

Flow Architecture

Building solutions with real-time information

Clemens Vasters, Twitter: @clemensv

Principal Architect, Microsoft Azure Messaging

Architect, Digital Circle, Borussia Mönchengladbach

Agenda

Flow-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

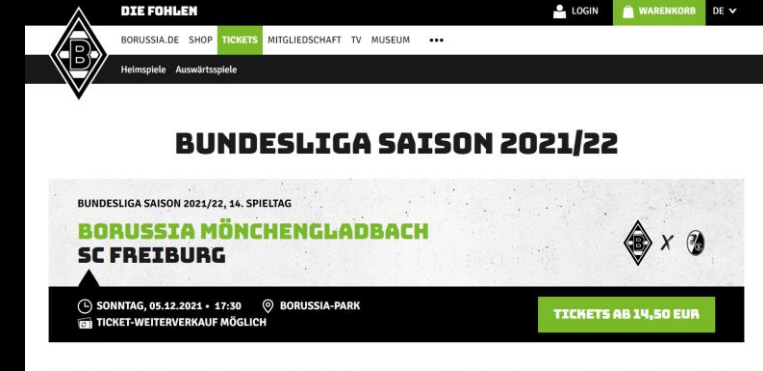
Automated access control: Axess



ticket codes

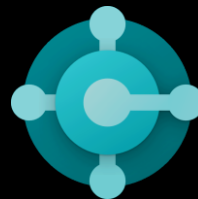


Ticket portal: Eventim



ticket sales

customer info



Microsoft Dynamics ERP



customer info



Single-Sign-On
Profile



Gaming - Telemetry

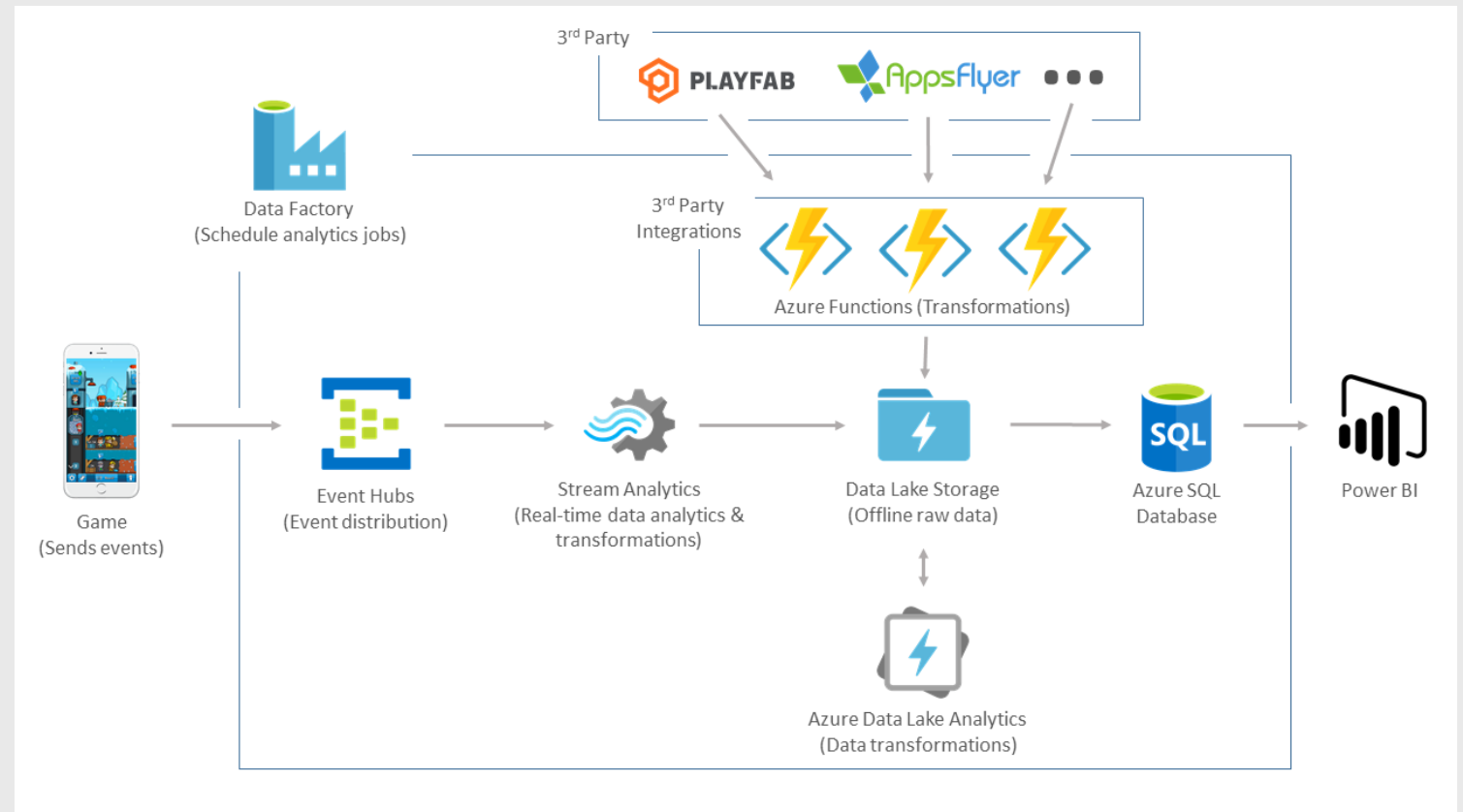
HALO

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.

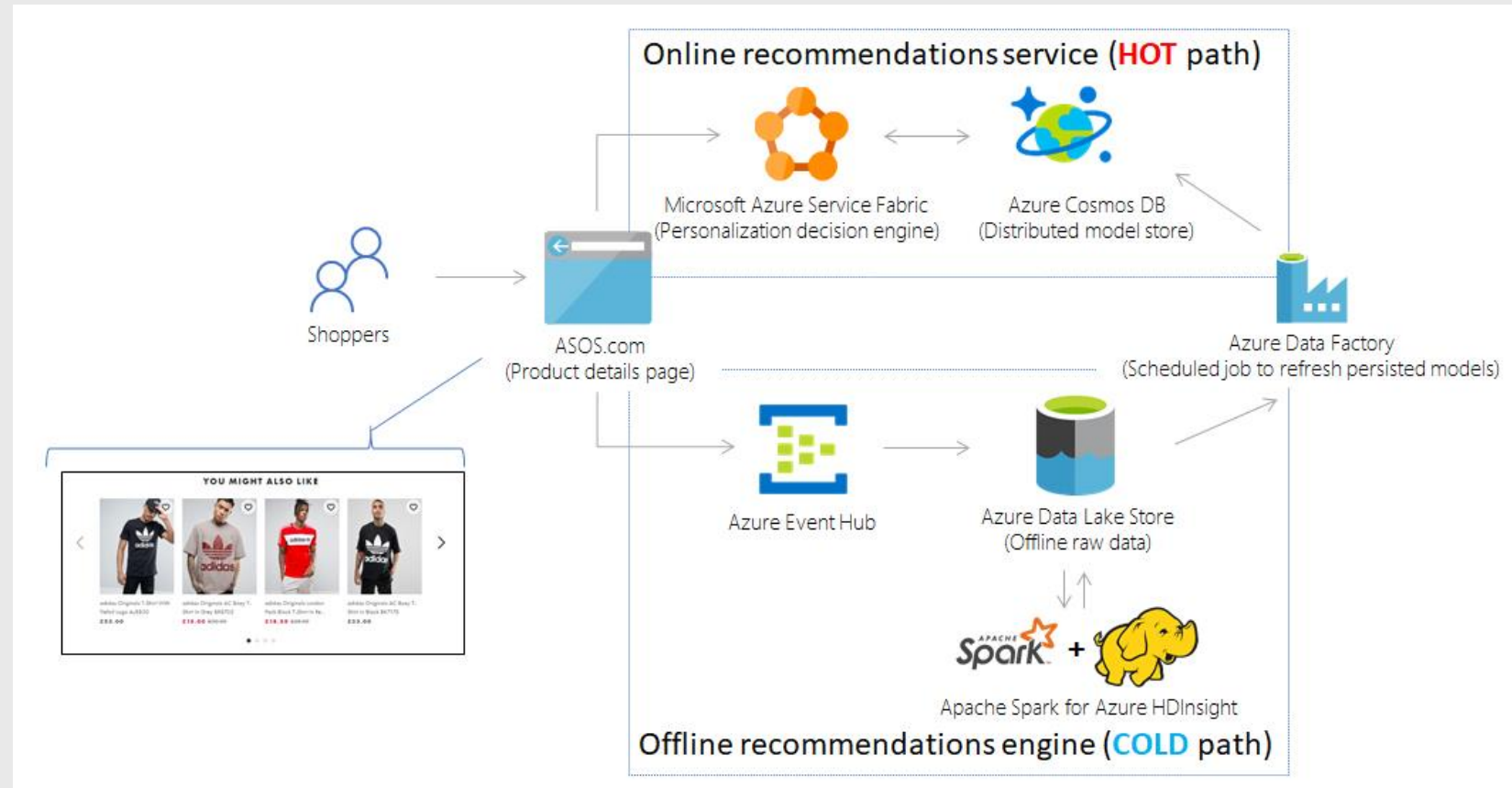


<https://customers.microsoft.com/en-us/story/fluffyfairygames-media-telco-azure>

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.



Automotive –Telematics

Bridgestone

Bridgestone started developing Tirematics on Azure in August 2016. The Tirematics 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 Tirematics system configuration diagram below for more details).



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.

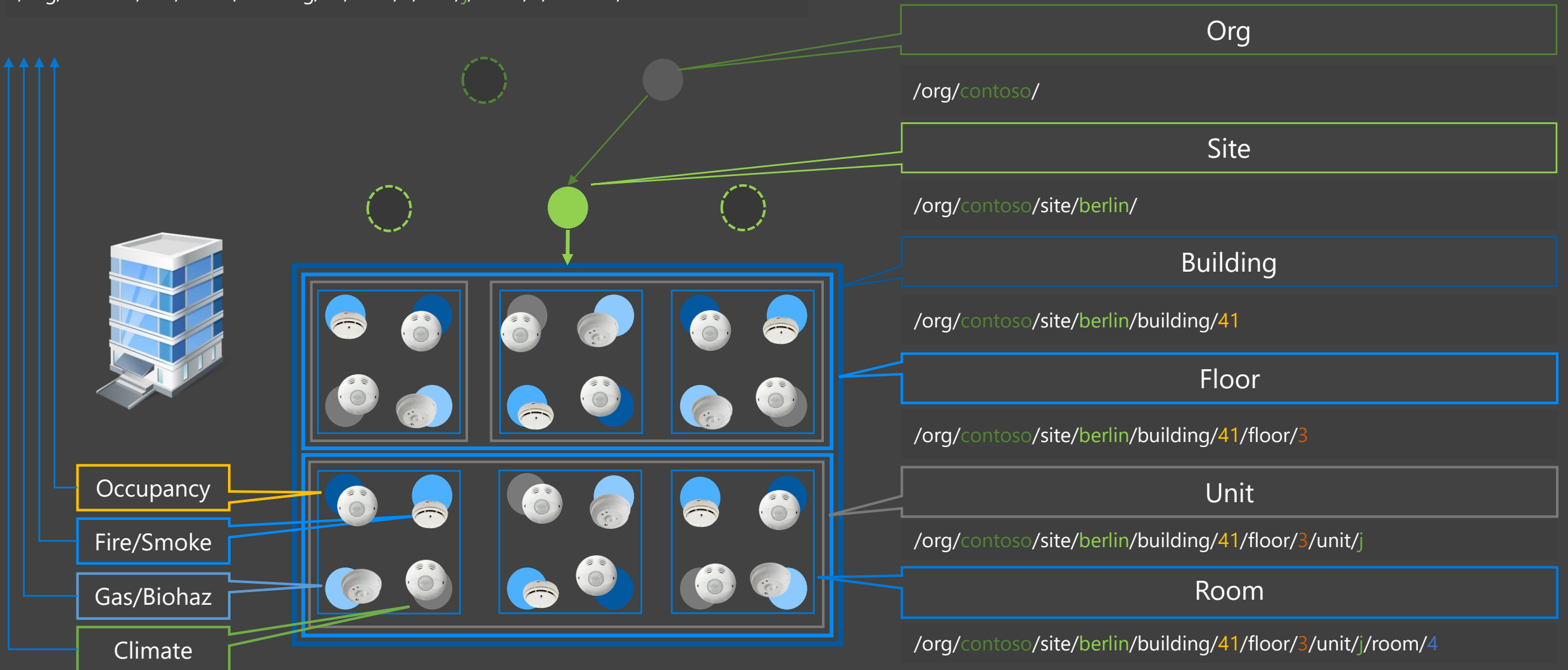
/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/occupancy

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

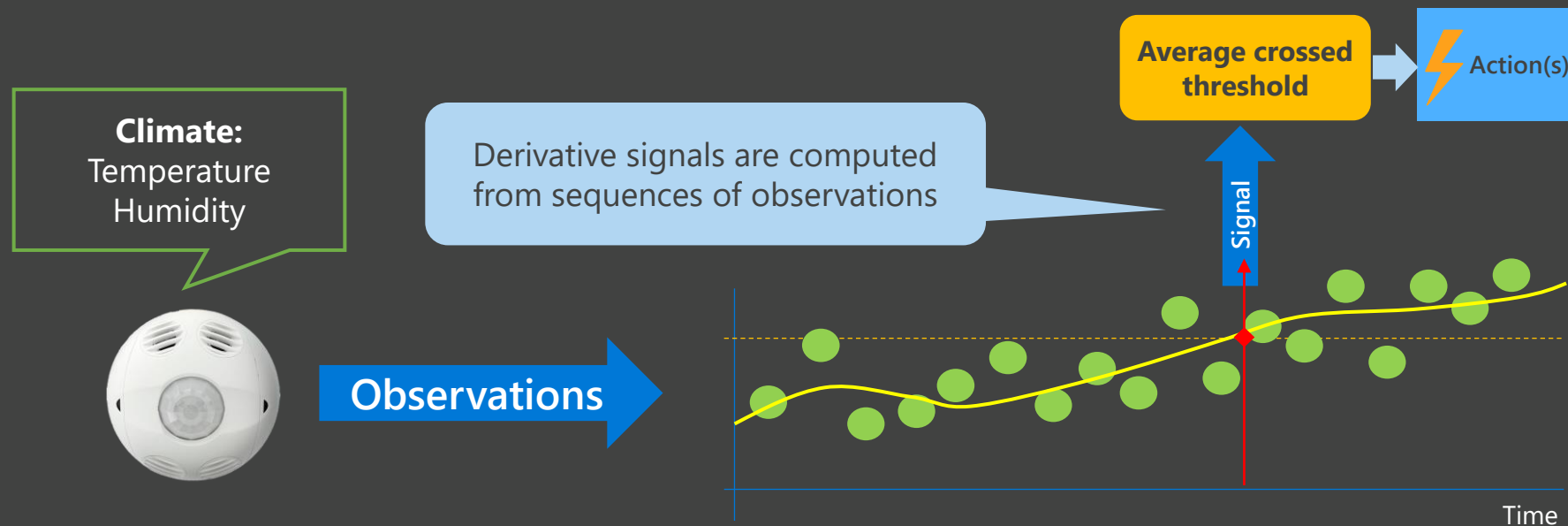
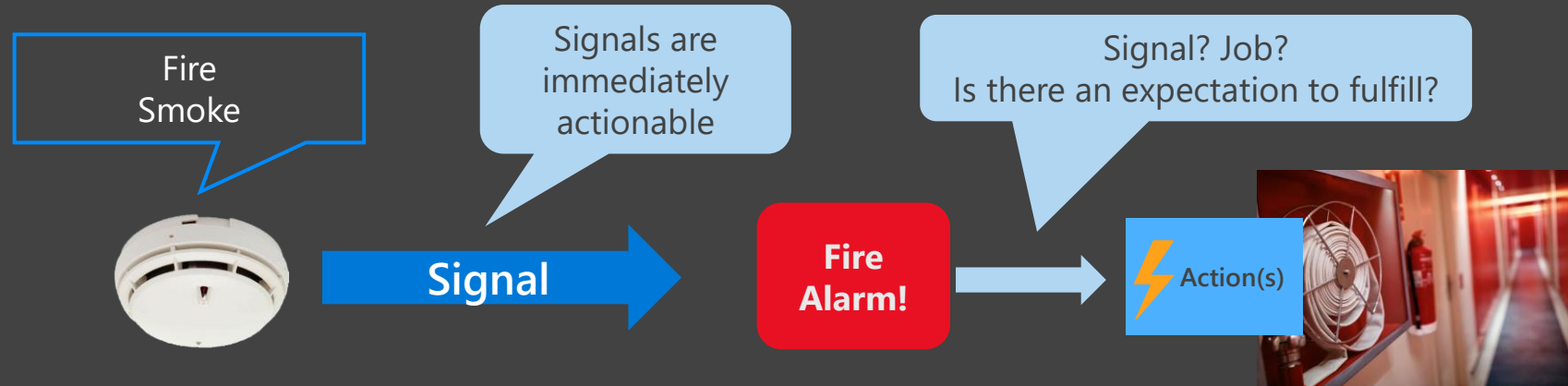
/org/contoso/site/berlin/building/41/floor/3/unit/j/room/4/sensors/gasbio

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

Events put signals into context



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

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.



Occupancy

Fire/Smoke

Gas/Biohaz

Climate

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

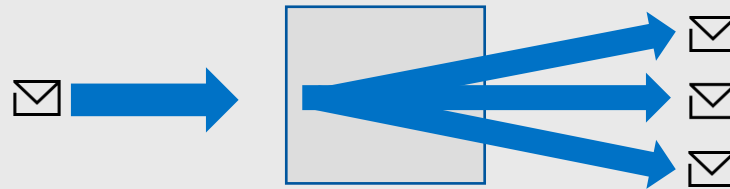
Room

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

Eventing and Messaging Service Patterns

Discrete Event Broker

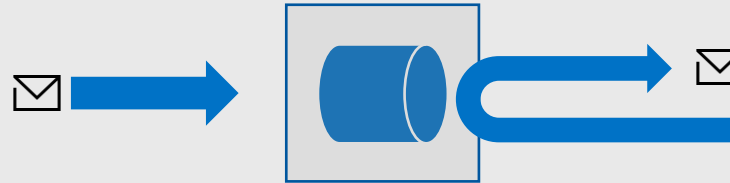
Azure Event Grid, AWS Event Bridge, Knative Eventing



Push-style distribution of discrete events to serverless workloads

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

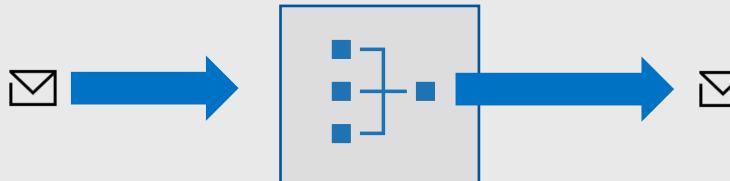
Azure Event Hubs, AWS Kinesis, Apache Kafka, Apache Pulsar, CNCF Pravega



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

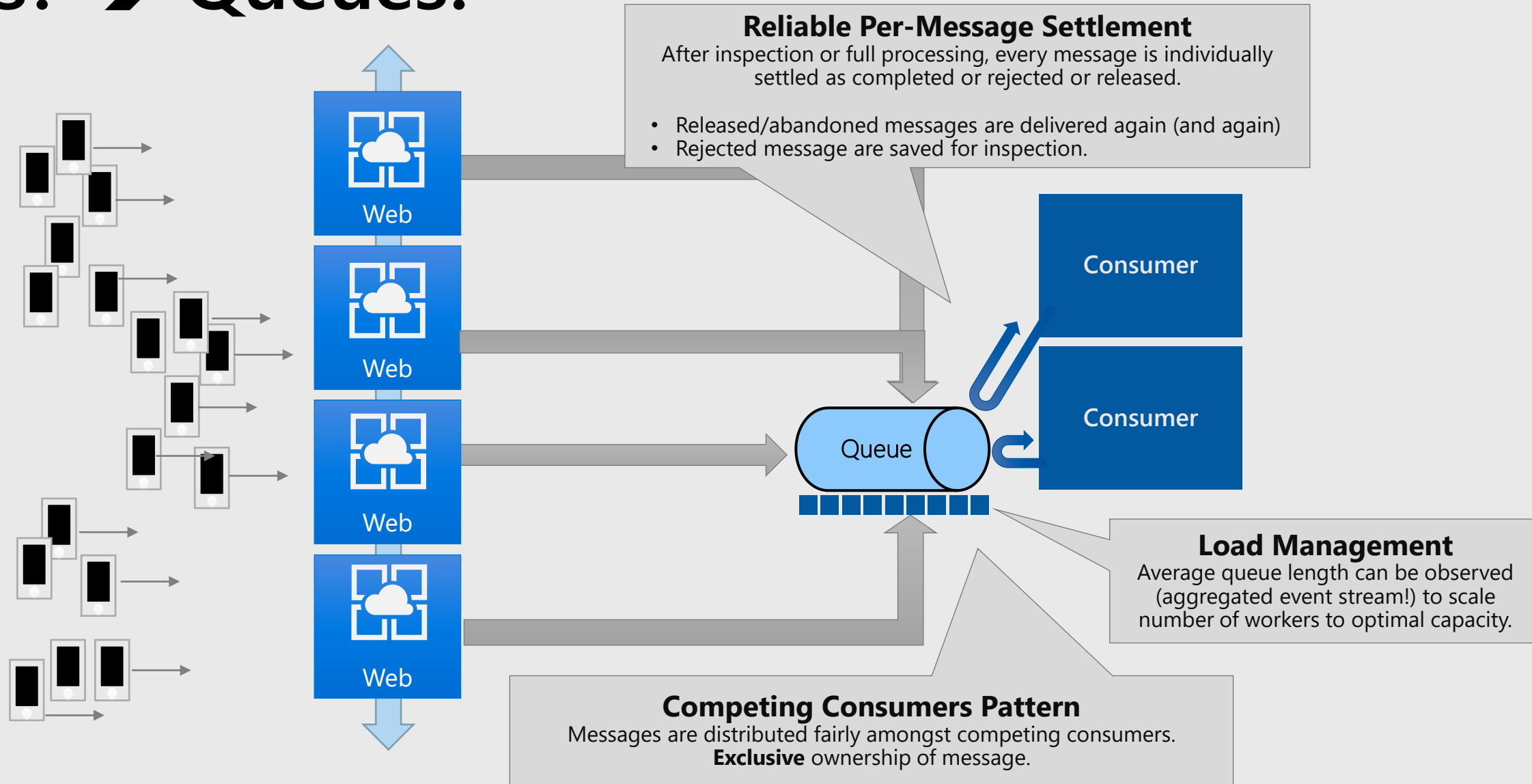
Event Stream Aggregator

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

Jobs? → Queues!



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

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

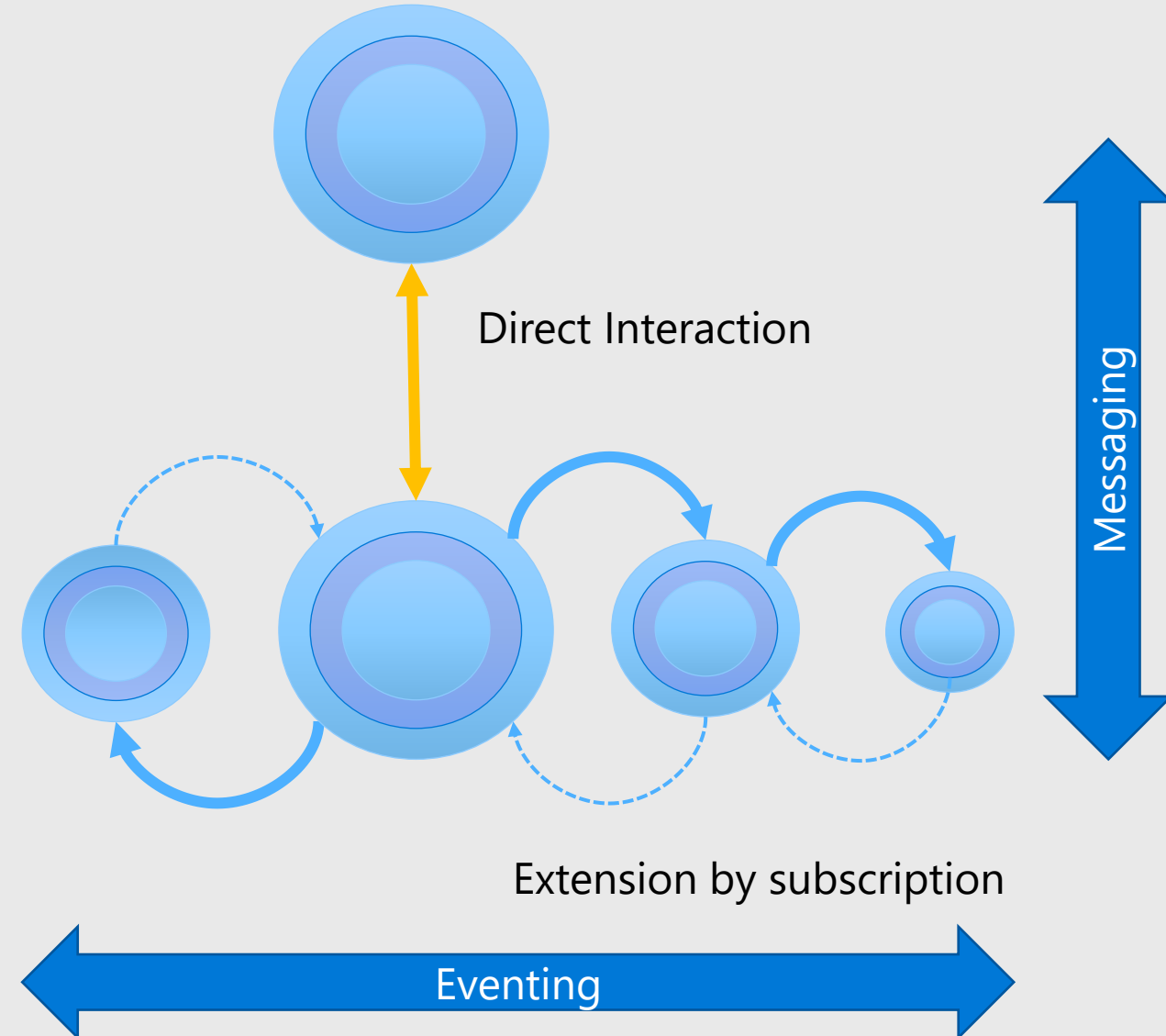
Modern apps use eventing and queue-based messaging

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

Commands, Requests

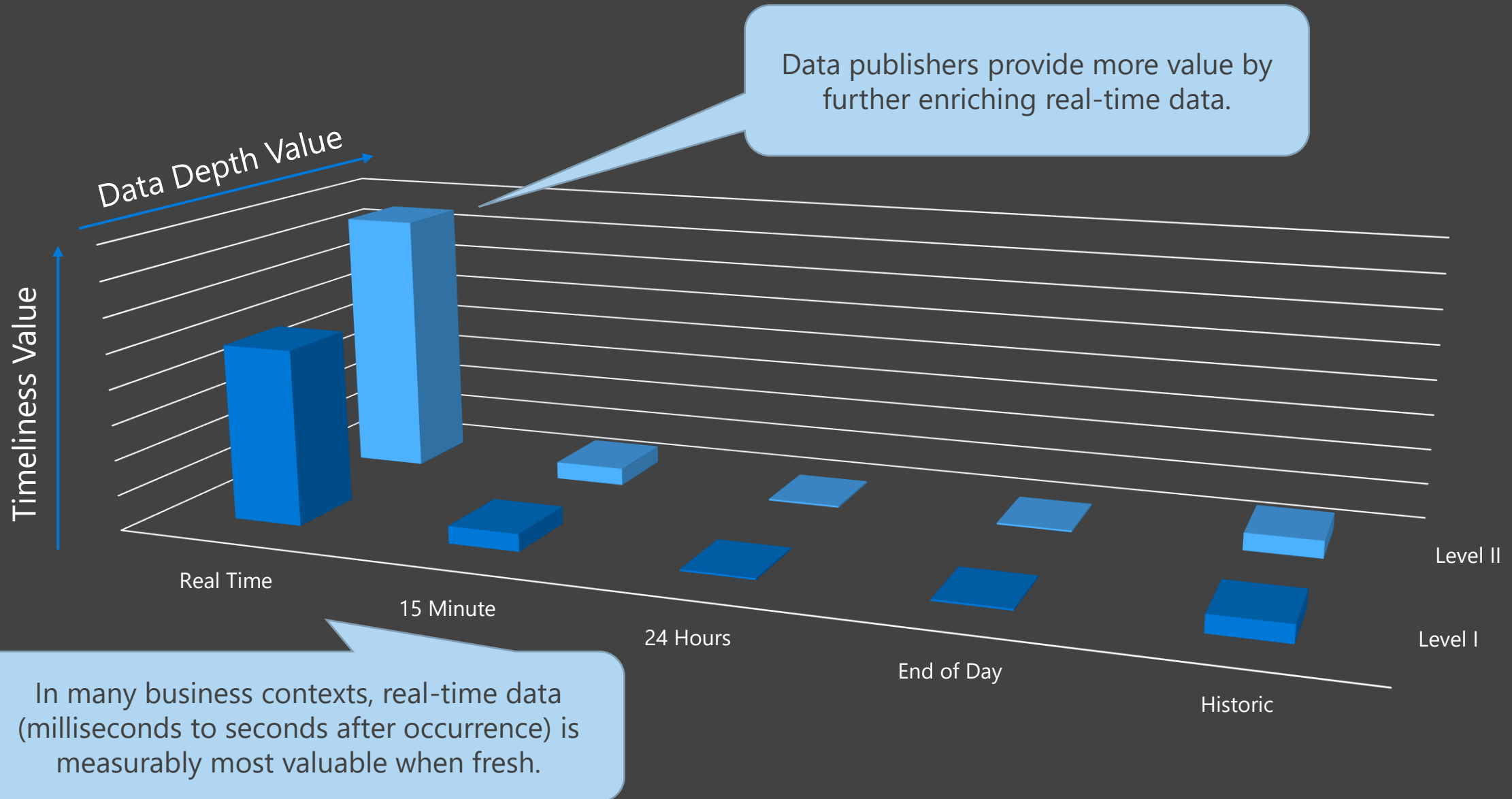
**Extensions react to events or
insights derived from event
streams emitted by services.**

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

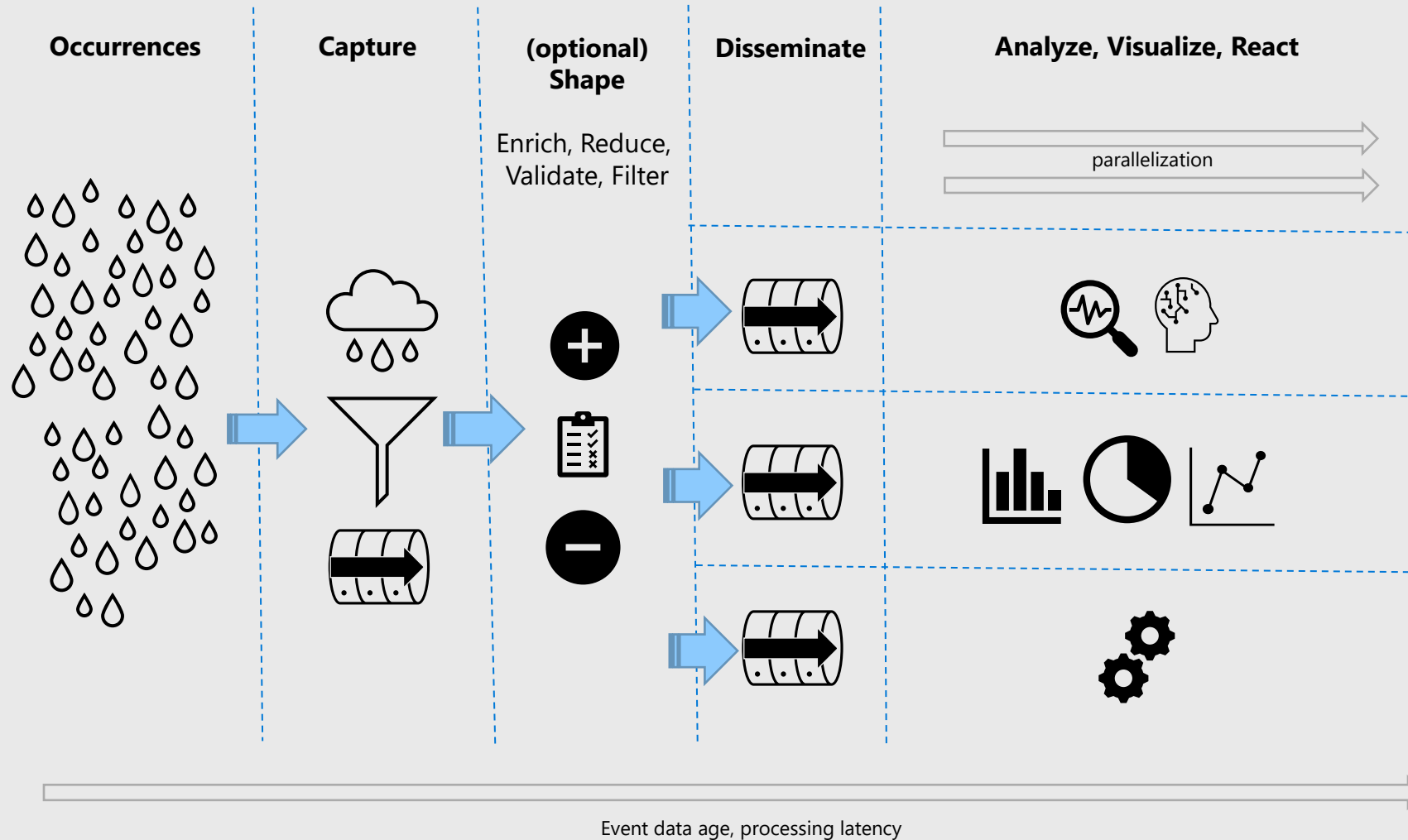


Event Streams and Time(-liness)

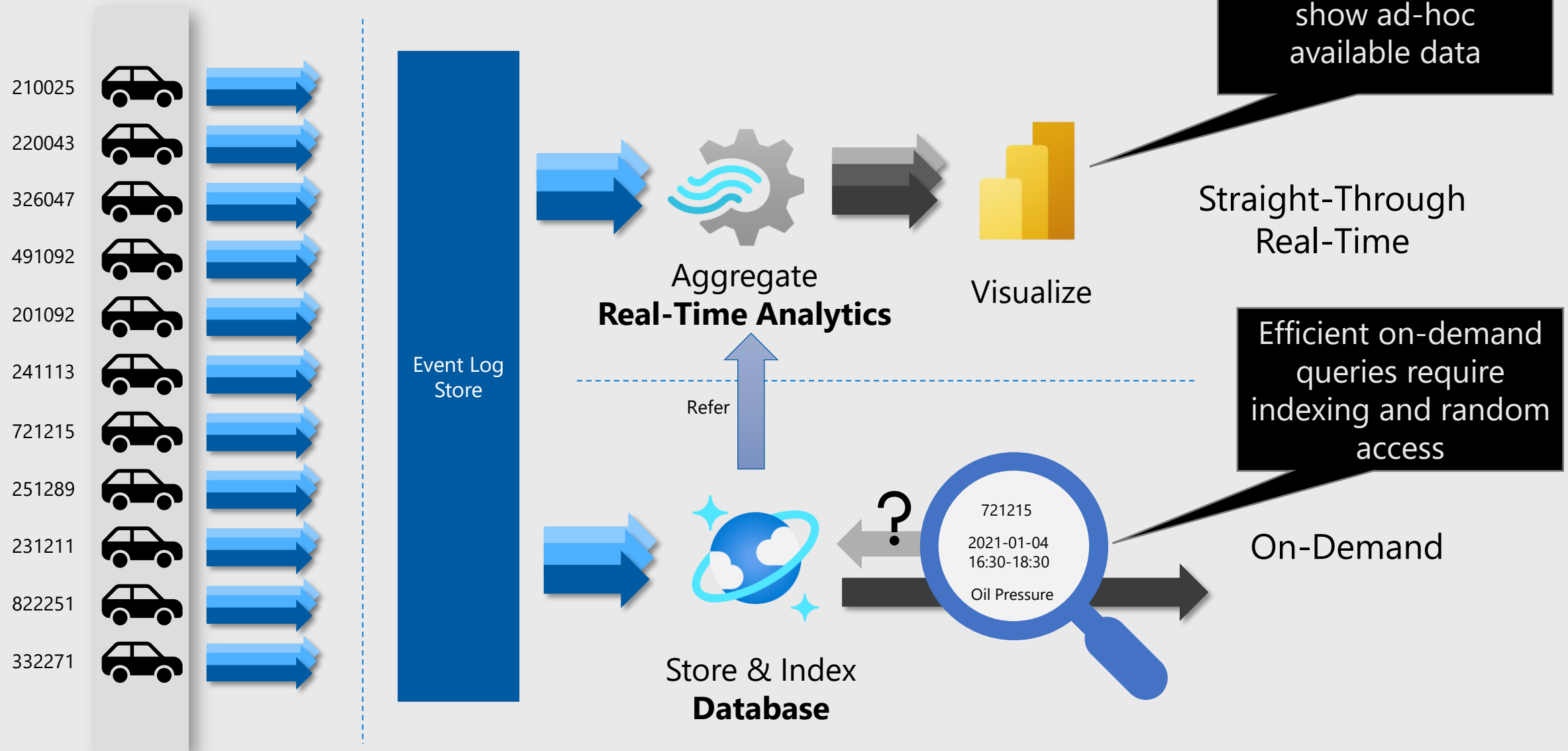
Event Data Value – Securities Markets



Velocity Matters → Parallelization Matters



Event Streams and Context

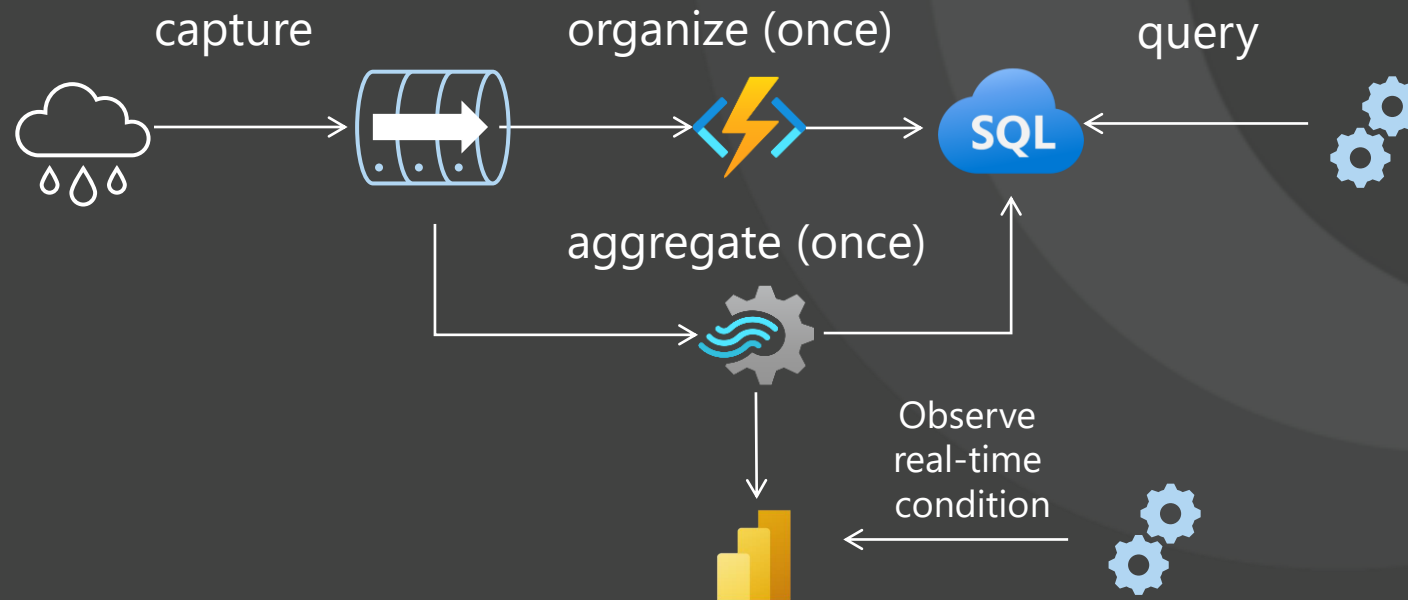


Event Stream Engines are lousy Data Lakes or Databases*

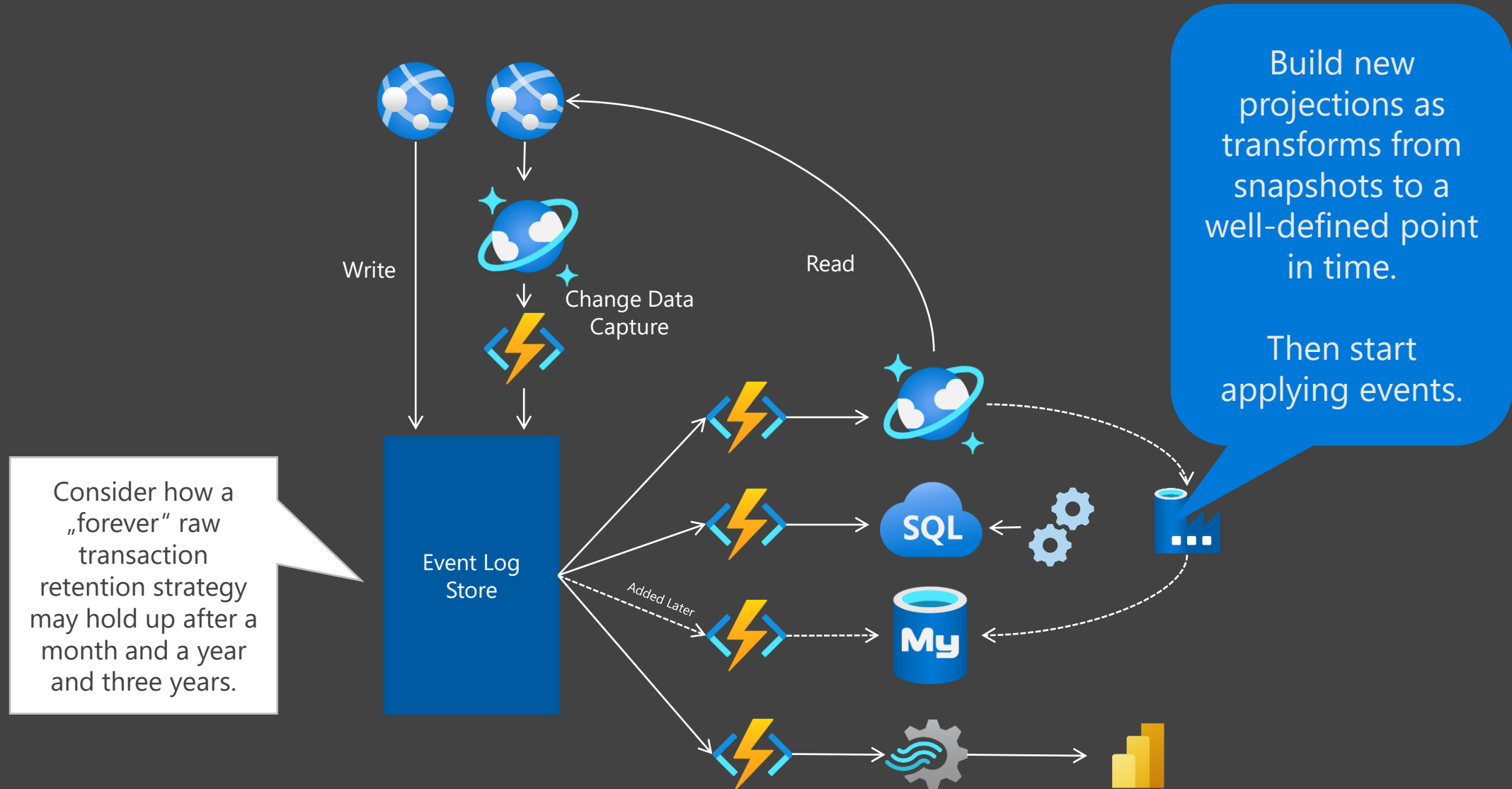
- A topic is a partitioned table
- Records are byte arrays
- Only lookup index is the time axis

*yes, Apache Kafka, too

Events are captured facts that won't change. Keep created projections in a database.

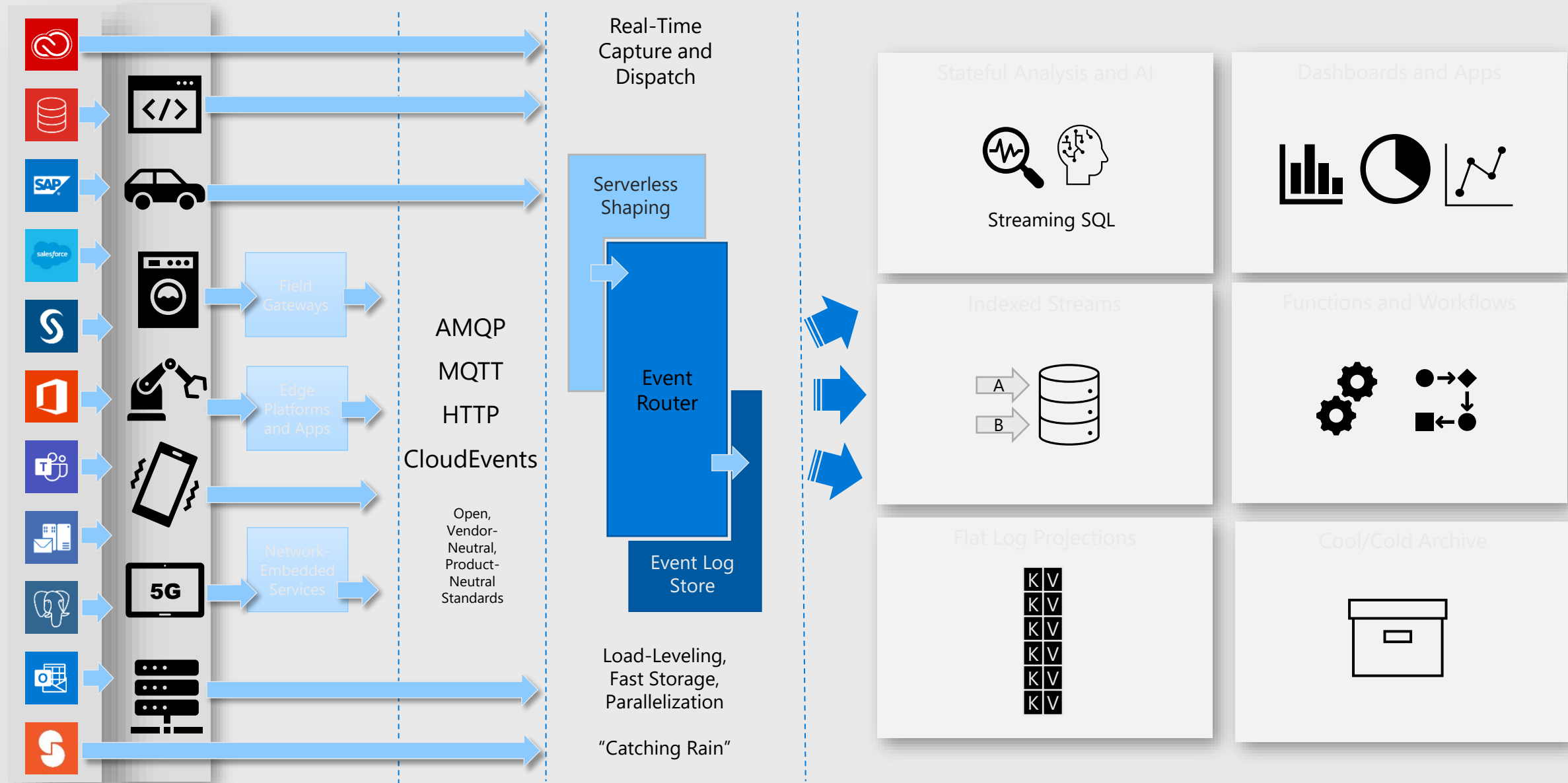


Event Sourcing



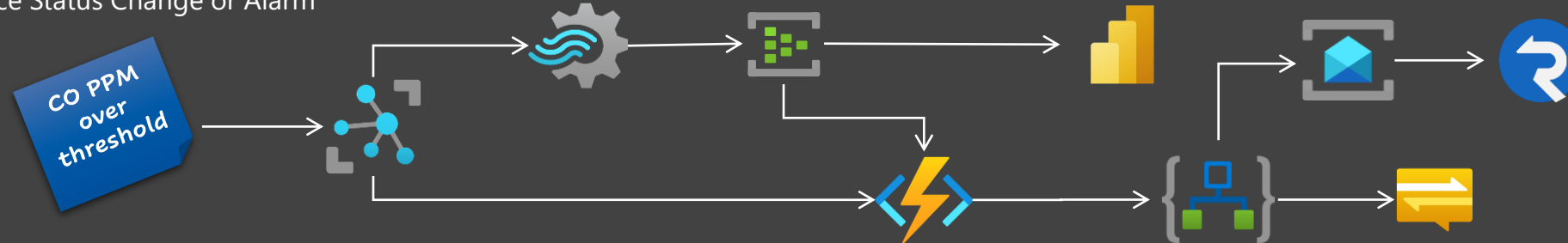
Event Journeys

Federation: Event Journeys

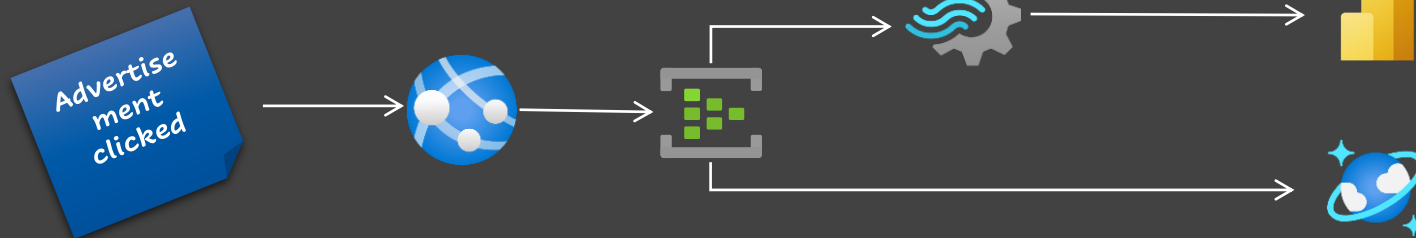


Event Journeys

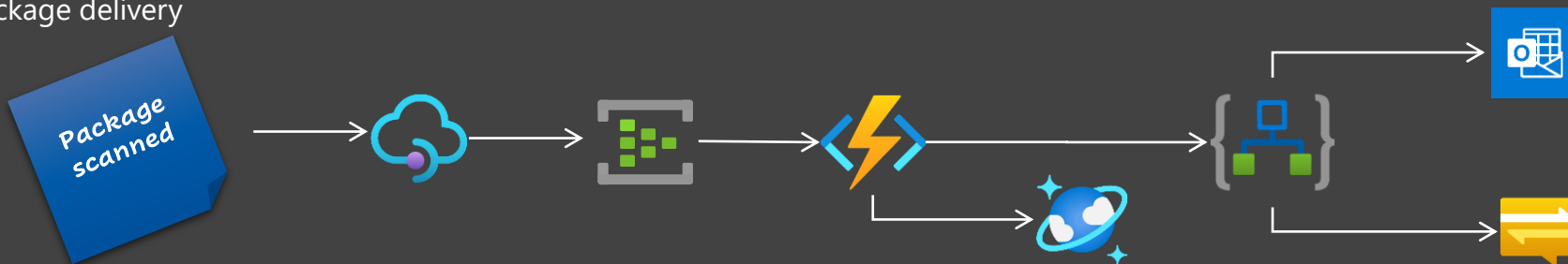
Device Status Change or Alarm



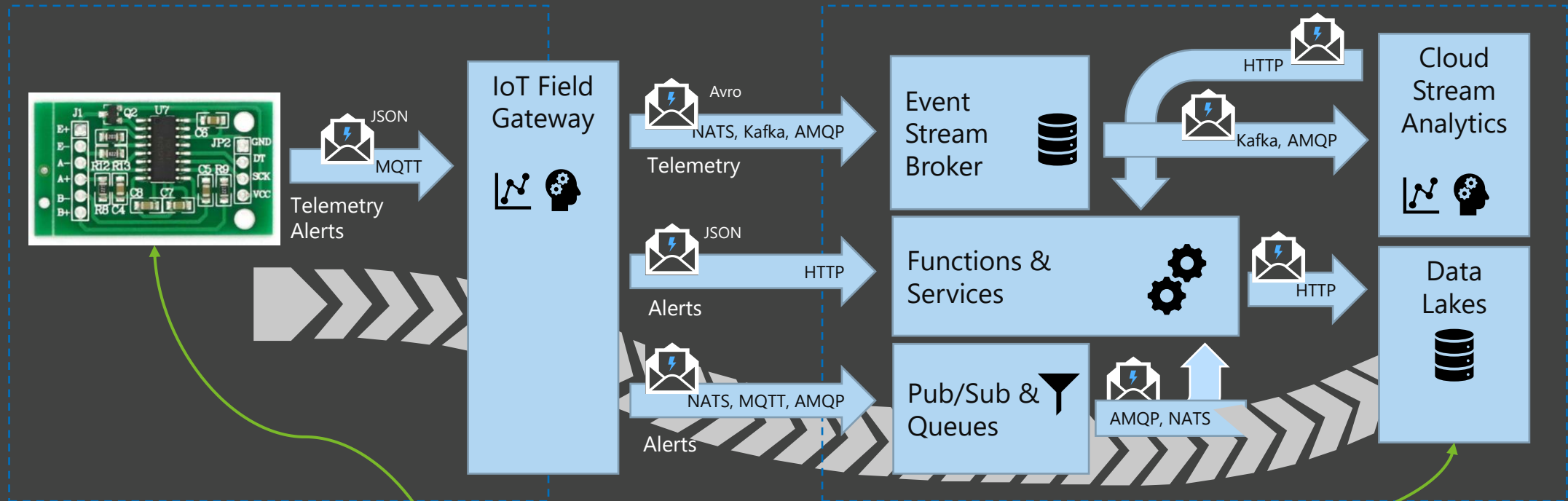
Web site visitor statistics



Package delivery



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?

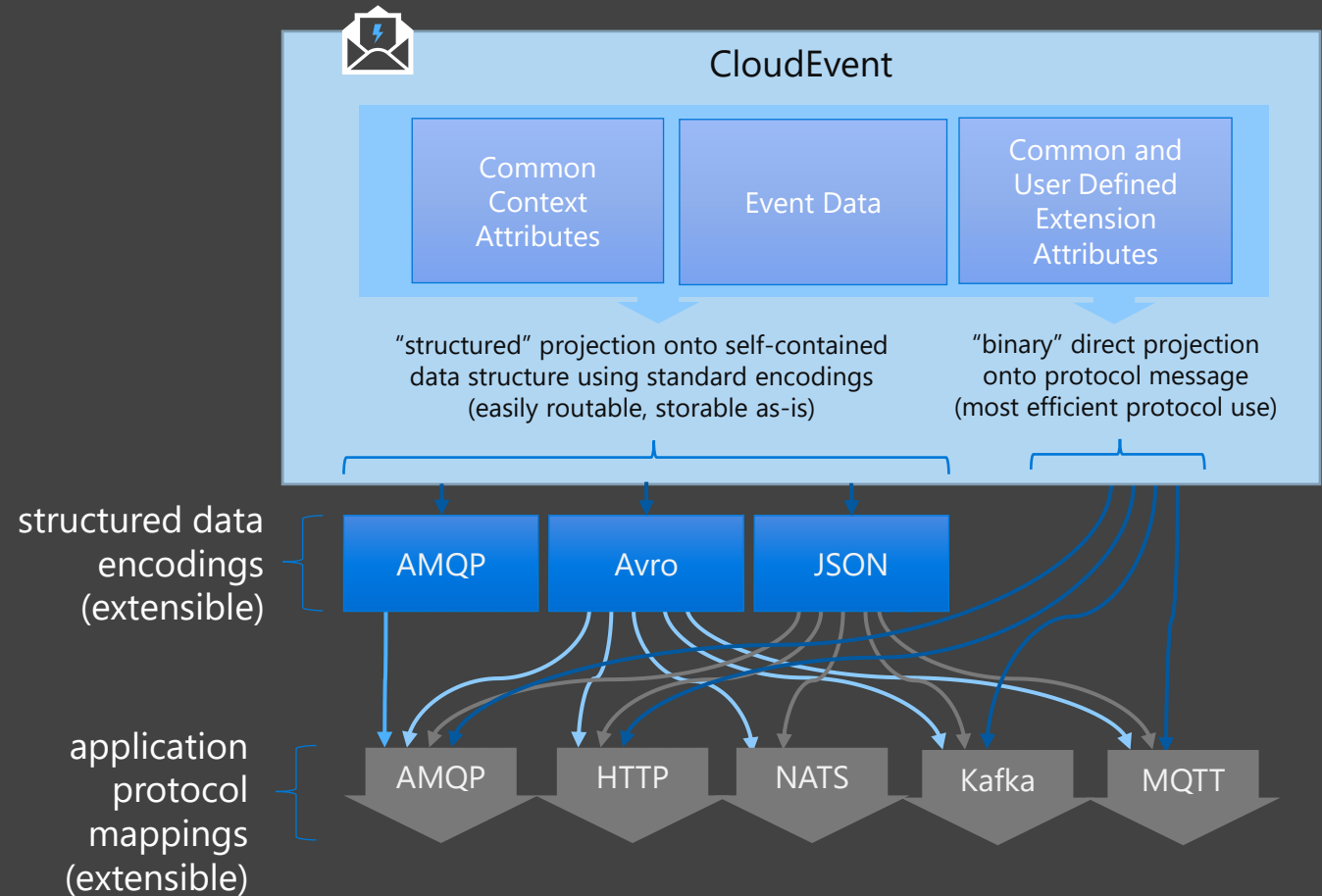
Binds to existing standard application protocols. Not a new protocol.

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

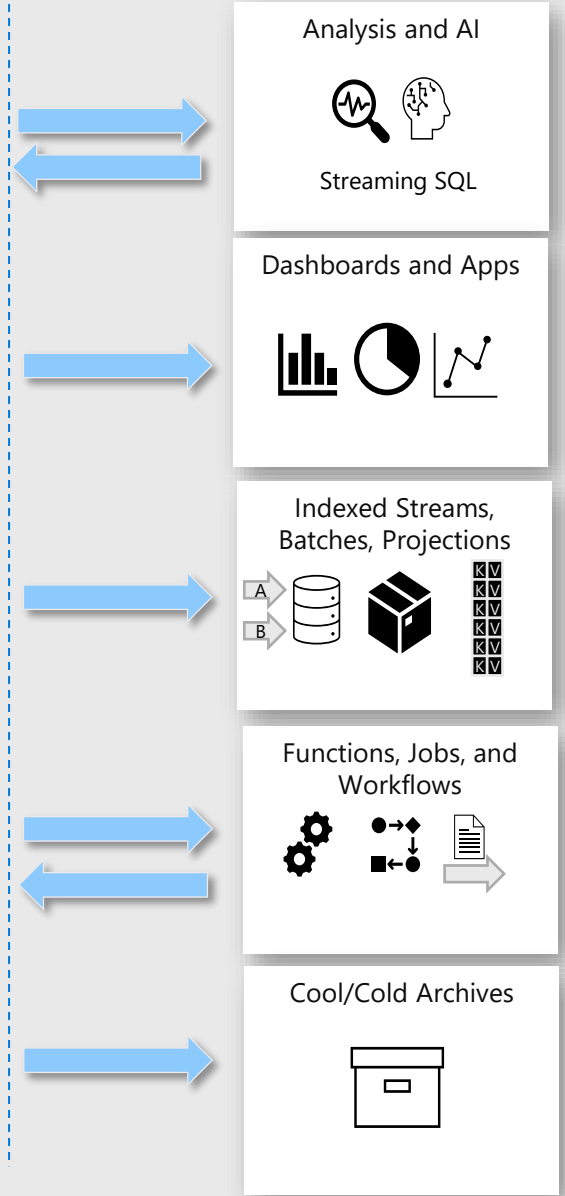
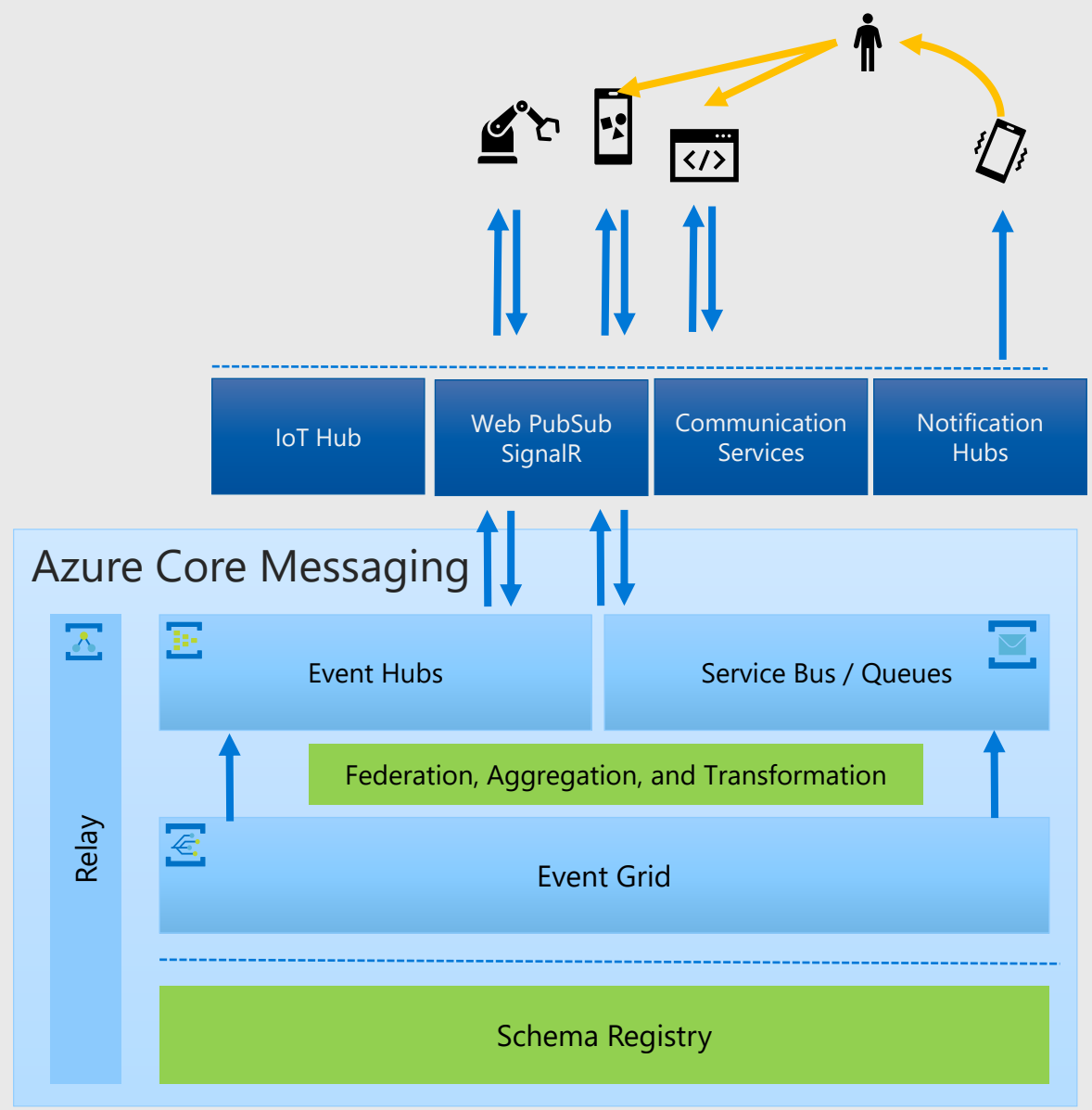
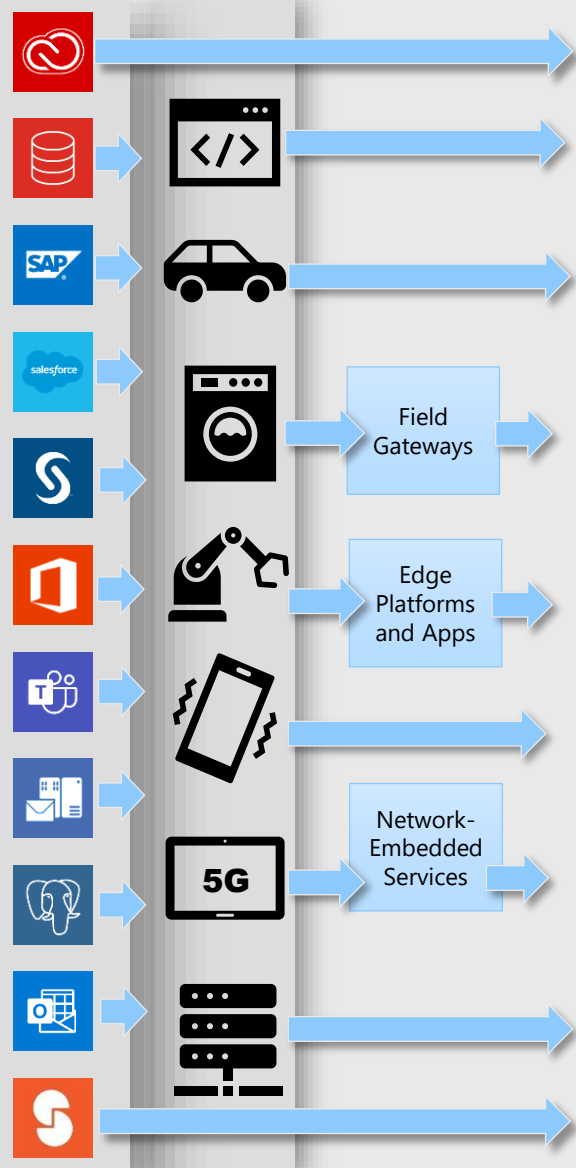
API definition for managing and discovering event templates at development time

Discovery

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

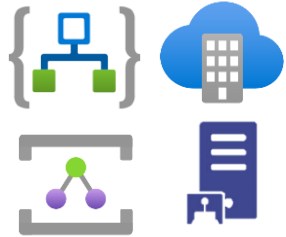
Subscription

API definition for subscribing to events across multiple protocols



Eventing on Azure Cloud

SaaS Platform Integration



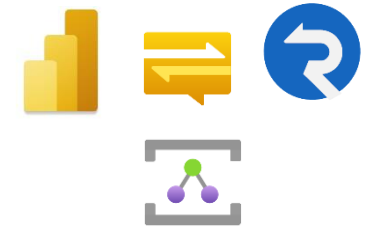
Open, Vendor-Neutral, Product-Neutral Protocols



Stateful Analysis and AI



Dashboards and Apps



EDI Integration



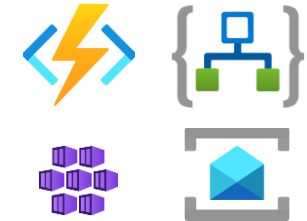
Event Stream Capture and Streamed Delivery



Indexed Streams and Batches



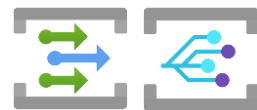
Functions, Jobs, and Workflows



Database Change Capture



Discrete Event Capture and Subscriber Delivery



Flat Log Projections



Cool/Cold Archives



