Microsoft Fabric Real-Time Intelligence for Data Engineers

Kamil Nowiński

Paweł Potasiński

Kamil Nowiński

Group Manager
Avanade UK & Ireland

SQL Day

Microsoft® Most Valuable Professional

Microsoft CERTIFIED DEVOPS ENGINEER EXPERT

Microsoft CERTIFIED AZURE DEVELOPER ASSOCIATE

databricks SOLUTION ARCHITECT CHAMPION

azureplayer.net
@KamilNowinski
@AzurePlayer.bsky.social
linkedin.com/in/KamilNowinski

FAVORITE STUFF:

Pawel Potasinski

Chief Technology Officer
InfiniteDATA Services

@PawelPotasinski.bsky.social

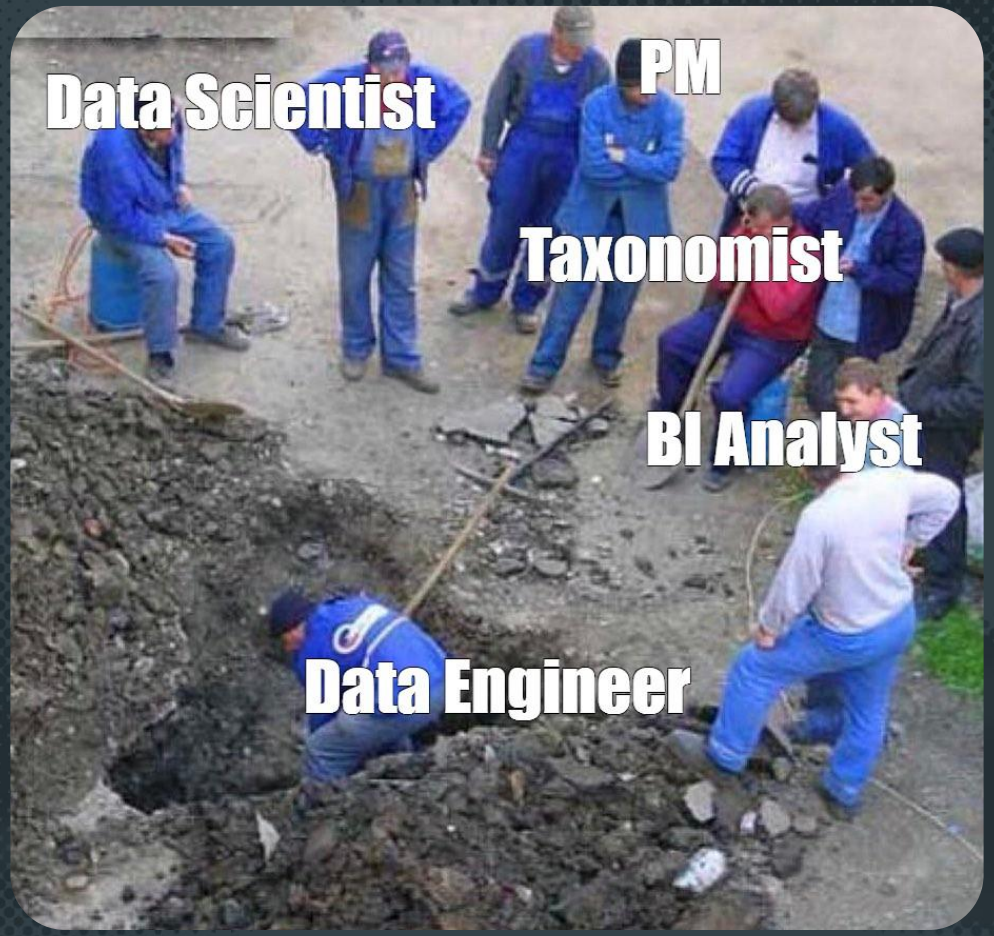
linkedin.com/in/PawelPotasinski

FAVORITE STUFF:

Agenda

- Introduction - data ingestion modes
- Overview of Fabric Real-Time Intelligence (RTI)
- Real-time data sources
- Fabric Eventstreams in use
- Selected unique capabilities of RTI
- Use cases for RTI
- Summary & resources

Data engineer’s life



Success criteria

On-time data delivery

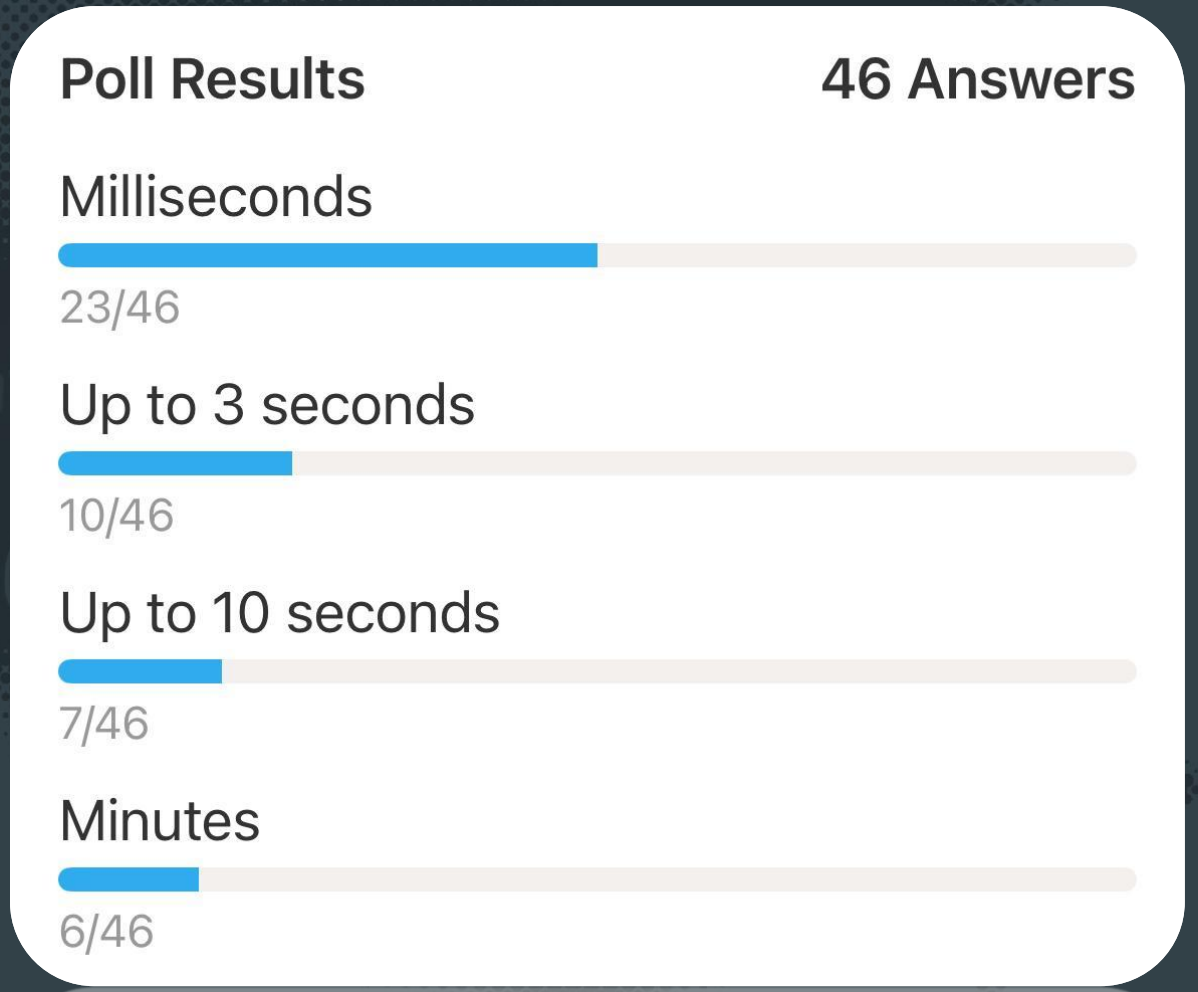
Change management

Good enough data quality

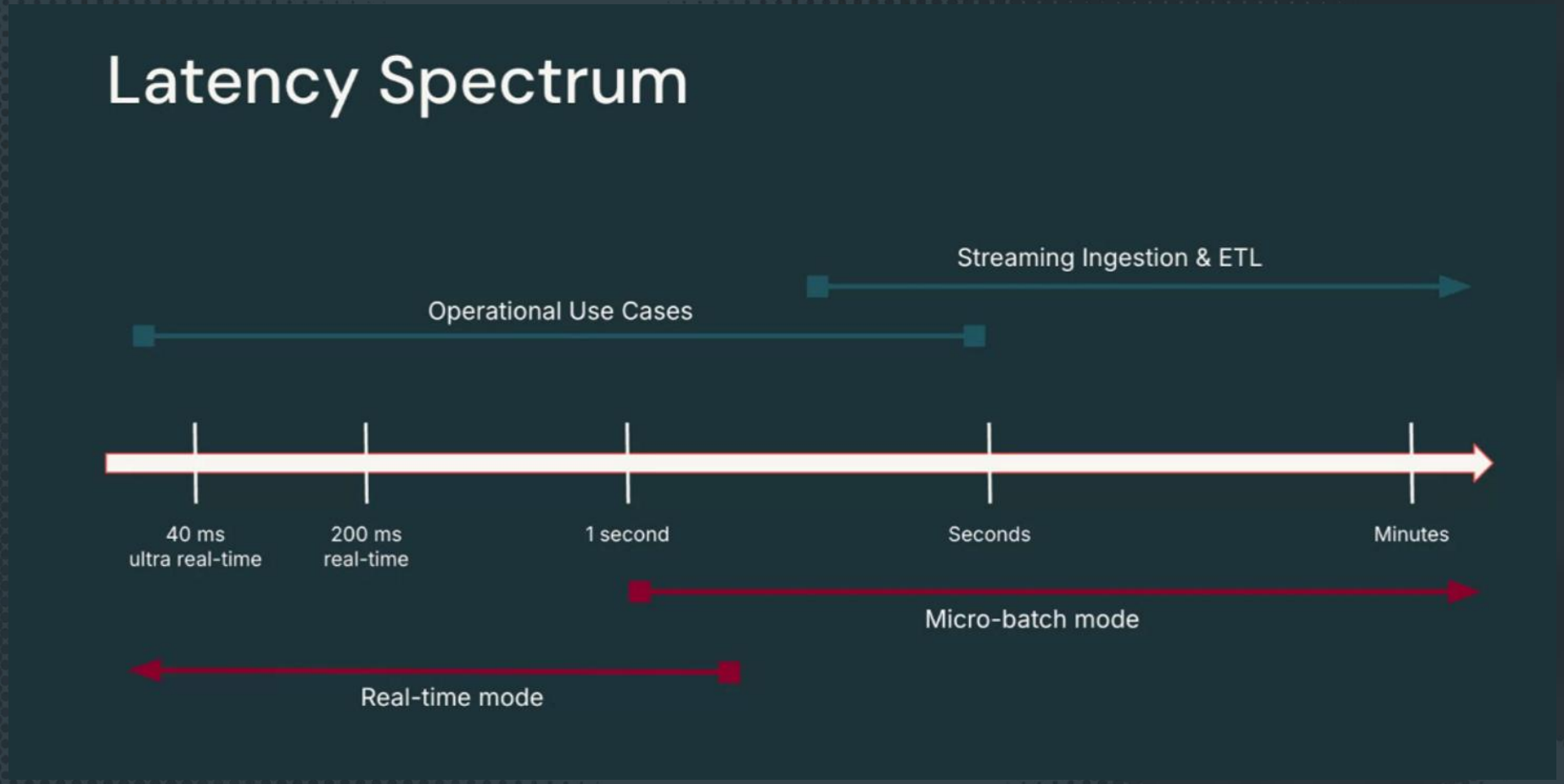
Acceptable cost

Whova - Survey

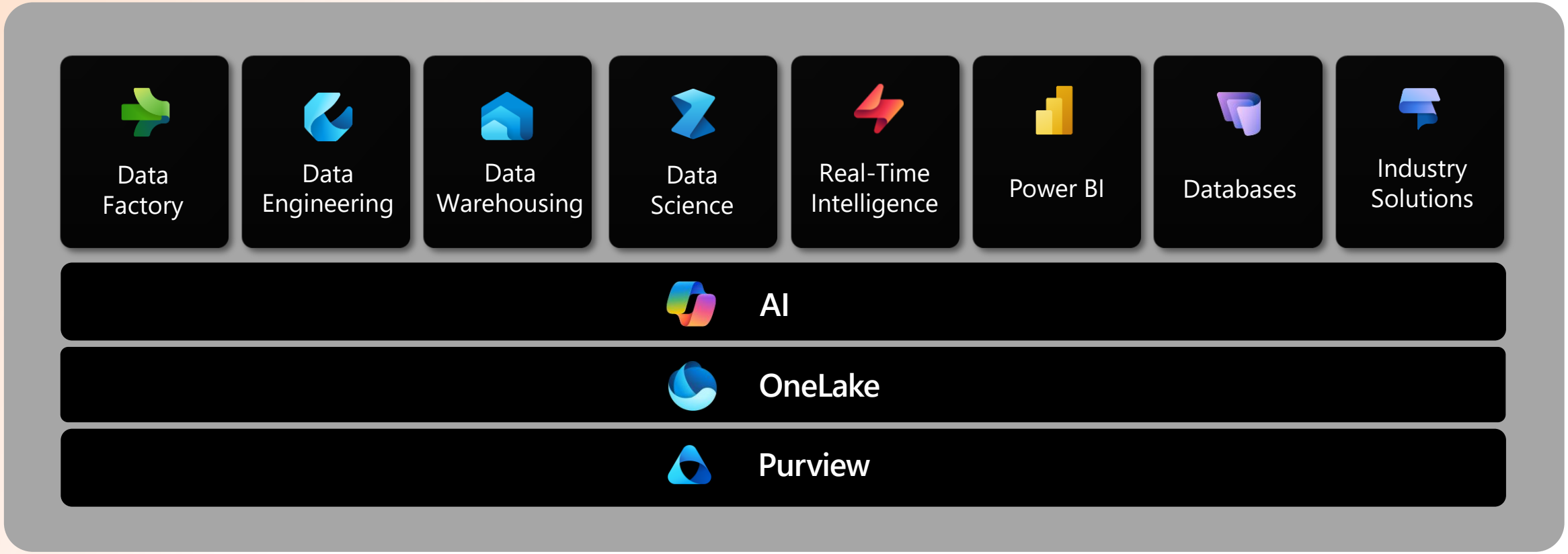
What does “rea-time analytics” mean for you in terms of data latency?



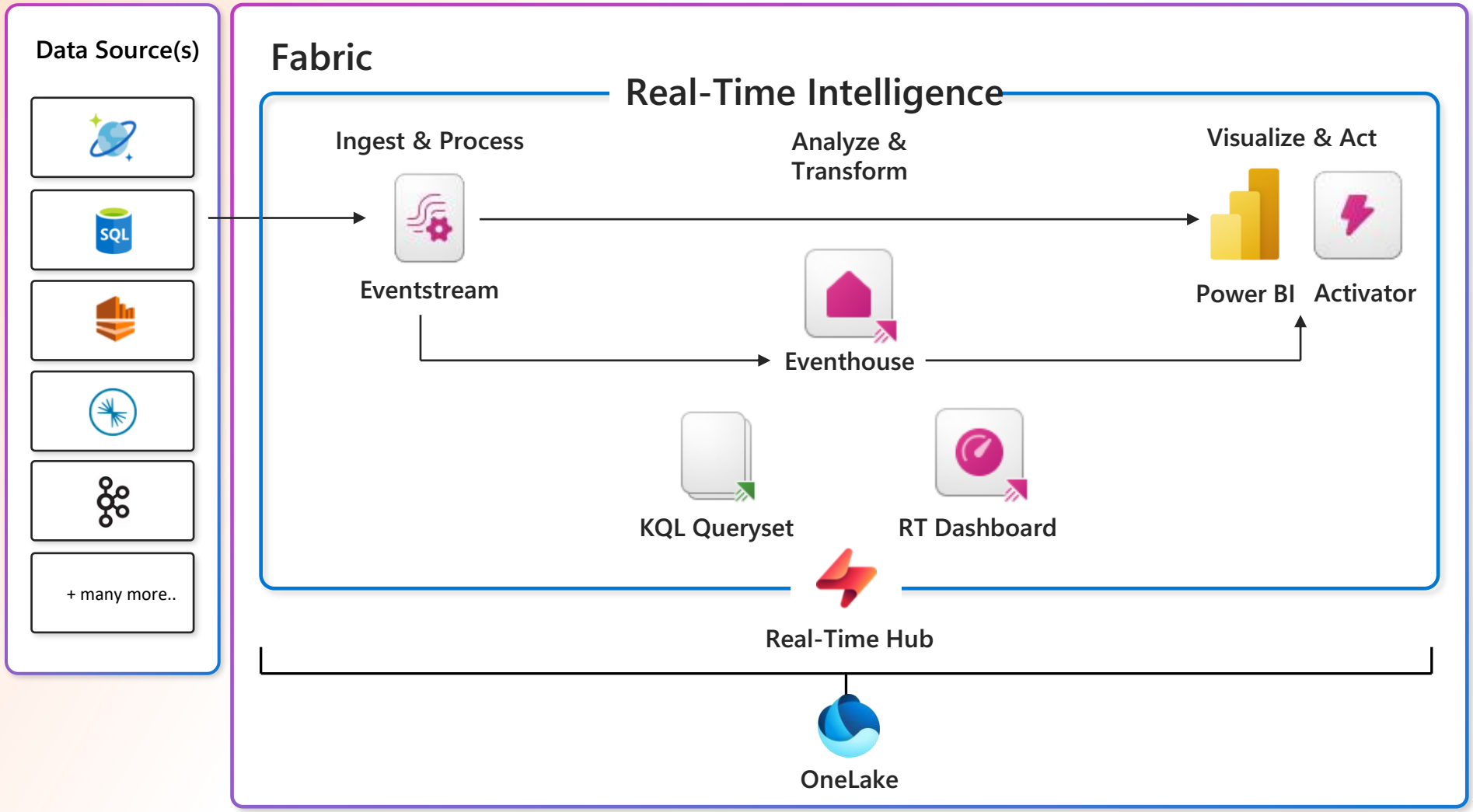
Data ingestion modes



Microsoft Fabric



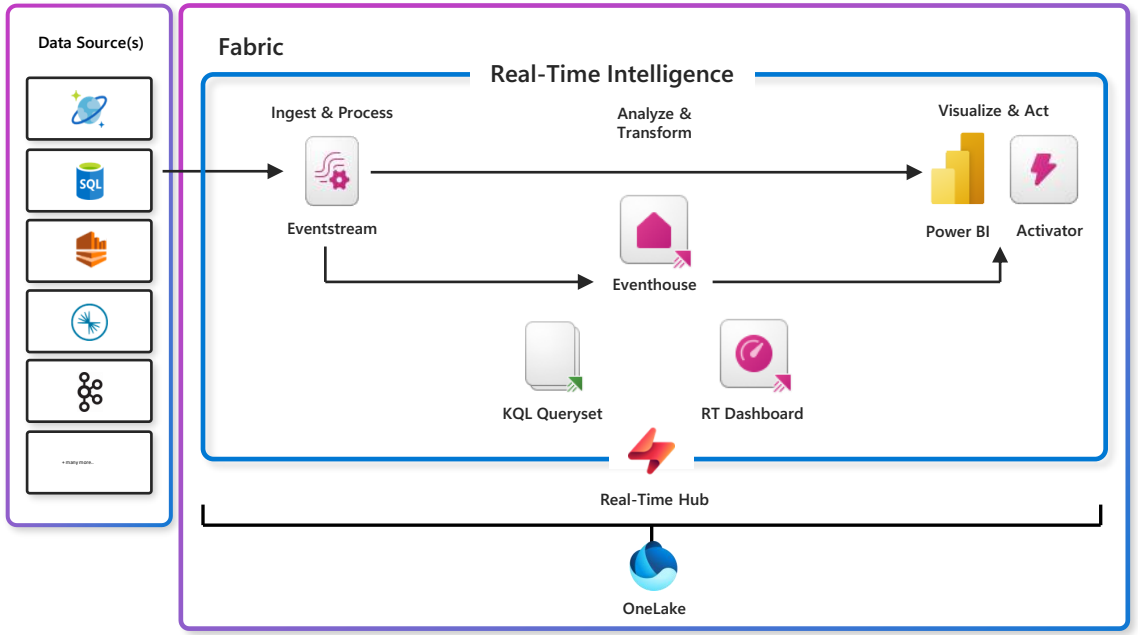
Overview of Fabric Real-Time Intelligence



Overview of Fabric Real-Time Intelligence

RTI item types

- **Eventstream**: Real-time data stream ingestion and transformation
- **Eventhouse**: Scalable storage and querying of real-time data
- **KQL Queryset**: A tool for creating and running queries in the KQL databases within an eventhouse
- **Real-Time Dashboard**: Interactive visualization and exploration of real-time data
- **Activator**: Automated actions triggered by real-time events

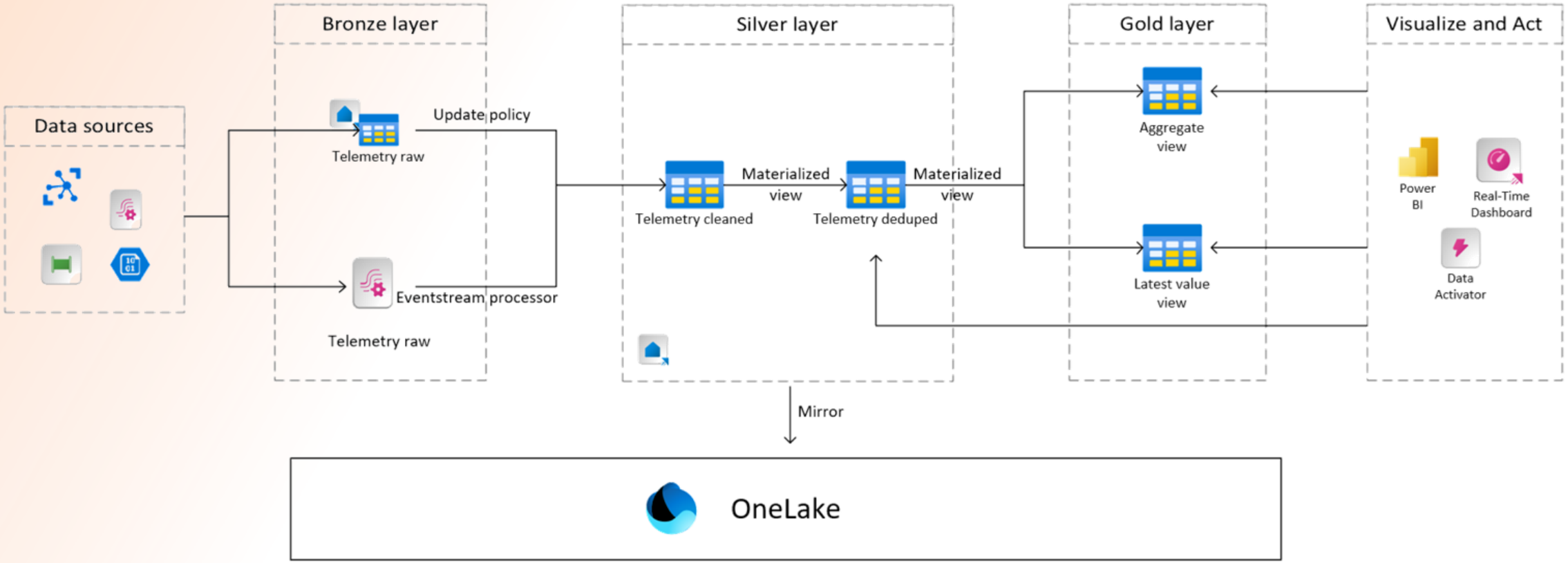


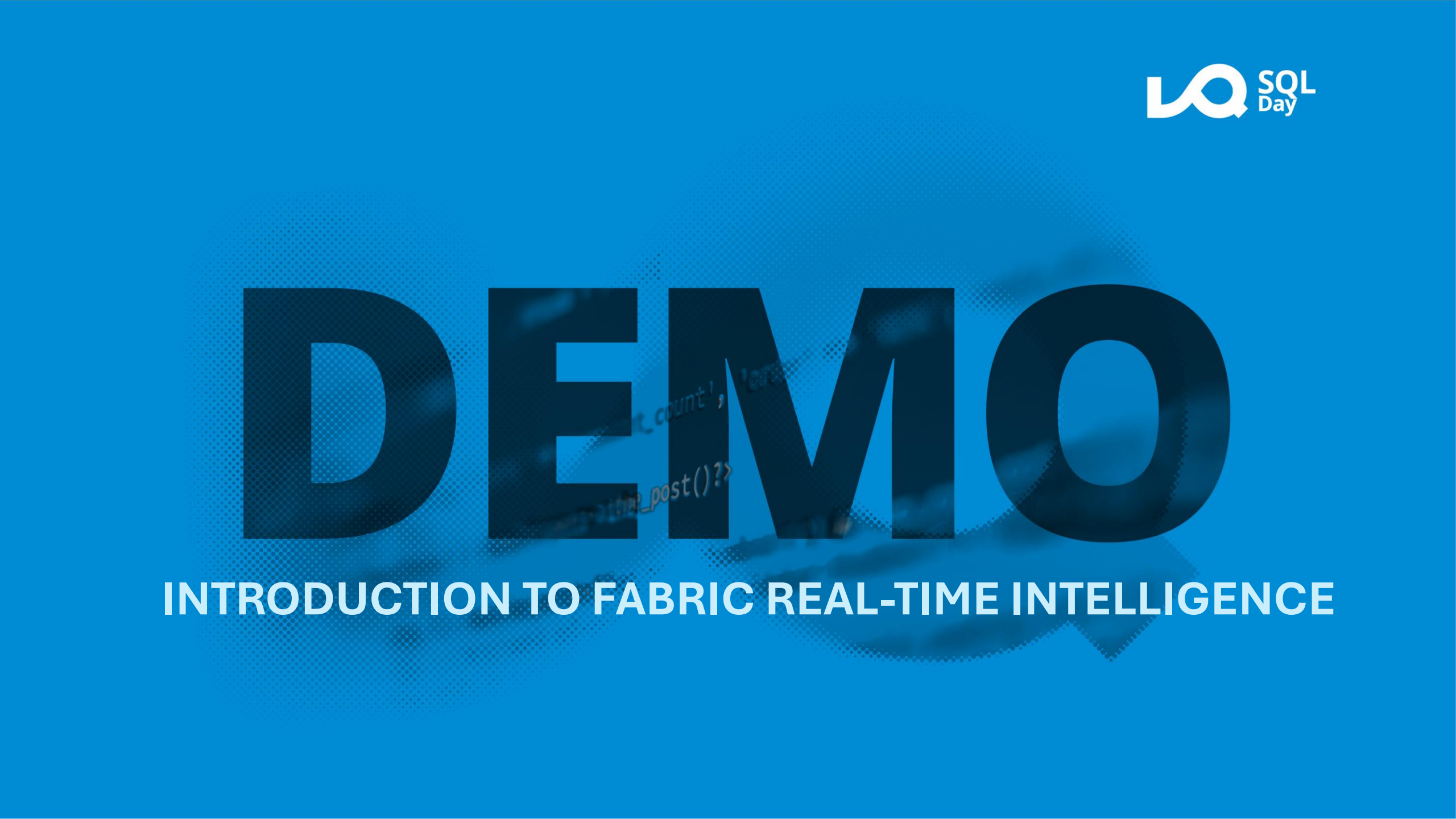
Overview of Fabric Real-Time Intelligence

Objects in KQL

- **Table:** Every data engineer knows what a table is, right? ☺
- **Function:** Built-in or user-defined function (stored or query-defined) for data transormation
- **View:** A virtual table (stored or query-defined) based on the KQL query result-set
- **Materialized view:** Aggregation query over a table or another materialized view, returning up-to-date result
- **Update policy:** An automation mechanism triggered when new data is written to a table

Event Medallion Architecture



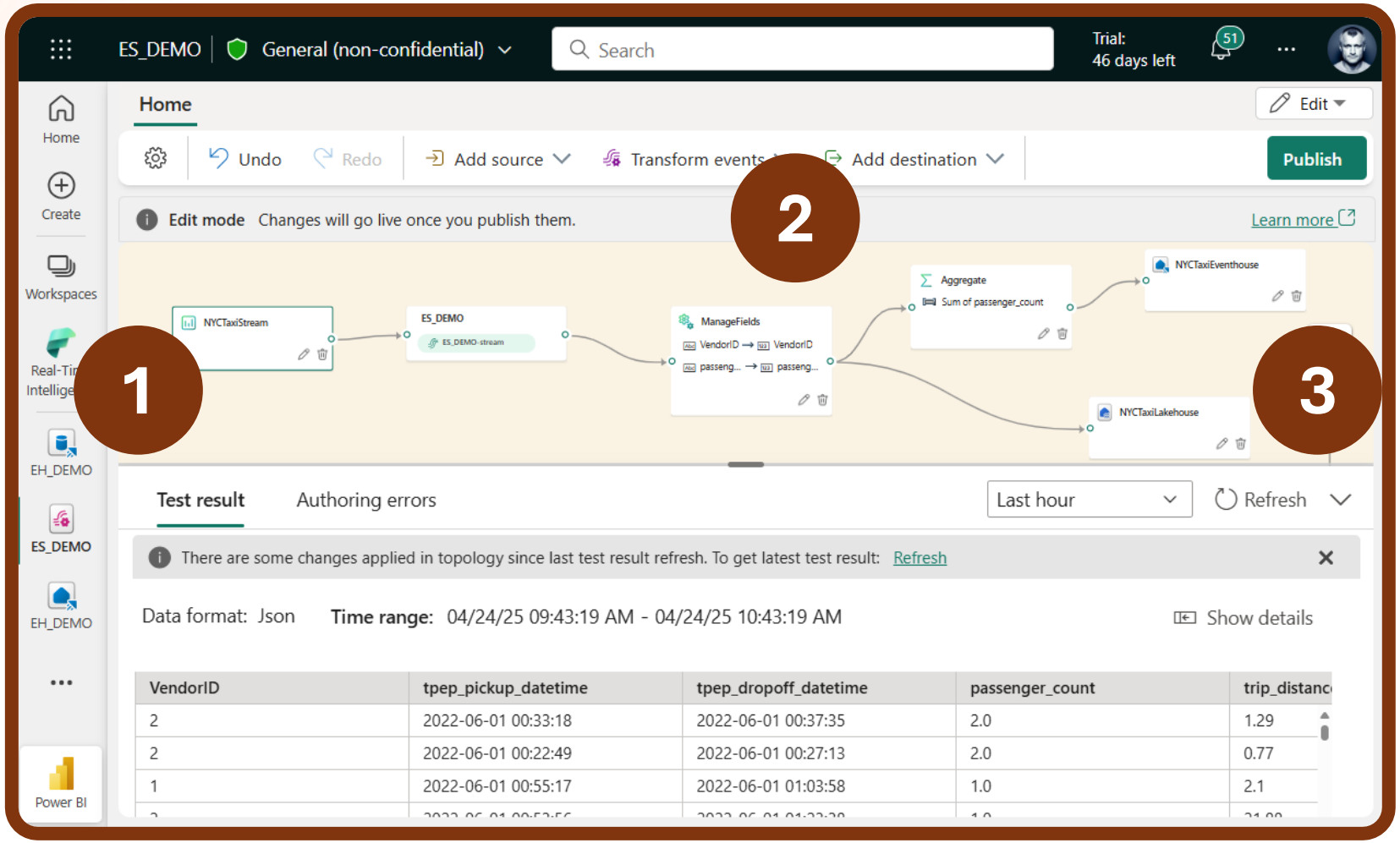

SQL Day
DEMO
INTRODUCTION TO FABRIC REAL-TIME INTELLIGENCE

Real-time data sources

- Streaming cloud services
  - Azure: Event Hubs, IoT Hub
  - GCP: Pub/Sub
  - AWS: Kinesis
- Change Data Capture (CDC) streams from databases
- Apache Kafka (e.g. Confluent)

Eventstreams in use

1. Sources
2. Transformations
3. Destinations



Eventstream sources and destinations

Sources		Destinations
• Azure Event Hub • Azure IoT Hub • Azure Event Grid Namespace • Azure Service Bus • Azure Data Explorer DB • Azure SQL DB Change Data Capture (CDC) • PostgreSQL DB CDC • MySQL DB CDC • Azure Cosmos DB CDC • Azure SQLMI DB CDC • SQL Server on VM DB CDC	• Google Cloud Pub/Sub • Amazon Kinesis Data Streams • Amazon MSK Kafka • Confluent Cloud Kafka • Apache Kafka • MQTT • Solace PubSub+ • Fabric Workspace Item events • Job events • OneLake events • Azure blob storage events • Custom endpoint • Sample data	• Eventhouse • Lakehouse • Custom endpoint • Derived Stream • Fabric Activator

Eventstream transformations

SQL Day

Transformations

- Filter
- Aggregate
- Union
- Expand
- Join
- Group by

Temporal Windows

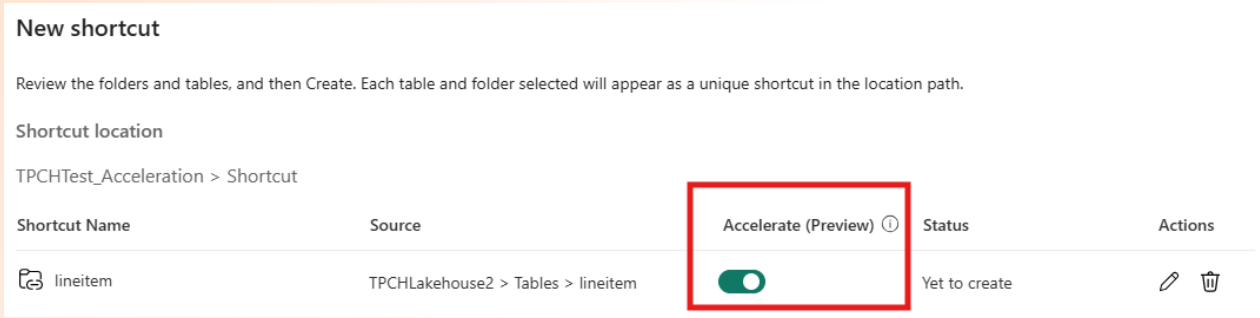
- Tumbling
- Sliding
- Session
- Hopping
- Snapshot

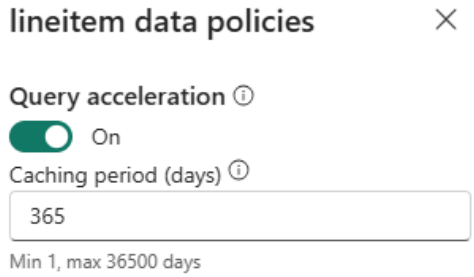
DEMO

1. LAMBDA ARCHITECTURE IN FABRIC
2. REAL-TIME ANALYTICS WITH REFERENCE DATA
3. FROM INSIGHTS TO ACTIONS WITH ACTIVATOR

Selected unique capabilities of RTI

Query acceleration over OneLake shortcuts





- High-performance queries over shortcuts in Eventhouse
- Policy on top of external delta tables - number of days to cache data

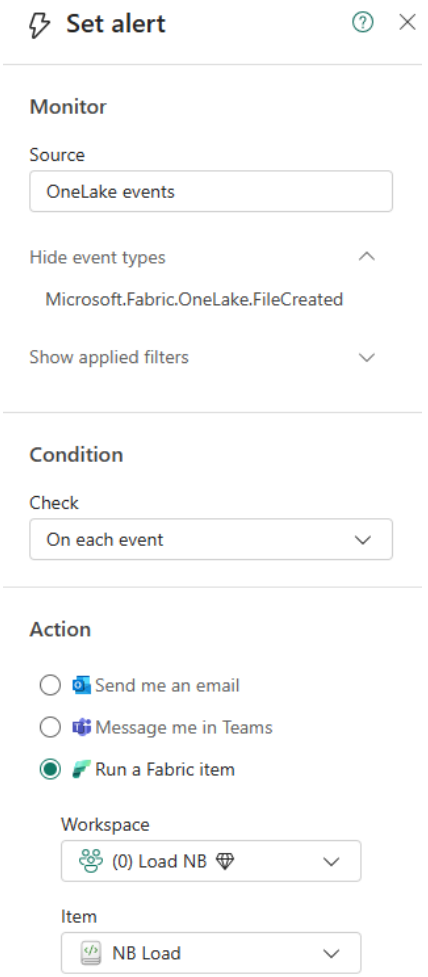
```
.show external table lineitems operations query_acceleration statistics
```

- Multiple sources supported
  - OneLake shortcuts (+all destinations supported by OneLake shortcuts), Azure Data Lake Store Gen1, Amazon S3, Google Cloud Services, Azure Blob Storage
- See performance benchmark: https://dcode.bi/blog/accelerate/

Selected unique capabilities of RTI

Fabric Events

- Events based on Fabric **monitor activities, OneLake operations** or actions on Fabric items
- Can be configured in **Real-Time hub**
- Alerts (activators) can be set to **trigger actions** based on events of specific types (e.g. new file added in specific lakehouse)
- **An action can be running a Fabric item**

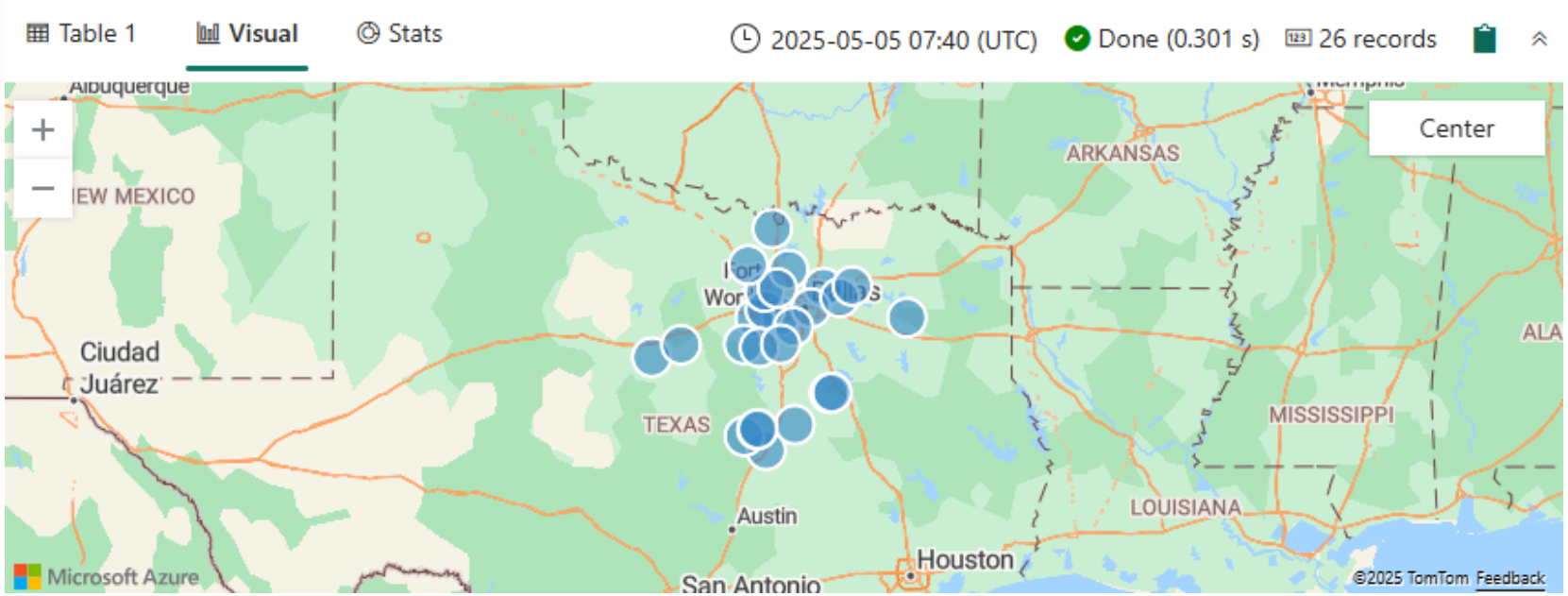


Selected unique capabilities of RTI

Geospatial analysis

- Functions
- Visualizations
- Clustering
- Joins

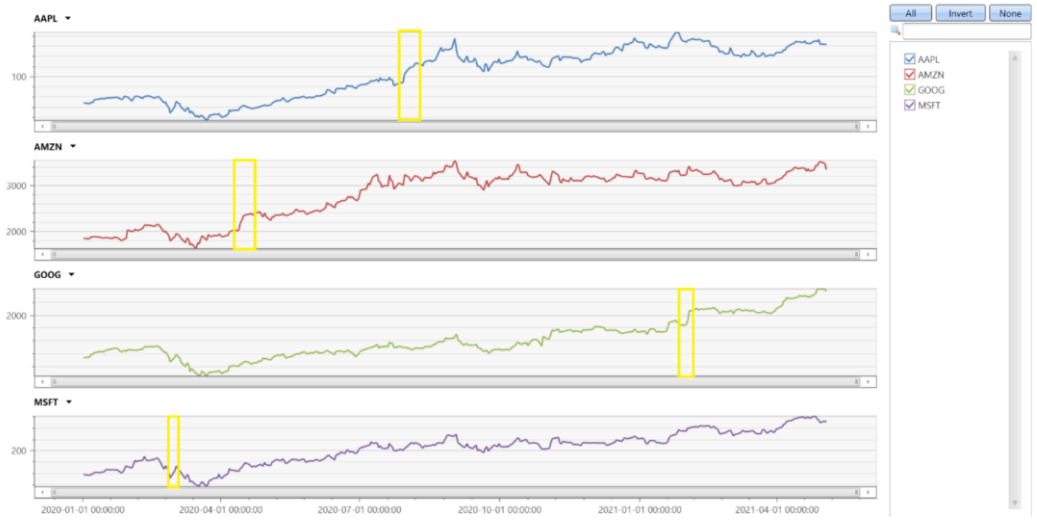
```
1  Weather
2  | where EventType in ('Tornado')
3  | where geo_distance_2points(-96.79802652188019, 32.77683864575861, BeginLon, BeginLat) <= 200000
4  | project BeginLon, BeginLat, EventType
5  | render scatterchart with (kind = map)
```



Selected unique capabilities of RTI

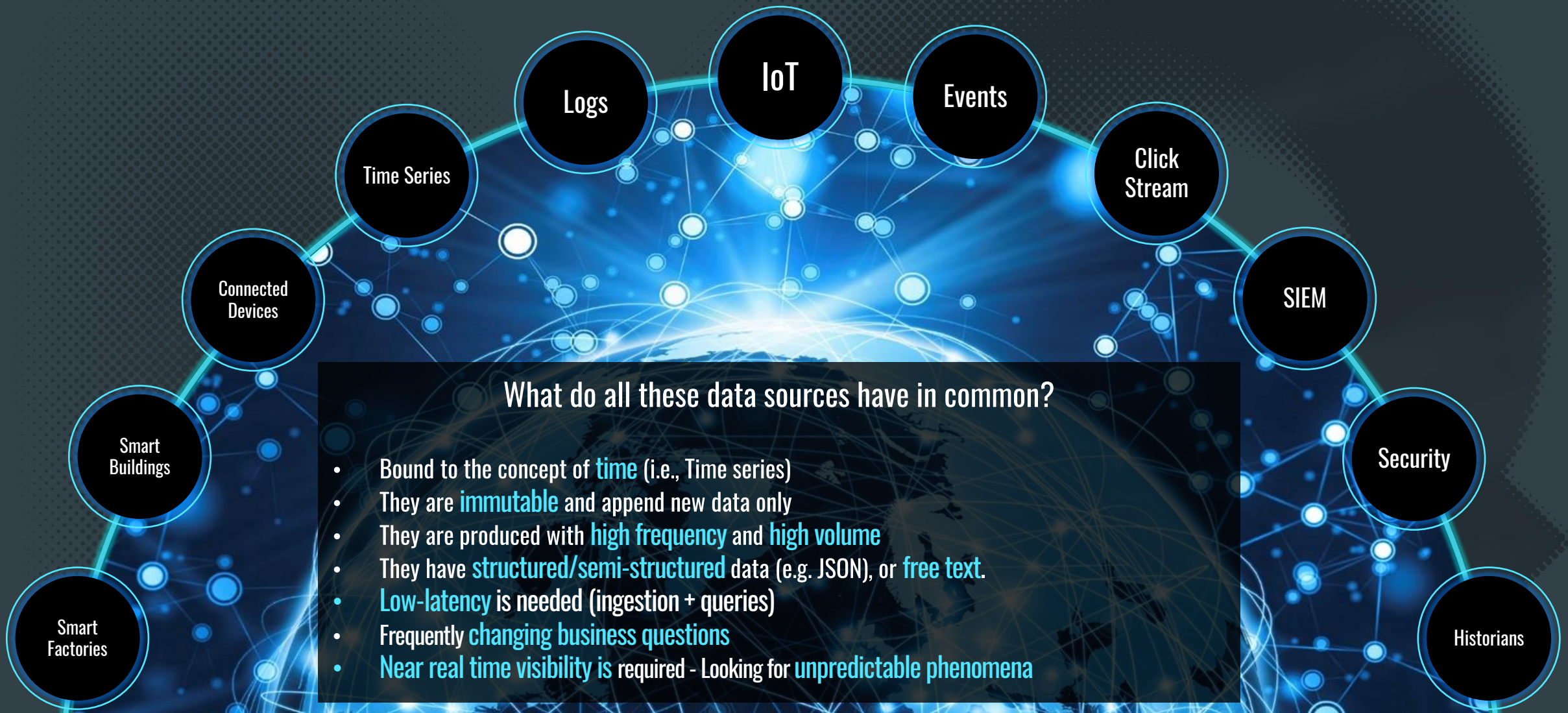
KQL built-in functions

```
TopStocksCleaned
| where DateTime >= datetime(2019-01-01) and Ticker in('AAPL', 'AMZN', 'GOOG', 'MSFT')
| partition by Ticker
(
    order by DateTime asc
    | extend pDate=prev(DateTime), pAdjClose=prev(AdjClose)
    | extend delta = AdjClose - pAdjClose
    | scan with_match_id=m_id declare(down:bool=false, step:string) with
    (
        //  if state of s1 is empty we require price increase, else continue as long as price doesn't decrease
        step s1: delta >= 0 and (delta > 0 or isnotnull(s1.delta)) => step = 's1';
        //   exit the 'rally' when price decrease, also forcing a single match
        step s2: delta < 0 and s2.down == false => down = true, step = 's2';
    )
)
| where step == 's1'   //  select only records with price increase
| summarize (start_date, start_AdjClose, start_delta)=arg_min(pDate, pAdjClose, delta), end_date=max(DateTime), trading_days=count(), total_delta=sum(delta) by Ticker, m_id
| extend delta_pct = total_delta*100.0/start_AdjClose
| summarize arg_max(delta_pct, *) by Ticker
| project Ticker, start_date, end_date, trading_days, delta_pct, start_AdjClose, total_delta
| order by delta_pct
```



Ticker	start_date	end_date	trading_days	delta_pct	start_AdjClose	total_delta
AAPL	2020-07-29	2020-08-07	7	20.752	93.75	19.455
AMZN	2020-04-13	2020-04-21	6	18.461	2040	376.610
MSFT	2020-02-28	2020-03-03	2	14.034	152.410	21.389
GOOG	2021-01-28	2021-02-03	4	12.422	1843.939	229.060

Use cases



What do all these data sources have in common?

- Bound to the concept of time (i.e., Time series)
- They are immutable and append new data only
- They are produced with high frequency and high volume
- They have structured/semi-structured data (e.g. JSON), or free text.
- Low-latency is needed (ingestion + queries)
- Frequently changing business questions
- Near real time visibility is required - Looking for unpredictable phenomena

Summary

- Always define what „real-time” means
- Fabric RTI provide capabilities for data engineers to handle real-time data
- Verify whether your data characteristics fit RTI
- Fabric RTI is NOT a replacement for Data Warehouse or Lakehouse

Resources

- Fabric RTI documentation: https://learn.microsoft.com/en-us/fabric/real-time-intelligence/
- Fabric RTI end-to-end tutorial: https://learn.microsoft.com/en-us/fabric/real-time-intelligence/tutorial-introduction
- Medallion architecture in Fabric RTI: https://blog.fabric.microsoft.com/en-us/blog/21597?ft=All
- Kusto Detective Agency: https://detective.kusto.io/
- Brian Bonk's blog: https://dcode.bi/blog/
- KQL.how: https://kql.how/

Data
Community