



# Conference Partners

Data Point Prague 2025



Data  
Brothers



GORDON & WEBSTER

CONSULTING AND INVESTMENT INC.





# Wi-Fi Access

Data Point Prague 2025

SSID: **DataPointPrague2025**

Password: **EnjoyFriends**

# Build A Fabric Real-Time Intelligence Solution in One Day

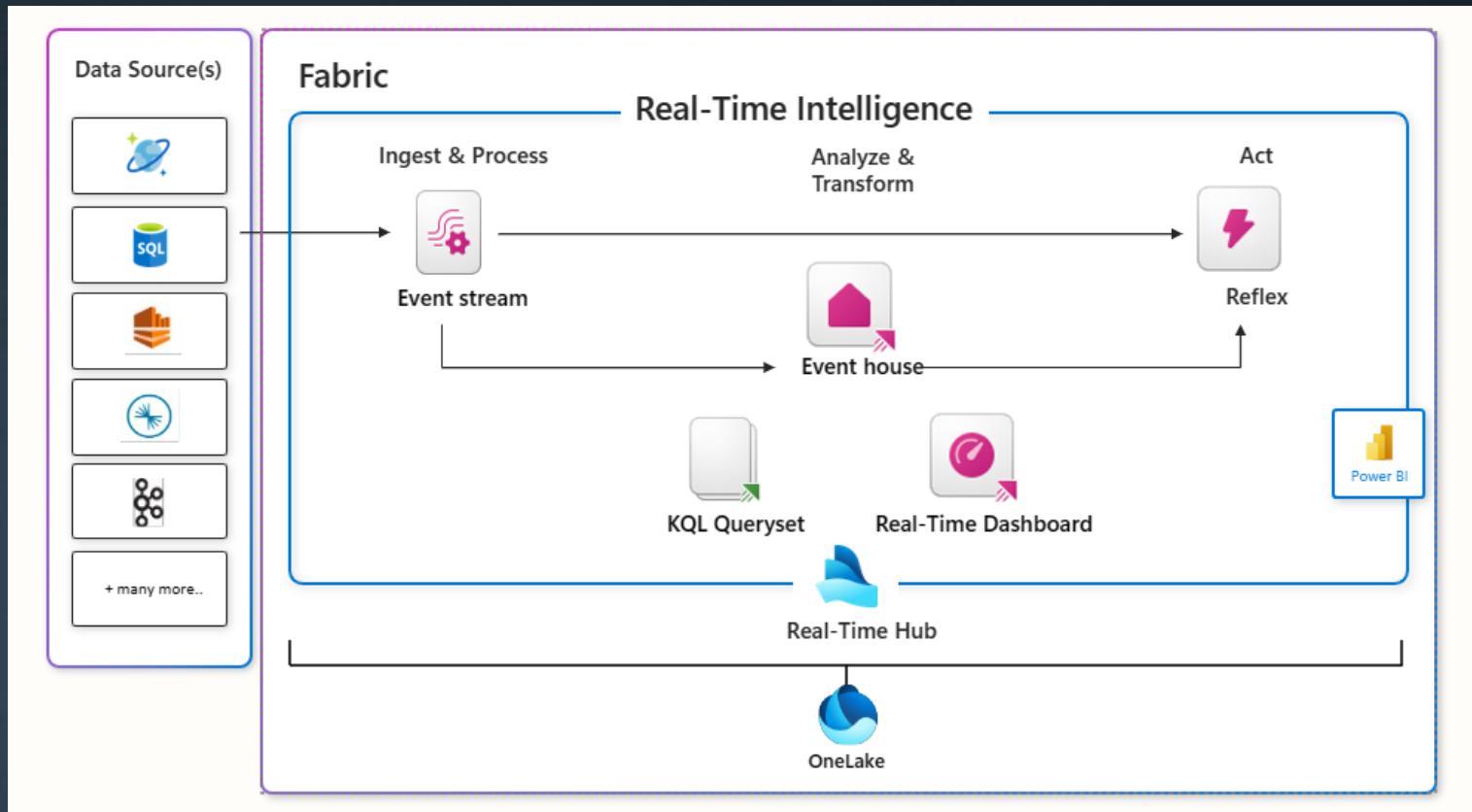
Johan Ludvig Brattås

# Agenda

- Introduction
- Overview of Fabric Real-Time Intelligence
- Eventstream & Real-Time hub
- Eventhouse
- Real-Time Dashboard
- Data Activator

# What is Fabric Real- Time Intelligence?

Powerful end to end  
solution for data in  
motion





There is a rapidly growing set of use cases that need ‘real-time’ speeds, generating decisions and actions at least **20 times faster** than the blink of an eye.”

Forbes, [“Understanding AI and ML in the real-time economy,” February 2024](#)



The world is exploding with high-granularity data...

# Telemetry – a key data for digital transformation



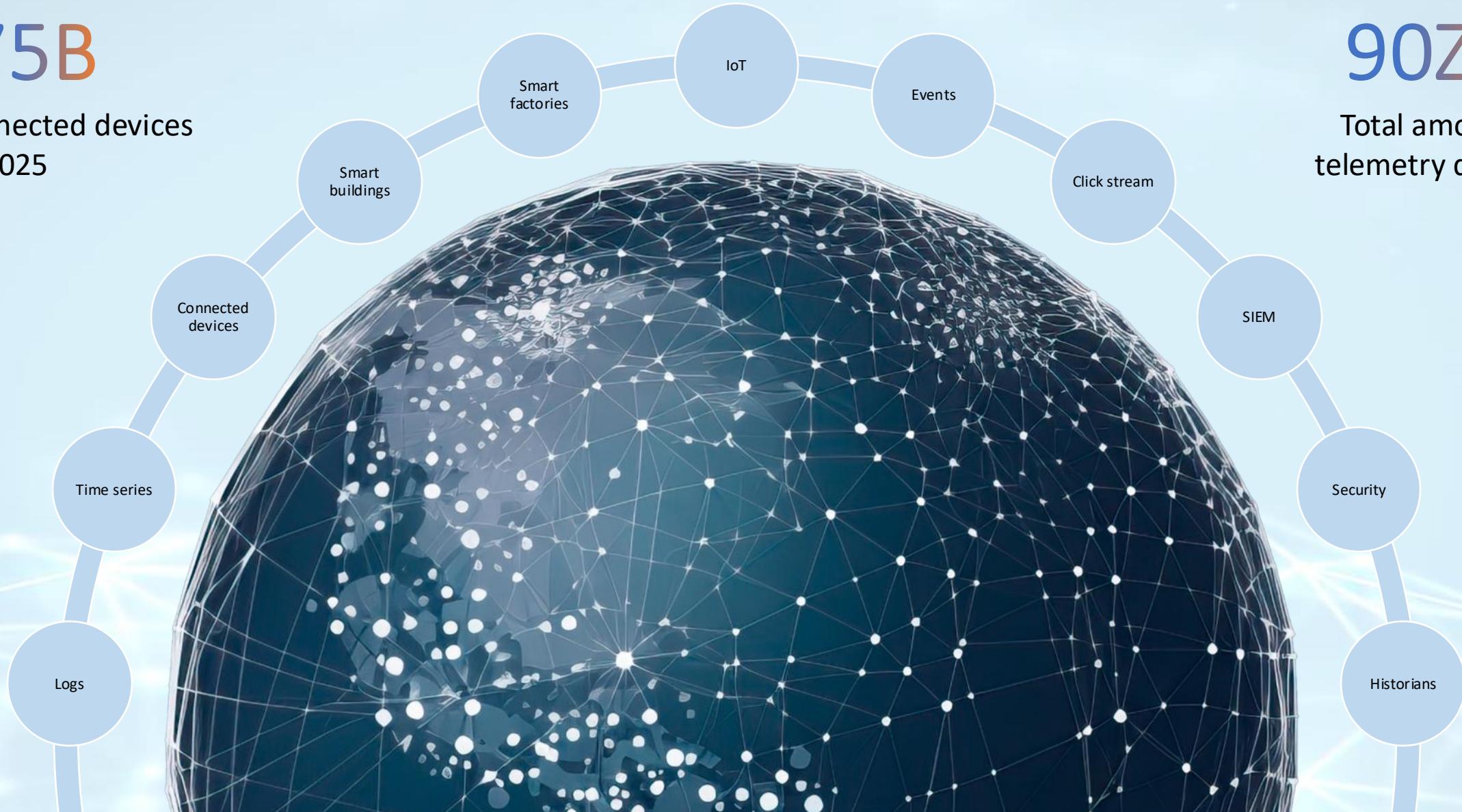
# Telemetry – a key data for digital transformation

75B

Connected devices  
by 2025

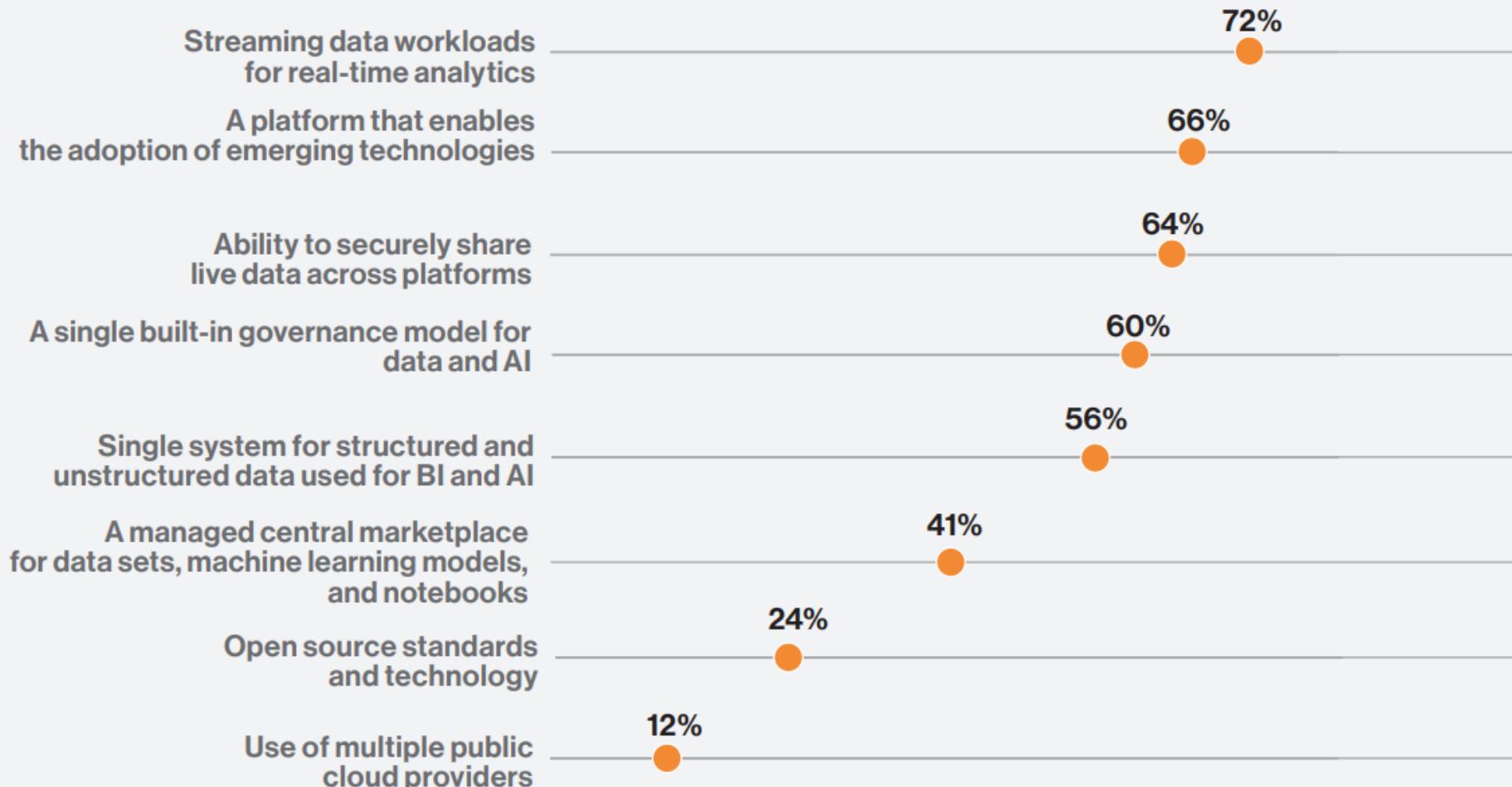
90ZB

Total amount of  
telemetry data by  
2025



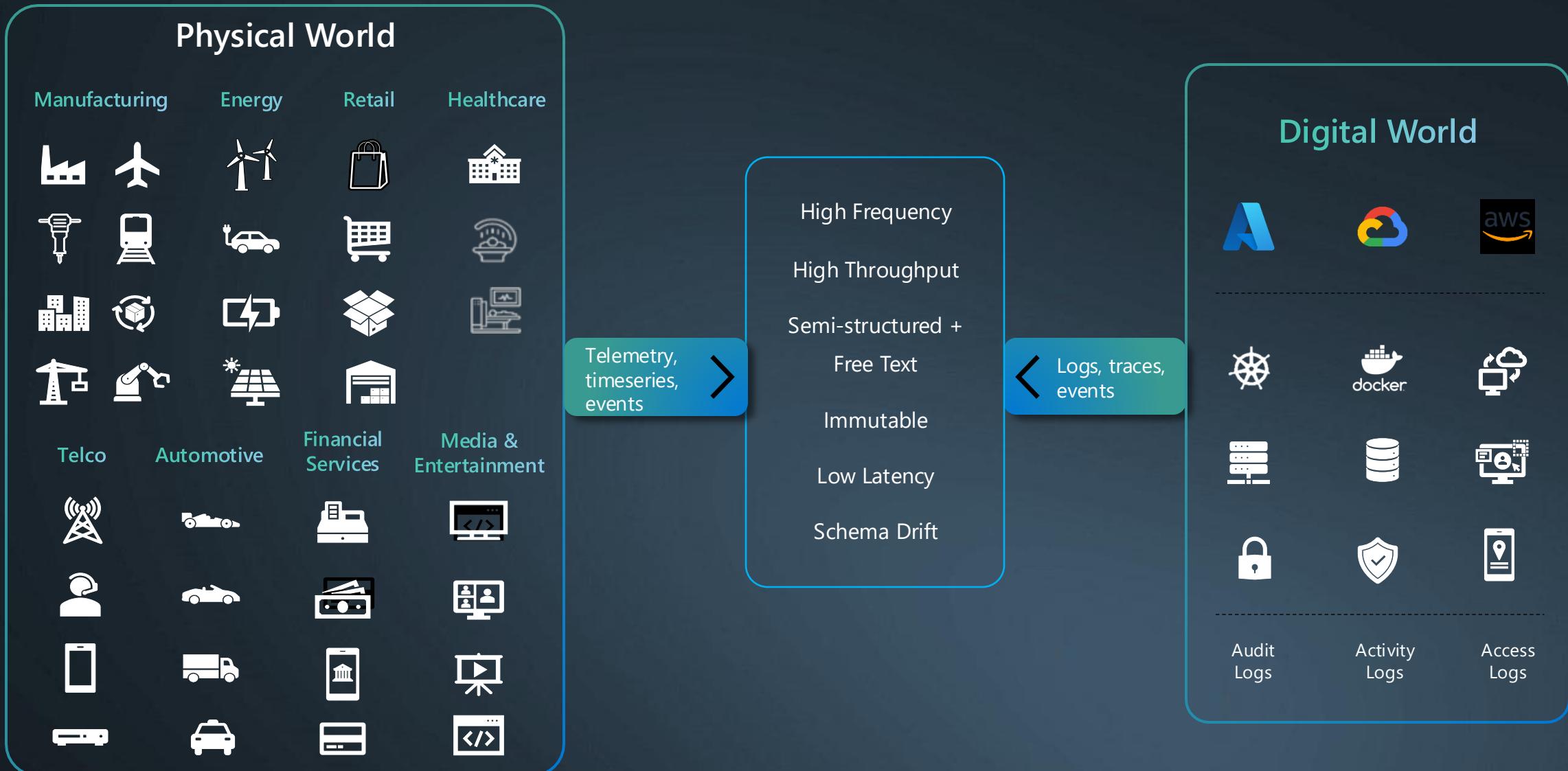
## Figure 3: Infrastructure attributes necessary for modernization

In the next two years, how important are the following to achieving your organization's overall technology goals? (percentage responding "very important")



Source: MIT Technology Review Insights survey, 2023

# Time based data: difficult to manage, yet critical for success





## Enterprise real-time data platforms



Azure Event Hubs  
Azure Event Grid  
Azure Stream Analytics  
Azure Data Explorer



## Self-service reporting and activation



OneLake  
Data Activator  
Power BI

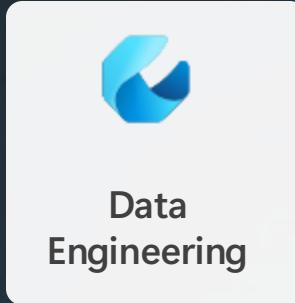


## Real-Time Intelligence in Microsoft Fabric

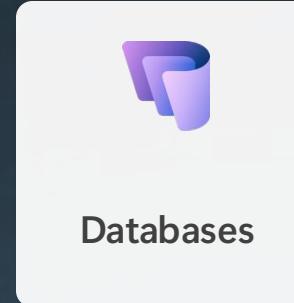
Fully integrated,  
no/low-code real-time SaaS  
data platform



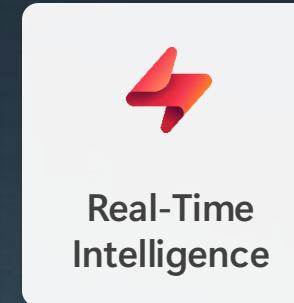
Data  
Factory



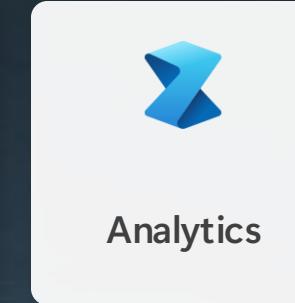
Data  
Engineering



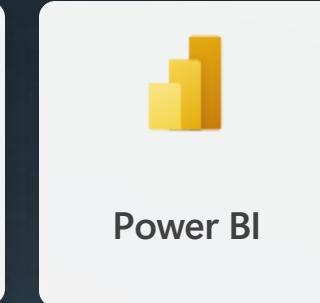
Databases



Real-Time  
Intelligence



Analytics



Power BI



Copilot in Fabric



OneLake



Microsoft Purview



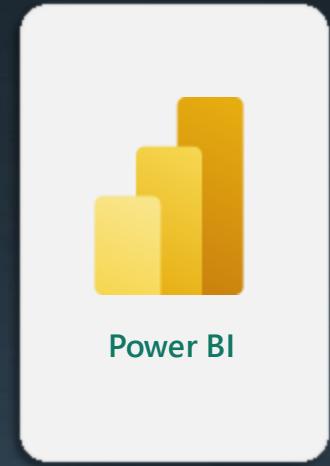
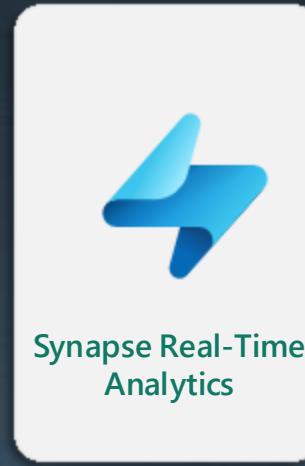
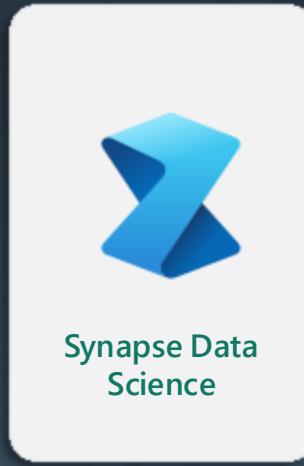
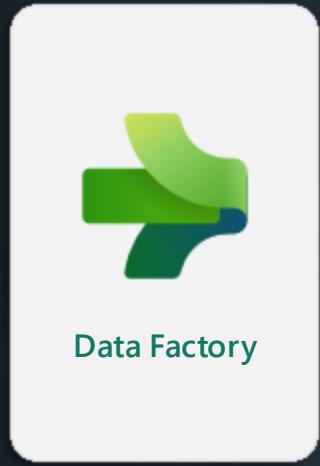
Real-Time Hub



OneLake Data Hub



# Microsoft Fabric – Initial release



OneLake



Data  
Factory



Data  
Engineering



Data  
Warehouse



Data  
Science



Real-Time  
intelligence



Power BI



AI



OneLake



Purview



Best-in-class Connectivity



Real-Time  
hub



Process



Analyze



Monitor

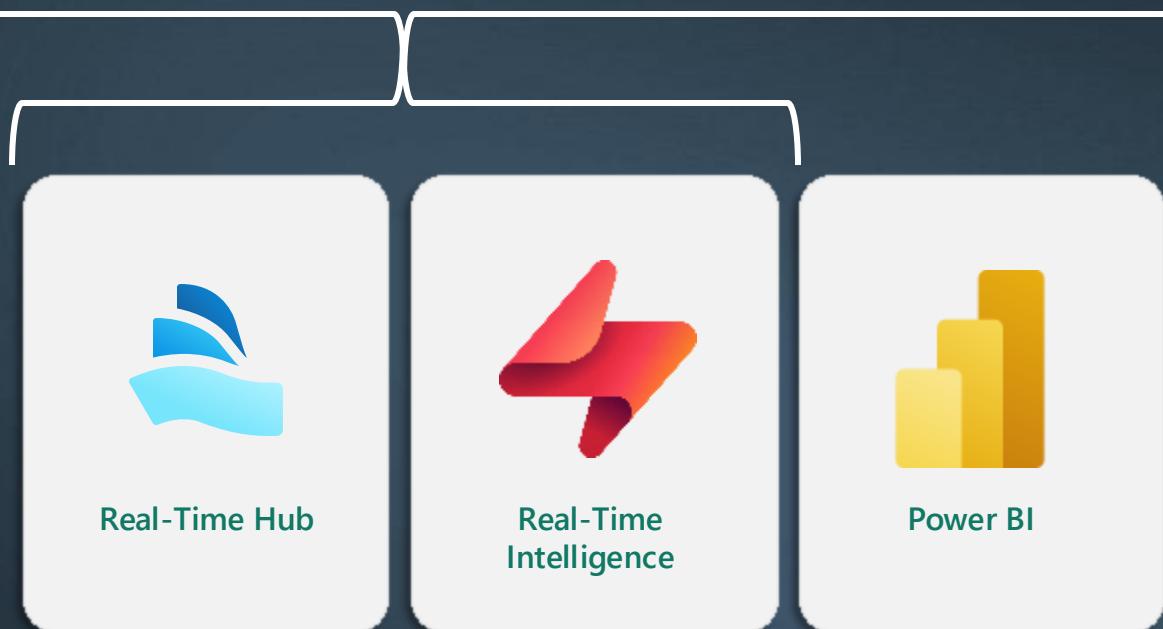


Act

★ AI-powered Intelligence



# Microsoft Fabric



## Automotive



Connected fleet applications

Autonomous Driving

Manufacturing + R&D

## Manufacturing



Improving Quality and Throughput

Predictive Maintenance

Inventory Prediction

## Logistics



Delivery tracking and routing

Warehouse management

Supply & demand operations

## Finance & Insurance



Finance Automation

Fraud Detection

Operational Efficiency

## Energy & Utilities



Station monitoring, energy leakage detection

Equipment Maintenance & Monitoring

Failure Monitoring

## Retail



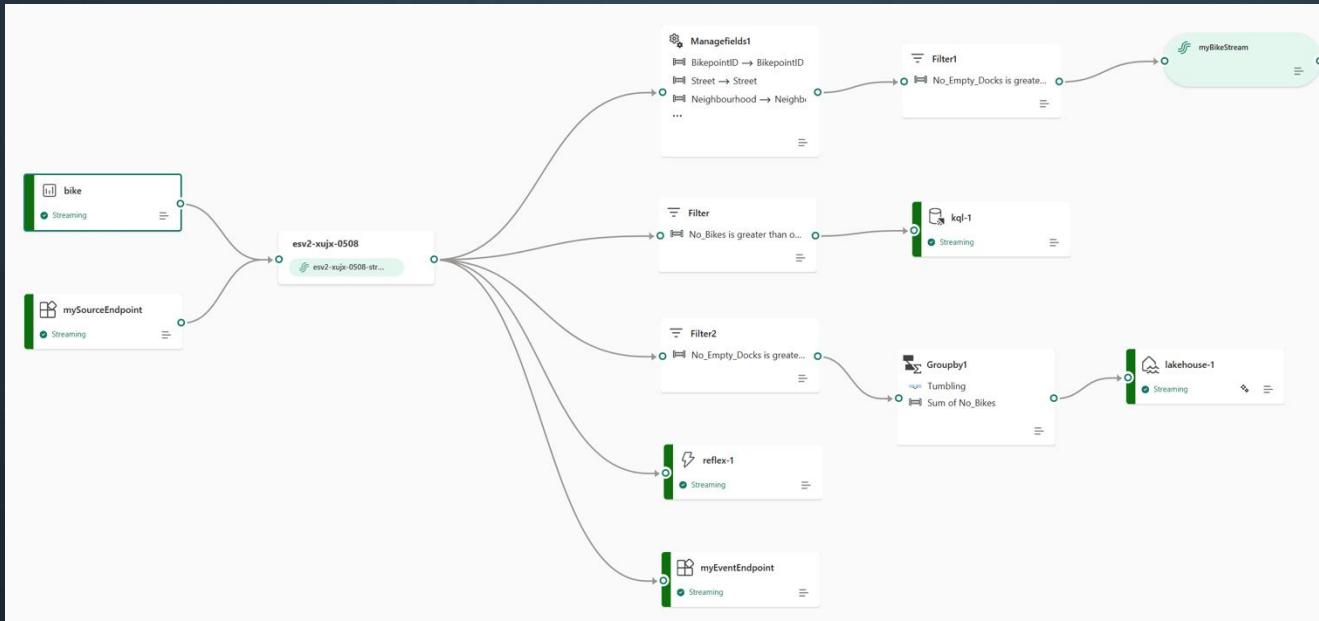
Inventory tracking

Promotions and buying experiences

Supply chain management

# Eventstream

- Low-code/No-code event transformation
- EventGrid & Stream Analytics bastard child
- Connect, Transform, Route



Eventstreams

# Eventstream Capacity

OPERATION IN CAPACITY METRICS APP	DESCRIPTION	OPERATION UNIT OF MEASURE	FABRIC CONSUMPTION RATE
Eventstream Per Hour	Flat charge* (per eventstream)	Per hour	0.222 CU per hour
Eventstream Data Traffic per GB	Data ingress & egress volume	Per GB	0.342 CU per hour per GB
Eventstream Processor Per Hour	Computing resources consumed by the processor	Per hour	0.778 CU per CPU hour**
OneLake Standard Storage	Used for extended retention (Includes 24-hour retention)	Per GB	\$0.023 per GB***
Eventstream Connectors Per vCore hour	Computing resources consumed by the connector	Per hour	0.611 CU per vCore hour

\*Eventstream is charged if Eventstream is not idle (i.e. data is not ingested or egressed for at least 2 hours)

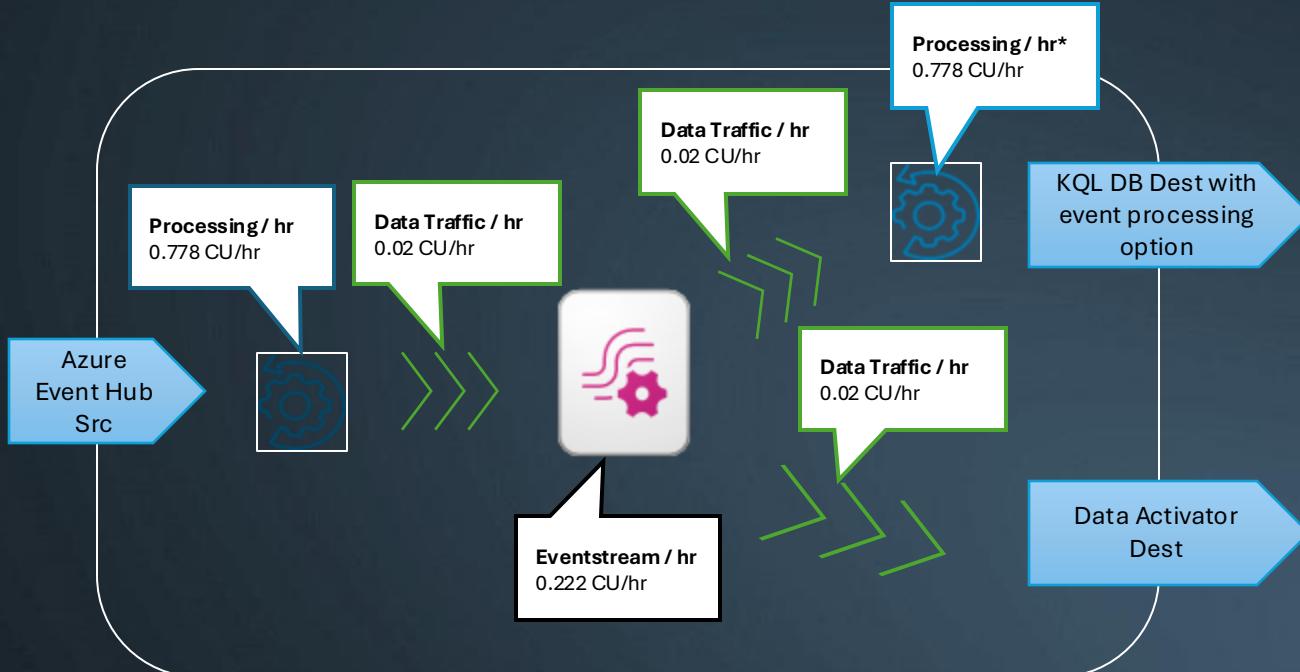
\*\*Processor base rate 2.333 CU per hour, starting at 1/3 CPU hour (0.778)

\*\*\*Refer to OneLake Storage pricing for more detail - [Microsoft Fabric - Pricing | Microsoft Azure](#)

[Monitor Microsoft Fabric event streams capacity consumption - Microsoft Fabric | Microsoft Learn](#)

# Eventstream usage scenario

An eventstream is ingesting from an Azure Event Hub source that is coming in at **1 MB/minute (0.059GB/hr)**. The data is transformed and filtered before sending to a KQL DB destination and also routed to a Data Activator destination.



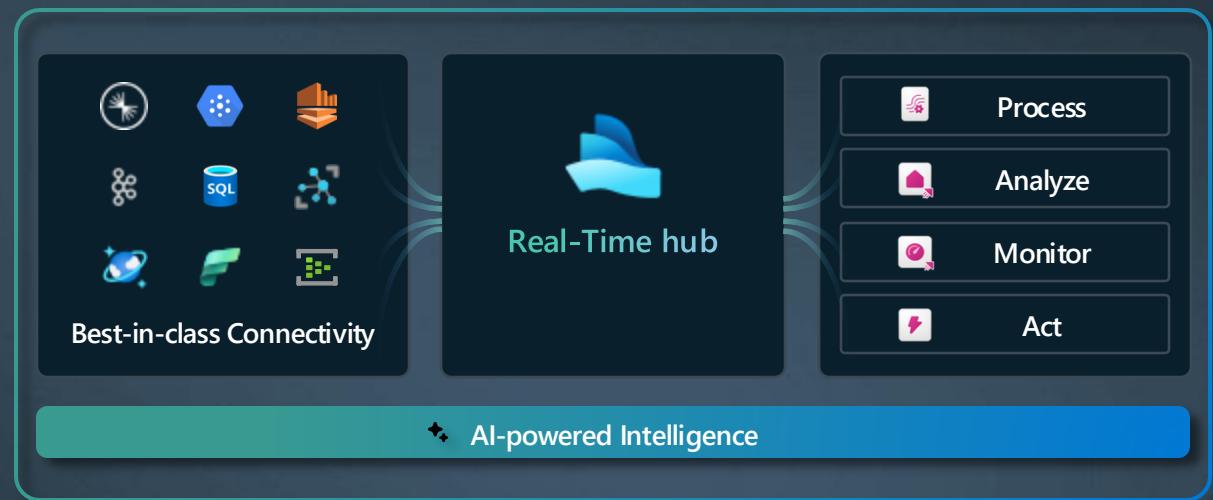
Total CU/hr: **1.838 CU**

\*Processing can cost more depending on transformation complexity (e.g. aggregation)

\*\*Does not include downstream costs for KQL DB and Data Activator

 Real-Time Hub

- Single catalog for data in motion
- Tenant-wide
- Connections
- Integrations
- Data-in-motion governance



# Eventhouse

The logical collection of KQL databases in Fabric



KQL Database



KQL Database



KQL Database

**Eventhouse**

A vibrant underwater photograph showing a school of small, dark fish swimming between two large, rocky coral reefs. Sunlight filters down from the surface in bright rays, creating a dappled light effect on the rocks and the sandy ocean floor. The water is a clear turquoise color.

DIVE INTO HISTORY

No, Kusto – not Cousteau!

Jaques Cousteau  
1910-1997



# KQL

- Developed by Microsoft engineers in Israel to handle Azure's need for fast log and telemetry analytics
- Released as Application Insights in 2016
- Data Explorer went GA 2019



# The Kusto Product Family

Solutions



Azure Monitor



Sentinel



Defender



Playfab



Product



Fabric Real-Time  
Intelligence Eventhouse



Azure Data  
Explorer



Kustainer

Tech Stack

Kusto

# KQL

- Columnar
  - very good compression
- Append-Only
- Elastic scale
- Low latency ingestion (200 Mb/s per node)
- Fully managed
  - indexed by default



# KQL

- RDBMS-like with databases, schemas & tables
- Powerful time series database
- Zero constraints
- Can be queried with SQL
- Better yet – the native language



# Intro to KQL

- Kusto Query Language

« ▶ Run ▾ ⏪ Recall ⚒ KQL tools ▾ adxcosmostest.eastus/adxbeers

```
1 //+ all+ data
2 BeerCheckins
3
```

BeerCheckins Stats

checkin_id	created_at	beer_name	beer_type	beer_ibu	beer_abv	flavors
600295668	2018-05-19 21:46:34.0000	Watou Tripel	Belgian Tripel	19	7.5	[{"flavor": "smooth"}]
600957579	2018-05-20 20:24:34.0000	Troubadour Magma	IPA - Belgian	45	9	[{"flavor": "hoppy"}]
600980213	2018-05-20 20:57:02.0000	Duvel	Belgian Strong Golden Ale	33	8.5	
601039251	2018-05-20 22:14:53.0000	Chouffe Soleil	Belgian Blonde	20	6	[{"flavor": "light"}]
601066988	2018-05-20 22:51:54.0000	Duvel	Belgian Strong Golden Ale	33	8.5	[{"flavor": "strong"}]
601849915	2018-05-23 19:11:38.0000	Orval	Pale Ale - Belgian	36	6.2	[{"flavor": "fruity"}]
601858859	2018-05-23 19:48:47.0000	Tripel Karmeliet	Belgian Tripel	16	8.4	[{"flavor": "strong"}]
601883612	2018-05-23 21:17:50.0000	Leffe Blonde / Blond	Belgian Blonde	20	6.6	[{"flavor": "light"}]
606238565	2018-06-02 15:17:25.0000	Maes Pils	Pilsner - Other	15	5.2	[{"flavor": "light"}]

```
13 // filtering
14 BeerCheckins
15 | where brewery_country == "Norway"
16
```

Table 1 Stats

checkin_id	created_at	beer_name	beer_type
> 919498731	2020-07-18 23:31:19.0000	Nitro Hot Chocolate ...	Stout - Imperial / Double...
> 1088786720	2021-10-16 22:07:42.0000	Møykjeddingi	IPA - Farmhouse
> 1089148830	2021-10-17 15:38:25.0000	Gølse Singelkveik Kve...	Traditional Ale
> 1091170649	2021-10-23 18:27:02.0000	STAFANGR New Engl...	IPA - New England / Hazy
> 1167989760	2022-06-11 18:49:26.0000	Solitude	IPA - New England / Hazy
> 1216484786	2022-10-30 15:47:30.0000	Å VESTLAND, VESTLA...	Pale Ale - American
> 1216489681	2022-10-30 16:06:10.0000	LEFT BEHIND - TRIPL...	IPA - Imperial / Double N...

```
18 BeerCheckins
19 | search todecimal(score) > 4.5
20 | project created_at, beer_name, beer_type, score
21
```

Table 1 Stats

created_at	beer_name	beer_type	score
> 2018-05-20 20:57:02.0000	Duvel	Belgian Strong Golden Ale	5
> 2018-05-20 22:51:54.0000	Duvel	Belgian Strong Golden Ale	5
> 2018-05-23 19:48:47.0000	Tripel Karmeliet	Belgian Tripel	5
> 2018-06-29 12:24:47.0000	Duvel	Belgian Strong Golden Ale	5
> 2018-07-06 20:00:39.0000	Duvel Tripel Hop Citra	Belgian Strong Golden Ale	5
> 2018-07-08 15:44:51.0000	Duvel Tripel Hop Citra (2018)	Belgian Strong Golden Ale	5
> 2018-07-15 20:18:49.0000	Vedett Extra Ordinary IPA	IPA - Belgian	4.75

```

4 // row count
5 BeerCheckins
6 | count
7

```

Table 1 Stats

Count
> 1,078

```

13 // group by
14 BeerCheckins
15 | summarize count() by bin(todatetime(created_at),30d)
16

```

Table 1 Stats

	created_at	count_
>	2018-05-17 00:00:00.0000	14
>	2018-06-16 00:00:00.0000	14
>	2018-07-16 00:00:00.0000	22
>	2018-08-15 00:00:00.0000	16
>	2018-09-14 00:00:00.0000	19
>	2018-10-14 00:00:00.0000	18

```

8 // sampling
9 BeerCheckins
10 | order by created_at desc
11 | take 10
12

```

Table 1 Stats

checkin_id	created_at	beer_name	beer_type	beeribu
> 1218337235	2022-11-05 22:09:59.0000	India Pale Ale Mosa...	IPA - American	0
> 1218240683	2022-11-05 20:33:14.0000	Zinne des Flandres	Pale Ale - Belgian	0
> 1218186931	2022-11-05 19:34:50.0000	Noire du Midi	Porter - Other	0
> 1217651189	2022-11-04 19:11:24.0000	Hercule Stout	Stout - Imperial / Double	44
> 1217577843	2022-11-04 15:38:45.0000	Monasterium Kloos...	Spiced / Herbed Beer	0
> 1217165844	2022-11-02 19:23:04.0000	Victoria	Belgian Strong Golden Ale	33
> 1216549442	2022-10-30 18:42:35.0000	HertenHaas	Belgian Tripel	35
> 1216503709	2022-10-30 16:52:17.0000	Trappist Westvleter...	Belgian Quadrupel	38
> 1216489681	2022-10-30 16:06:10.0000	LEFT BEHIND - TRIP...	IPA - Imperial / Double N...	0
> 1216484786	2022-10-30 15:47:30.0000	Å VESTLAND, VEST...	Pale Ale - American	28

Add connection

Filter...

Storm\_Events

PopulationData

- State (string)
- Population (long)

StormEvents

- StartTime (datetime)
- EndTime (datetime)
- EpisodeId (int)
- EventId (int)
- State (string)
- EventType (string)
- InjuriesDirect (int)
- InjuriesIndirect (int)
- DeathsDirect (int)
- DeathsIndirect (int)
- DamageProperty (int)
- DamageCrops (int)
- Source (string)

Run Recall KQL tools help/Samples

```
1 StormEvents
2 | join PopulationData on $left.State == $right.State
3 | project StartTime, EndTime, State, EventType, Population
4 | limit 15
```

Table 1 Add visual Stats

StartTime	EndTime	State	EventType	Population
2007-01-01 08:13:00.0000	2007-01-01 14:00:00.0000	DELAWARE	Flash Flood	982,049
2007-02-01 00:00:00.0000	2007-02-02 13:00:00.0000	ALASKA	Flood	727,951
2007-07-10 12:31:00.0000	2007-07-10 12:32:00.0000	DISTRICT OF COLUMBIA	Thunderstorm Wind	709,951
2007-08-05 17:05:00.0000	2007-08-05 17:20:00.0000	IDAHO	Hail	1,823,590
2007-08-17 22:00:00.0000	2007-08-17 22:00:00.0000	CONNECTICUT	Thunderstorm Wind	3,559,050
2007-08-21 17:11:00.0000	2007-08-21 17:11:00.0000	COLORADO	Hail	5,826,180
2007-08-24 17:28:00.0000	2007-08-24 17:28:00.0000	INDIANA	Thunderstorm Wind	6,768,940
2007-09-18 20:00:00.0000	2007-09-19 18:00:00.0000	FLORIDA	Heavy Rain	21,711,200
2007-12-01 00:00:00.0000	2007-12-01 01:00:00.0000	CALIFORNIA	Flash Flood	39,562,900
2007-12-01 10:40:00.0000	2007-12-01 16:00:00.0000	ARIZONA	Flash Flood	7,399,410
2007-12-08 06:40:00.0000	2007-12-08 13:01:00.0000	HAWAII	Flash Flood	1,411,150
2007-12-09 16:00:00.0000	2007-12-09 16:00:00.0000	ARKANSAS	Lightning	3,025,880
2007-12-15 18:00:00.0000	2007-12-15 19:00:00.0000	ALABAMA	Heavy Rain	4,918,690
2007-12-23 02:15:00.0000	2007-12-23 02:15:00.0000	ILLINOIS	Thunderstorm Wind	12,620,600
2007-12-30 16:00:00.0000	2007-12-30 16:05:00.0000	GEORGIA	Thunderstorm Wind	10,723,700

different join types

innerunique (default) → only  
first match is included (like the  
SSIS lookup component)

inner

leftouter, rightouter

fullouter

leftanti, rightanti

leftsemi, rightsemi

```
22 // line chart
23 BeerCheckins
24 | summarize count() by bin(todatetime(created_at),30d)
25 | render timechart
26
```

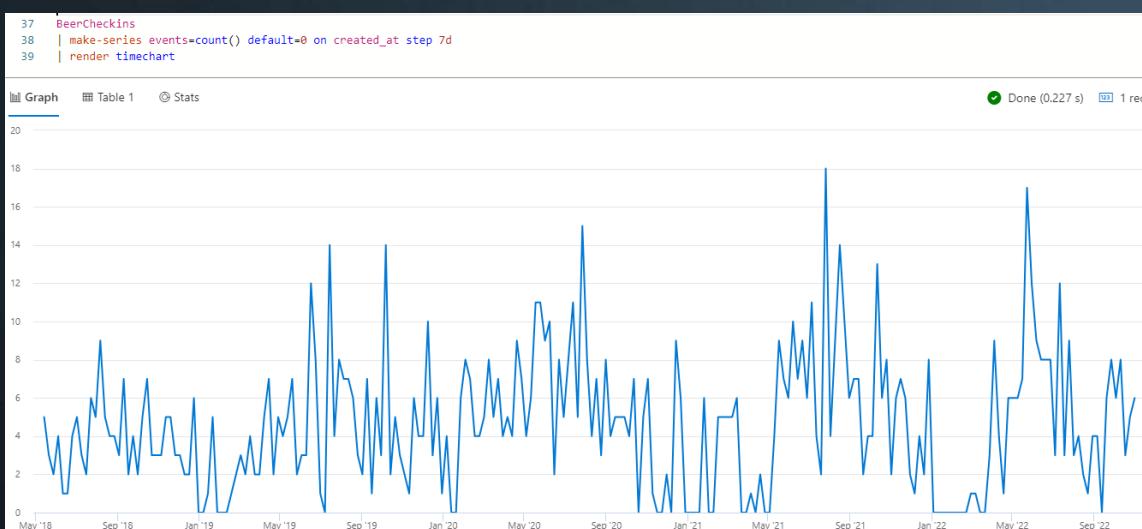


```
37  BeerCheckins
38  | make-series events=count() default=0 on created_at step 1d
```

Table 1 Stats

events created\_at

[1,4,0,0,3,0,0,0,0,0,0,0,0,2,0,0,0,0,1,2,1,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,2,0,2,0,1,0,1,0,3,0,0,0,0,1,2,0,0,0,0,1,1,... ["2018-05-19T00:00:00.000000Z", "2018-05-20T00:00:00.000000Z", "2018-05-21T00:00:00.000000Z", "2018-05-22T00:00:00.000000Z", ...]



you can also  
calculate moving averages  
apply linear regression  
detect patterns  
forecast

```
34 BeerCheckins  
35 | project checkin_id, flavors
```

Table 1 ⚡ Stats

	checkin_id	flavors
>	1216549442	[{"flavor": "belgiany"}, {"flavor": "sweet"}, {"flavor": "hoppy"}]
>	1214306625	[{"flavor": "floral"}, {"flavor": "strong"}, {"flavor": "spiced"}]
>	1210260383	[{"flavor": "dry"}, {"flavor": "sour"}, {"flavor": "tart"}]
>	1210205708	[{"flavor": "sweet"}, {"flavor": "strong"}, {"flavor": "smooth"}]
>	1209867243	[{"flavor": "soft"}, {"flavor": "light"}, {"flavor": "dry"}]
>	1206767994	[{"flavor": "herbal"}, {"flavor": "strong"}]
>	1205300638	[{"flavor": "fruity"}, {"flavor": "dry"}, {"flavor": "sour"}]
>	1203891699	[{"flavor": "sweet"}, {"flavor": "strong"}]
>	1198657148	[{"flavor": "clean"}, {"flavor": "light"}, {"flavor": "smooth"}]
>	1193418465	[{"flavor": "malty"}, {"flavor": "ginger"}, {"flavor": "spiced"}]
>	1190374989	



```
27 // expand  
28 BeerCheckins  
29 | project checkin_id, flavors  
30 | mv-expand flavors  
31 | extend flavor = tostring(flavors.flavor)  
32 | project-away flavors
```

Table 1 ⚡ Stats

	checkin_id	flavor
>	1216549442	belgiany
>	1216549442	sweet
>	1216549442	hoppy
>	1214306625	floral
>	1214306625	strong
>	1210260383	spiced
>	1210260383	dry
>	1210260383	sour
>	1210260383	tart

```
1 EXPLAIN
2 SELECT
3     beer_type
4     ,count(checkin_id) AS nbrBeers
5 FROM BeerCheckins
6 WHERE brewery_country like 'bel%'
7 GROUP BY beer_type
```

Table 1 Stats

Search Done (0.570 s) 123 1 records

Query

BeerCheckins | where (brewery\_country startswith "bel") | summarize nbrBeers=toint(countif(notnull(checkin\_id))) by beer\_type | project beer\_type, nbrBeers

# Eventhouse Capacity

## Kusto UpTime

- The number of seconds that your Eventhouse is active in relation to the number of virtual cores used
- After 5 minutes of inactivity the Eventhouse will no longer report Kusto UpTime
- Example: An active 8 core KQL DB will report 240 seconds of Kusto UpTime every 30 seconds

# KQL DB Storage

## **One Lake Cache Storage**

Premium storage that is utilized to provide the fastest query response times.

Controlled via the cache policy

Comparable to the Azure ADLS (Azure Data Lake Storage) premium tier.

\*\* Note: Using minimum capacity setting on Eventhouse results in 100% Kusto UpTime but no longer charges for OneLake Cache Storage

## **One Lake Standard Storage**

Standard storage that is used to persist and store all queryable data.

Controlled via the retention policy

Comparable to the Azure ADLS (Azure Data Lake Storage) hot tier.

# Eventhouse Usage Scenario



Eventhouse  
4.25 Cores  
100 GB in Cache  
500 GB Total Storage

## Capacity via Activity

Queries or Command or Ingestion  
80% active would need 3.4 CUs and require  
min 4 CUs



## Storage

100 GB at Premium Storage (~\$0.20/GB)  
500 GB at Standard Storage (~\$0.02/GB)



# LUNCH BREAK

---

Workshop starting again at 13:00

# Real-Time Dashboard

Visualizations optimized for streaming,  
timeseries, event-based data

Complimentary to Power BI

Low-latency refresh

Based on KQL

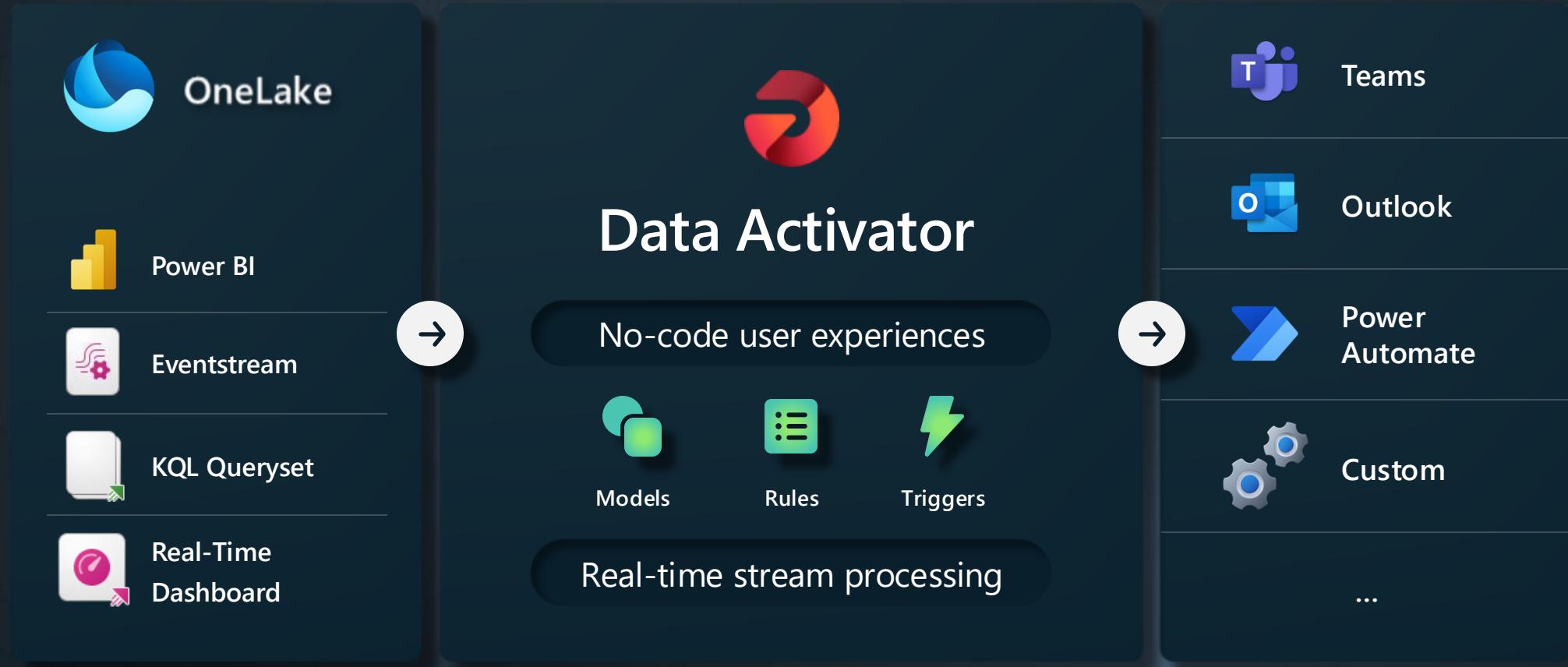
Parameterized

Cross-interactions and filters



Real-Time  
Dashboards

# Data Activator: Trigger actions on all your data, from one place

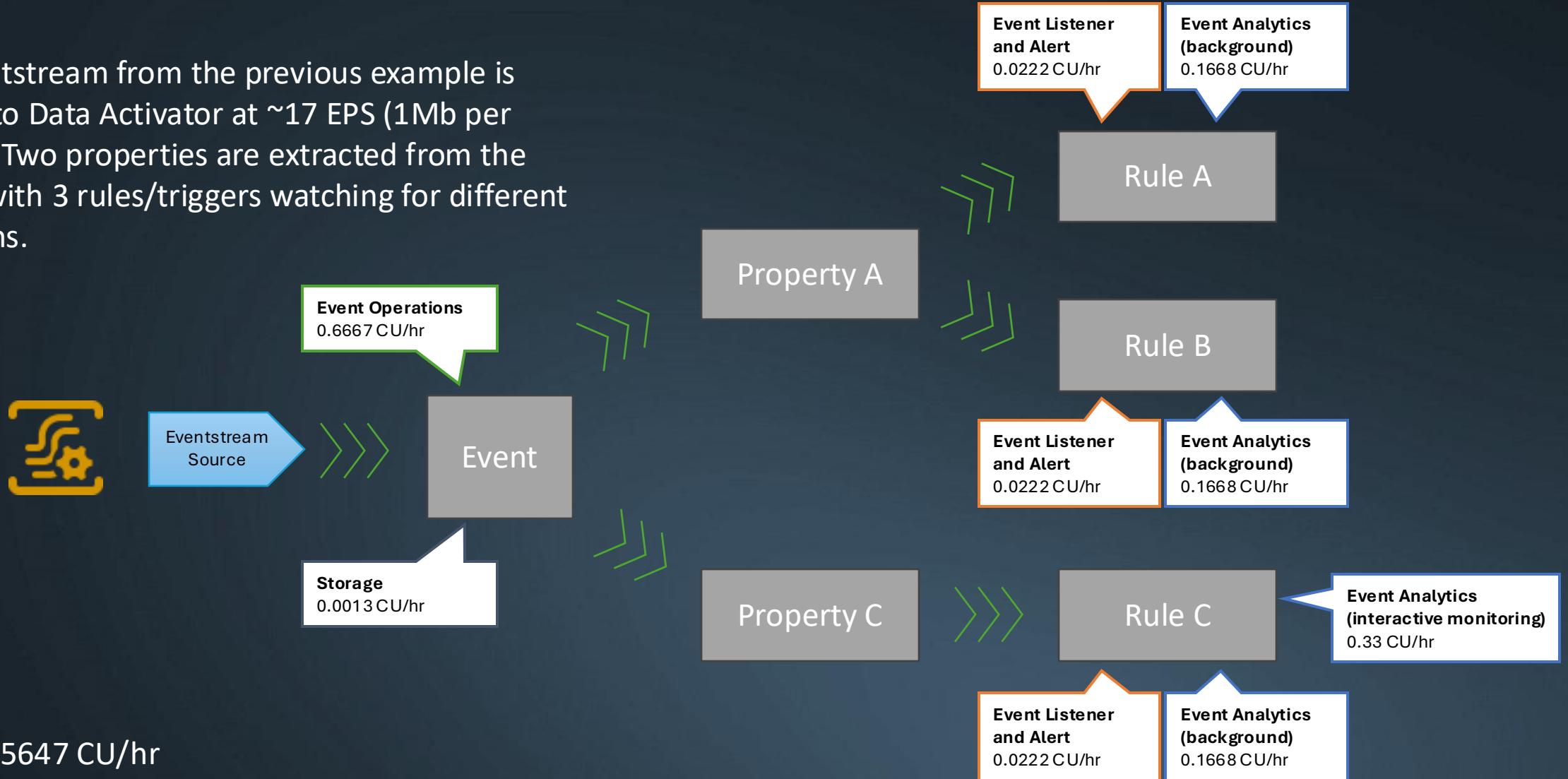


# Data Activator – Consumption Meters

BILLING METER	DESCRIPTION	OPERATION UNIT OF MEASURE	FABRIC CONSUMPTION RATE
Real-Time Intelligence – Event Listener & Alert	Flat charge per rule (trigger)	Per hour	0.0222 CU
Real-Time Intelligence – Event Operations	Data ingress	Per M events*	11.111 CU
Data Activator – Event Analytics	Events processed to evaluate rules, run queries etc.	Per M events	2.78 CU
Storage		Per GB/ month	1.333 CU

\*Event size 0-1Kb = 1 event, 1-2Kb = 2 events, 2-3Kb = 3 events etc.

The eventstream from the previous example is sending to Data Activator at ~17 EPS (1Mb per minute). Two properties are extracted from the events, with 3 rules/triggers watching for different conditions.

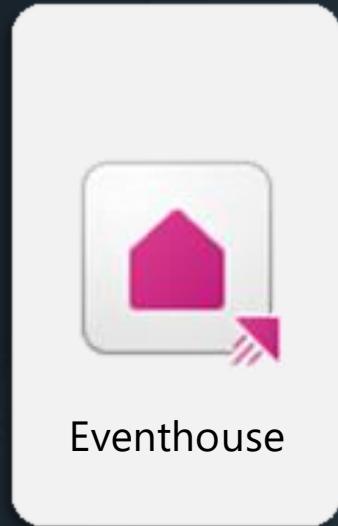




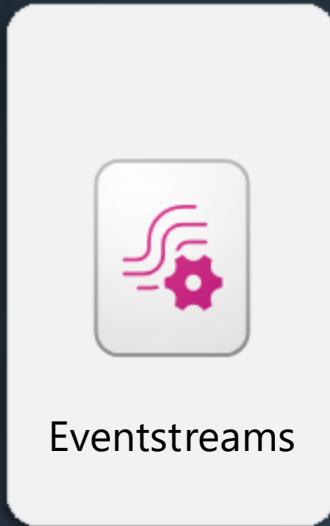
# Hands-on Lab time!



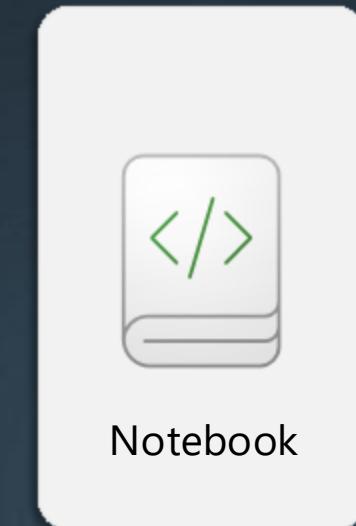
# Lab Introduction



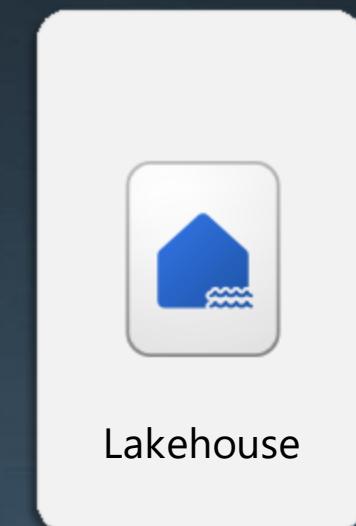
Eventhouse



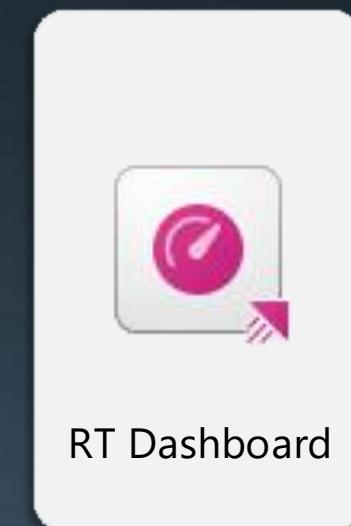
Eventstreams



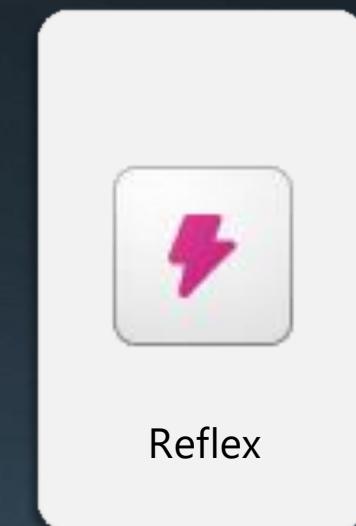
Notebook



Lakehouse



RT Dashboard

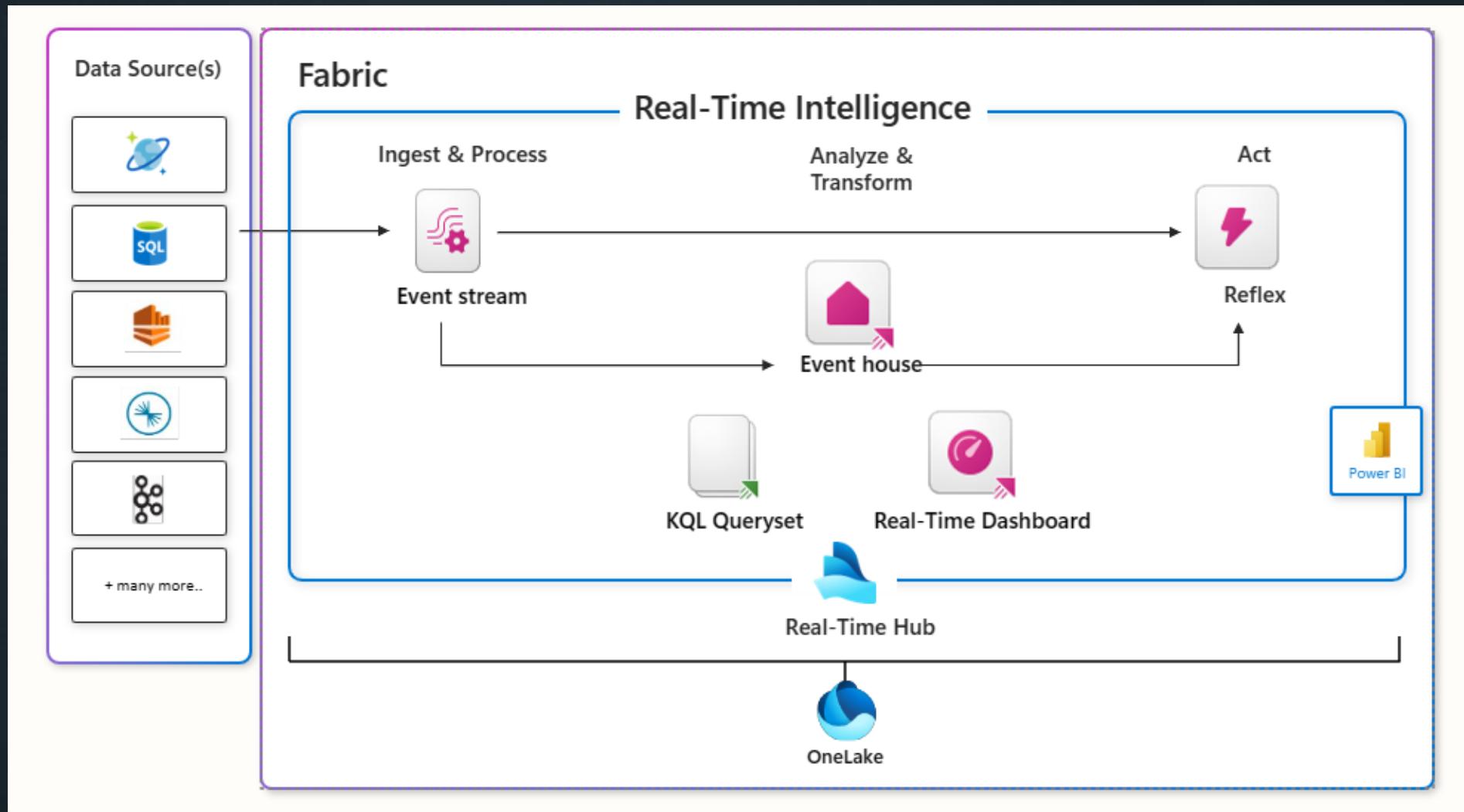


Reflex

<https://aka.ms/FabConRTITutorial>

Perform Section 5 part 1-14

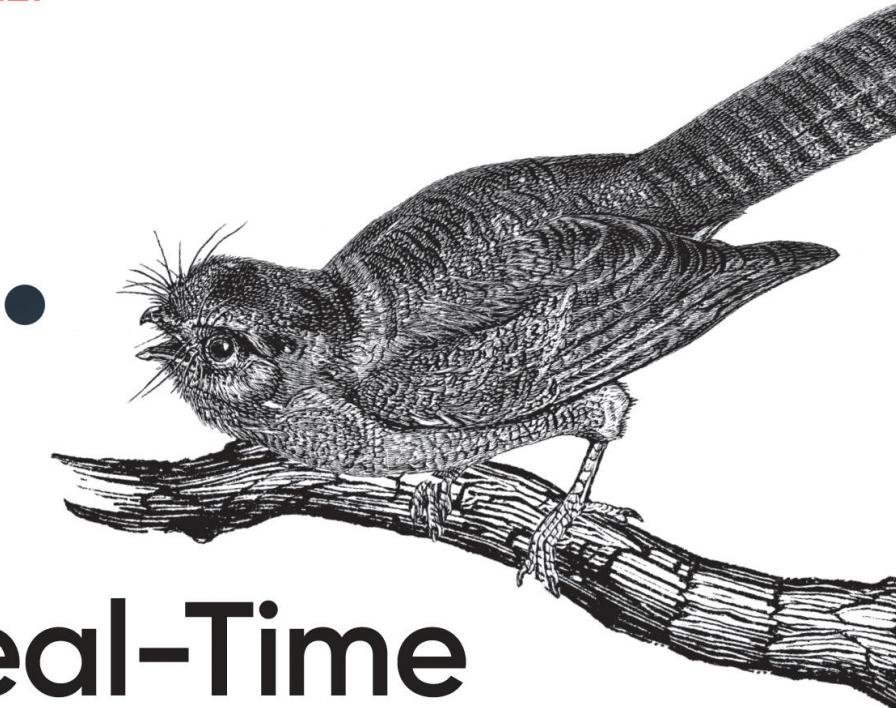
# Wrapping up



# Build the data foundation for digital twins and AI solutions



O'REILLY®



# Real-Time Intelligence with Microsoft Fabric

Empowering Data-Driven Decisions  
in the Era of AI

Johan Ludvig Brattås  
& Frank Geisler



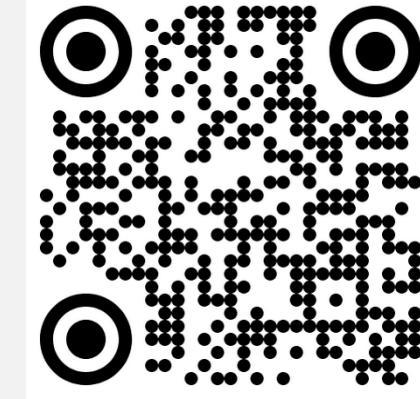
# Johan Ludvig Brattås

## Director, Deloitte

 /johanludvig

 @SQLClause.no

 jbrattas@deloitte.com



GitHub

### Chronic volunteer

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

### When not geeking out over new tech

Teaching coeliacs how to bake gluten free  
Baking  
Hiking  
Gardening