Microsoft

Azure

Simplify application integration with Event-Driven Architecture

Clemens Vasters, Twitter: @clemensv

Principal Architect, Microsoft Azure Messaging

Agenda

Event-based integration. Examples.

Job, Signals, and Streams

Event Streams and Timeliness

Event Journeys

TICKETING INTEGRATION FLOW

BORUSSIA-PARK, MÖNCHENGLADBACH

DIE FOHLEN

UNSER ZUHAUSE

DER BORUSSIA-PARK

BORUSSIA-PARK
PLATZ FÜR
54.022
ZUSCHAUER

2.064
BUSINESS-SEATS

684
LOGEN-PLÄTZE

45
LOGEN

ÜBER
20 MIO.
BESUCHER
SEIT STADION-
ERÖFFNUNG

KONZERTE
(U.A. BRUCE SPRINGSTEEN,
ELTON JOHN UND
HERBERT GRÖNEMEYER)

EVENTLOCATION
MIT 900
EXTERNEN VER-
ANSTALTUNGEN
PRO JAHR

Santander
FOHLEN
STALL
MIT PLATZ
FÜR 24
JUGENDSPIELER

FOHLEN
SHOP
MIT
900 QM
LADENFLÄCHE

Santander
FOHLEN
CAMPUS
DAS NACHWUCHS-
LEISTUNGSZENTRUM

FOHLEN
WELT
DAS INTERAKTIVE
VEREINSMUSEUM
1.150 QM

H4-HOTEL
MIT
125
ZIMMERN &
6 SUITEN

ÄRZTE- UND REHA-
ZENTRUM MEDICAL
PARK AUF ÜBER
1.400 QM

DIE PRAXIS.
UNSERE VEREINS-
ÄRZTE AUF
600 QM

BIERGARTEN MIT
4.000 QM
UND PLATZ FÜR
CA. 2.500
FANS

FOHLEN
SPORTSBAR
MIT
300 QM
UND PLATZ FÜR
300 GÄSTE

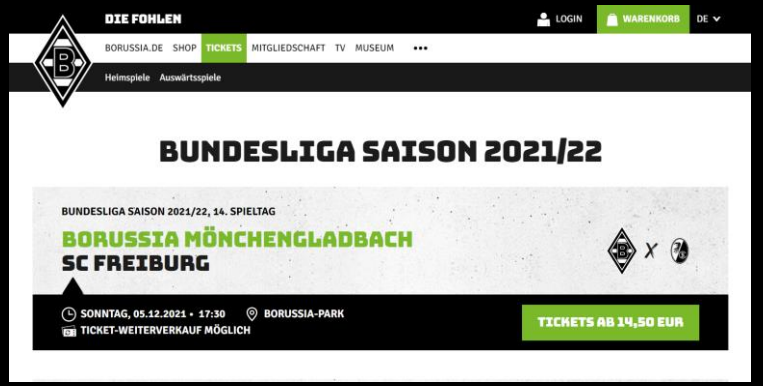
TICKETING INTEGRATION FLOW

BORUSSIA / EVENTIM / AXESS

Automated access control: Axess

ticket codes

Ticket portal: Eventim



ticket sales

customer info

Live Reporting

ERP

customer info

Single-Sign-On
Profile

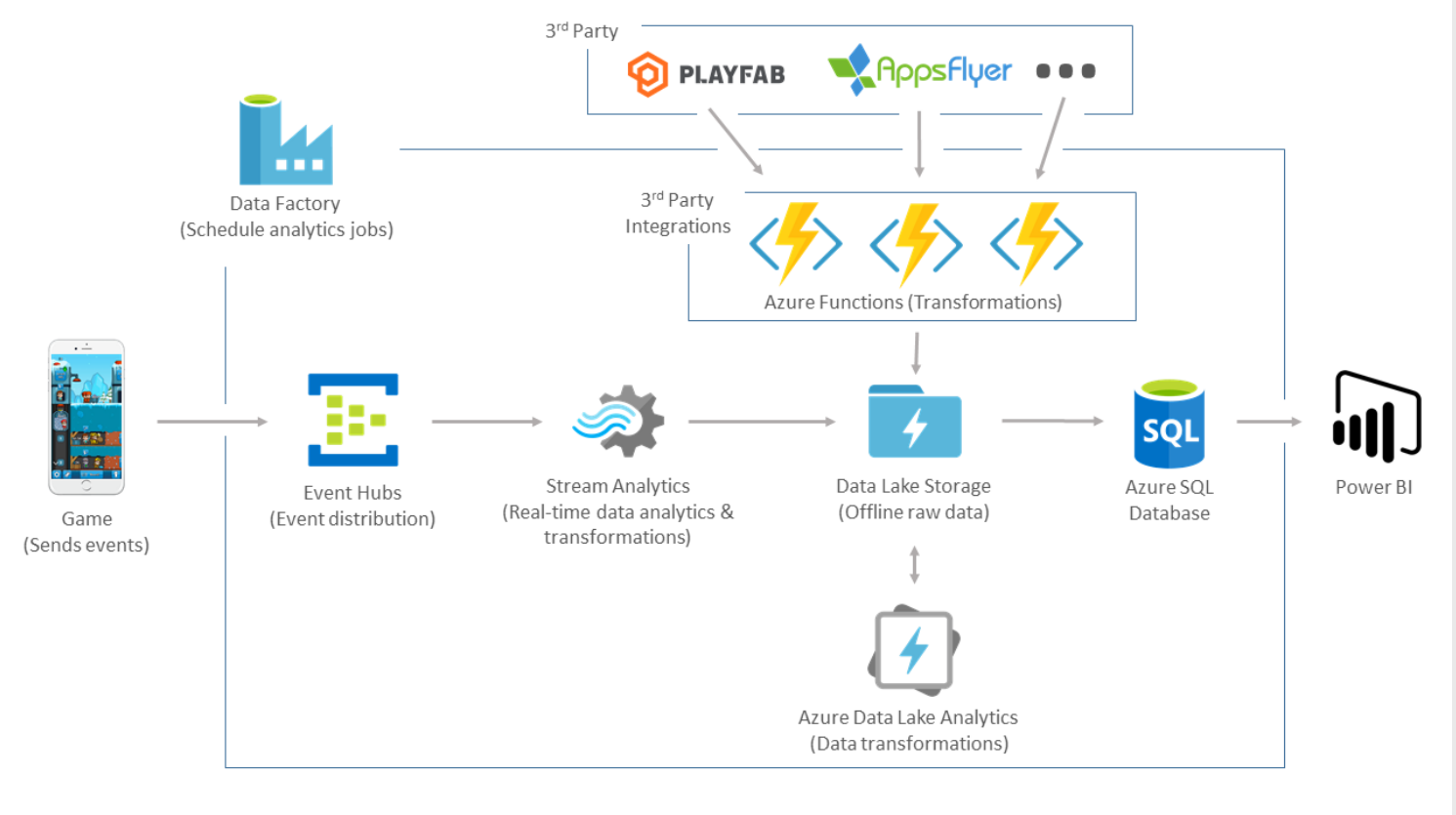
Gaming - Telemetry

343 Industries was (and is) a close partner during design and initial implementation of Azure Event Hubs for multiple Halo titles.

(Halo 4 launched on Service Bus)

Kolibri Games

With help from the Microsoft Azure gaming staff, Kolibri built that pipeline in just a few weeks. It uses Azure Event Hubs for data ingress, capturing every time a player clicks to build or buy something in the game. Azure Stream Analytics grabs the events from Event Hubs, normalizes the data, and puts it in Azure Data Lake Store. Azure Data Factory is used as the data integration service, where all the data sources and ETL workflows are scheduled, orchestrated, and monitored.

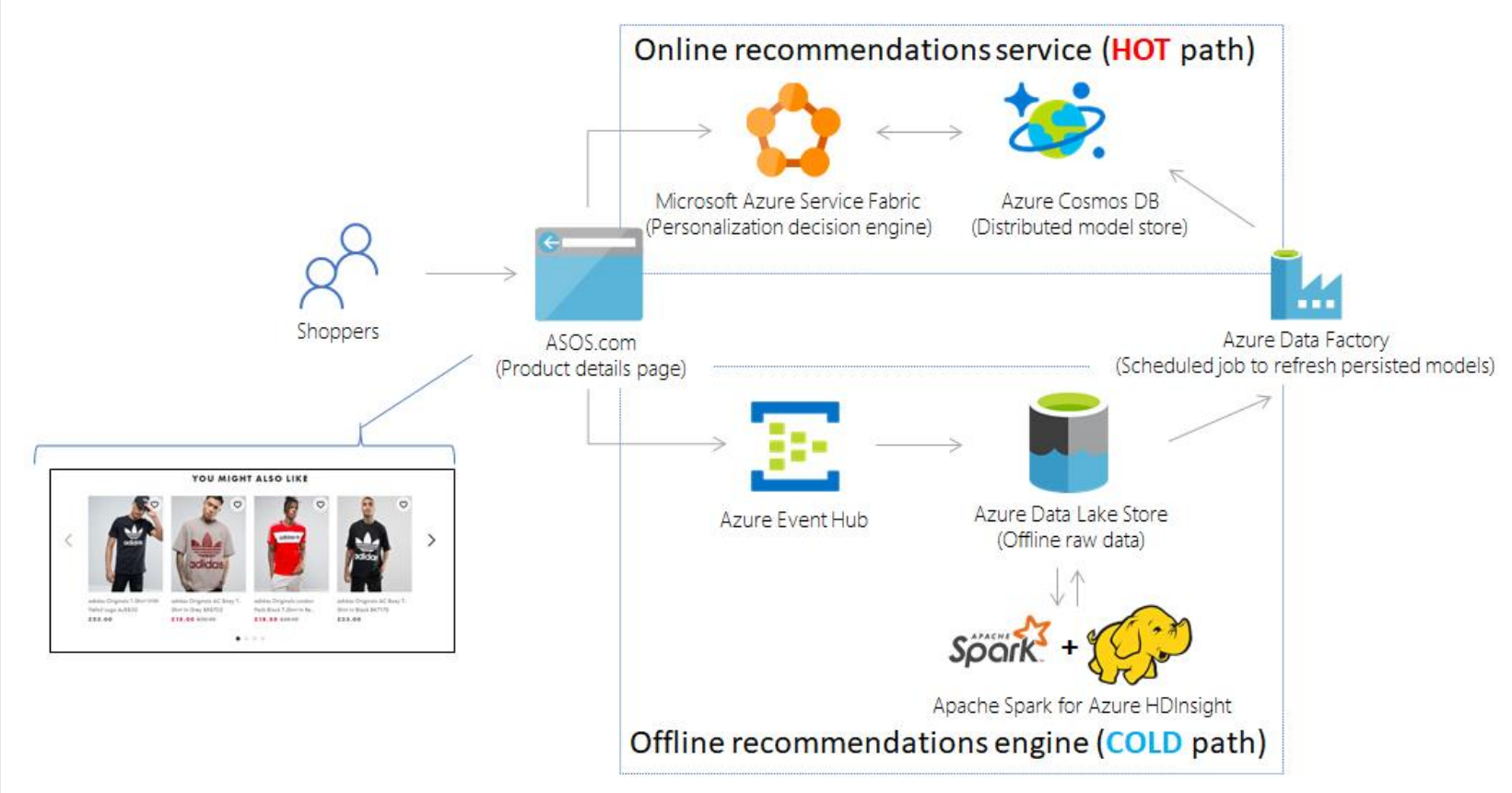


https://customers.microsoft.com/en-us/story/fluffyfairygames-media-telco-azure

Retail – Recommendations

ASOS

The offline recommendations engine is used to batch-process user telemetry to build multiple machine learning models to be hosted by the online recommendations service. User interaction telemetry is stored in Azure Data Lake Store for long-term storage. Competing versions of the user and product vector models are generated, using the Apache Spark MLlib machine learning library in Azure HDInsight using Python LightFM and TensorFlow. These are then bulk-loaded by Azure Data Factory into Azure Cosmos DB.

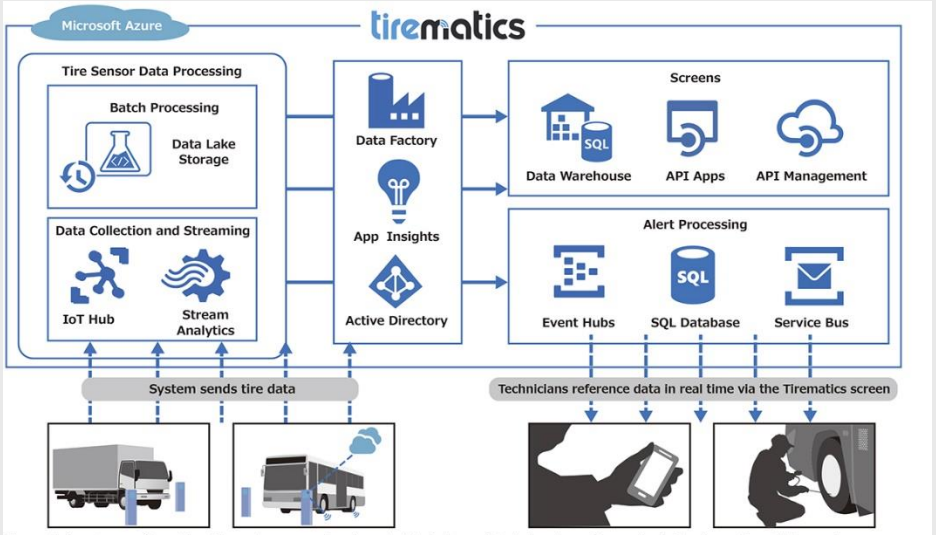


https://customers.microsoft.com/en-us/story/asos-retail-and-consumer-goods-azure

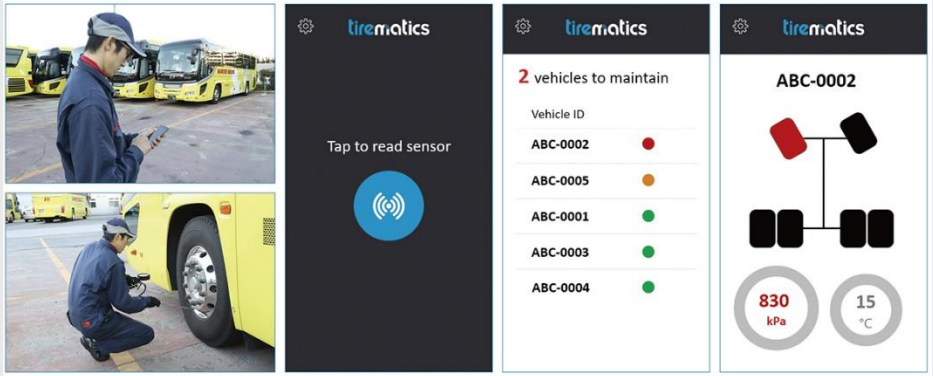
Automotive –Telematics

Bridgestone

Bridgestone started developing Trematics on Azure in August 2016. The Trematics architecture comprises Azure IoT Hub, Azure Data Lake, Azure Stream Analytics, Azure Event Hubs, and more. Tire sensors first send data to a local system at a vehicle maintenance site. The system then uses mobile phone networks to send the data to Azure IoT Hub and stores it on Azure Data Lake. Next, Azure Stream Analytics analyzes the data in real time. If the system detects an abnormality, Azure Event Hubs sends an alert (see the Trematics system configuration diagram below for more details).



Trematics' system configuration. The system comprises Azure IoT Hub, Azure Data Lake, Azure Stream Analytics, Azure Event Hubs, and more.



A technician checks tires. Trematics' information can be checked with a smartphone. Technicians can check the air pressure etc. of each tire by simply tapping the screen* and selecting a vehicle. In addition, the tire layout shows which tires have air pressure problems. *The screens above are mockups.

https://customers.microsoft.com/en-us/story/724108-bridgestone-japan-azure-iot-chemicals-agrochemicals-en-jp

Signals, Streams, and Jobs

***Signal*:** The capture of an **occurrence** (statement of fact) during the operation of a software system

***Event*:** A data record expressing a signal and its context**.** The context is expressed in metadata annotating the signal.

***Event Stream*:** A chronological sequence of events belonging to the same context.

Job: *Not an event*. A description of a task that needs to be performed by some party. Preferably just once.

Signals


Calendar entry created
Meeting scheduled
Meeting attendee accepted
Inventory item added to shelf
Payment collected
Calendar appointment reminder
Meeting attendee invited
Inventory item removed from shelf
Inventory item added
Application deployment initiated
Change committed
Change proposal verification succeeded
Change proposed
Change proposal commented
Mouse cursor moved
Inventory item scanned for sale
Aircraft rotated
Application deployment succeeded
Pizza ordered
Aircraft position report
Application deployment failed
Vehicle passed induction loop
Eye focus tracked
Sales lead added
Delivery address changed
Pizza prepared
Aircraft squawk code change
CO PPM over threshold
Room humidity read
Advertisement clicked
Database record inserted
Database record updated
Room temperature read
Smoke detected
Virtual machine started
Engine oil pressure read
Package scanned
Pizza delivered
Aircraft landed
Stock trade order issued
Venue seat ticket reserved
CPU utilization read
Package delivered
File changed
Object detected ahead
Infrared movement detected
Engine coolant temperature read
Engine RPM read
Stock trade order executed
Account transfer scheduled
Venue seat ticket released
Package picked up
Venue seat ticket purchased
Object velocity differential calculated
Visual object classified
Engine oil pressure threshold alert
Tank pressure read
Stock trade order settled
Account transfer completed
File created
Package loaded
Package routed

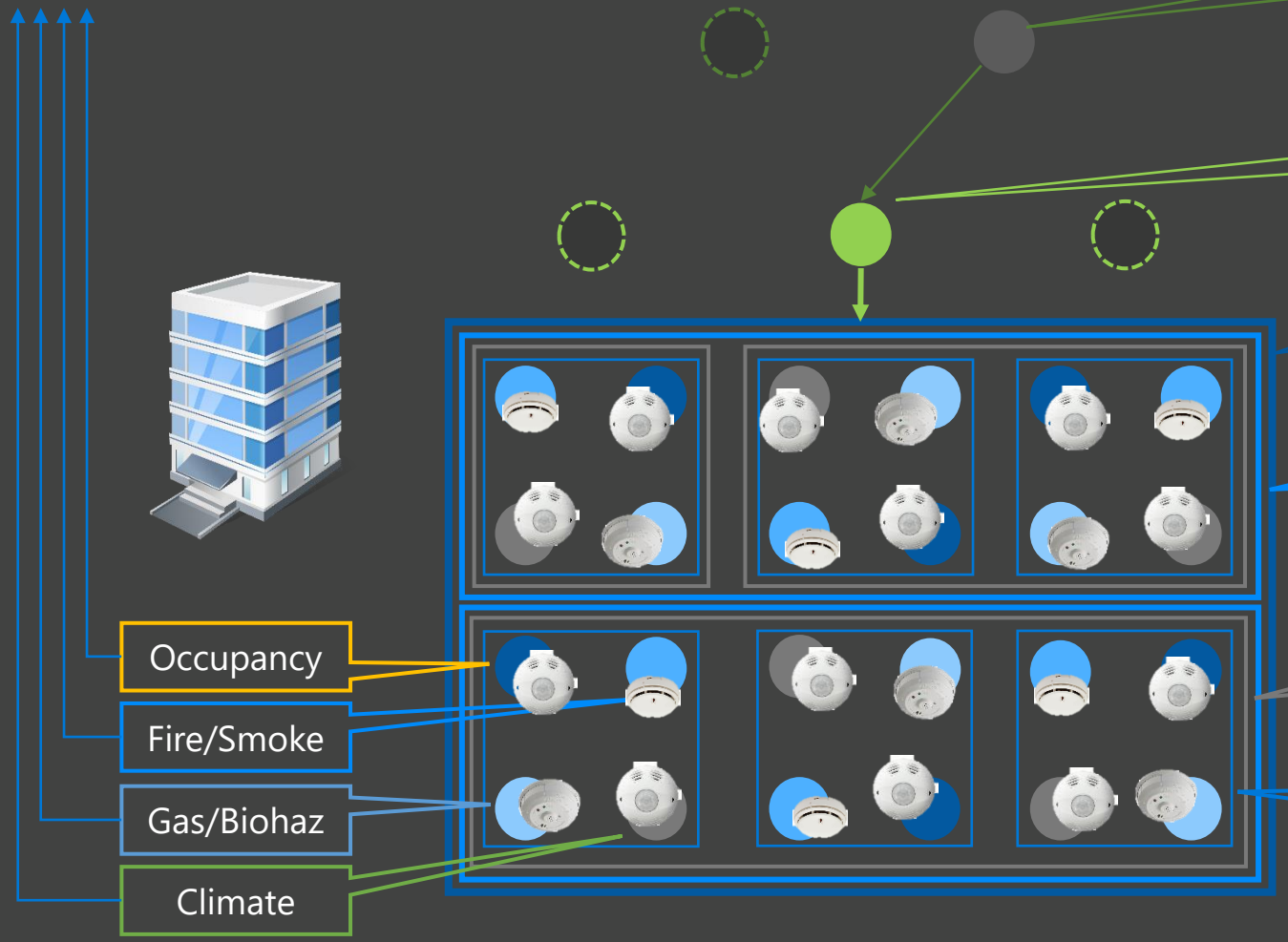
/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/occupancy

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/fire

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/gasbio

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/climate

Events put signals into context



Org

/org/contoso/

Site

/org/contoso/site/berlin/

Building

/org/contoso/site/berlin/building/41

Floor

/org/contoso/site/berlin/building/41/floor/3

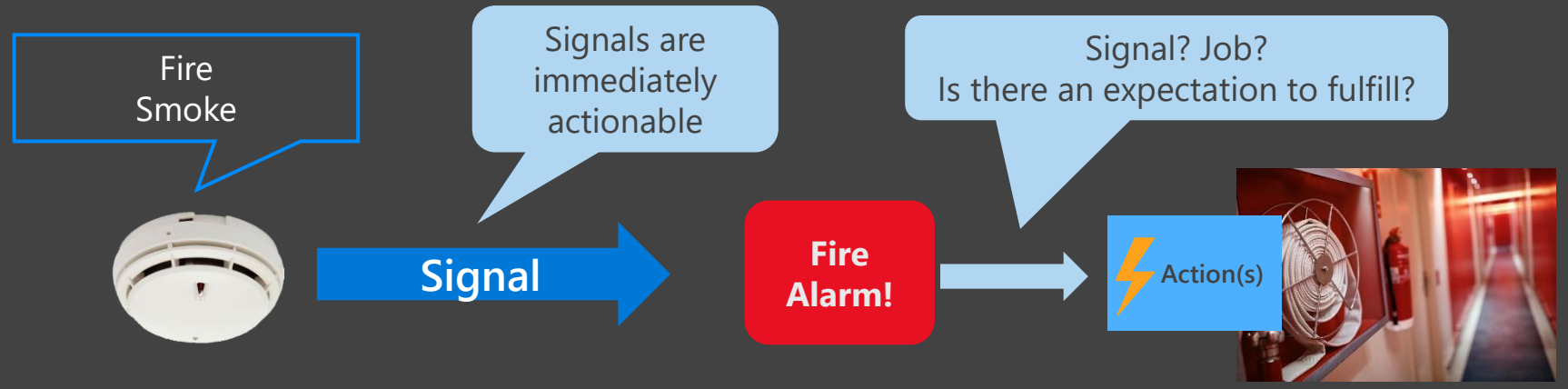
Unit

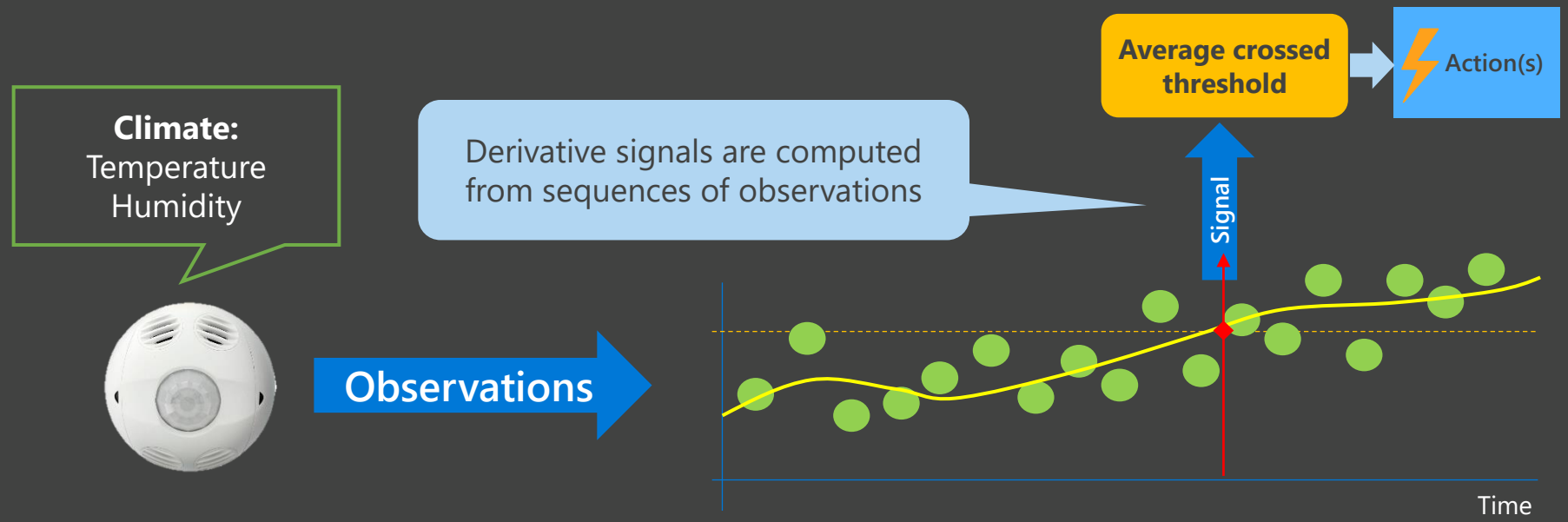
/org/contoso/site/berlin/building/41/floor/3/unit/j

Room

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4

Events: Observations, Signals, Jobs





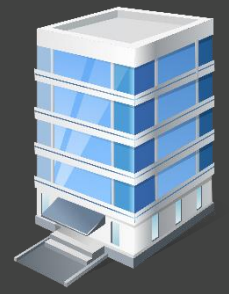
Observations, signals, and derivative jobs

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/occupancy

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/fire

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/gasbio

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/climate



Occupancy

Fire/Smoke

Gas/Biohaz

Climate

Analytics Questions:

- Are there people in room 41 3J/4?
- What room is unoccupied in building 41?
- Is there a fire alarm at the site?
- How is the air quality on the lab floor?
- What's the temp in tenant unit 41 3J?

Derivative signals and reactive actions:

- Signal evacuation and alert the Fire Department if any fire or gas/biohaz sensor on site goes into an alert state.
- Adjust floor HVAC when average temp on any building floor deviates by +/- 2C from 20C.
- Alert Security when unexpected occupancy is detected in Unit 41 3J.

Org

/org/contoso/

Site

/org/contoso/site/berlin/

Building

/org/contoso/site/berlin/building/41

Floor

/org/contoso/site/berlin/building/41/floor/3

Unit

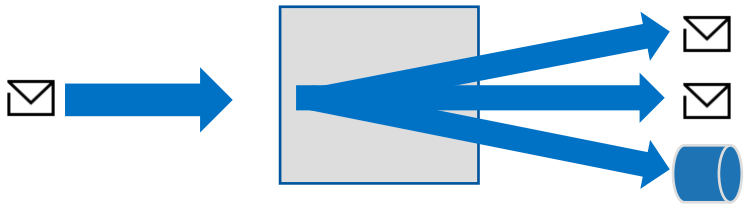
/org/contoso/site/berlin/building/41/floor/3/unit/j

Room

/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4

Discrete Event Router

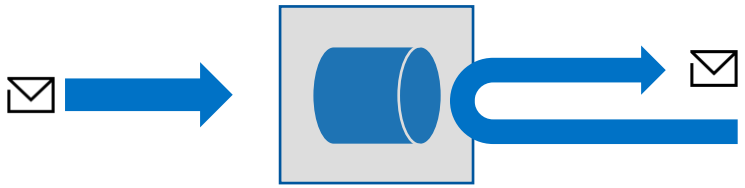
Azure Event Grid, AWS Event Bridge, Knative Eventing



Push-style distribution of discrete events to serverless workloads or other messaging infrastructures

Queue Pub/Sub Broker

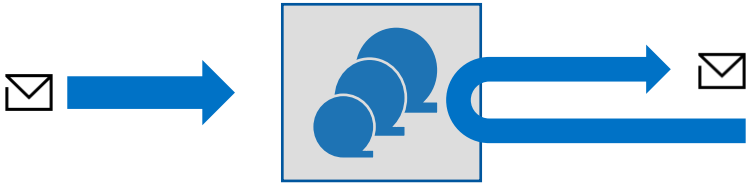
Azure Service Bus, AWS SQS/SNS, Google PubSub, Apache ActiveMQ, RabbitMQ, IBM MQ



Pull-style, queue-based transfer of jobs and control via message queues and topics

Event Stream Engine

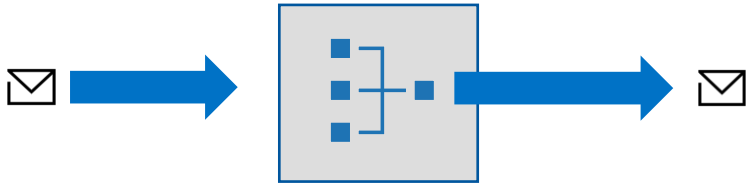
Azure Event Hubs, AWS Kinesis, Apache Kafka, Apache Pulsar, CNCF Pravega



Partitioned, high-volume, tape-drive-style sequential recording and unlimited, pull-style re-reads of event streams.

Event Stream Aggregator

Azure Stream Analytics, AWS Kinesis Analytics, Apache Samza, Apache Flink, etc.



Stateful processing of event streams yielding event streams and discrete events as continuous output

Event Streaming is not "modern" and Queues are not "traditional"

Both are patterns of state-of-the art messaging infrastructures.

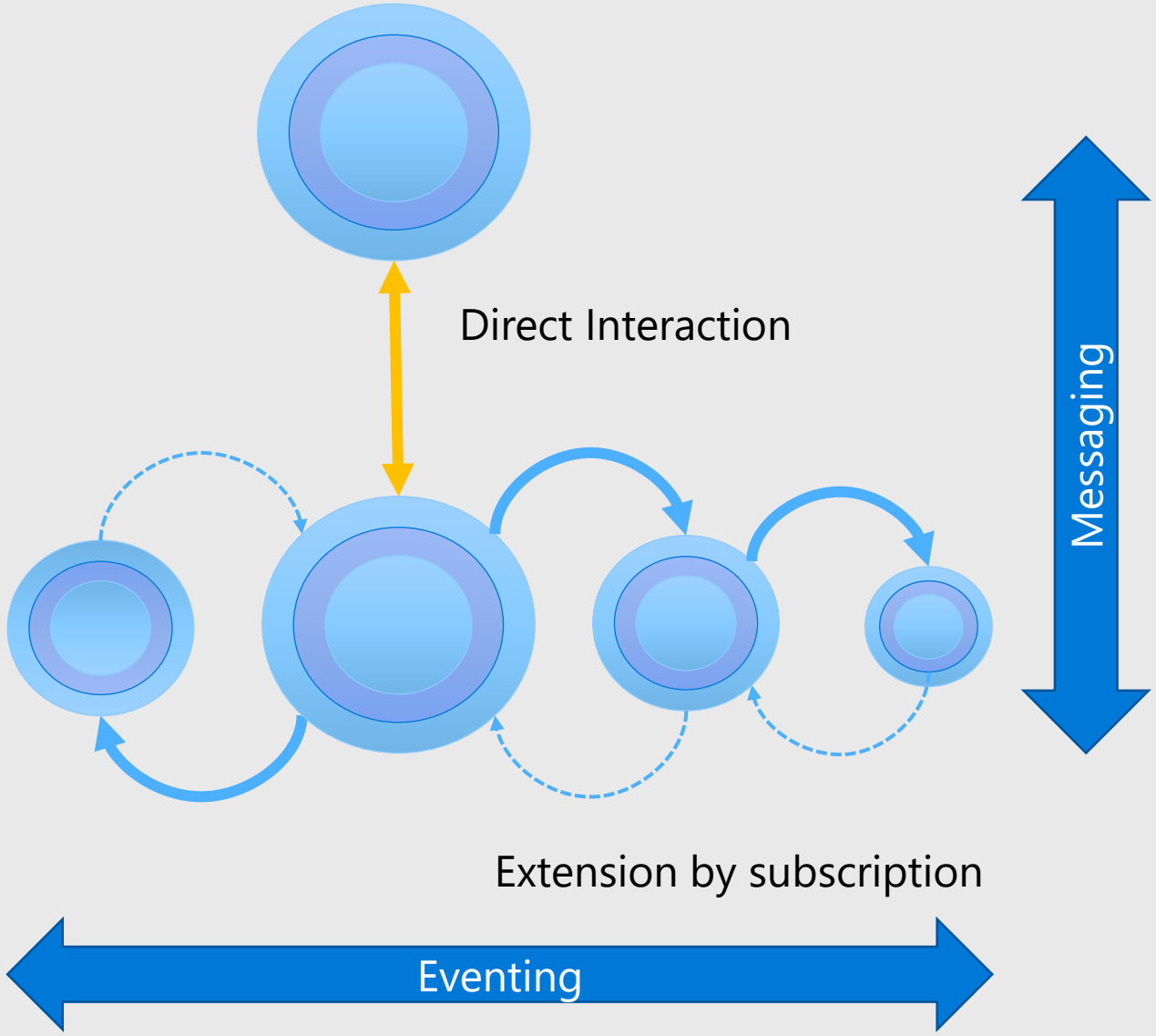
Modern apps use eventing and queue-based messaging

“Core” functions of services require direct, point-to-point, RPC or queue-based interaction:

Commands, Requests

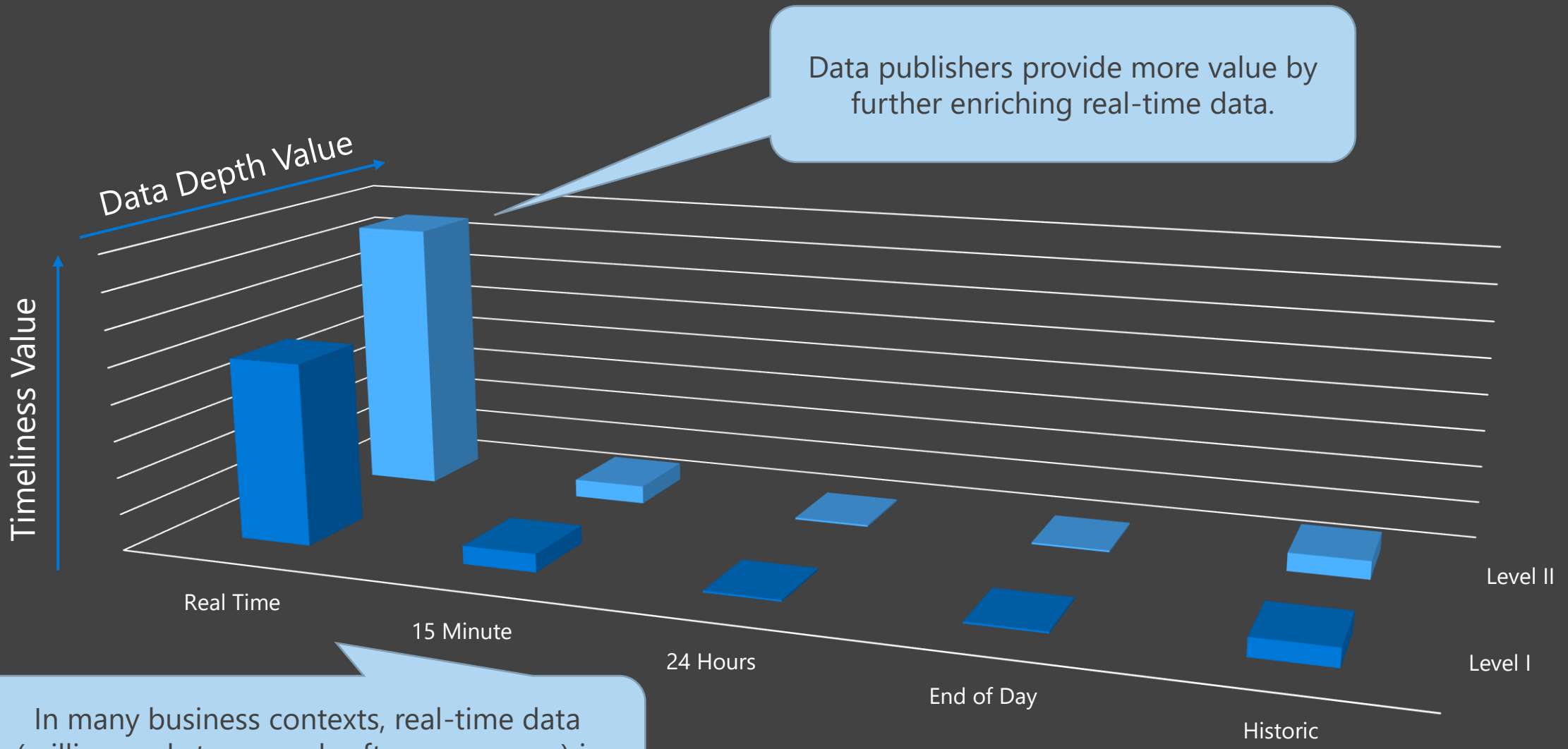
Extensions react to events or insights derived from event streams emitted by services.

Might turn to the emitting service to ask for details or perform actions.

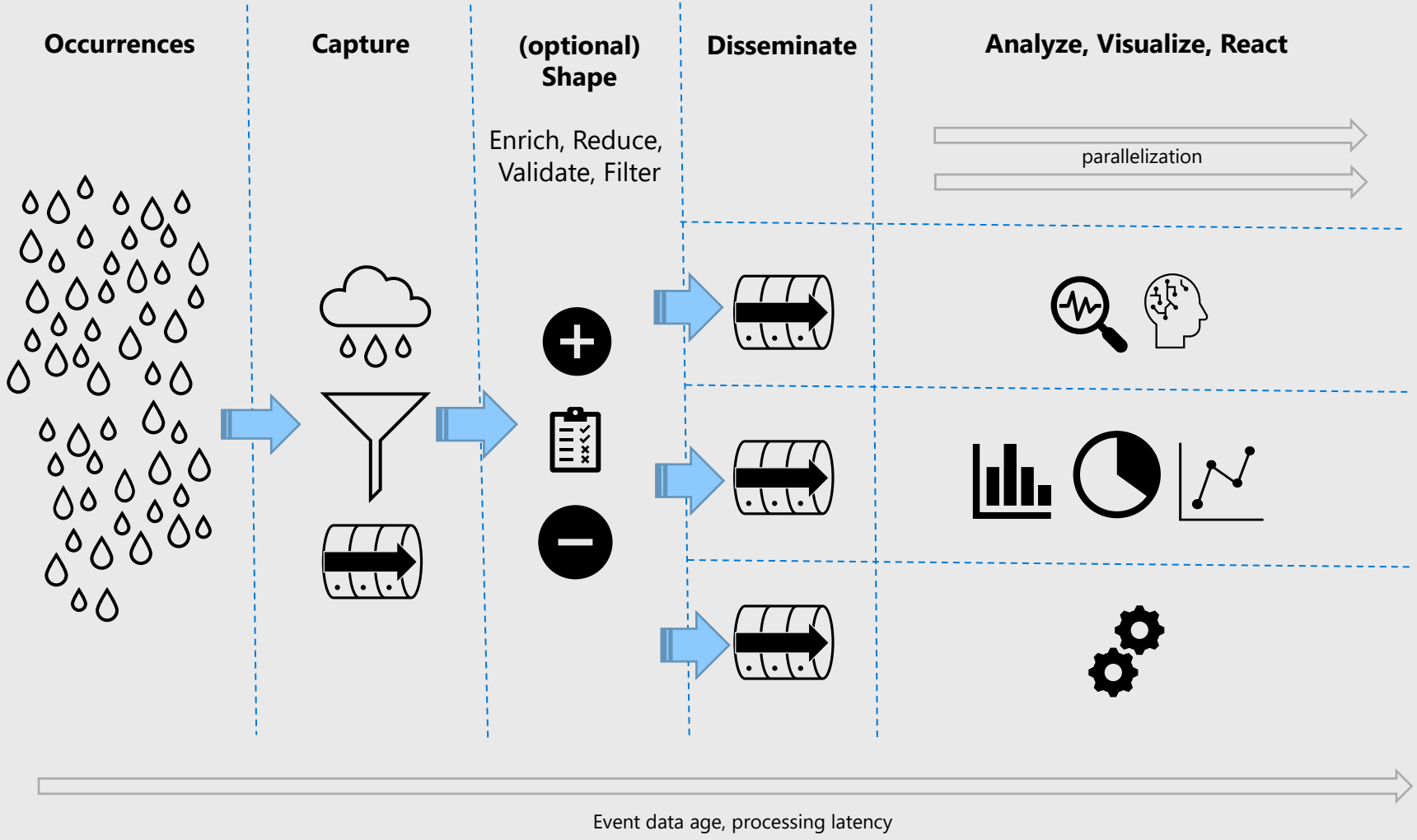


Event Streams and Time(-liness)

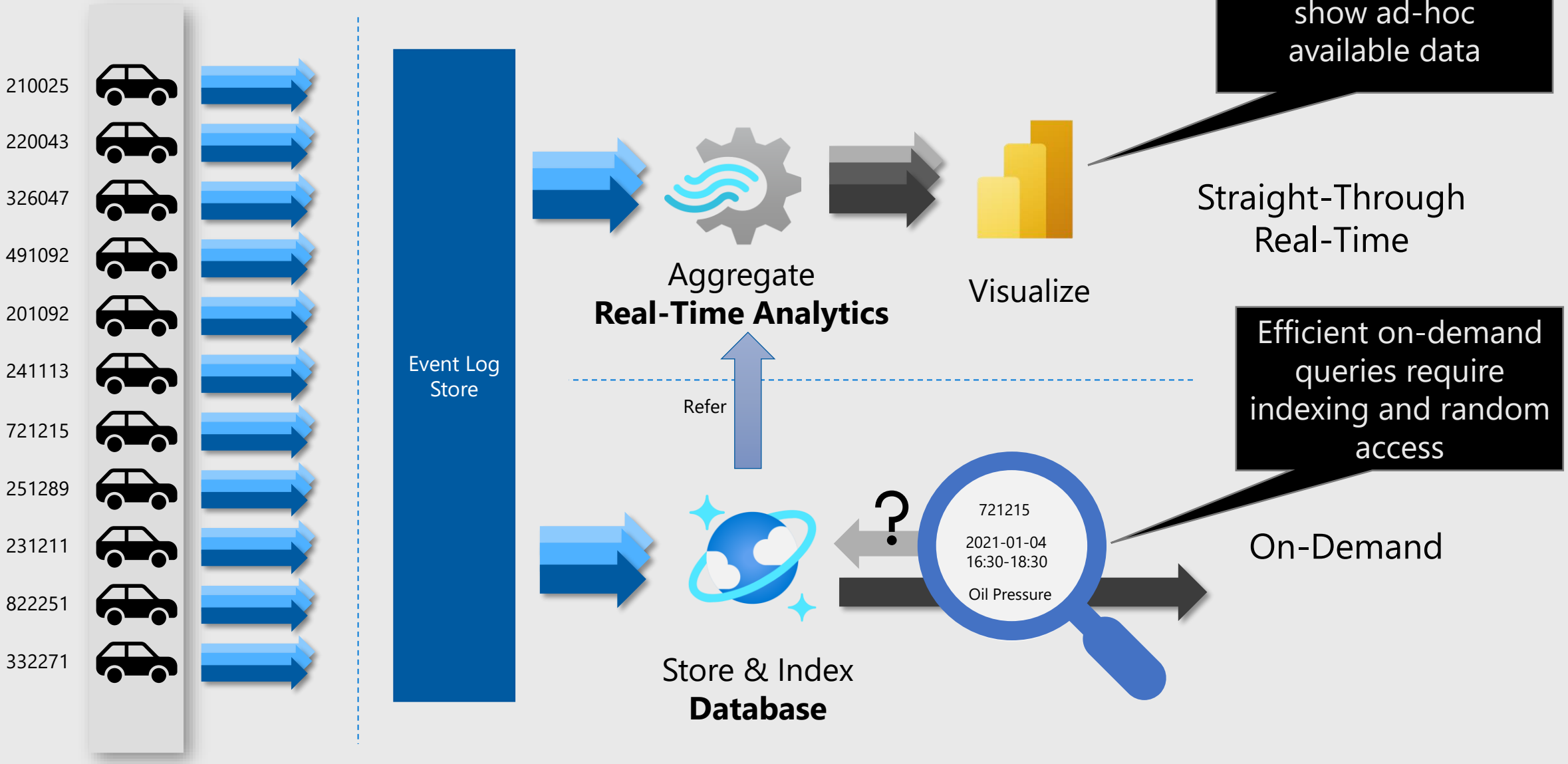
Event Data Value – Securities Markets



Velocity Matters ➔ Parallelization Matters

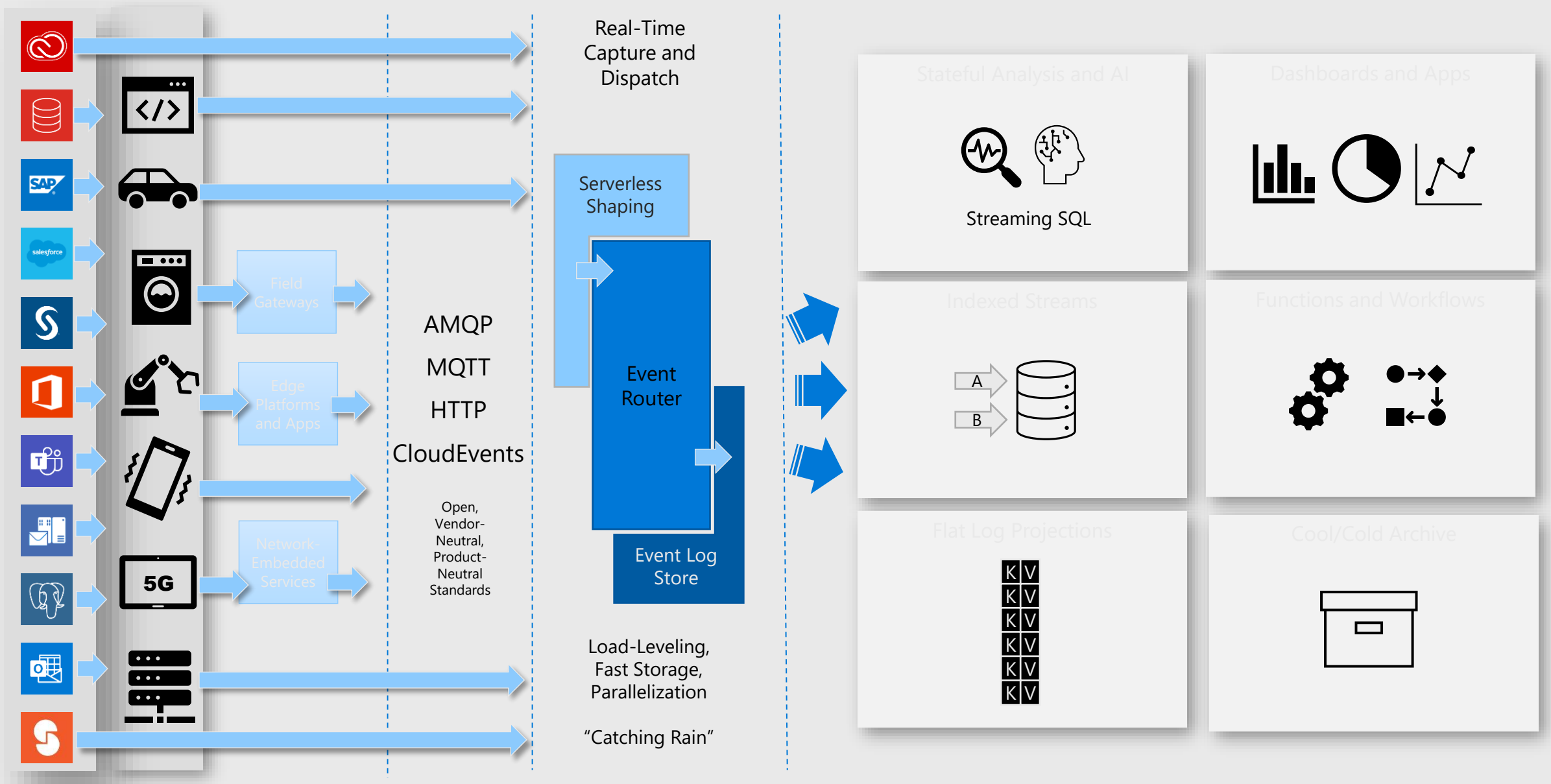


Event Streams and Context

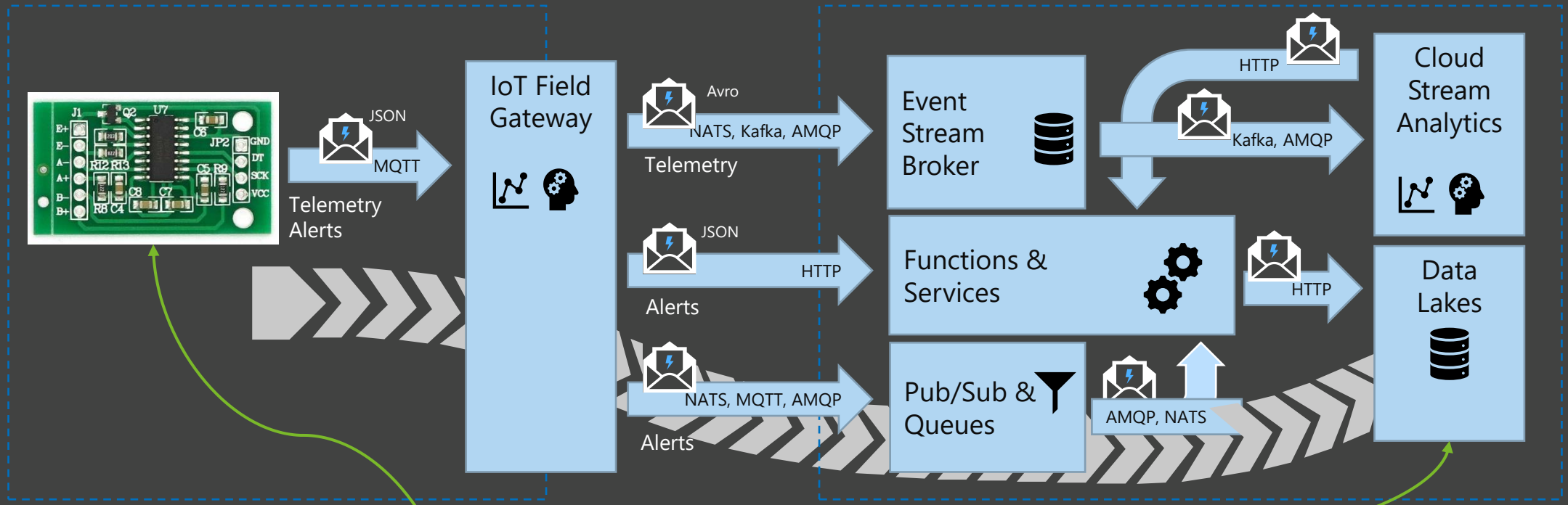


Event Journeys

Federation: Event Journeys



Why Standards? Why CNCF CloudEvents?



- Event data is often routed via multiple hops and often using different protocols and including intermediaries that are not under a single party's control
- How is what gets sent here easily routed to and stored here in hybrid edge/cloud and multi-cloud systems?
- How can we ensure that no critical information is lost in spite of the differences?

Why CNCF CloudEvents?

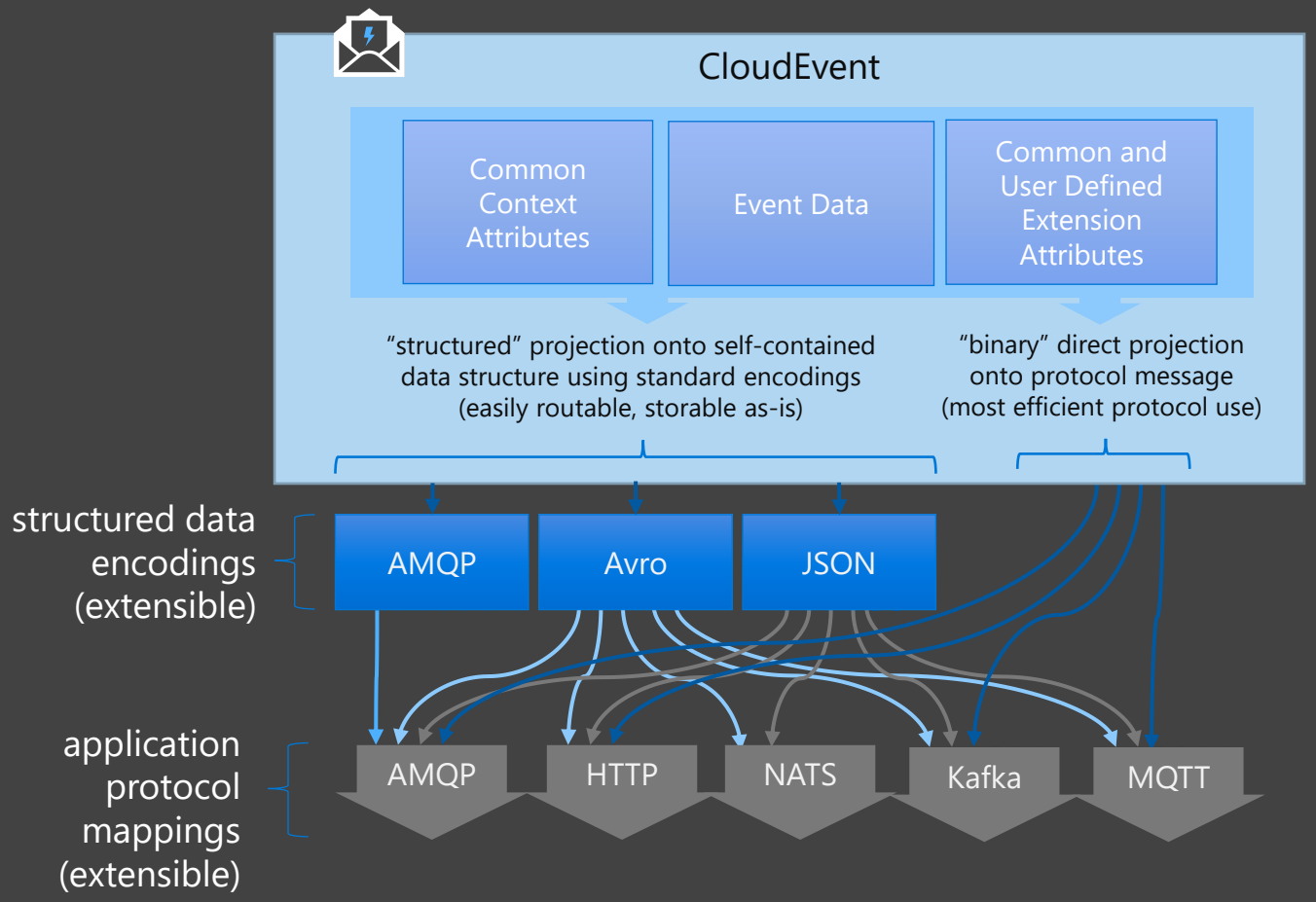
Binds to existing standard application protocols. Not a new protocol.

Does not try to abstract away protocols but leverages each for its strengths

Integrates with existing messaging and eventing stacks

Leverages existing data encodings and is easy to adapt to new ones (Protobuf, CBOR, MsgPack, etc.)

Allows for protocol switching and transcoding on multi-hop routes



CNCF CloudEvents Current Work

Schema Registry

API definition for managing schema documents used for serialization and validation in eventing and messaging scenarios.

Event Catalog

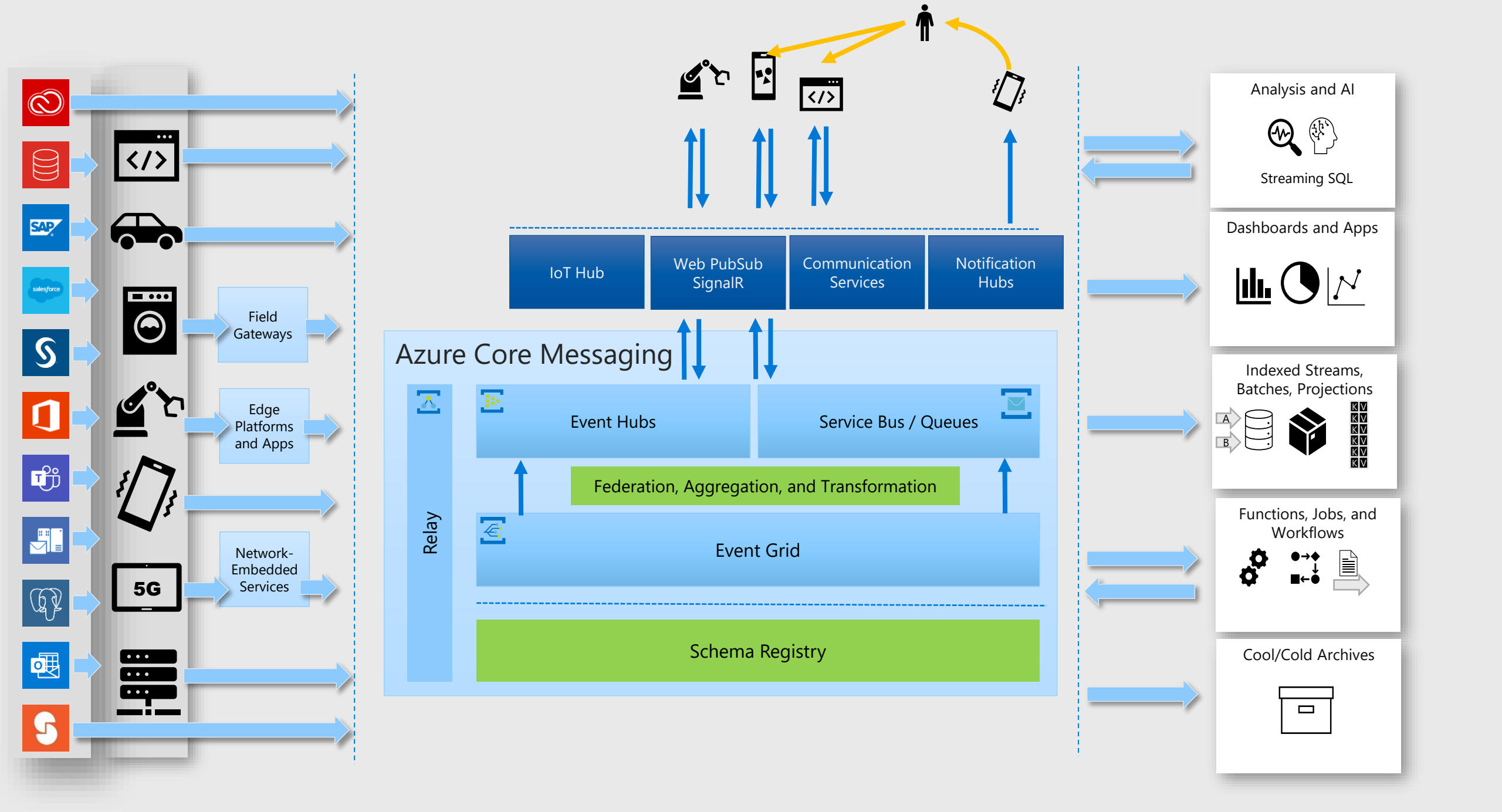
API definition for managing and discovering event templates at development time

Discovery

API definition for sharing and discovering event subscription points and sources at runtime

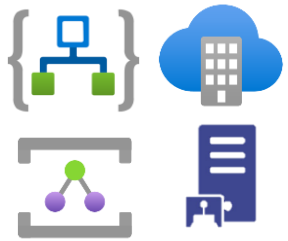
Subscription

API definition for subscribing to events across multiple protocols


Field Gateways
Edge Platforms and Apps
5G
Network-Embedded Services
IoT Hub
Web PubSub SignalR
Communication Services
Notification Hubs
Azure Core Messaging
Relay
Event Hubs
Service Bus / Queues
Federation, Aggregation, and Transformation
Event Grid
Schema Registry
Analysis and AI
Streaming SQL
Dashboards and Apps
Indexed Streams, Batches, Projections
Functions, Jobs, and Workflows
Cool/Cold Archives

Eventing on Azure Cloud

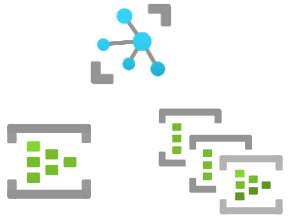
SaaS Platform Integration



Open, Vendor-Neutral, Product-Neutral Protocols

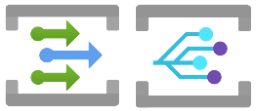
EDI Integration

Event Stream Capture and Streamed Delivery

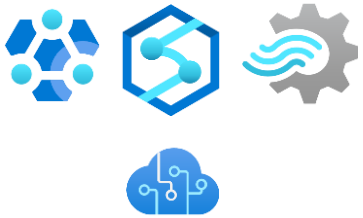


Database Change Capture

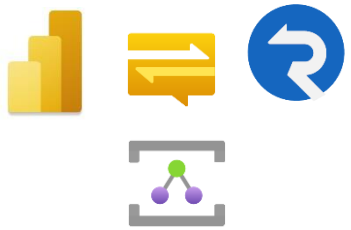
Discrete Event Capture and Subscriber Delivery



Stateful Analysis and AI

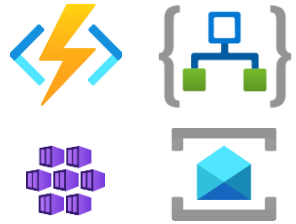


Dashboards and Apps



Indexed Streams and Batches

Functions, Jobs, and Workflows



Flat Log Projections

Cool/Cold Archives