Microsoft Fabric

#FABCONSQLCON2026

FABCON
Microsoft Fabric
COMMUNITY CONFERENCE

SQLCON
Microsoft SQL
COMMUNITY CONFERENCE

ATLANTA MARCH 16 - 20, 2026

Mastering Eventhouse: Architecture, Patterns, and Real-World Scenarios

Christopher Schmidt – Principal PM

Brad Watts – Principal PM

Learning Objectives

RTI Overview

Overview: Modern Medallion Architecture

Best Practices each Layer (Bronze, Silver, Gold)
le and resilient schema evolution in real-time.

Lessons Learned

Wrap- Up

Microsoft Fabric

The unified data platform for AI transformation

Data Factory

Analytics

Databases

Real-Time Intelligence

IQ

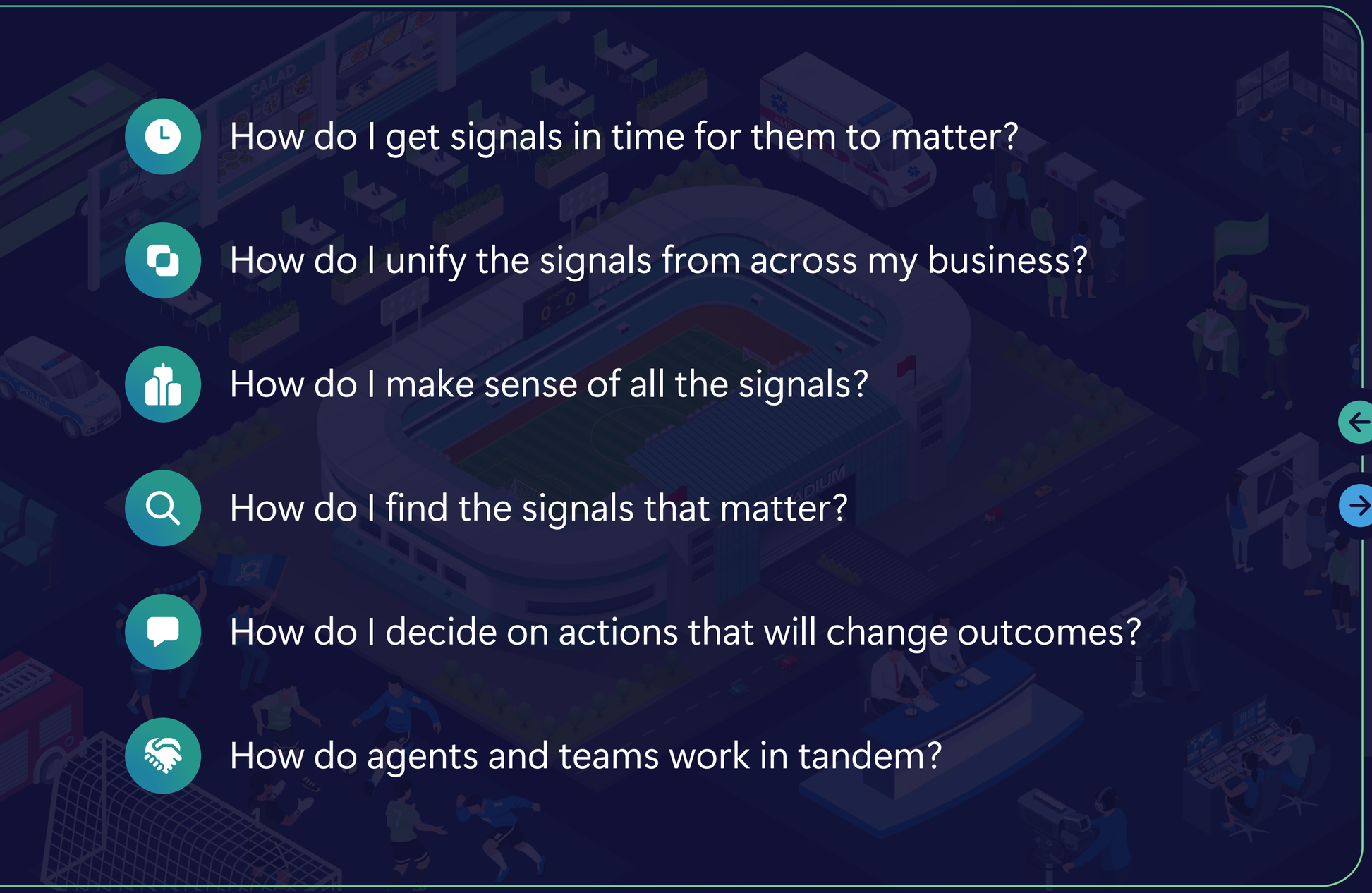
Power BI

Fabric Platform

Copilot

OneLake

Governance



- How do I get signals in time for them to matter?
- How do I unify the signals from across my business?
- How do I make sense of all the signals?
- How do I find the signals that matter?
- How do I decide on actions that will change outcomes?
- How do agents and teams work in tandem?

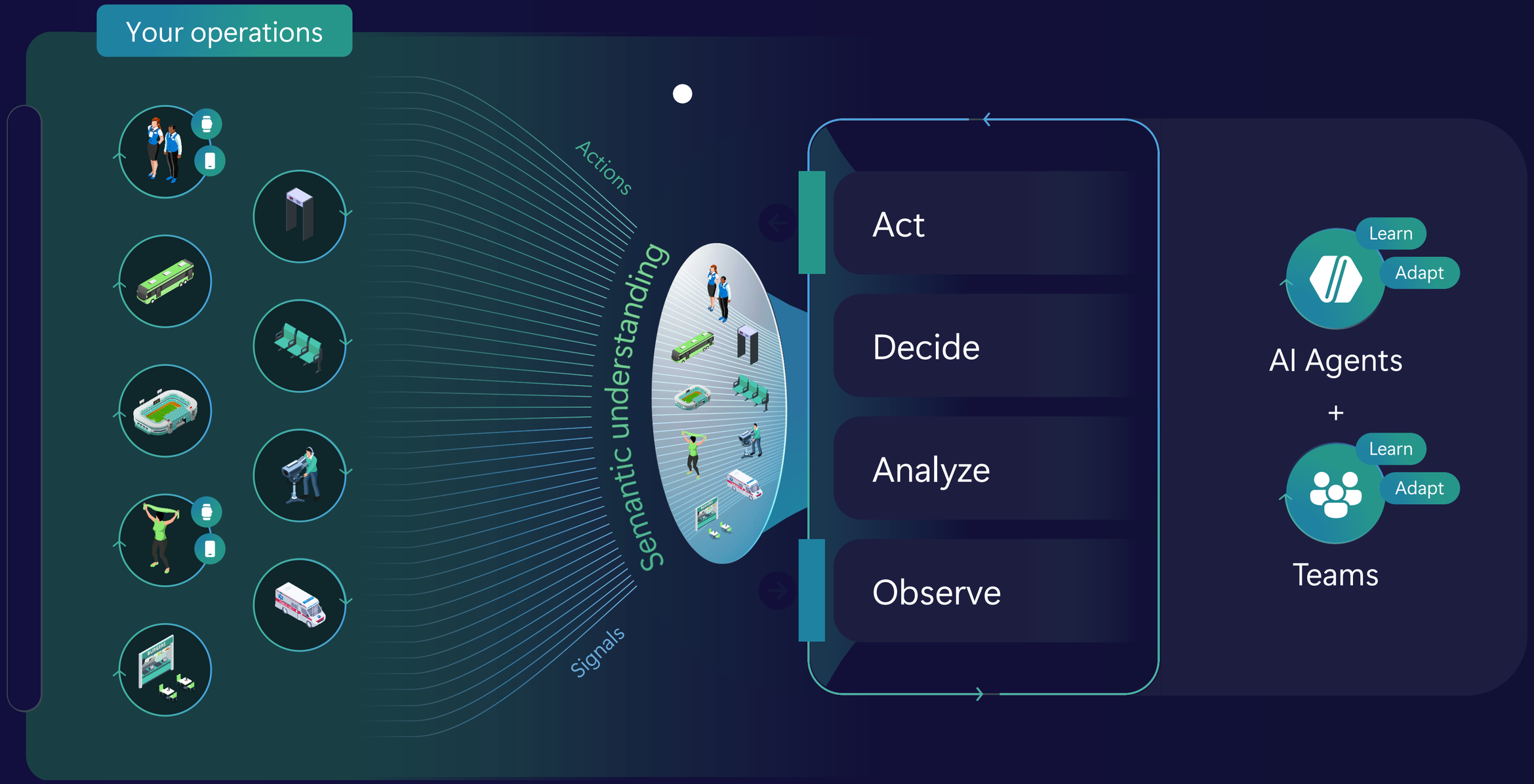
Actions

Signals

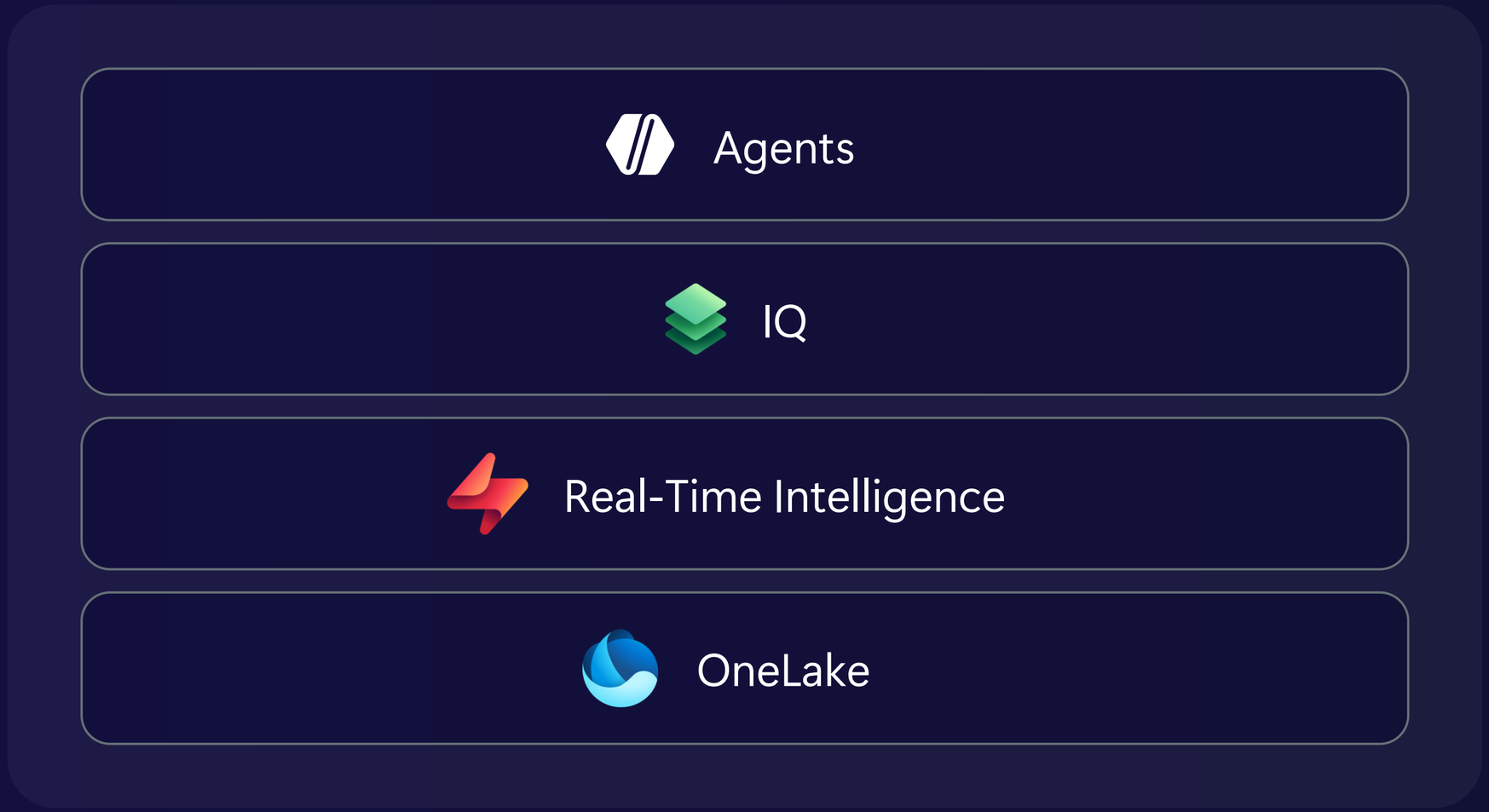
AI Agents

+

Teams


Your operations
Actions
Semantic understanding
Signals
Act
Decide
Analyze
Observe
Learn
Adapt
AI Agents
+
Learn
Adapt
Teams

The unified intelligence platform for AI



Real-Time Intelligence in Microsoft Fabric

Stream

Analyze

Model

Visualize

Act

Fabric Platform

Copilot

OneLake

Real-Time Hub

Governance

Modern Medallion Architecture in Eventhouse

RTI Modern Medallion Architecture

Eventhouse Architecture Essentials

Unified Ingestion Engine

Eventhouse supports both micro-batch and streaming ingestion patterns for flexible data processing.

Dynamic Column Flexibility

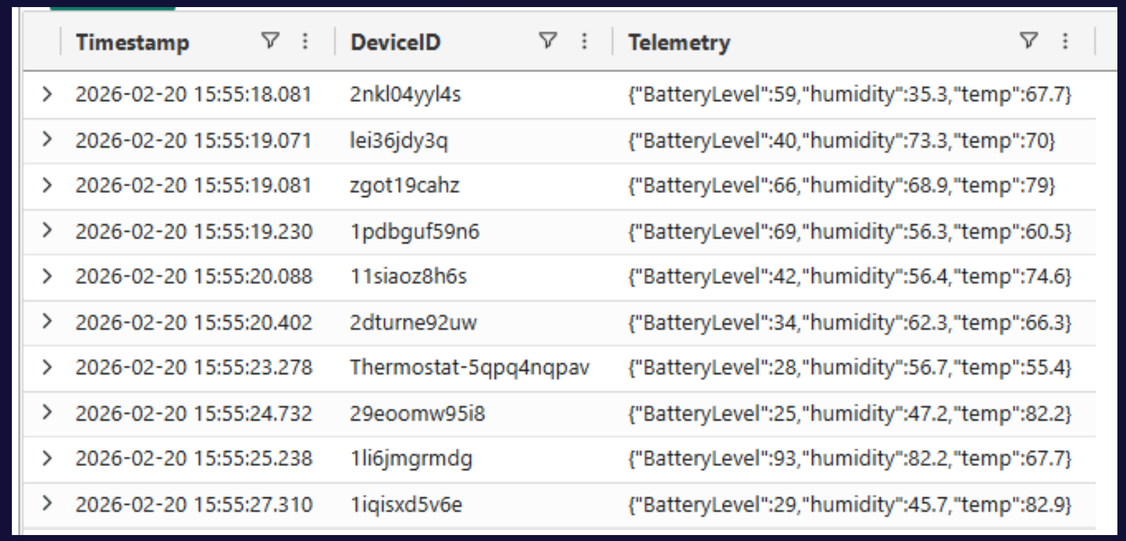
Supports heterogeneous JSON payloads with minimal preprocessing through dynamic columns.

Efficient Data Management

Uses deduplication and incremental refresh via materialized views for optimized analytical tables.

Bronze Layer

- Data Retention
  - Do I need transformations
  - Is this the analytical layer
- Schema
  - How much do I flatten out at this layer
- Streaming or Batch Ingestion
  - What is my latency requirement
  - Will my transformations contain a lookup



Timestamp	DeviceID	Telemetry
2026-02-20 15:55:18.081	2nkl04yyl4s	{"BatteryLevel":59,"humidity":35.3,"temp":67.7}
2026-02-20 15:55:19.071	lei36jdy3q	{"BatteryLevel":40,"humidity":73.3,"temp":70}
2026-02-20 15:55:19.081	zgot19cahz	{"BatteryLevel":66,"humidity":68.9,"temp":79}
2026-02-20 15:55:19.230	1pdbguf59n6	{"BatteryLevel":69,"humidity":56.3,"temp":60.5}
2026-02-20 15:55:20.088	11siaoz8h6s	{"BatteryLevel":42,"humidity":56.4,"temp":74.6}
2026-02-20 15:55:20.402	2dturne92uw	{"BatteryLevel":34,"humidity":62.3,"temp":66.3}
2026-02-20 15:55:23.278	Thermostat-5qpq4nqpav	{"BatteryLevel":28,"humidity":56.7,"temp":55.4}
2026-02-20 15:55:24.732	29eoomw95i8	{"BatteryLevel":25,"humidity":47.2,"temp":82.2}
2026-02-20 15:55:25.238	1li6jmgrmdg	{"BatteryLevel":93,"humidity":82.2,"temp":67.7}
2026-02-20 15:55:27.310	1iqisxd5v6e	{"BatteryLevel":29,"humidity":45.7,"temp":82.9}

Ingestion Patterns

- **Unmanaged (Streaming) ingestion * RTI Default**
- Low latency (<10 second) ingestion path
- Ideal for real-time dashboards, anomaly detection, operational monitoring Uses Direct Ingestion through SDK, Event Hub, Eventstream, or Kusto Data Management service
- Can experience backpressure or rate throttling at extreme ingest volumes
- Raw events land quickly → must handle schema drift, dedup, transformations downstream
- Best when freshness matters more than ingestion cost efficiency

- **Managed (batch) ingestion *ADX Default**
- **Optimized, micro-batch ingestion** managed by Kusto ingestion service
- Ideal for **high-volume**, **cost-efficient**, **reliable** ingestion workloads Files (Parquet, CSV, JSON,XML, etc.) ingested from **Blob/ADLS** via managed identity
- Kusto automatically handles **partitioning**, **compression**, **indexing**, **data mapping**
- Supports **large payloads** and **schema evolution** via mappings / update policies
- Best when **cost & throughput matter more** than sub-second latency

Silver Layer

- Data Retention
  - Is this my analytical layer?
  - Do I have duplicates?
- Transformations
  - What addition context is needed?
  - Do I have all the columns I need?
    - Add columns based on logic
    - When A, B, and C then Column is J else it's I
  - Normalization
    - Case, Remove text, etc..

```
.create-or-alter function ExtractThermostatData() {
ThermostatBronze
| lookup materialized_view('DimDevicesMV') on $left.DeviceID==$right.DeviceId
| extend additonalInfo=bag_remove_keys(Telemetry, dynamic(['BatteryLevel', 'temp','TelemetryType','humidity']))
| project
Timestamp,
DeviceID,
BatteryLevel = tolong(Telemetry.['BatteryLevel']),
Temp =  toreal(Telemetry.['temp']),
Humidity =  toreal(Telemetry.['humidity']),
Model,
SerialNumber,
additonalInfo
}
```

Timestamp	DeviceId	BatteryLevel	Temp	Humidity	Model	SerialNumber	AdditionalInfo
2026-02-20 19:56:41.801	Occupancy	23					{"DwellTime1":6.7,"DwellTime2":11.122584
2026-02-20 19:56:42.059	Thermostat-5qpq4nqpav	60	66.9	61.5	A401	91030a82-73d7-4e78-97ac	{}
2026-02-20 19:56:44.114	2o6dde7n3is	72	75.3	65.8	A402	5c6c703a-d3f6-4beb-b714	{}
2026-02-20 19:56:45.868	2iy38jg2usg	36	51.1	62.1	A402	dc5eeb2b-9cad-42e2-9833	{}
2026-02-20 19:56:46.122	1iqisxd5v6e	32	52.2	72	A402	523ae10b-1507-4d6f-8519	{}
2026-02-20 19:56:46.243	Thermostat-1l7ry5cvclu	16	68.7	53.3	A500	a1f323f0-afcc-4239-bba3-9	{}
2026-02-20 19:56:49.044	2dturne92uw	38	73.3	65	A402	da5b28ba-9536-4564-bfc7	{}
2026-02-20 19:56:49.354	1v08xevlmtu	70	64.2	67.4	A402	38a5bd19-fbf7-45da-b19c	{}
2026-02-20 19:56:51.289	1v9ep03gana	31	63.7	49.6	A401	87449729-deae-464e-9415	{}
2026-02-20 19:56:51.930	Thermostat-3osu0vwx19	17	61.5	62.5	A401	a46702e3-24e4-4298-8332	{}
2026-02-20 19:56:53.414	24eb4jjj47x	19	83.2	46.2	A402	bb471e6f-bdd5-45dc-a1b0	{}
2026-02-20 19:56:53.418	23yjjd1921h	78	71.7	66.1	A401	b2e7ef63-404f-45b2-a6db	{}
2026-02-20 19:56:54.968	1y7je8g2lst	79	82	72.2	A500	6a60237d-343f-4665-b14e	{}
2026-02-20 19:56:55.121	1ld4a112aww	36	53.7	42.7	A500	650116c1-ba01-4c44-934a	{}
2026-02-20 19:56:55.478	s6v47prs8a	53	71.3	72.1	A402	5aafa66c-20f9-4f6b-92be-1	{}
2026-02-20 19:56:55.484	Thermostat-58fkpk9kqe	33	71.1	39	A500	e6ca809c-cf54-4f92-b683-	{}
2026-02-20 19:56:55.484	1x6b1xh31bi	13	67.7	56.9	A401	4a90cbc0-4d5b-4b6d-a037	{}
2026-02-20 19:56:55.888	Thermostat	44	74.7	65.9	A500	8b2fbf7f-e4cf-48ef-a789-7	{}
2026-02-20 19:57:01.014	14pfvg53lg6	30	75.2	46.9	A500	56a0773d-a833-41ba-a51a	{}

Lets Talk about Materialized Views

- Common Use Case
  - Dedup Records (take-any)
  - Aggregate Records/Downsample (bin)
  - Latest Value (arg_max timestamp)
  - Update Records (arg_max ingestion time)
- Considerations
  - Aggregate MV can run on either a table or a Dedup MV (take-any)
  - Dedup MV is typically recommended
  - Update Records MV is powerful but no Aggregate MV on top
- Best Practice Query
  - Query with materialize_view(‘MV Name’)

Gold Layer

- What are my analytical needs?
  - Do I have trending needs over long periods?
  - Are there visuals/alerts that need to be immediate?
- What are my performance needs?
  - Do I pre-aggregate or aggregate at query time?
- What are my retention/caching needs?
  - Can I down-sample for long term retention?

```
// [NOTE!] the materialized view will be created with no 'backfill'
.create-or-alter materialized-view  Thermostats_Hourly_Gold_MV on materialized-view ThermostatSilverDedupMV {
    ThermostatSilverDedupMV
    | summarize avg_Battery_Level=avg(BatteryLevel),avg_Temp=avg(Temp), avg_Humidity=avg(Humidity)
       by DeviceId,Model,SerialNumber, bin(Timestamp,1h)
}
```

Additional Tips

Real-Time + Historical Hybrid Queries

Combine live stream and historical archive in a single KQL query:

- Architecture
- Hot layer: Eventhouse hot cache (recent hours/days)
- Warm layer: Eventhouse cold storage (weeks/months)
- Cold layer: OneLake shortcut → directly queryable via KQL
- Single Eventhouse — no data movement, no separate query engine

- KQL union pattern
- let live   = SensorEvents | where ingestion_time() > ago(1h);
- let history = SensorEvents_Archive | where EventTime > ago(90d);
- union live, history | summarize … by bin(EventTime, 1h)

- Common use cases
- Anomaly vs. historical baseline | Trend vs. real-time state
- Regulatory: audit trail + live ops view in one dashboard
- Predictive: training features from history + real-time inference input

- OneLake shortcut key benefit
- External Delta/Parquet tables (from pipelines, Spark, Dataflow) immediately queryable
- No ETL back into Eventhouse — query-time federation

JSON handling

- **Supports flexible ingestion of semi-structured data**
  - **Raw json, nested object, arrays, schema on read patterns**
- **Dynamic columns for schema agility**
  - **Store unpredictable JSON properties in dynamic type without breaking ingestion**
- **Mapping-based extraction for structure**
  Use JSON mappings to project fields into typed columns (string, long, datetime, etc.).
- **Schema Evolution without breakage**
  New fields → automatically land in dynamic; update policies convert selectively.
- **Efficient querying & indexing**
  Extract with todynamic(), bag_unpack(), dot notation; add indexing strategies for high-frequency fields.
- **Best practice: wide table + dynamic + update policy**
  One canonical table; handle unpredictable fields dynamically; curate stable schema over time.

Cost Optimization Best Practice

Granular Retention Policy

- Optimize retention policy per table
- Retention policy determines how much compressed data is stored in OneLake Storage

Granular Cache Policy

- Each Eventhouse size has a certain amount of hot cache. If you store more then it will scale up
- The caching policy determines how much data is stored in OneLake Premium storage and can affect the size of your Eventhouse

Example

- 0 Days retention on raw table
- 1 Day retention on silver table + 1 Day Hot Cache
- 30 Days retention on dedup materialized view + 7 Days Hot Cache
- 365 Days retention on hourly aggregate materialized view + 90 Days Hot Cache

Materialized Views

- **Down sample: Hourly/Daily Aggregates for example**
- **Don’t overuse**
  - **Don’t create an aggregate for everything!**
  - **Example: Create an hourly aggregate and at query time convert to daily aggregate**

AI Integration: Grounding Intelligence with Real-Time Data

Production patterns connecting Eventhouse to AI workloads:

- Pattern A — Real-Time RAG (Retrieval-Augmented Generation)
- Eventhouse stores product/device/customer context as structured knowledge
- KQL query retrieves relevant context rows at inference time
- LLM prompt = user question + live KQL context → grounded, fresh answers
- No stale data problem: context is always pulled from live Eventhouse

- Pattern B — Copilot in Real-Time Dashboards
- Power BI / RTD Copilot can explain anomalies using Eventhouse-backed queries
- KQL queries auto-generated by Copilot; results narrated in natural language

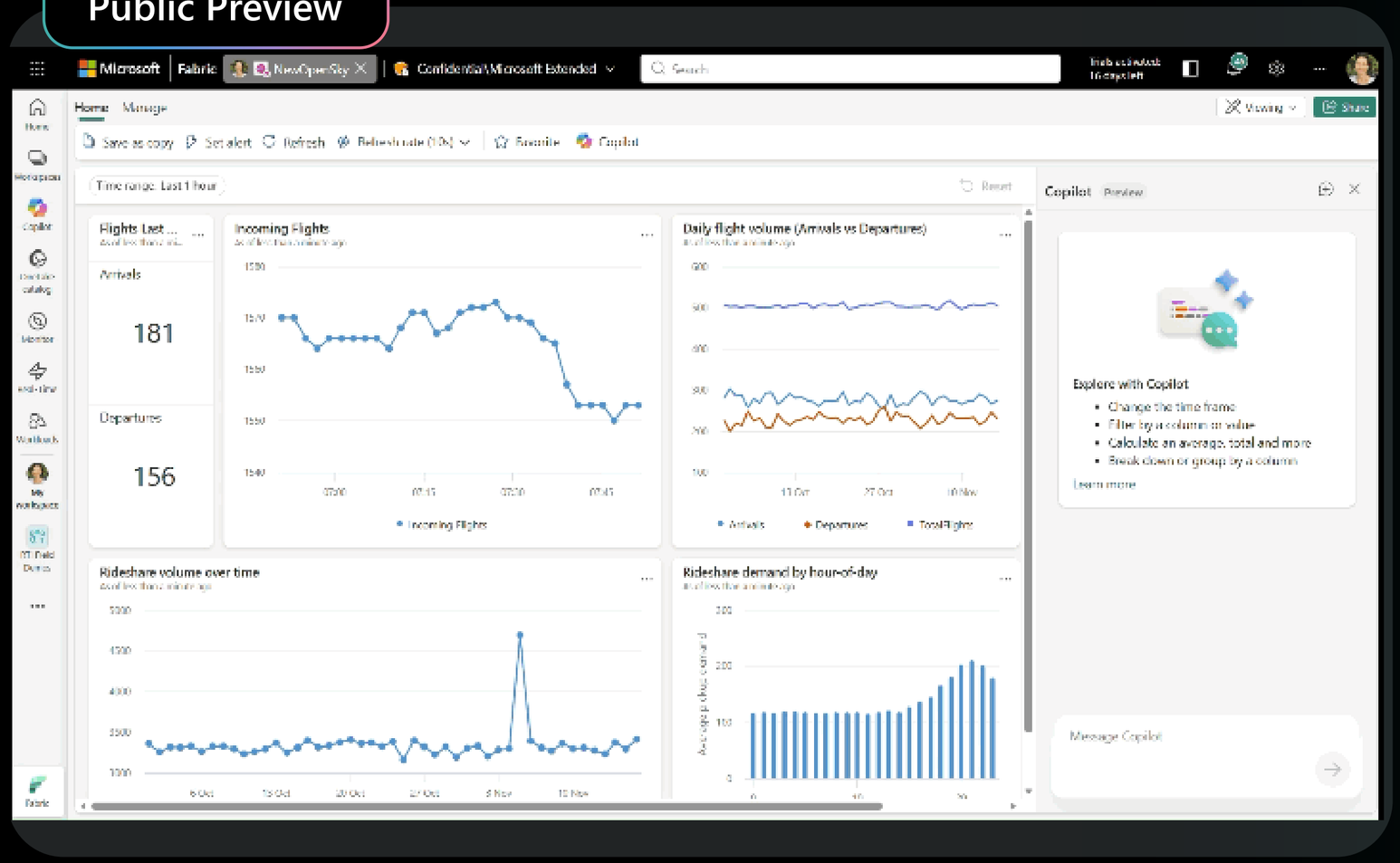
- Pattern C — Agentic Workflows
- AI agent calls Eventhouse via KQL MCP / REST API as a tool
- Agent retrieves latest device state, compares to historical norm, decides action
- Eventhouse as the 'memory' layer: persistent, queryable, time-aware
- Operations Agents poll data directly

- Key enabler
- Eventhouse + OneLake shortcut = structured real-time + unstructured archive
- One query surface for all AI context retrieval — no vector DB required for structured data

Real-Time Intelligence

Explore your live data with natural language using Copilot

Public Preview



Ask in natural language and get instant insights: explore your live data without writing queries, making this accessible to business users

Turn monitoring into exploration into action: ask questions and uncover deeper insights on the spot to address risks and opportunities

Keep the high-value insights: save Copilot-generated visuals directly to your real-time dashboard

Home | Manage | Viewing | Share

Save as copy | Set alert | Refresh | Refresh rate (10s) | Favorite | Copilot

Home | Workspaces | Copilot | OneLake catalog | Monitor | Real-Time | Workloads | My workspace | RTI Field Demos | ...

Time range: **Last 1 hour** | Reset

Flights Last ...
As of less than a mi...

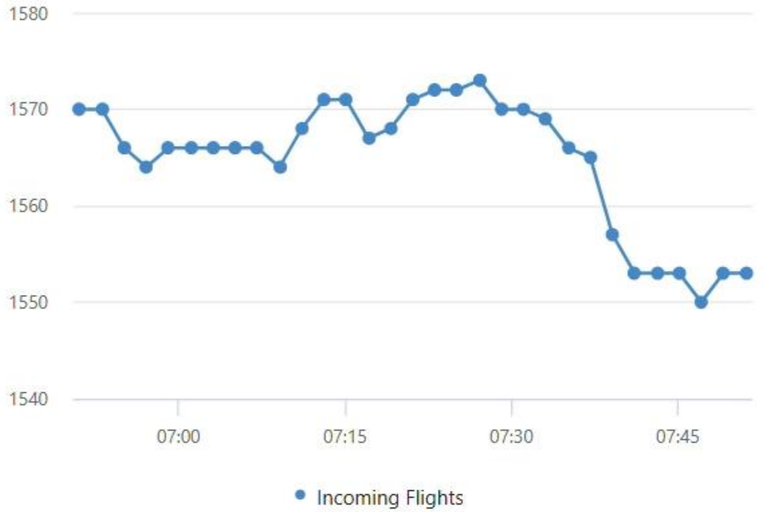
Arrivals

181

Departures

156

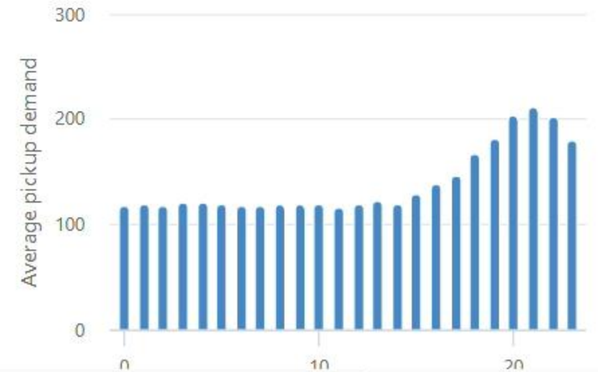
Incoming Flights
As of less than a minute ago



Daily flight volume (Arrivals vs Departures)
As of less than a minute ago

Rideshare volume over time
As of less than a minute ago

Rideshare demand by hour-of-day
As of less than a minute ago



Copilot Preview

Explore with Copilot

- Change the time frame
- Filter by a column or value
- Calculate an average, total and more
- Break down or group by a column

Learn more

Message Copilot

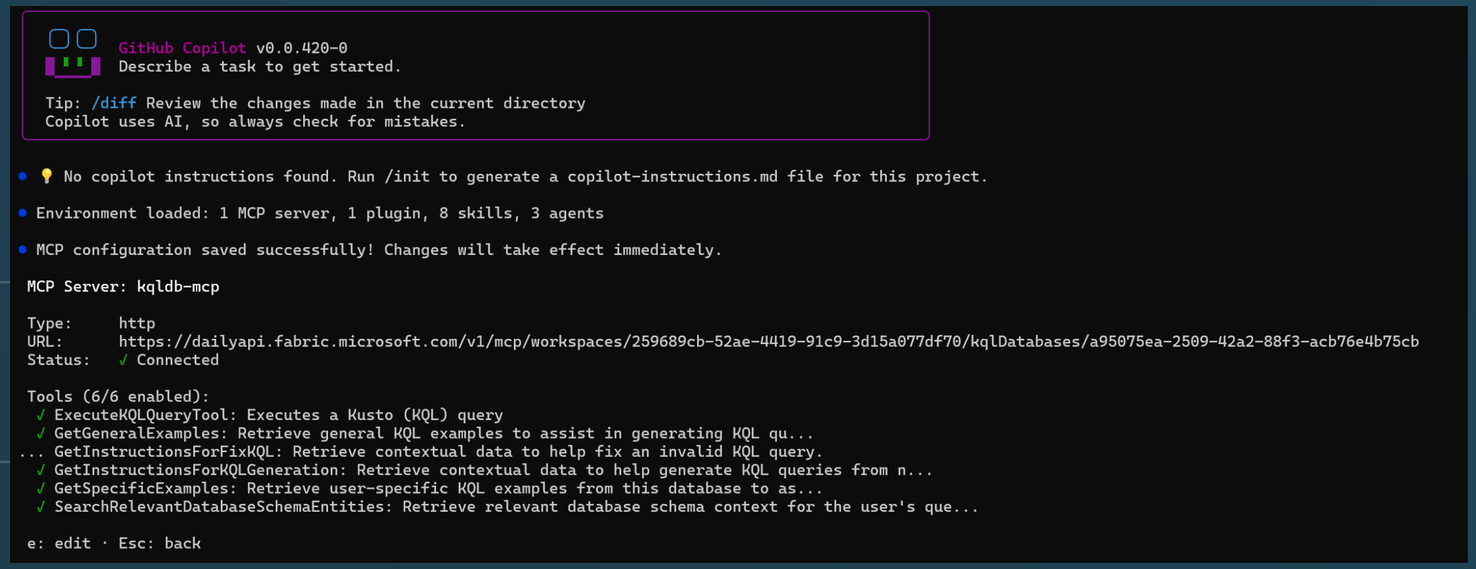
Remote MCP for Eventhouse

Public Preview

Generating KQL queries from natural language conversation

Explore schema in Eventhouse

Natural language to insights

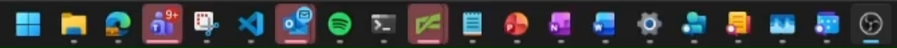


Command Prompt | Windows PowerShell | Windows PowerShell | Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\radennis>
```

Your Production Patterns Checklist

Apply these six patterns immediately — no new infrastructure required:

- ☑ Scale — Use managed ingestion + materialized views for throughput workloads
- Right-size hot cache; use per-table retention to control cost

- ☑ Decision Loop — Wire KQL availability query → Fabric Activator → Teams / webhook
- Pre-compute decision tables in update policies — keep Activator lean

- ☑ Hybrid Query — Use OneLake shortcuts + KQL union to span real-time and archive
- No data movement — query-time federation across hot, warm, cold

- ☑ Schema Agility — Ingest raw JSON to dynamic column → update policy to curated table
- New fields land automatically; zero pipeline downtime

- ☑ RLS Multitenancy — Single canonical table + metadata filter = zero data duplication
- Audit-friendly; rule changes require no schema changes

- ☑ AI Grounding — KQL as real-time context retrieval for LLM / Copilot prompts
- Eventhouse = persistent, queryable, time-aware agent memory

Sound off.
The mic is all yours.
Influence the product roadmap.

Join the Fabric User Panel

Share your feedback directly with our Fabric product group and researchers.

https://aka.ms/JoinFabricUserPanel

Join the SQL User Panel

Influence our SQL roadmap and ensure it meets your real-life needs

https://aka.ms/JoinSQLUserPanel

Get hands-on	Complete the tutorial	aka.ms/realtimetutorial	aka.ms/ontology-tutorial
	Play Kusto Detective Agency	aka.ms/realtimedetective	
Engage	Ask questions on the forum	aka.ms/realtimeforum	aka.ms/fabric-iq-forum
	Submit ideas and vote	aka.ms/realtimeideas	aka.ms/fabric-iq-ideas
	Use the docs	aka.ms/realtimedocs	aka.ms/fabric-iq-overview
Learn more	Complete the learning path	aka.ms/realtimelearningpath	
	Get certified	aka.ms/realtimeskill	
	Read the blog	aka.ms/realtimeblogs	aka.ms/fabric-iq-blogs
Stay updated	Check the release plan	aka.ms/realtimereleaseplan	aka.ms/fabric-iq-release
	Follow on LinkedIn	aka.ms/realtimelinkedin	
Learn about Fabric	Watch on YouTube	aka.ms/fabricyoutube	
	Find customer success stories	aka.ms/fabric-customer-success	

Get hands-on with Fabric RTI and Fabric IQ

100+ workshops delivered across 20+ countries. Now it’s your turn!

Gain hands-on skills: real-time routing, monitoring, alerting, and AI-powered analytics

aka.ms/nextRTIAD

Learn more about ontologies and IQ in the playground:

aka.ms/ontology-playground

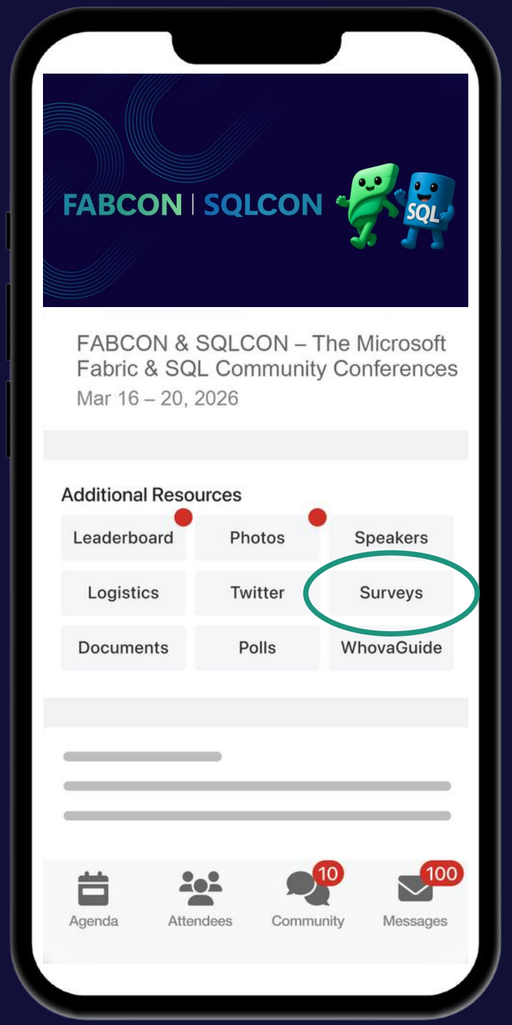
Get hands-on today!

Related sessions at FabCon

Session ID	Title	Date/Time
10727186	CORENOTE: Create a Single Source of Truth for Your Data, AI, and Actions	Wed, Mar 18, 3:05 PM–4:05 PM
1072716	Transform Your Business with Fabric Real-Time Intelligence	Wed, Mar 18, 4:25 PM–5:25 PM
1094675	Fabric IQ: Unlock Enterprise AI with a Unified Semantic Layer	Thu, Mar 19, 10:10 AM–11:10 AM
1094315	Connect Your Data with Fabric Graph: Why Relationships Matter for AI	Thu, Mar 19, 2:00 PM–3:00 PM
1081361	Master Eventhouse Patterns for Real-Time Intelligence at Scale	Thu, Mar 19, 4:15 PM–5:15 PM
1094445	Simplify Geospatial Analytics with Real-Time Maps in Fabric	Thu, Mar 19, 4:15 PM–5:15 PM
1025145	Build Trustworthy Real-Time AI Applications with Eventstream in Real-Time Intelligence	Fri, Mar 20, 10:10 AM–11:10 AM
1080563	Build Your First Digital Employee: A Guide to Operations Agents in Fabric	Thu, Mar 19, 11:30 AM–12:30 PM
1065937	Automate at Scale with Event-Driven Architectures and Business Events	Fri, Mar 20, 3:15 PM–4:15 PM

How was the session?

Complete Session Surveys in Whova for your chance to WIN PRIZES!



Get Two Fabric Certifications for FREE

Attendees of FABCON can take the Fabric Analytics Engineer or Fabric Data Engineer exam for free. Be part of the 2 fastest growing role-based certifications in Microsoft history.

Request your voucher by March 23, 2026.

https://aka.ms/fabcon/cert100