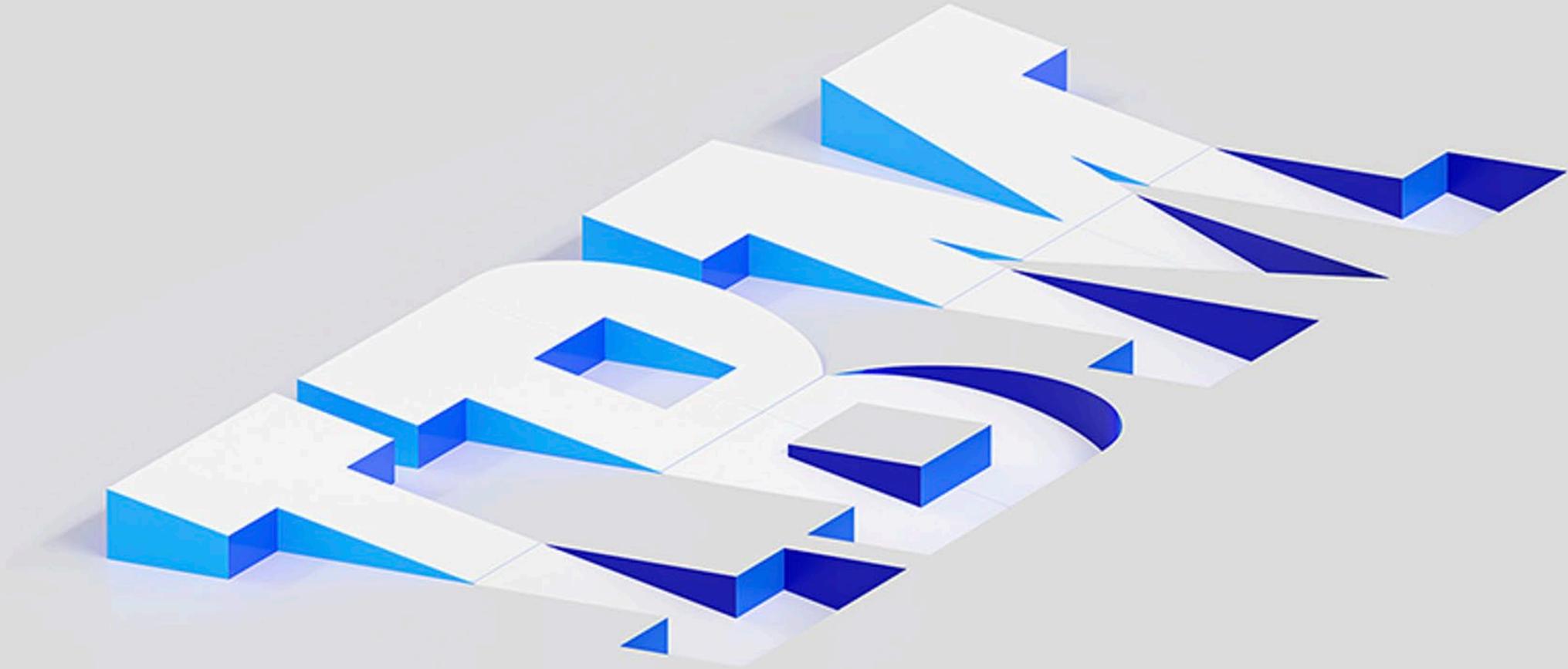


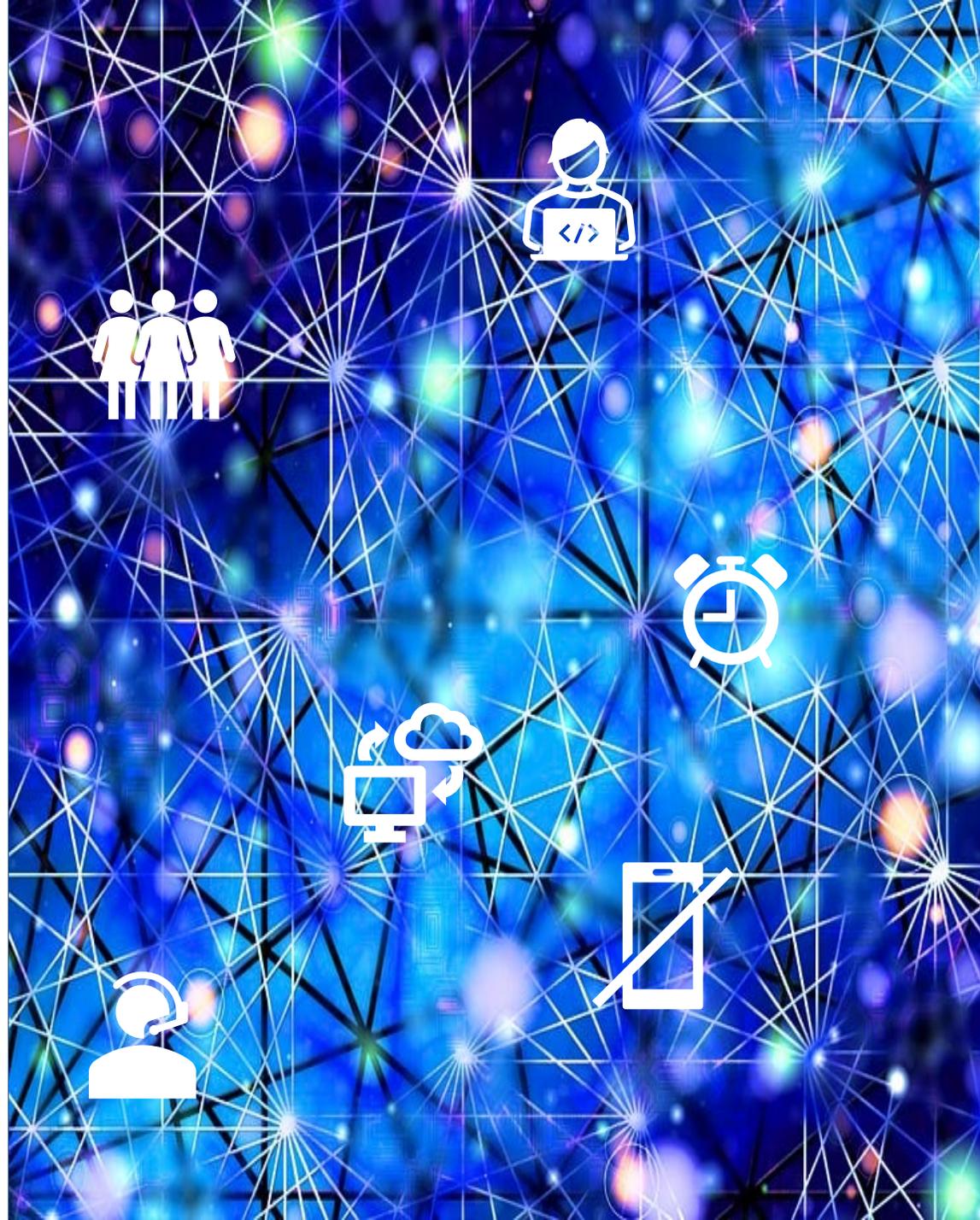
The future of the enterprise is event-driven



Matt Sunley
Program Director, Product Management, IBM Integration
sunleym@us.ibm.com



If you knew **what** happened
in your business **in the last**
5 minutes, what would you
do differently?



Capitalize on opportunities and address threats **in real-time**

Maximize
Sales



How can you build a 360-degree customer view to present the right offer as they are **in the store**?



How do you optimize your supply chain to meet changes in demand or costs **right now**?

Resilient
Operations

Customer
Protection



How do you detect orders outside normal buying patterns **before they are processed**?



How can you respond to disruption and adjust passenger travel plans **as it happens**?

Responsive
Experiences

Event-driven fuels the autonomous enterprise



Collect key business events as they happen

Get events from their source to wherever they're needed.

Allow applications to listen to dynamic rather than static data.



Derive continuous intelligence and insight

Constantly evaluate the impact of events in combination with their related context.

Allow applications to gain critical insights whilst an event is at its most relevant.



Automate to improve business outcomes

Make smarter decisions and automate actions when it matters most.

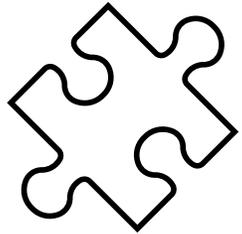
Allow applications to take the next best action immediately when important events occur.

The Event-Driven Business Challenge



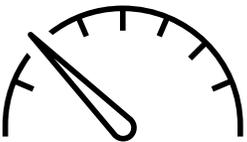
Event Overload

Today, companies juggle 10,000+ business events each day. These business events, however, remain siloed in various data sources.



Lack of Event Connectivity

With no efficient way to connect the dots between countless business events, you lose out on the opportunity to efficiently identify key patterns emerging.

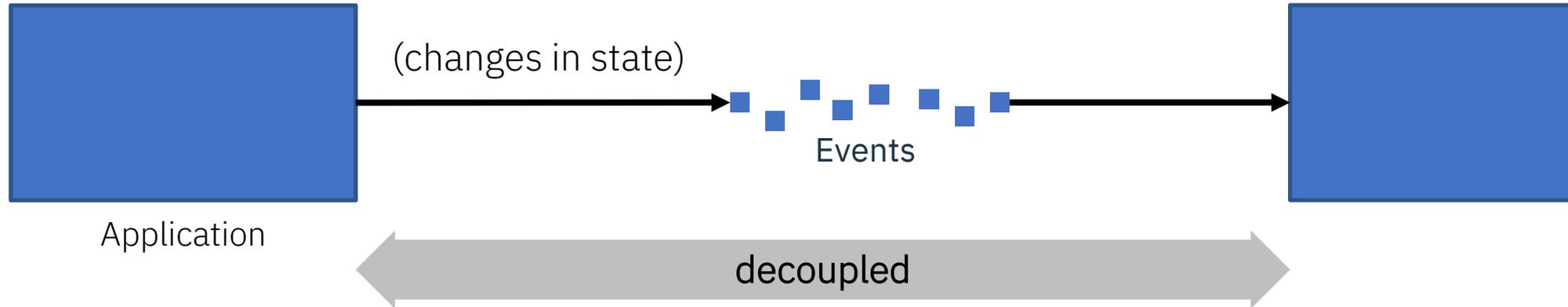


Slow Response to Business Dynamics

The longer it takes to identify key patterns, the more delayed your action will be. This could cause your business to suffer unnecessary financial, customer and overall reputational loss.

Event Driven Architecture (EDA)

Focuses on “changes in state”, rather than “current state”

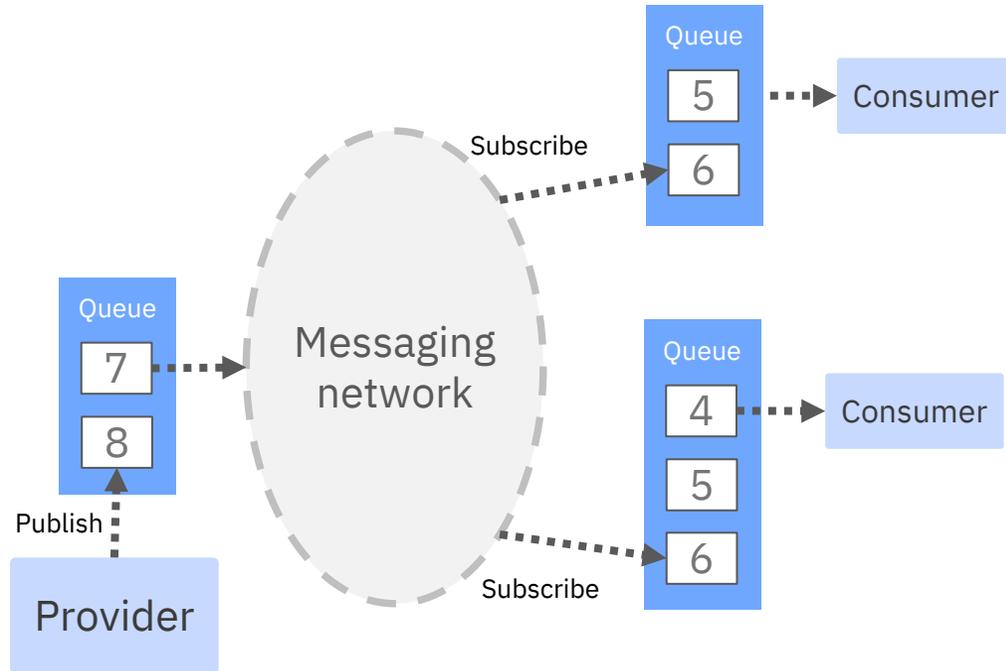


EDA is not new! Conceptually EDA has been present in computing since the very beginning. It can be underpinned by any mechanism that enables event data to be stored and read at independent times. Queues (e.g. MQ), logs (e.g. Kafka), database tables, files etc.

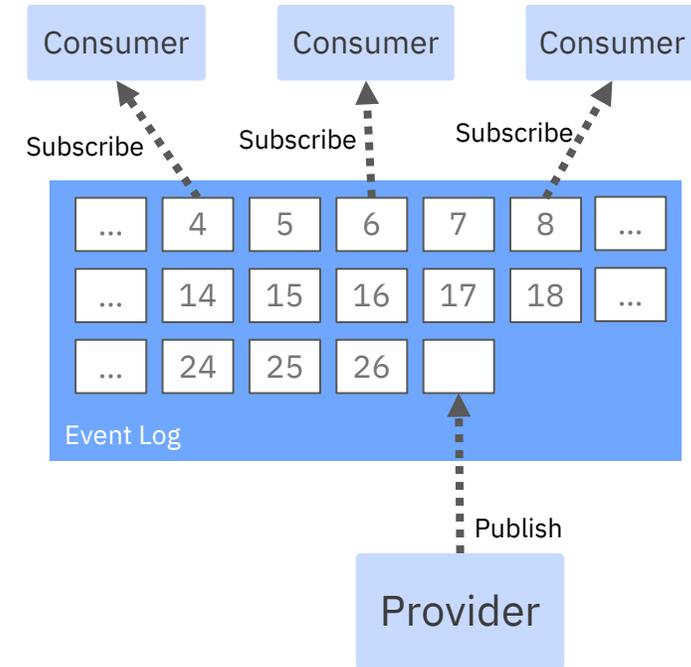
EDA was originally primarily about **decoupling components** to enable improving change agility, performance optimization, resilience and more.

Modern day EDA moves well beyond this, increasingly making use of **event processing** to gain insight directly from the event streams rather than just processing them at the target system.

Event Distribution Layer



The publish/subscribe pattern is decades old and is one of many interaction patterns implemented by message queue-based technologies such as IBM MQ.



Apache Kafka uses only the publish/subscribe pattern, using a persisted “event log”.

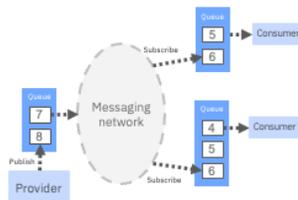
How has the use of the term EDA change over time?

Initially EDA was mostly focused on passing events via an asynchronous “**messaging**” transports such as IBM MQ. This style is still extremely pervasive, with new use cases cropping up regularly.

- Events in this style are **transitory** and primarily a way of moving data from one place to another.
- Much emphasis is placed on the ability to deliver data with high levels of assurance (**once only**) and security.
- Messaging often specializes in simplifying communication across **disparate platforms** and environments.

Messaging is used primarily as a transport mechanism.

Messaging is well suited to both events, and commands.

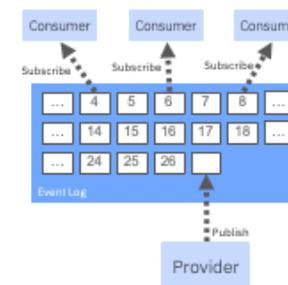


Apache Kafka popularized the idea of an “**event log**” where the events are retained even after being read. This enables a different set of patterns that make use of this event history:

- The retained event log enables alternative ways to create the **stateless components** favored by cloud native application designs.
- There is an increased interest in **stream processing** such as rolling accumulations over a window of time, or complex event processing to derive insight more rapidly.
- Use cases for historical **replay** of events such as testing, simulations, training AI models, auditing and more.

The event log can be seen as an alternative choice for persistence.

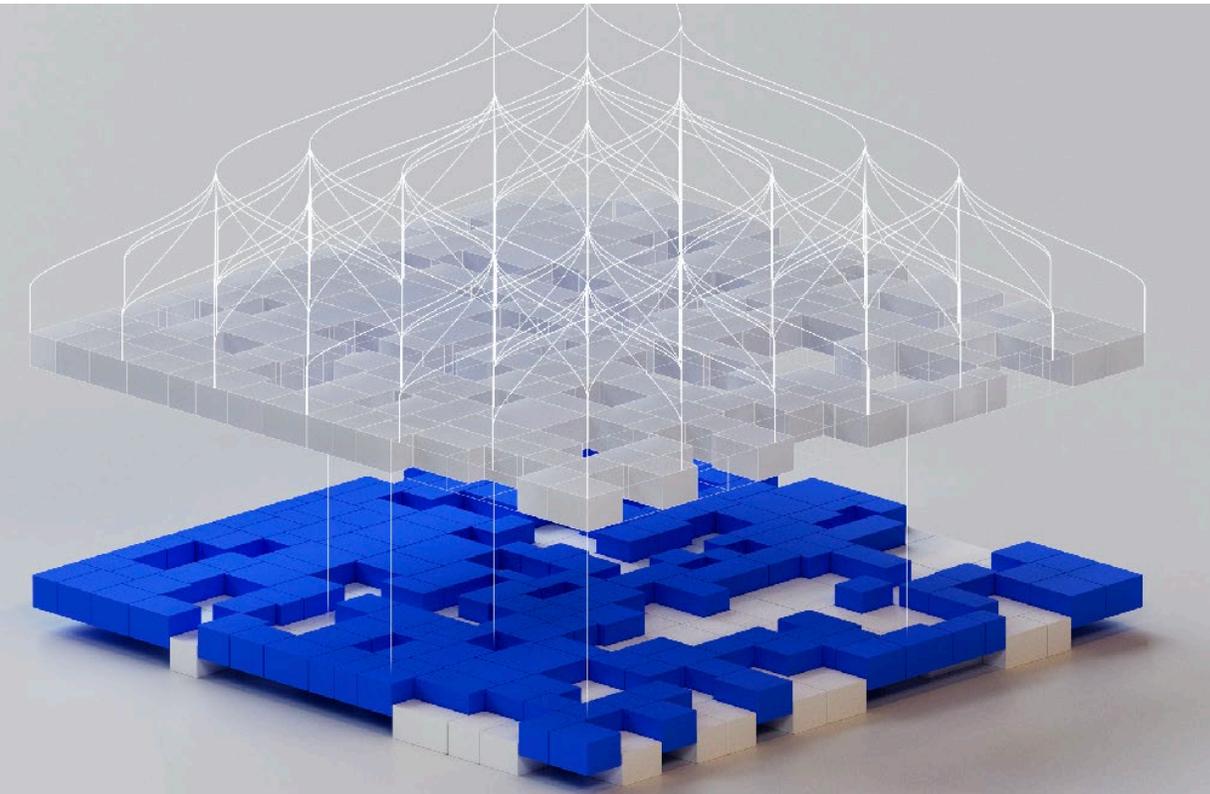
An event log is well suited to events, but less so for commands.



The event-driven market
is rapidly growing

IDC views event-driven software
as “one of the fastest-growing
segments of *Intelligent Process
Automation*”

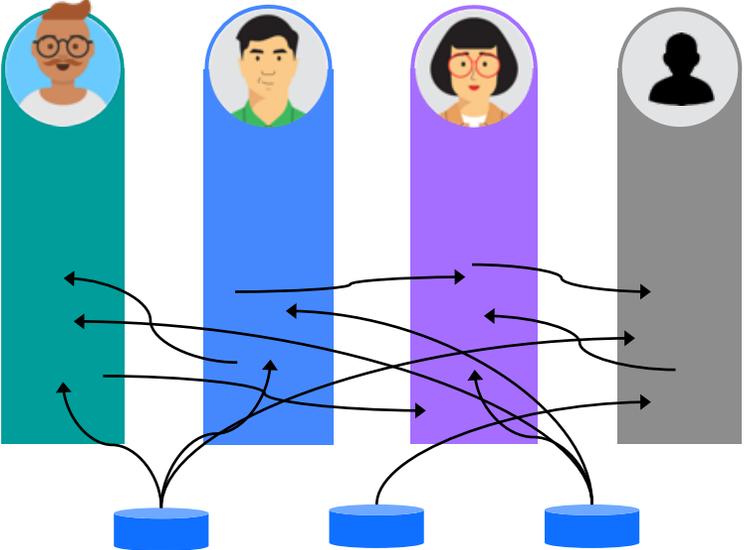
>20% CAGR
to \$7.0 billion by 2026



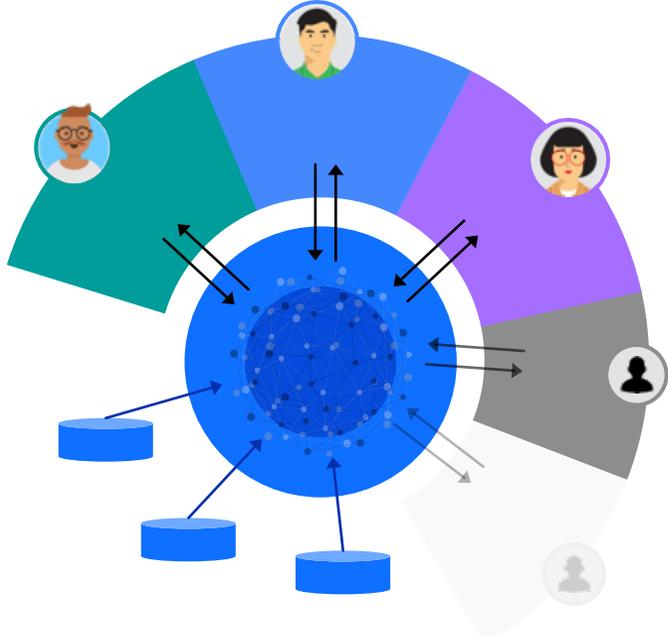
IBM's Point of View

From Individual Projects to Common Foundation

Siloed use of events, within individual projects, across different domains



Events available across the enterprise to all who need them

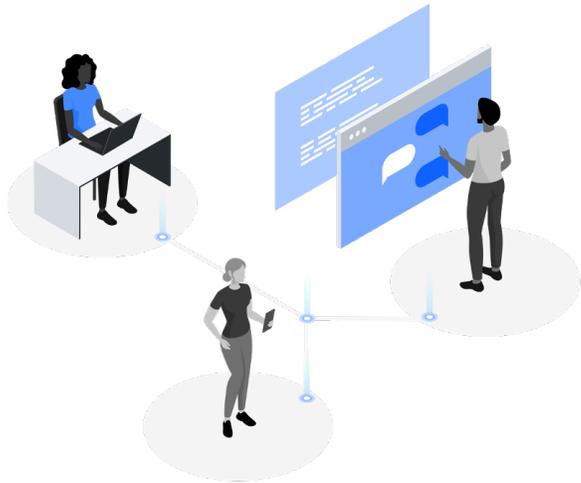


- Brittle point to point connections
- Difficult to manage
- Slow to create new integrations
- Inefficient; same events moved multiple times
- Impossible to track lineage
- Exponential scaling problem

- ✓ Clearly defined and versioned interfaces
- ✓ Control points for enterprise policies
- ✓ Self-service; increasing agility
- ✓ Efficient data movement
- ✓ Clear visibility of lineage
- ✓ Scale linearly

IBM Event Integration Strategy

Provide a fully composable set of capabilities to enable users to easily work with events and benefit from a consistent architecture, allowing organizations to prioritize responsiveness and adaptability to their clients.



Event Distribution

Built on open technologies, an **enterprise-grade packaging of Apache Kafka** with value-add services to assist connectivity and deployment across organizations at scale.

Event Discovery

Enable existing events to be **discovered and consumed by any user**, and manage your event sources like APIs to securely reuse them across the enterprise.

Event Processing

Make working with events intuitive and simple so users can understand and react to real-time business situations. **Remove the need to write SQL** and increase productivity.



Event Distribution

IBM Event Streams is fully supported Apache Kafka® with value-add capabilities

IBM Event Streams

Award Winning User Experience

Operator Ops Tooling - Strimzi

Schema Registry - APICurio

Geo-replication – MM2

Connector Catalog

Unrivalled MQ connectivity

24 x7 Support

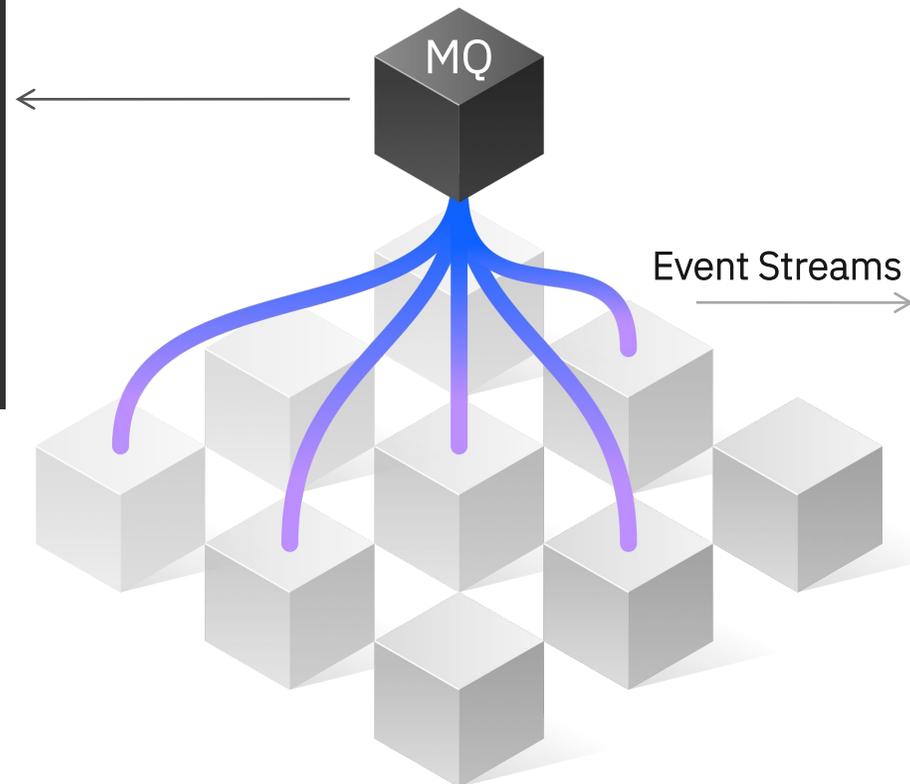


MQ can Drive Situation Detection and Innovative Applications

MQ is a key source of real-time events representing the transactions, changes and interactions occurring in a business.

Tapping into MQ for Event Streaming enables sharing across the enterprise and new responsive applications.

- Stock movements
- In store transactions
- Online orders
- Supply chain interactions
- Customer details



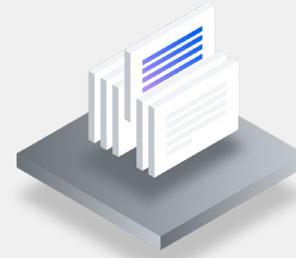
- In-store reservation confirmation text message
- Online “collect in store” order triggers real-time offer
- Suspicious return behavior triggers additional checks
- Ad-hoc “personal shopper” experience based on availability
- A “did you forget” checkout suggestion service
Personalized offers based on order history
- Optimized inventory “rolling stock checks”
- Real-time feedback for “in-store collection” offers

Event Discovery

Event Endpoint Management accelerates the implementation of event-driven and situational applications by making the events that drive them accessible to everyone

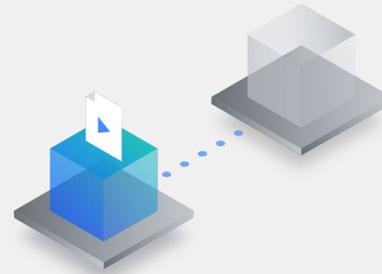
Providing a common management facility where streams of events can be:

- Described in a standardized way using AsyncAPI
- Published in a searchable portal
- Advertised for others to gain self-service access based on the applied policies



Event Distribution

- Described
- Discoverable
- Decentralized
- Decoupled



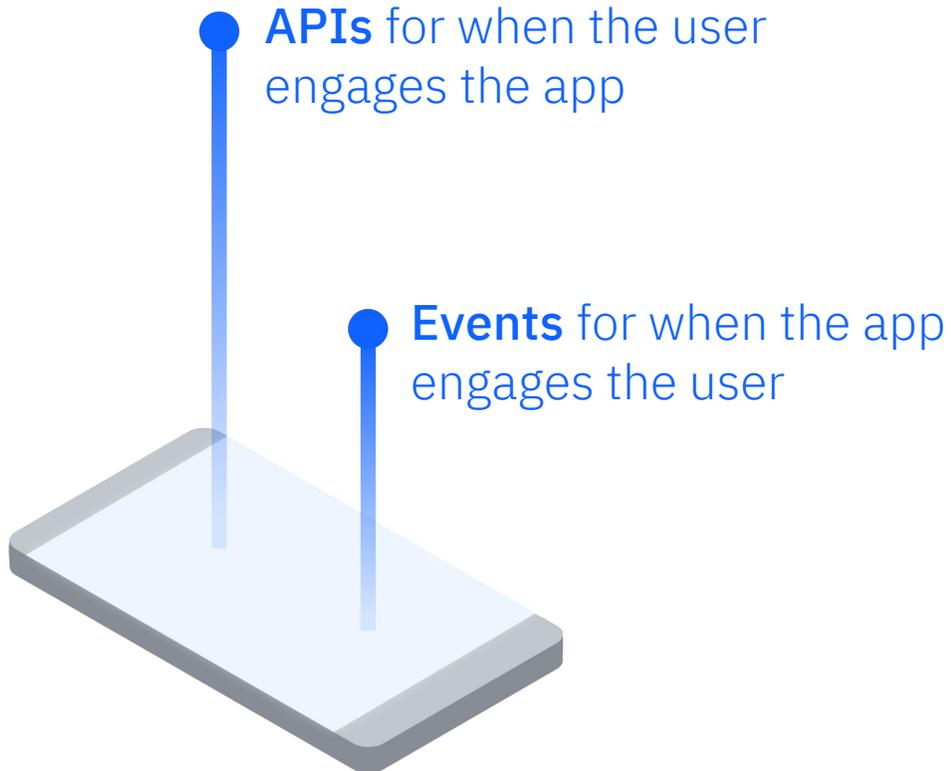
Responsive experiences



Engaging and Intelligent apps

Building great user experiences requires both APIs and Events

..but uncontrolled use of events suffer the same pain points addressed by API management



Access control

How to ensure that only the right users can access important enterprise data?

Workload management

How to prevent disrupting existing work and critical systems being swamped?

Enterprise reuse

How to share events for multiple use cases, maximizing reuse and avoid duplication?

Agile development

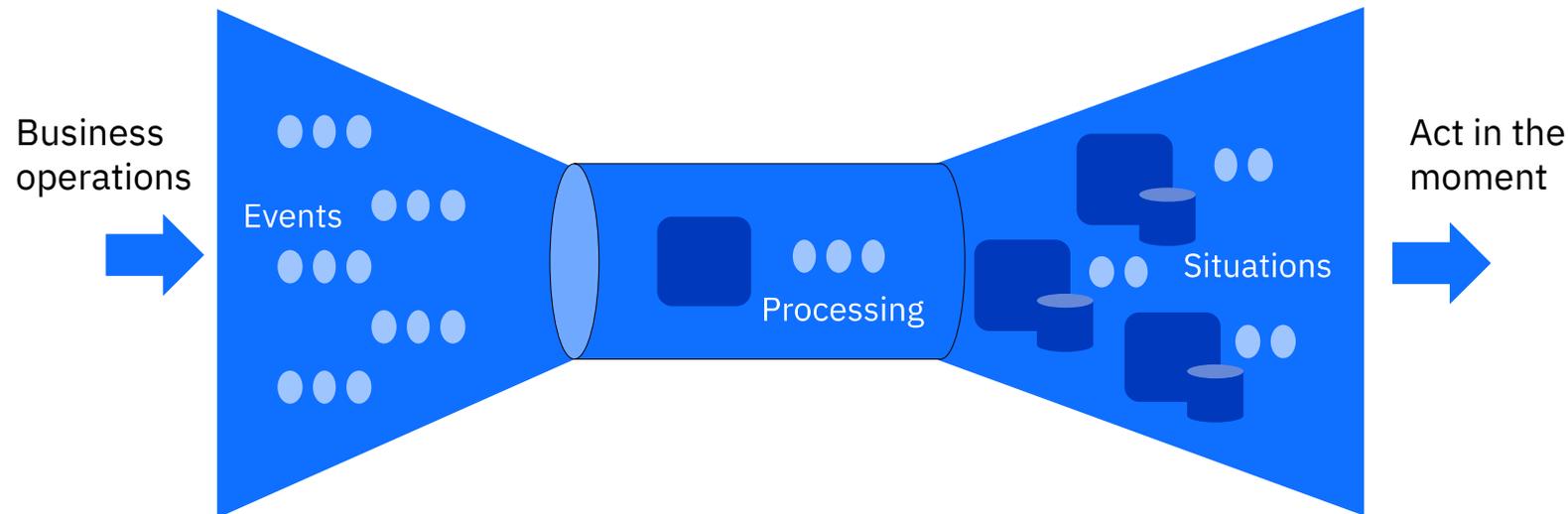
How to empower users to discover available events and request self-service access?

Event Processing

Events represent the dynamics of business operations

We can **process** streams of events, by:

- Combining, enriching, transforming, aggregating streams of events
- Correlating events over time on business context
- Identifying patterns, analyzing trends, detecting anomalies...



Events can then used to detect business-relevant **situations** that need to be handled

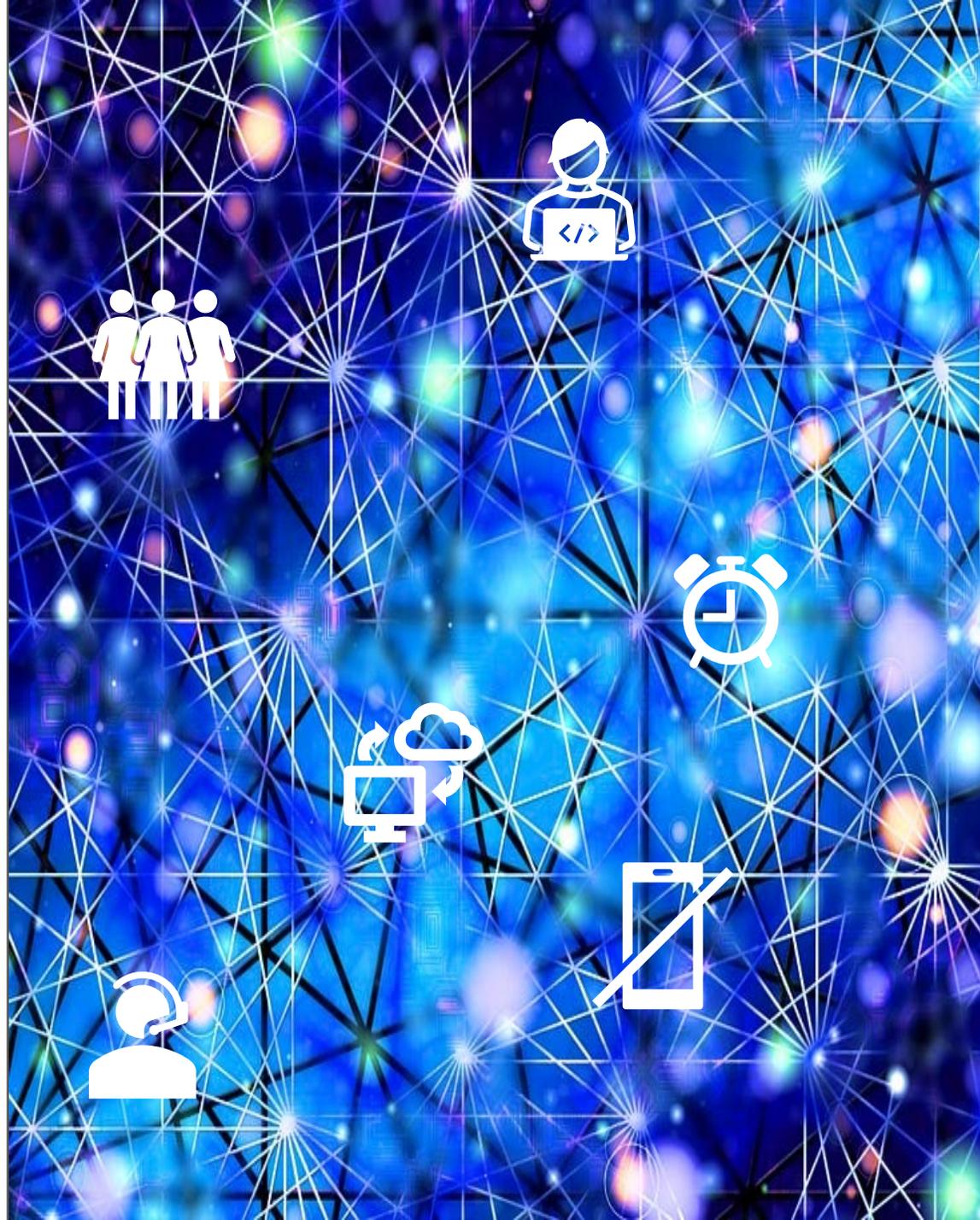
As-is

Existing event processing tools are complex and technical, limiting the speed at which businesses can tackle their event-driven use cases

To-be

Working with events becomes so simple and intuitive that a broad range of users can understand and react to situations as they occur

If you knew **what** happened
in your business **in the last**
5 minutes, what would you
do differently?



Unlock business events to activate automation

1

Detect

critical trends and business dynamics

2

Act

with speed and significant impact

3

Automate

decisions for better outcomes



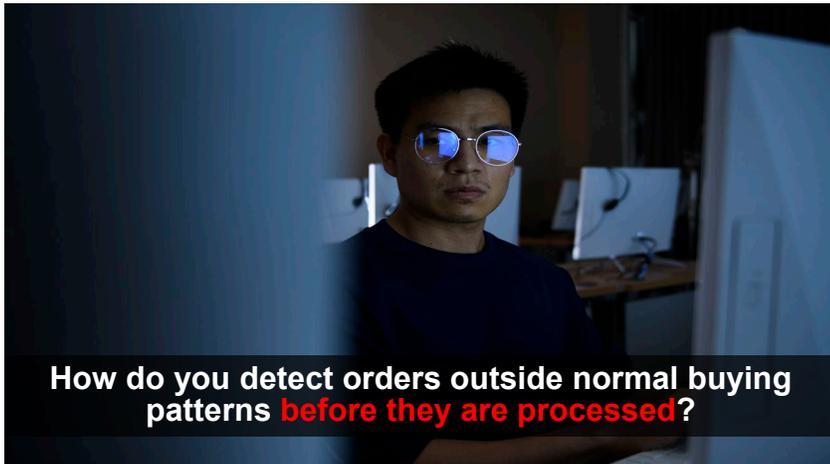
Capitalize on opportunities and address threats **in real-time**

Maximize
Sales



Resilient
Operations

Customer
Protection



Responsive
Experiences

