Real-Time Intelligence

In Microsoft Fabric

The world is exploding with
event data & data in motion...

How do you turn it into a
competitive edge?

Automotive

Manufacturing

Logistics

Finance & Insurance

Energy & Utilities

Retail

The negative impacts of stale data do not discriminate across industries

- Poor decision-making
- Financial loss
- Reduced operational efficiency
- Impaired data integrity
- Risk of non-compliance
- Compromised user experience

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

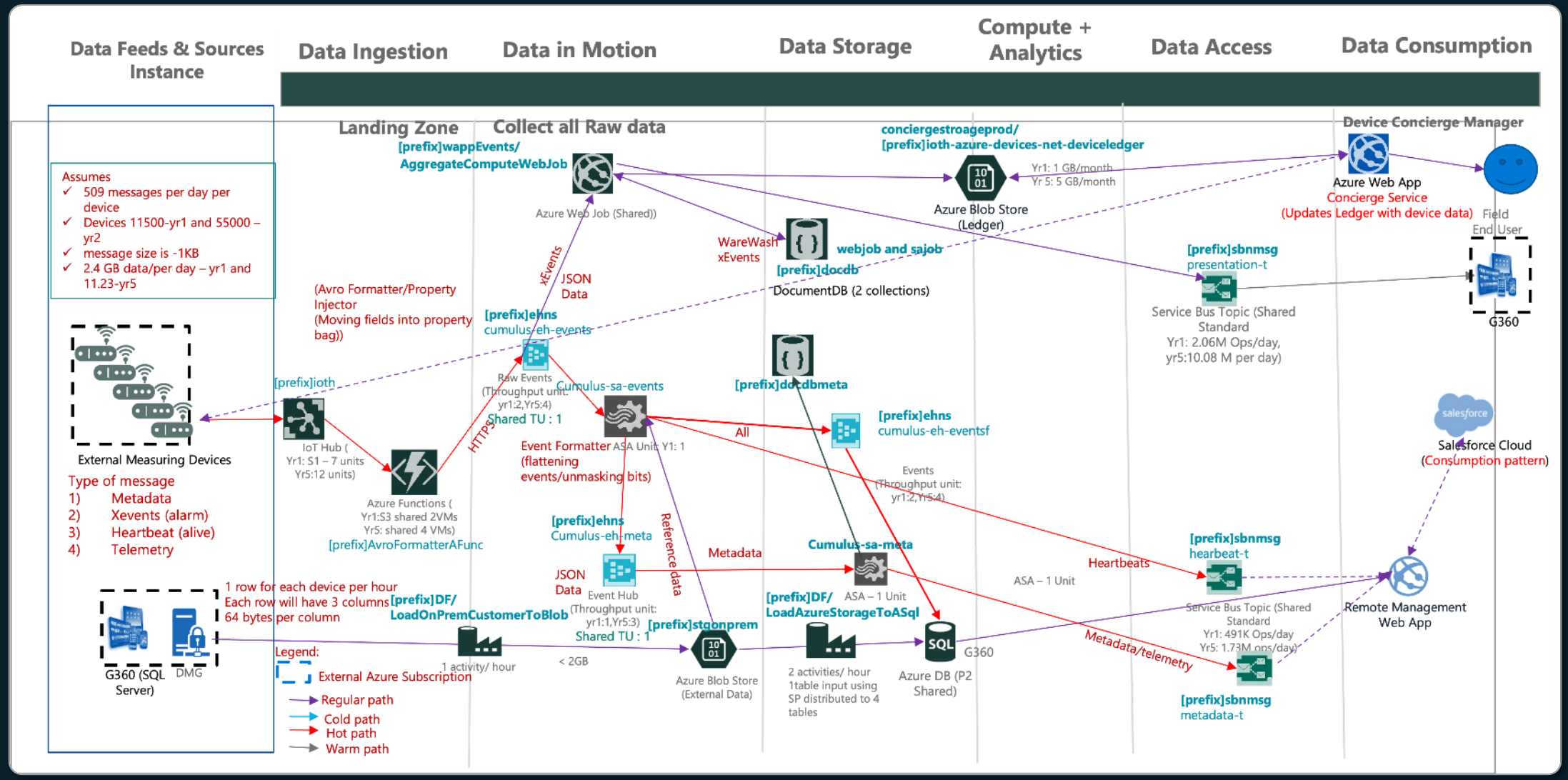
Energy & Utilities

- Station monitoring, energy leakage detection
- Equipment Maintenance & Monitoring
- Failure Monitoring

Retail

- Inventory tracking
- Promotions and buying experiences
- Supply chain management

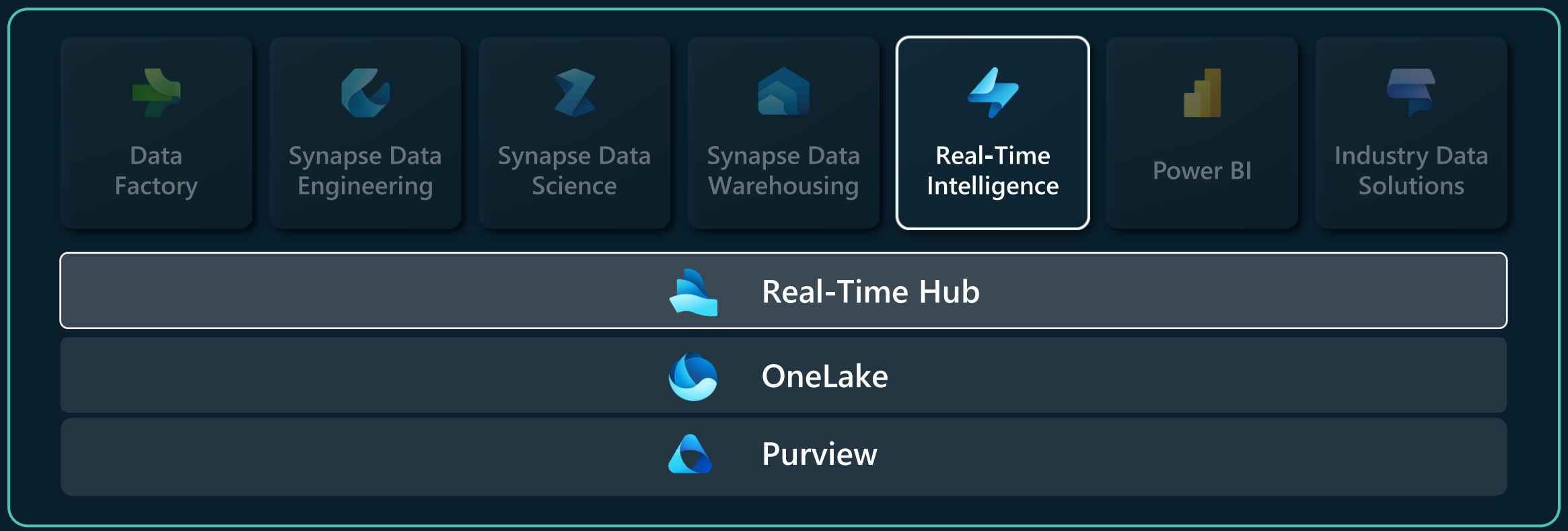
Hard to build real-time solutions today



Real-Time Intelligence

Microsoft Fabric

The unified data platform for the era of AI



Real-Time Intelligence in Microsoft Fabric

Microsoft Azure

Enterprise real-time data platforms

Azure Event Hub

Azure Event Grid

Azure Stream Analytics

Azure Data Explorer

+

Self-service reporting and activation

OneLake

Data Activator

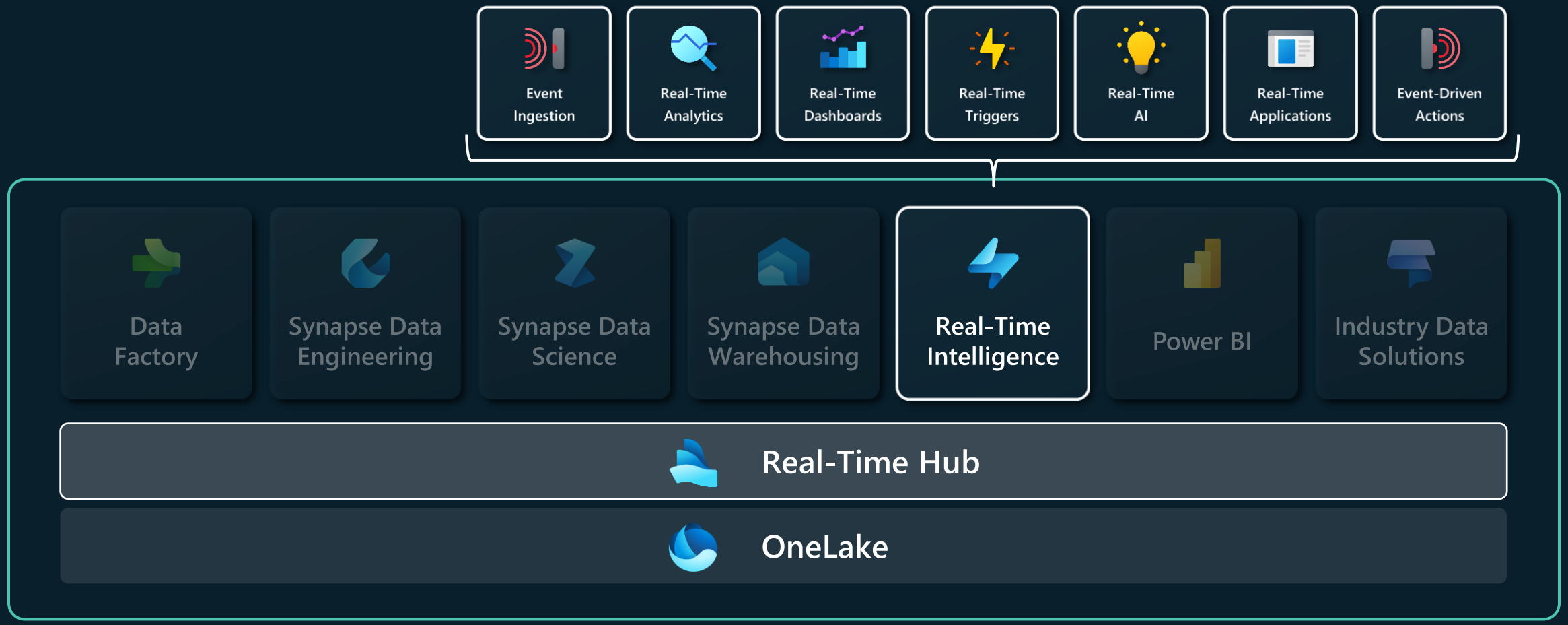
Power BI

=

Real-Time Intelligence in Microsoft Fabric

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

Real-Time Intelligence in Microsoft Fabric



Real-Time Intelligence in Microsoft Fabric

Best-in-class Connectivity

Real-Time Hub

Process

Analyze

Monitor

Act

OneLake

AI-powered Intelligence

Real-Time Intelligence in Microsoft Fabric

Complete SaaS solution

- Everything, unified
- SaaS experience
- Lightweight modeling

Single data estate & open

- Events out-of-the-box
- Open data format
- Data rests in OneLake

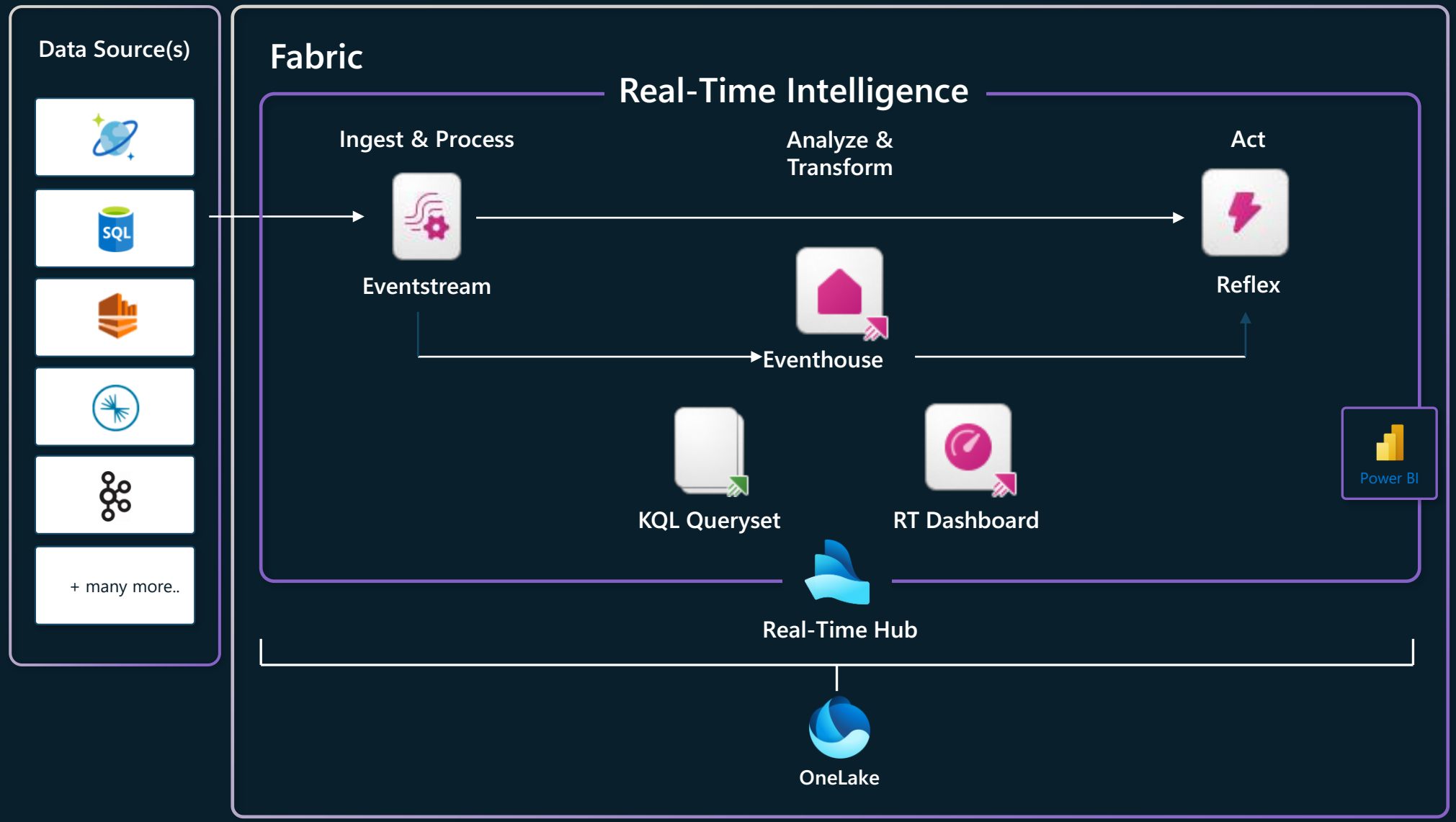
Rapid app development

- Reuse of models
- Multi-tenant security
- Event pub-sub

Real-time AI insights

- Non-obvious insights
- Automatic learning
- Copilot acceleration

Components of Fabric's Real-Time Intelligence



Real-Time Intelligence in Microsoft Fabric

Ingest & Process

- **Streaming sources** including AMQP, Kafka, Azure Event Hubs and more
- **No-code experience** including event processing
- **Event routing** to Eventhouse, Reflex and other Fabric entities

Analyze

- **Timeseries database** with unlimited scale (query, ingestion, storage)
- **Transformation on read** of complicated data structure
- **Copilot** for generating queries

Act

- **Take actions automatically** when patterns or conditions are detected
- **Light-weight modeling** empowering the business user

Real-Time Hub

Central place to discover data and launch all related tasks

Connected Factory

An end-to-end Real-Time Intelligence experience

Ingest & process

1. Over 1M/1 HR IIoT events, 30k tags, from 40 factories are streamed in real time with sub-second latency
2. Contextualization data (OPC asset metadata, shift details, weather, component cost, etc.) mirrored into Fabric OneLake

Analyze & transform

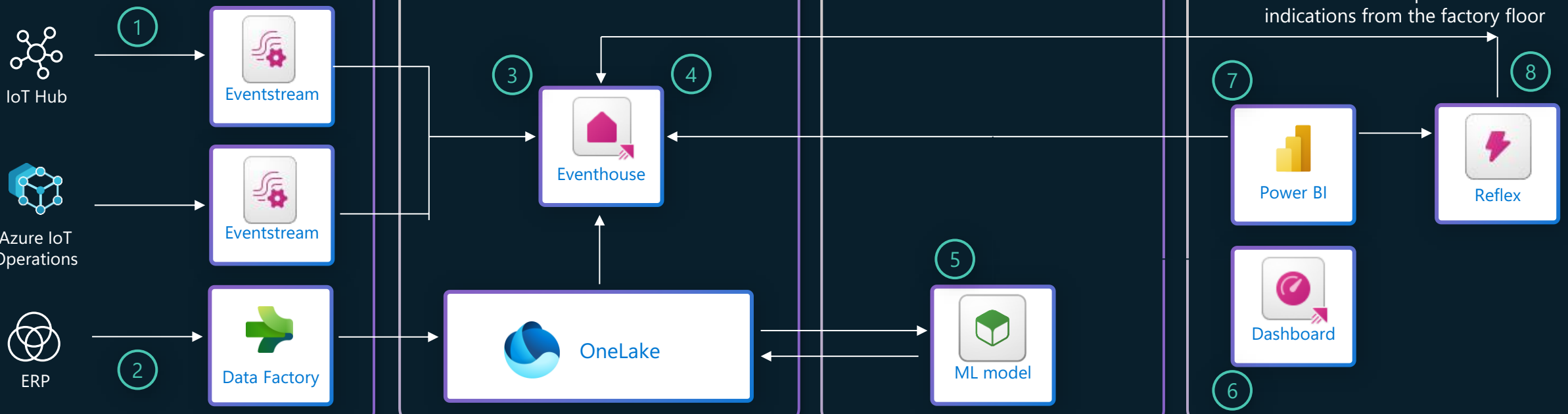
3. IIoT events are enriched in-motion with the Contextualization data and asset hierarchy from OneLake, providing clean and meaningful views
4. Enriched data is aggregated on the fly for long-term view

Train

5. Build, train, and score ML models in real time, persisting the scored data in OneLake

Visualize and activate

6. Real time dashboard of all factories with ability to easily shift from low granularity to high granularity view
7. Enhance organizational BI reports with high granularity enriched data
8. Notify users and activate automation in response to live indications from the factory floor



What is a Kusto query?

A Kusto query is a read-only request to process data and return results.

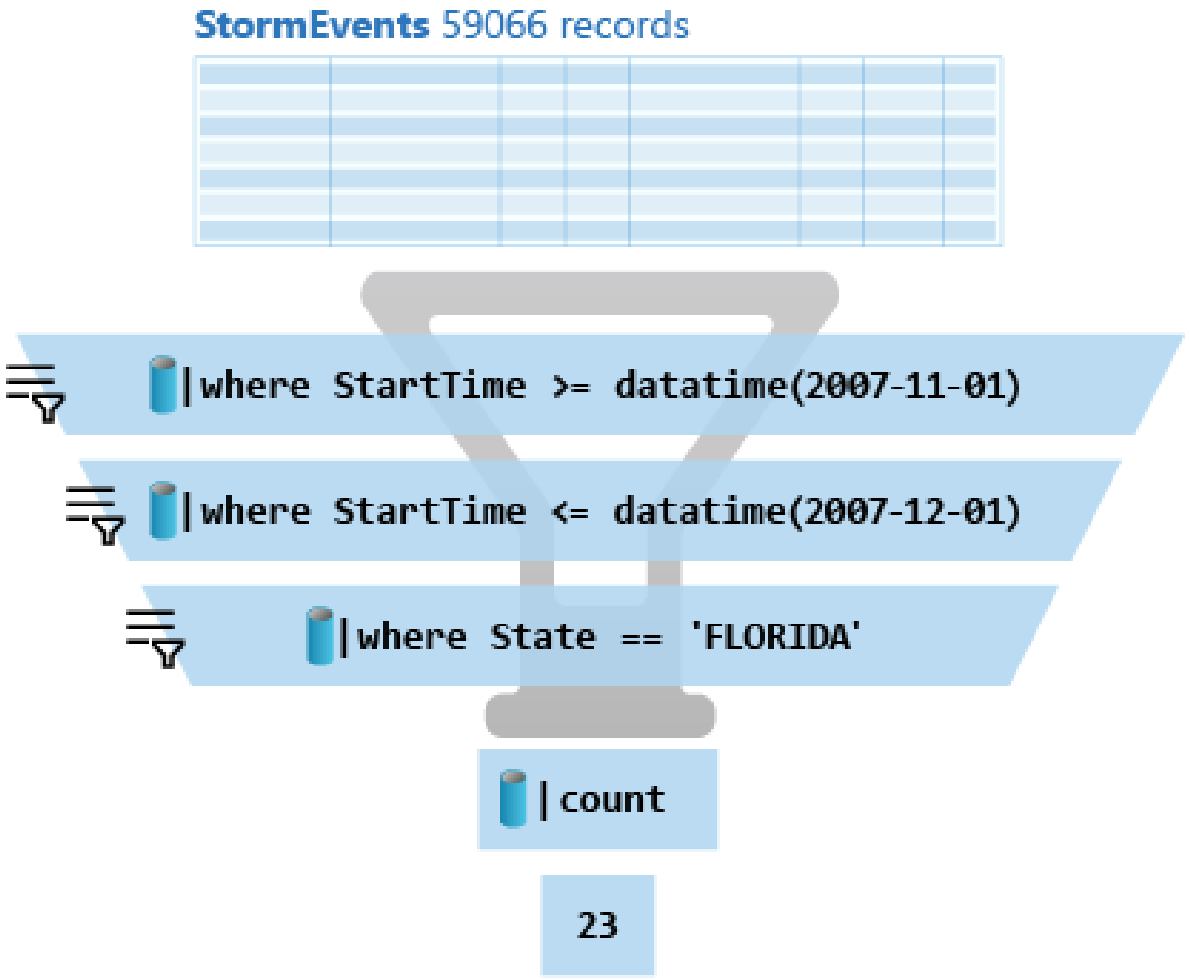
Has one or more query statements and returns data in a tabular or graph format.

Statements are sequenced by a pipe (|).
Data flows, or is piped, from one operator to the next.

It's like a funnel, where you start out with an entire data table.

The data is filtered/manipulated at each step and then fed into the following step.

Each time the data passes through another operator, it's filtered, rearranged, or summarized.



KQL Concepts

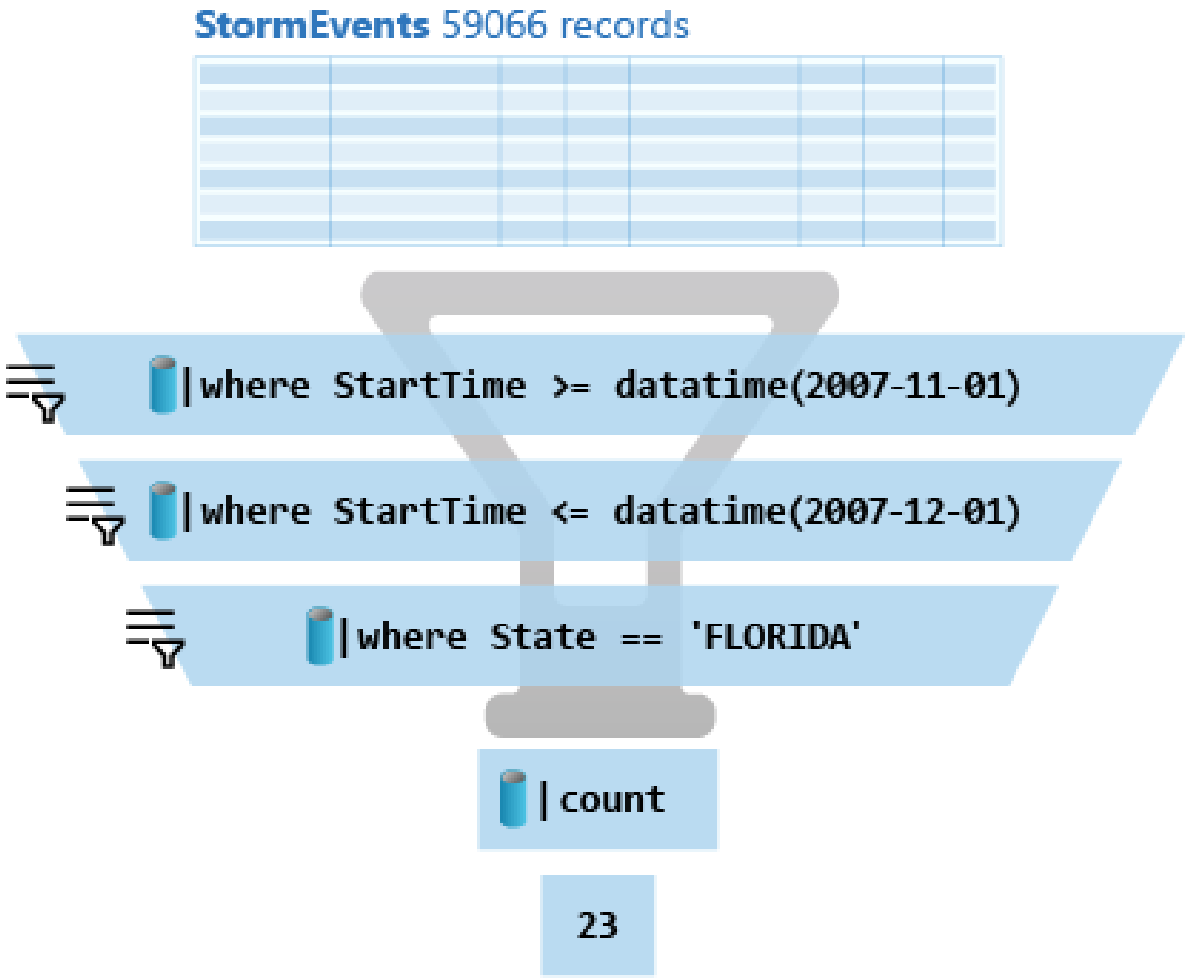
Relational operators
filters, union, joins, aggregations, …
Each operator consumes **tabular input**
and produces a **tabular output**

Commands
Can be combined with '|' (**pipe**)

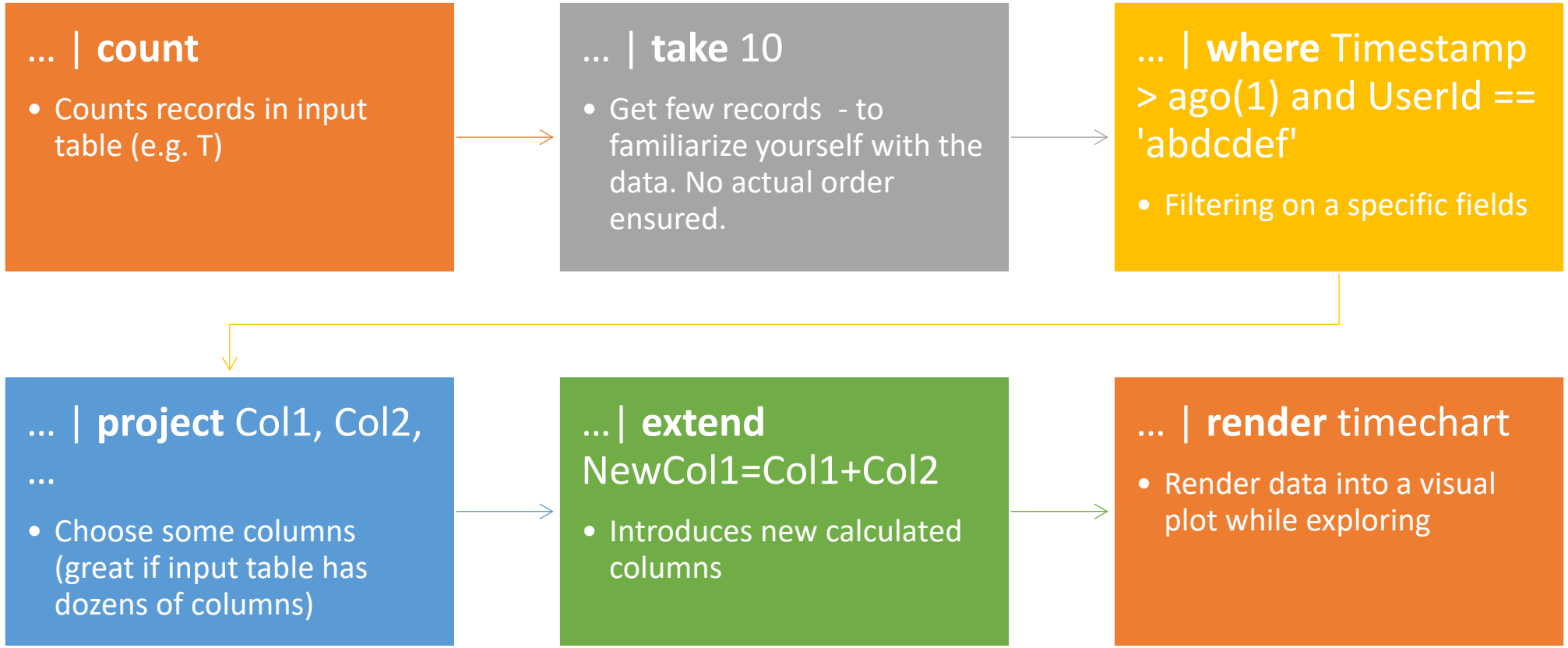
Similarities
OS shell, Linq, functional SQL…

Agility
Queries are easy to **write, read, change**

Statements
Single statement query
Use 'let' for reusing statements
Multi-statement (';') queries



KQL Basic Operators for data exploration



SQL to KQL

Try the 'EXPLAIN' operator as follows:

```
EXPLAIN SELECT
COUNT_BIG(*) as C
FROM LogisticsTelemetry
```

Use SQL to KQL Cheat Sheet

Growth mindset 😇

Schema

- **Schema is**
- Relational, Lightweight, Dynamic

- **Databases**
- Authorization boundary
- Transaction boundary
- But not query boundary!
- Supporting cross-database and cross-cluster queries

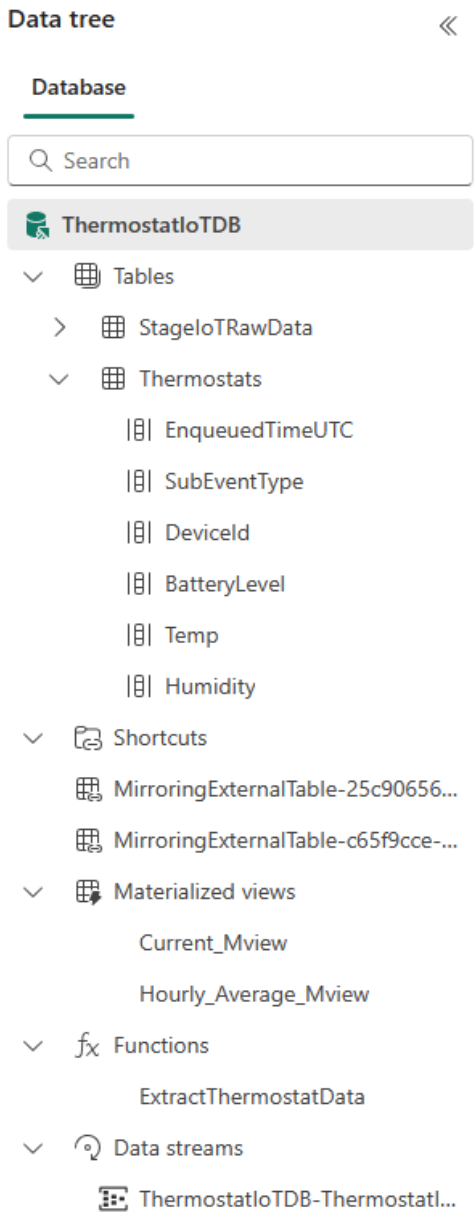
- **Tables**
- Rectangular

- **Columns**
- Supported types: boolean, integer, real, decimal, dates, timespan, string, dynamic (JSON)

- **Shortcuts** (external data)

- **Stored functions** (views)

- **Materialized views**



Time Series Analysis

make-series
Operator for creating set of (time) series

Large family of functions can be applied
- Element-wise operations
- Filtering (signal processing)
- Statistics
- Regressions
- Seasonality detection/validation
- Anomaly Detection
- Forecasting
- mv-apply operator

Optimized for bulks of time series

avg_shock_series	enqueuedTime
[-0.014102389298512921, 0.054822081643445778, -0.024335543043808073, 0.0073609799437805252, -0.034215766864223458, -0.031322885028220365, 0.007474701289064049,	["2022-01-25T16:53:16.9056584Z", "2022-01-25T17:03:16.9056584Z", "2022-01-25T17:13:16.9056584Z", "2022-01-25T17:23:16.9056584Z", "2022-01-25T17:33:16.9056584Z", "2022-01-25T17:43:16.9056584Z", "2022-01-25T17:53:16.9056584Z",