Building Event-driven Architectures in Azure

Johan Ludvig Brattås, Director @ BDO

Working With Event-Driven Data in Microsoft Fabric

Matt

Catalog Your Data In Motion With Fabric Real-Time Hub & Eventstreams

Wednesday, 19 Nov

11.30 AM – 12.30 PM

Abhi

Analyze Your Real-Time Data With Fabric Eventhouse & KQL

Thursday, 20 Nov

10.15 AM – 11.15 AM

Kristyna

Unlock Real-Time Insights With Fabric Real-Time Dashboards and Kusto-Powered Power BI Reports

Thursday, 20 Nov

2.30 PM – 3.45 PM

Frank

Activator and Further Functionality Within Fabric Real-Time Intelligence

Friday, 21 Nov

10.15 AM – 11.15 AM

Johan

Building Event-Driven Architectures in Azure and Fabric

Friday, 21 Nov

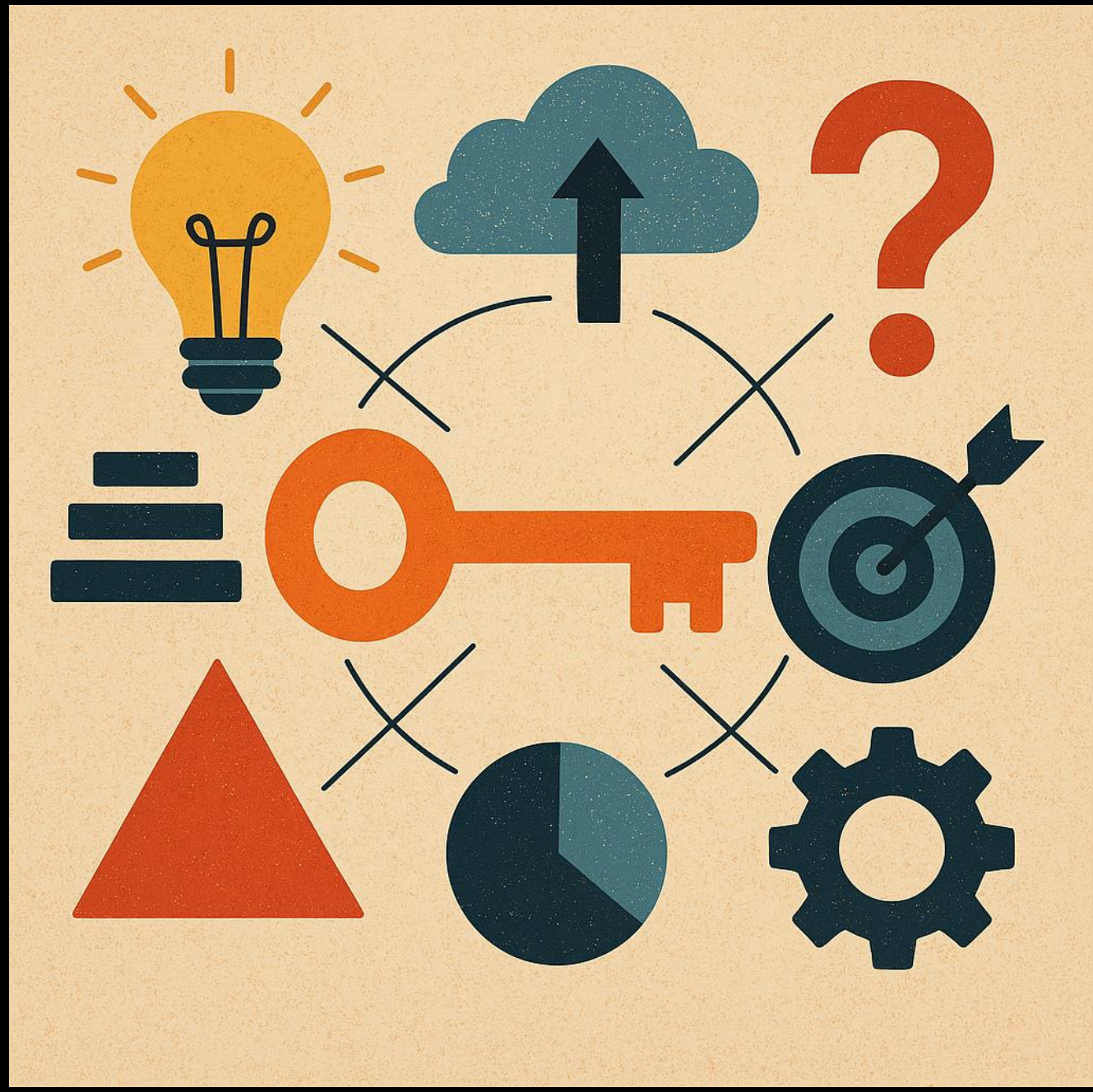
2.00 PM – 3.15 PM

Agenda

- Key terminology
- Event-driven Architecture
- Azure & Fabric event services
- What to choose when
- Some event-driven architectures

Key Terminology

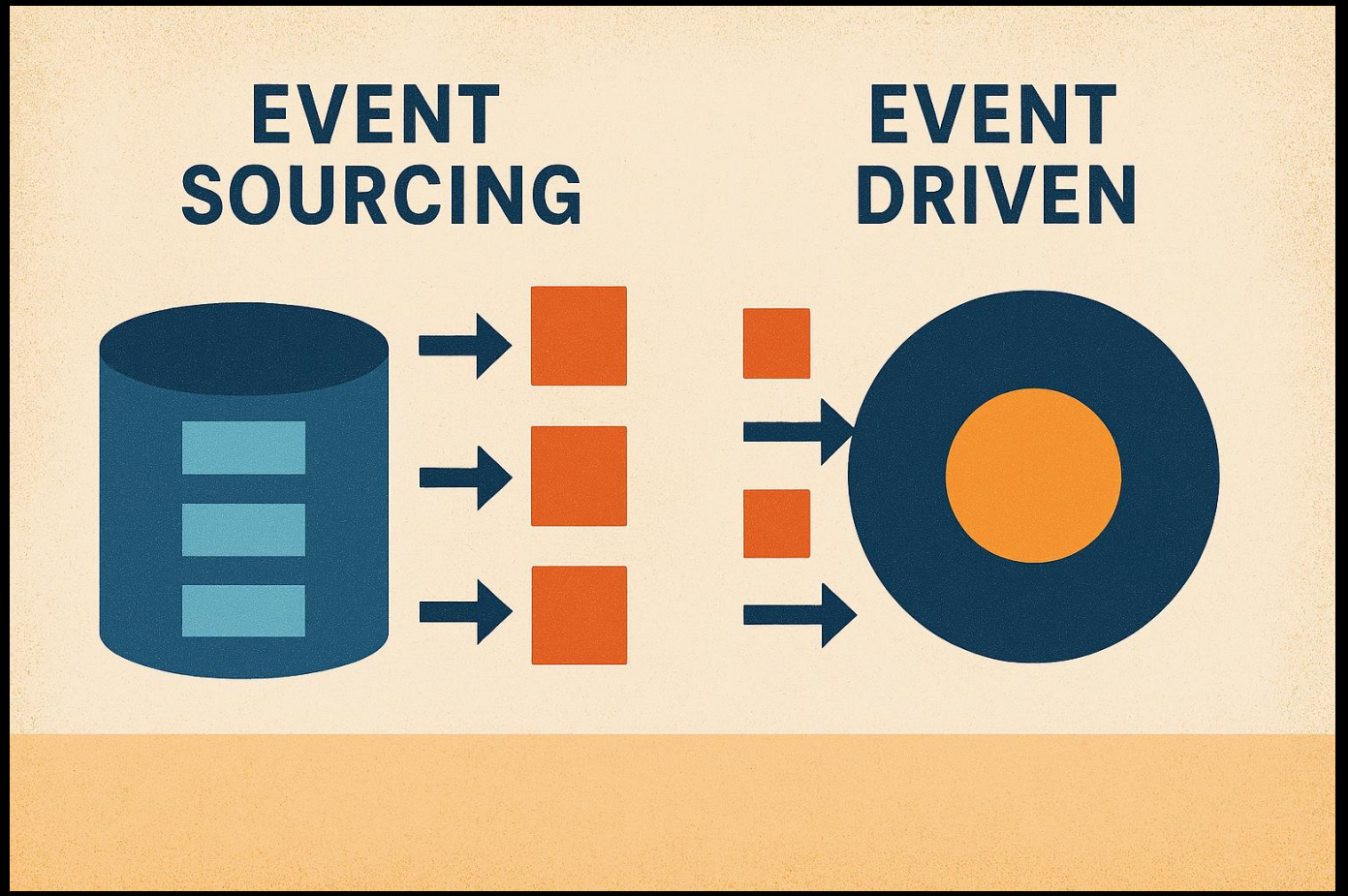
- Message: Generic payload of information from one system to another
- Event: A significant change of state
- Event Notification: Often called an «event» but is the immutable, historic message about an event
- Command: A message from one system to another with information on action to be taken



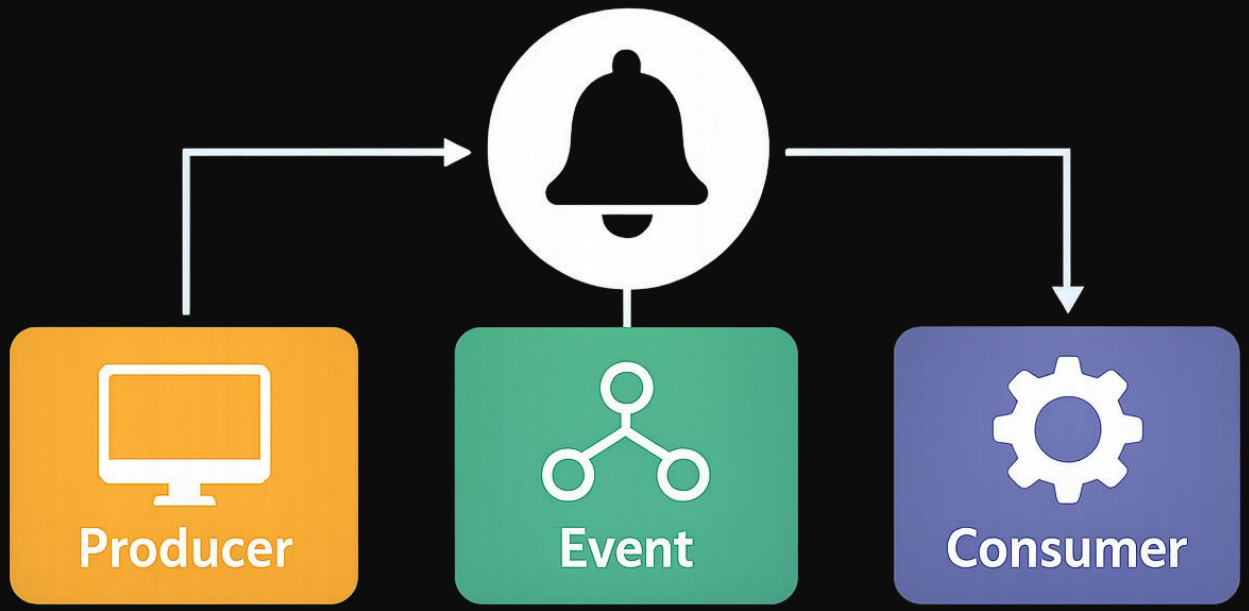
Key Terminology

Event Sourcing: A data-persistence pattern where events are the source of truth for state

Event-driven Architecture: A design pattern focusing on communication between services/components



EVENT-DRIVEN ARCHITECTURE



Event-driven architecture is not just for analytics

Three main components:

- Producer
- Event Channel
- Consumer

Asynchronous, decoupled communication between systems.

Central enabler of microservices and other distributed systems

Event-Driven Architecture

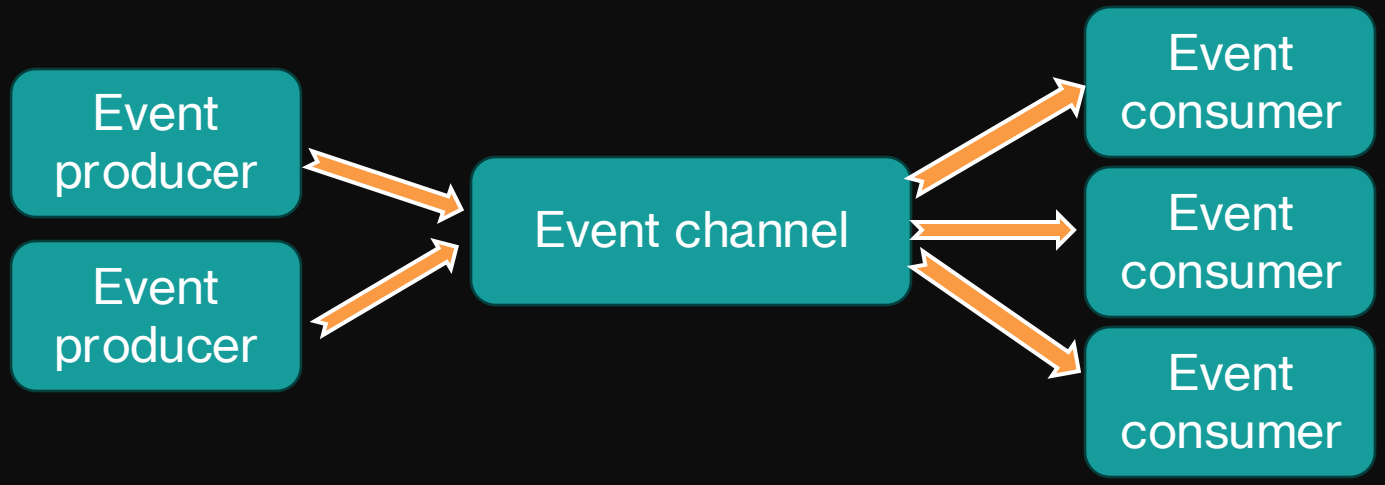
Producers are decoupled

Every consumer is decoupled

Event channels allow all consumers to see all events

The opposite of Competing Consumer pattern

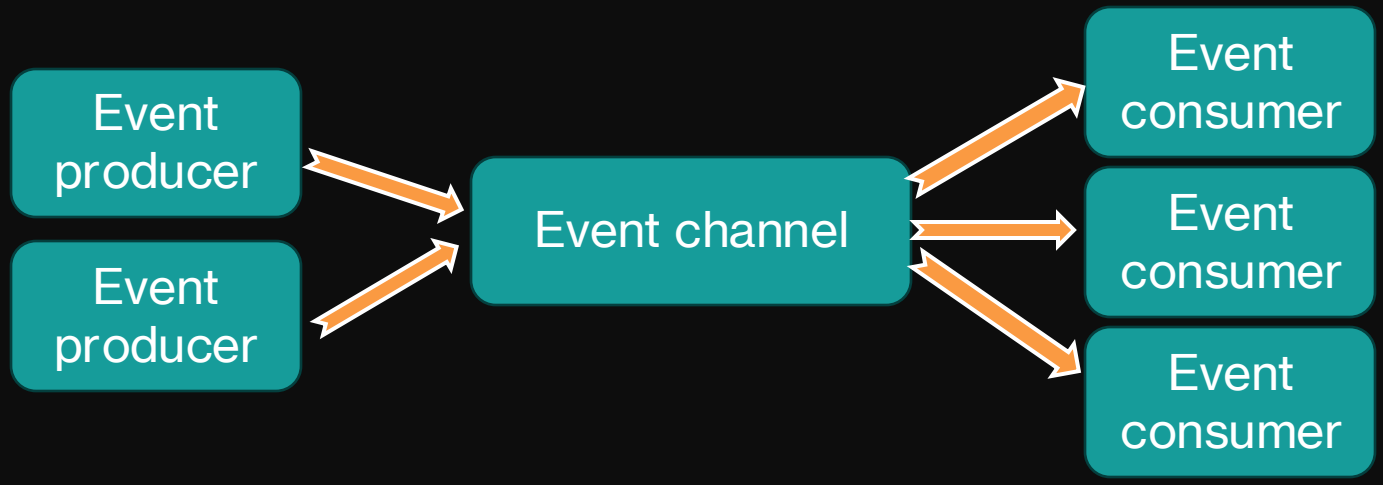
Two event models:
Publish-Subscribe – or PUB/SUB
Eventstreaming



Event-Driven Architecture

Consumer side variations

- Simple event processing
- Basic event correlation
- Complex event processing
- Event stream processing

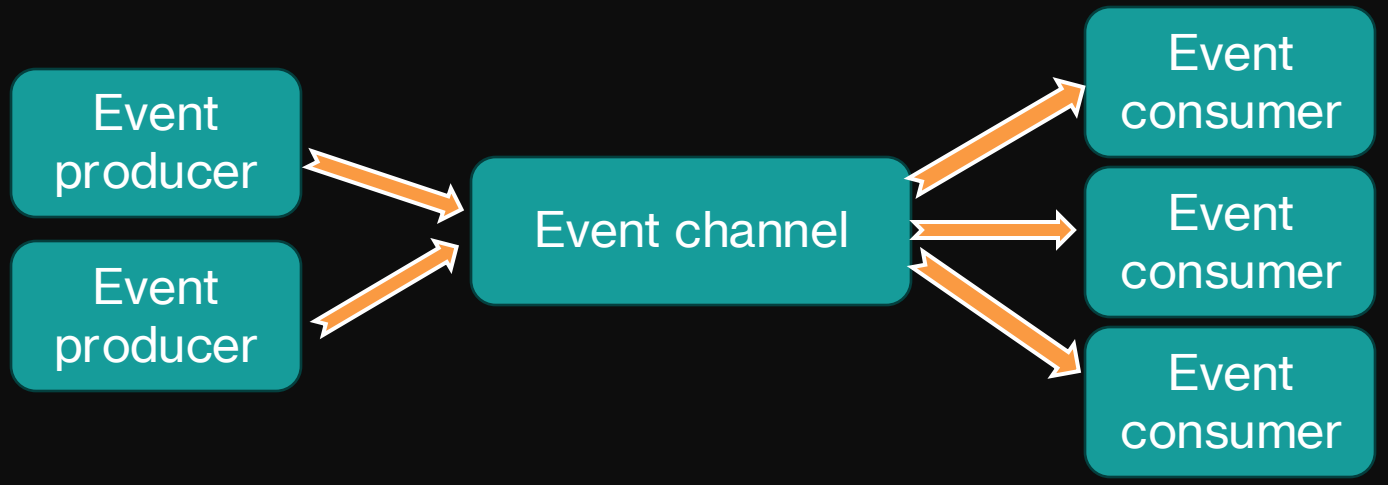


Event-Driven Architecture

Primary toplogies

- Broker topology
  - Broadcast events to everyone
  - Highly decoupled
  - For simple event processing

- Mediator topology
  - Event mediator manages and controls flow
  - Maintains state
  - Broadcasts commands, not notifications
  - Broadcasts only to designated channels



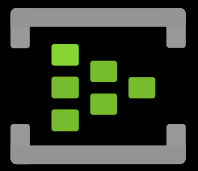
Event-Driven Architecture

When to use Event-driven architecture

- Multiple systems needs the same events
- Real-time processing
- Complex event processing
- High volume + high velocity
- Need for decoupling for scalability and reliability

- Have analytical workloads that don’t fit in traditional scheduling

Event Services in Azure



Eventhubs – for Eventstreaming scenarios. High throughput, messages up to 5 MB

Eventgrid – for PUB/SUB scenarios. Supports both PULL and PUSH scenarios. Can communicate over MQTT and HTTPS

Servicebus – Enterprise Messaging service. Not the most used in event-driven architectures

IOT Hub– for Eventstreaming scenarios with specific needs. High throughput, small messages.

Event Services in Azure

Functions – flexible and powerful. Can be used in analytical worklads on event triggers as well as simple event processing with Eventgrid or Servicebus.

Logic Apps – low code/no code alternative to Functions.

Stream Analytics – stream processor. Often used in complex event processing scenarios and sometimes in event stream processing

Data Explorer – powerful event store. Fits both event sourcing and event-driven architectures. Can be used for simple event processing and analytical loads as well as event stream processing...

Event Services in Fabric

Eventhouse / KQL DB – powerful event store. Fits both event sourcing and event-driven architectures. Can be used for simple event processing and analytical loads as well as event stream processing...

Eventstream – Swiss Army knife... Works both for PUB/SUB and event streaming. And for most variations

Data Activator – takes action. Triggers on specified commands or events. Low code/No code

What services should I use?

Azure or Fabric?

For Azure:

Event channel: Default choice of Event hub unless need for PUB/SUB or IoT specific functionality

For complex event streaming – Stream Analytics

Simple event processing and analytical workloads – Functions unless need for no code/low code

Simple processing and need for event store Data Explorer can be a fit

What services should I use?

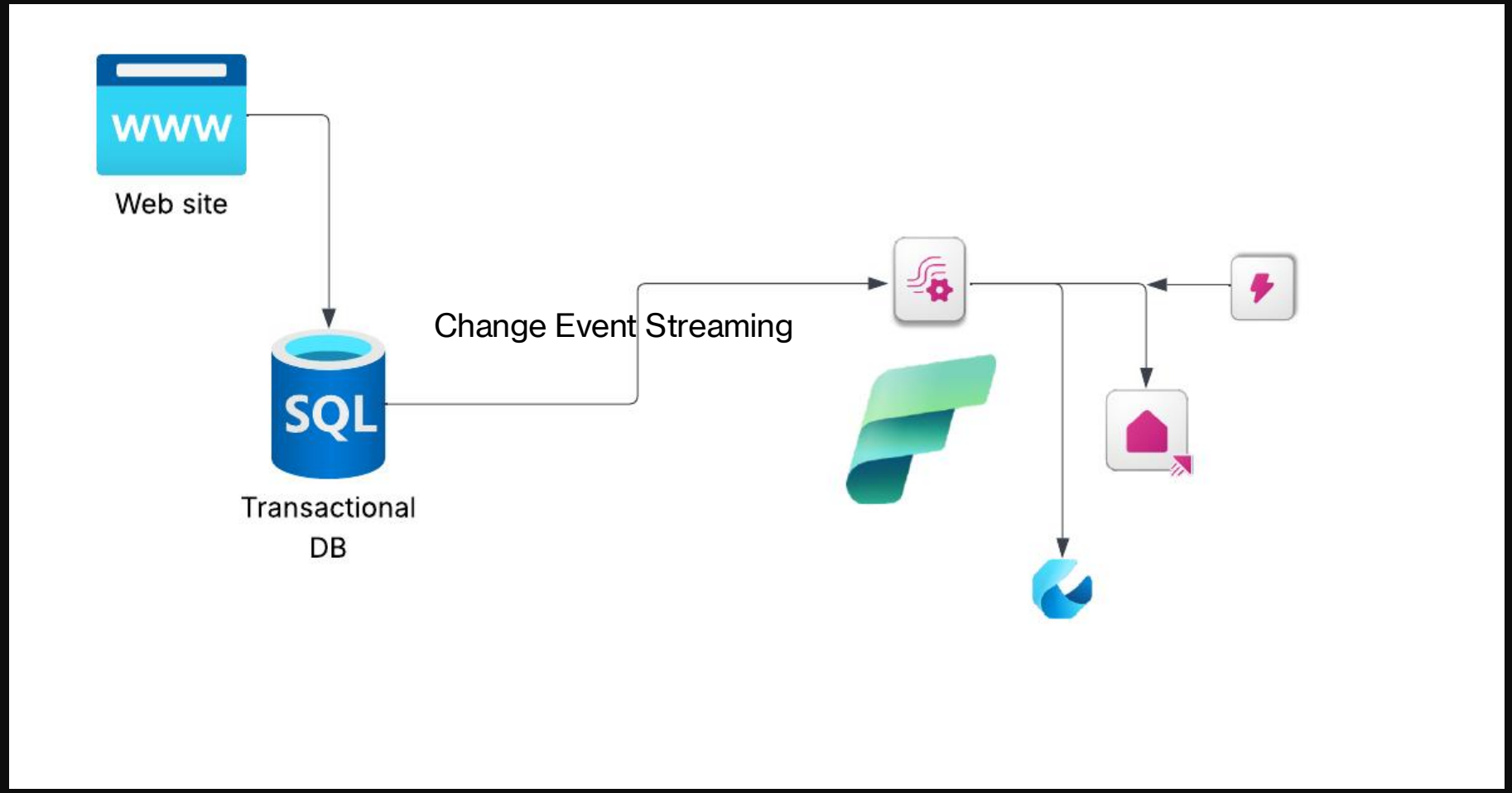
Azure or Fabric?

For Fabric the services rarely overlap.

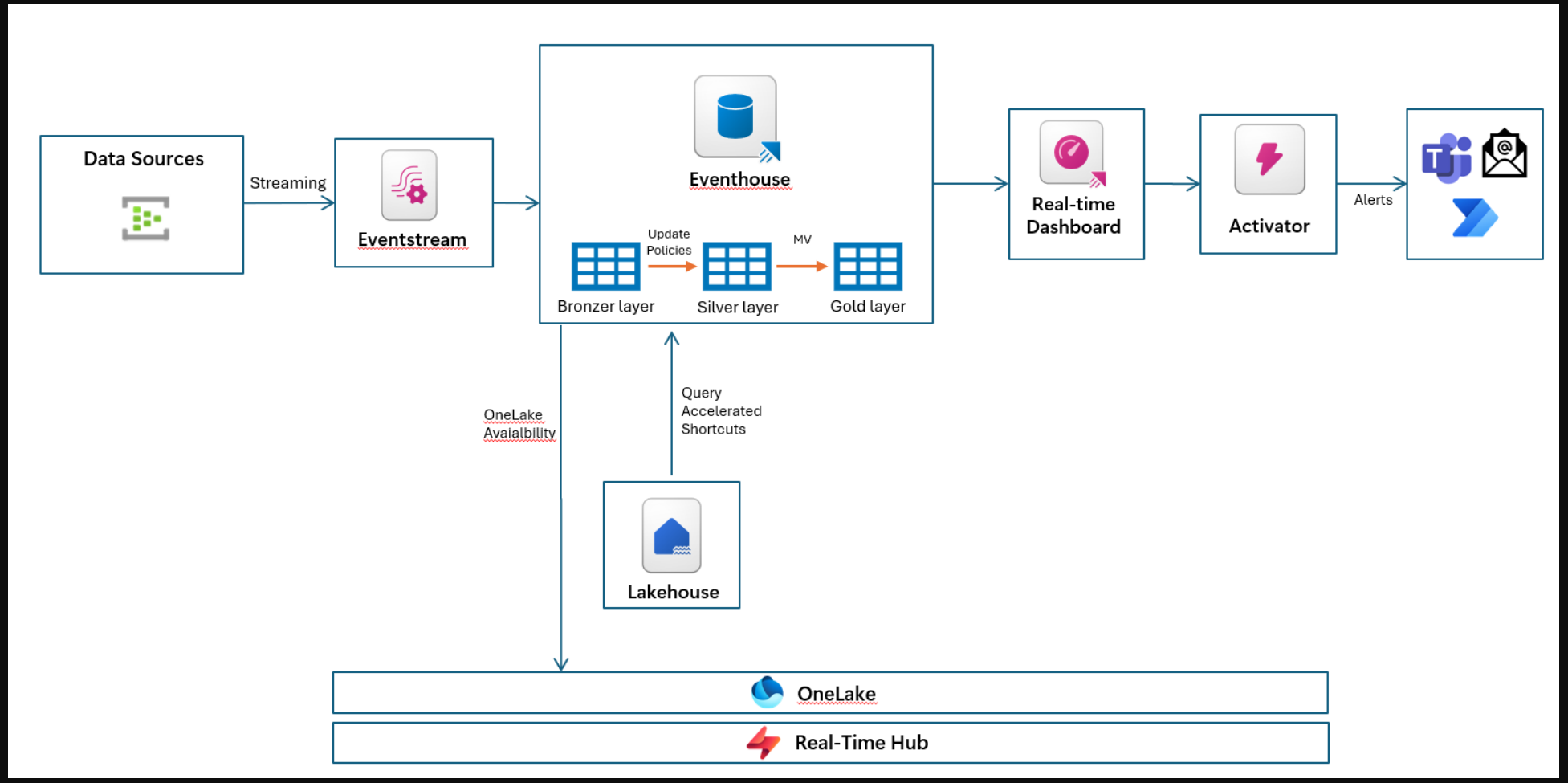
- Do I have an external event channel?
- Which consumer variation patterns do I have?
- Do I need to take action on events?
- Do I need an event store?

- If no external event channel, complex or event stream scenarios – then Eventstream has a place
- If external event channel and need for an event store KQL DB can be a good fit
- Taking actions on events – such as triggering a pipeline then Activator

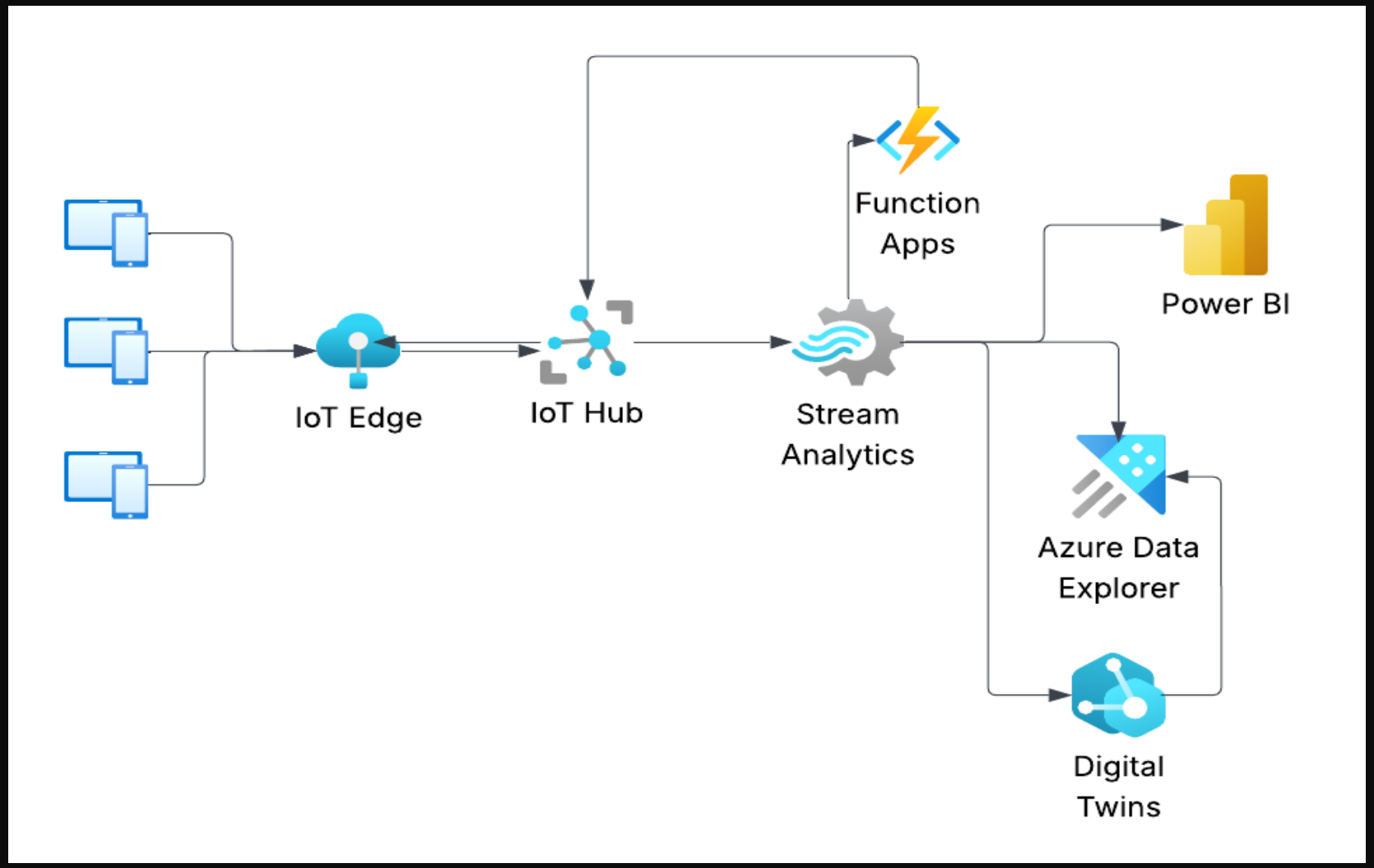
Event-Driven Architecture – Transactional to Analytics



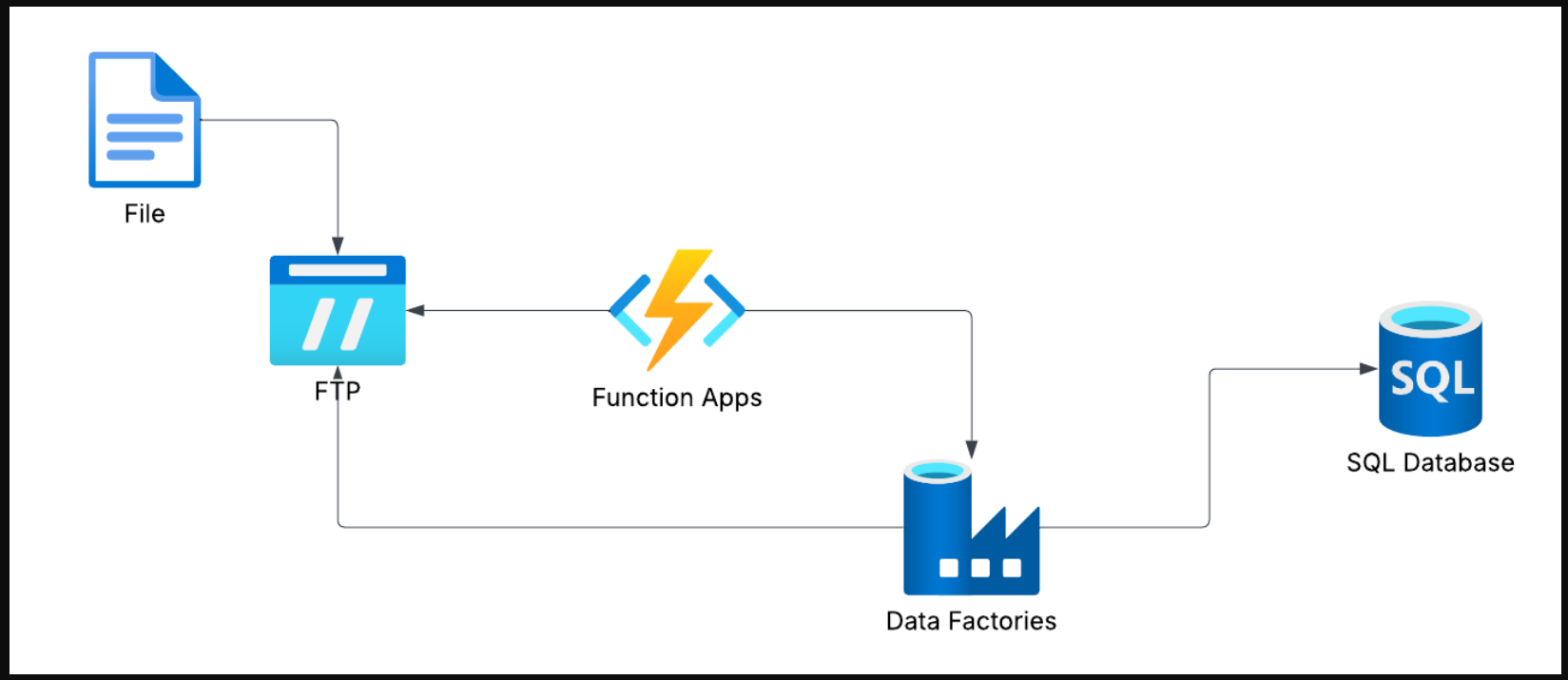
Event-Driven Architecture – Transactional to Analytics



Event-Driven Architecture – IoT



Event-Driven Architecture – Analytical workload




Demo

Your feedback is important to us

Evaluate this session at:

passdatacommunitysummit.com/evaluations

Johan Ludvig Brattås

Director, BDO

/johanludvig

@sqlclause.no

johan@brattas.no

GitHub

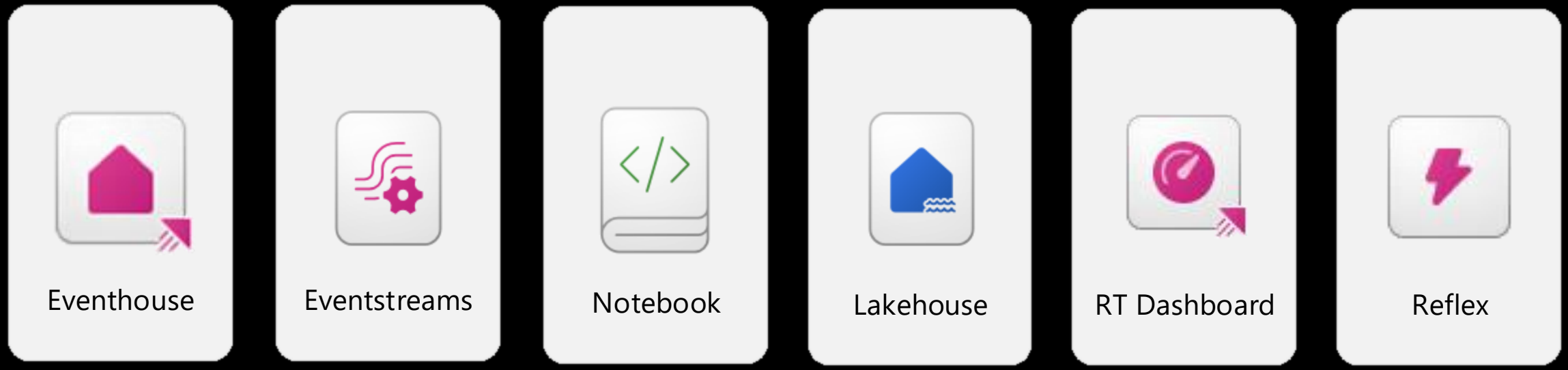
Chronic volunteer

Co-organizer – DataSaturday Oslo
President – MDPUG Oslo
Frequent voulenteer in general

When not geeking out over new tech

Teaching coeliacs how to bake gluten free
Baking
Hiking
Gardening

Try the Real-Time Intelligence lab at home



https://aka.ms/FabConRTITutorial