



Microsoft

Microsoft Fabric

1000s of Trucks a Day in Fabric

Hiram Fleitas

Senior Cloud Solution Architect, Microsoft Data & AI

<https://aka.ms/hiram>



Agenda

01

1000s of Trucks a Day

02

App-trimble2kusto

03

Trimble-FabricRTI

04

Trip Viewer

05

The Opportunity with
Streaming Data

06

Real-Time Intelligence in
Microsoft Fabric

07

A Complete SaaS Solution

08

A Single Place for Data in
Motion

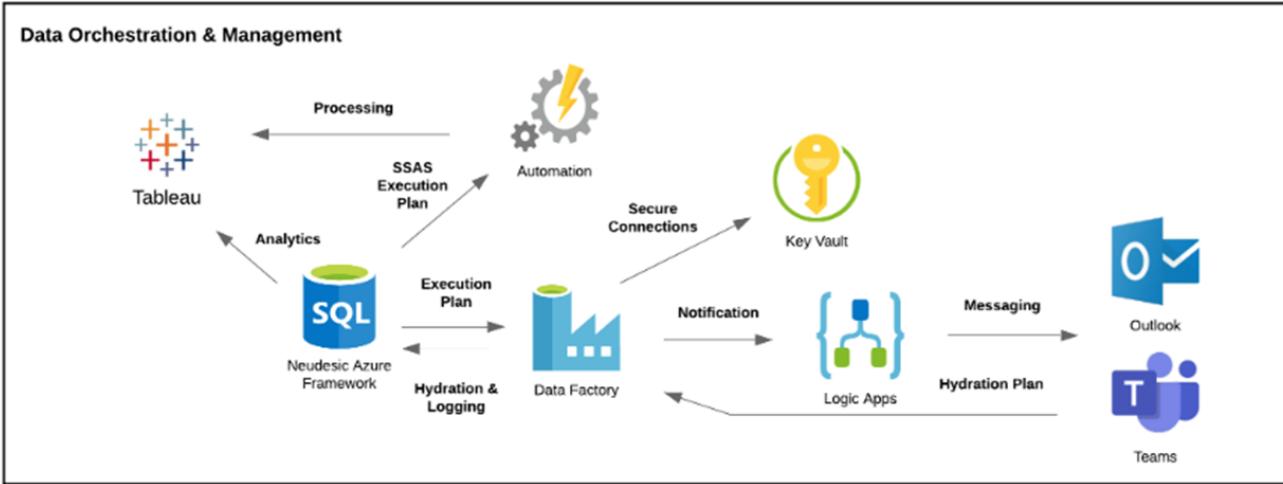
09

What We're Hearing From
Customers

10

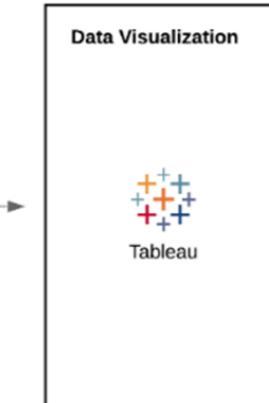
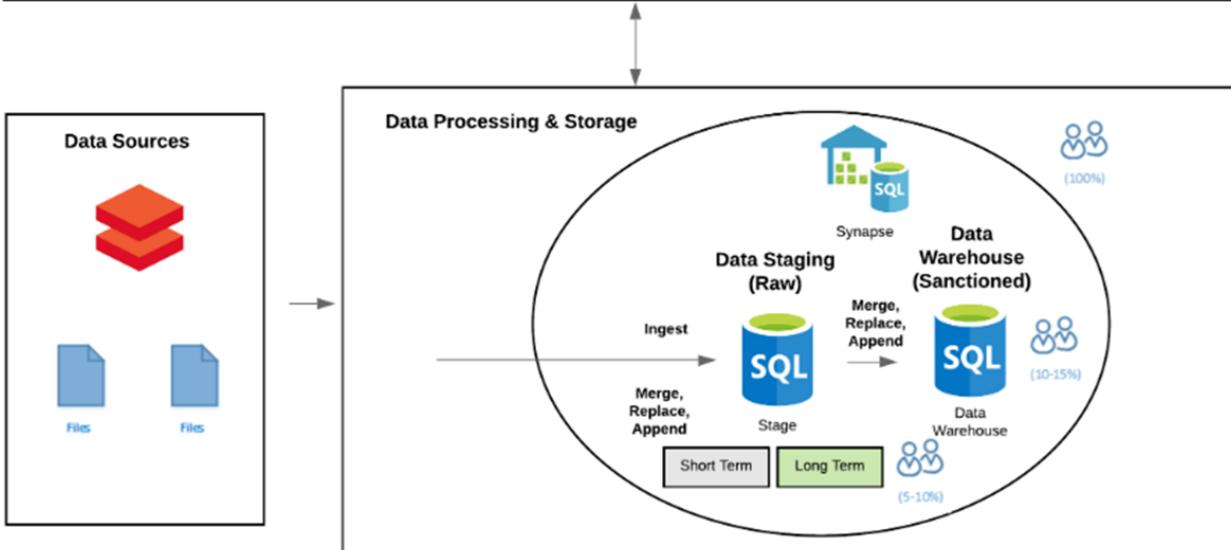
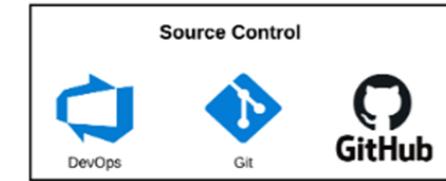
Get Started Today

Architecture Before Microsoft Fabric

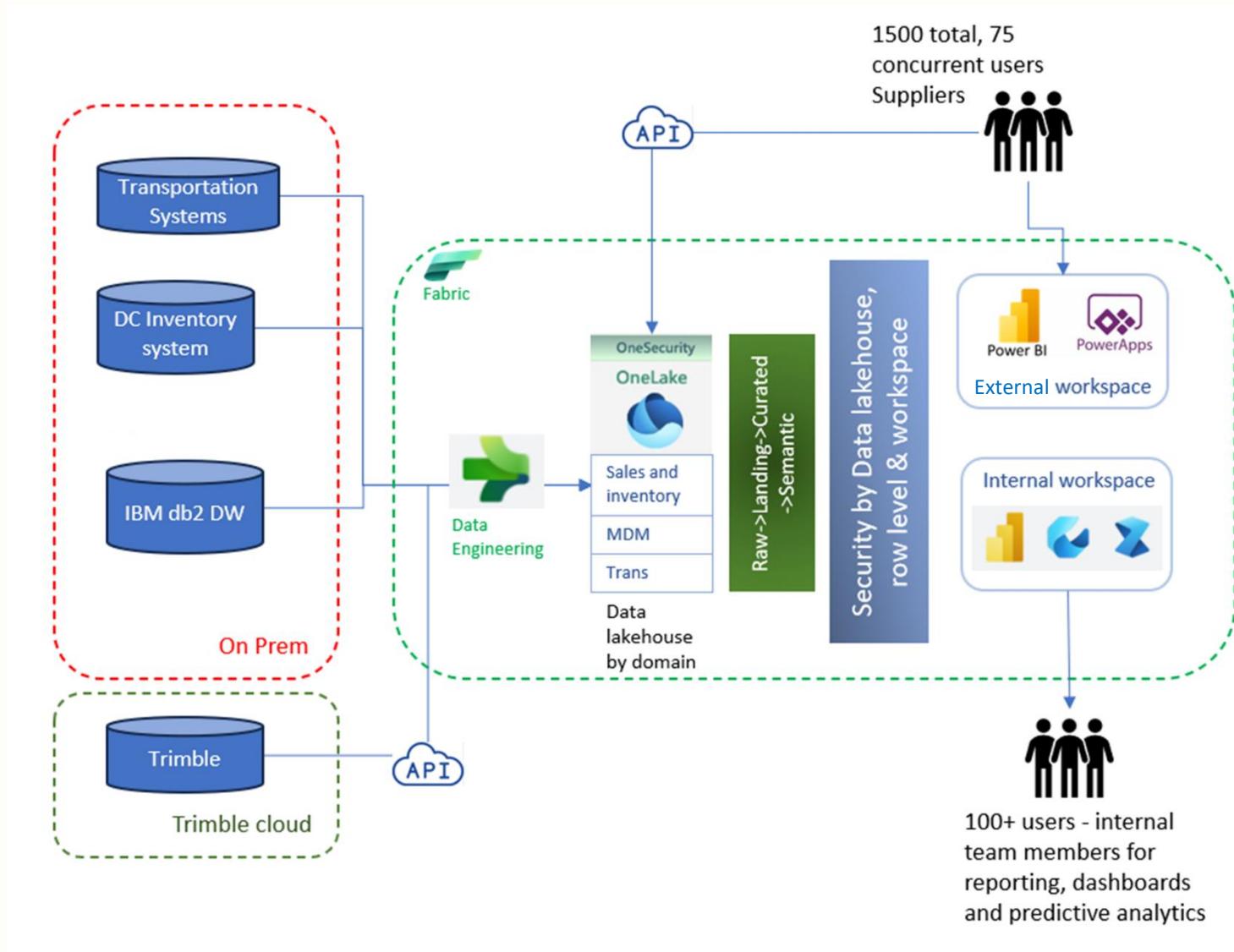


Highlights:

- Data & AI Accelerator / Framework
- Big Data Processing (2TB +)
- Structured & Unstructured Data
- Managed Data Lake
- Databricks is a great option for Data Science Teams
- Databricks Logging Directly to Framework
- Low Cost Blob Storage
- Self-Service Points (Analysis Services, Data Warehouse, Data Lake)

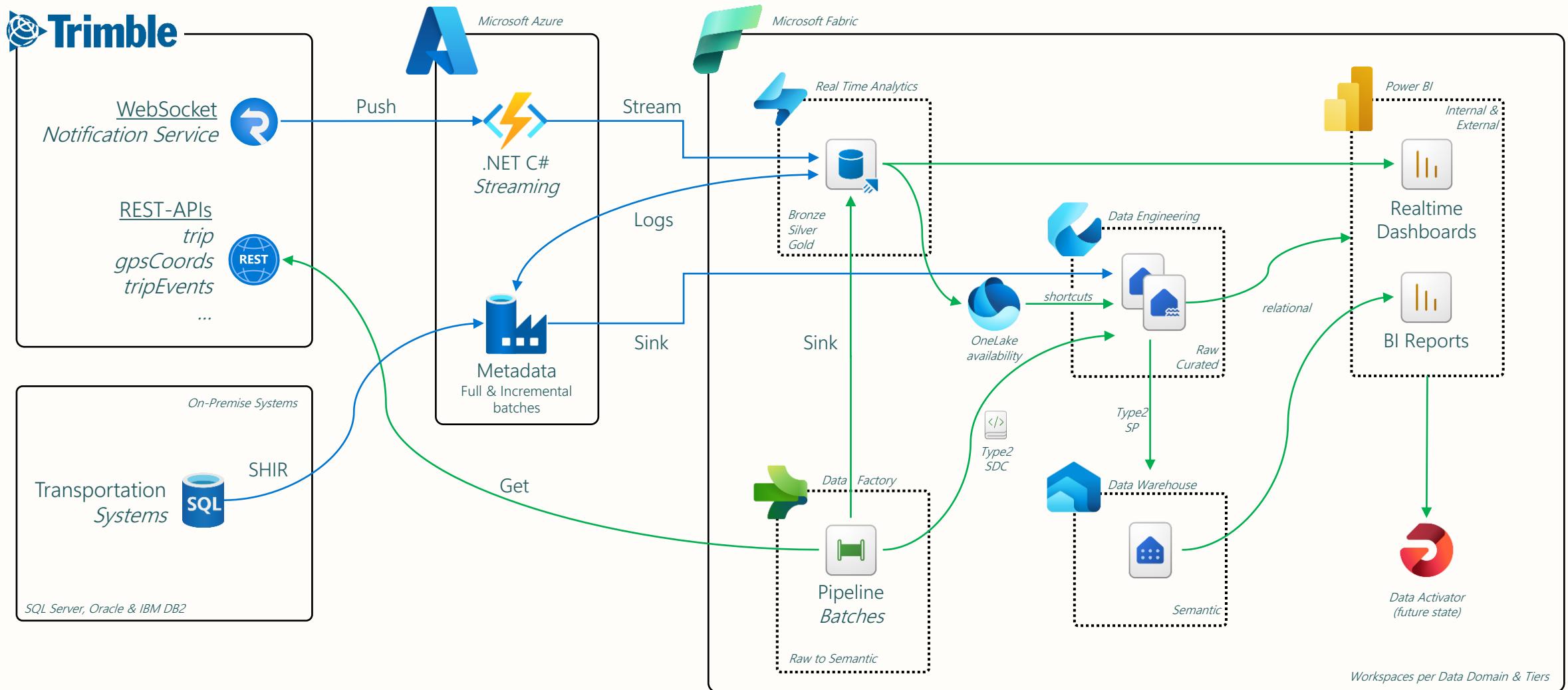


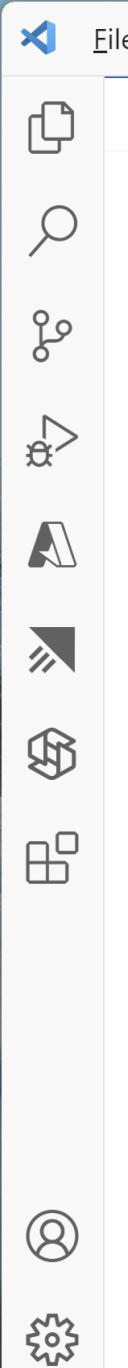
Vision with Microsoft Fabric



Solution Architecture with Microsoft Fabric

Transportation Data Domain





C# Program.cs X

notificationsvc > C# Program.cs

```
1  using Microsoft.AspNetCore.SignalR.Client;
2  using Kusto.Data;
3  using Kusto.Data.Net.Client;
4  using Kusto.Data.Common;
5  using Kusto.Ingest;
6  using System.Data;
7
8  /// <summary>
9  /// Example of using WebSocket, Trimble Notification Service, Kusto SDK, and SignalR to stream messages into Kusto.
10 /// References:
11 /// https://developer.trimblemaps.com/restful-apis/trip-management/notifications-service/
12 /// https://github.com/Azure/azure-kusto-samples-dotnet
13 /// </summary>
14 /// <remarks>
15 /// This sample assumes:
16 /// 1. A valid Trimble API key with the TM add-on.
17 /// 2. Your Kusto cluster has streaming enabled. (Streaming is enabled by default for Fabric KQL Database & ADX Free-Personal Clusters)
18 /// 3. Your Kusto database has a table with the following schema:
19 ///     .create table TrimbleNotificationRaw (message:dynamic)
20 /// </remarks>
21
22
23 // params required
24 string apiKey = "your-api-key";
25 string clusterUri = "your-kql-clusterUri";
26 string database = "your-kql-database";
27 string table = "your-kql-table";
28
29
```

The screenshot shows a Microsoft Visual Studio IDE window with the following details:

- Top Bar:** File, Edit, Selection, View, Go, ...
- Search Bar:** app-trimble2kusto
- Toolbars:** Standard toolbar icons (New, Open, Save, Print, etc.)
- Code Editor:** A C# file named "Program.cs" is open. The code is as follows:

```
29 // kusto client settings
30 var clusterKcsb = new KustoConnectionStringBuilder(clusterUri)
31 | .WithAadUserPromptAuthentication();
32 var ingestClient = KustoIngestFactory
33 | .CreateStreamingIngestClient(clusterKcsb);
34 var ingestProps = new KustoIngestionProperties(database, table) {
35     Format = DataSourceFormat.txt
36 };
37
38
39
40 // signalR connection to trimble notification service
41 var connection = new HubConnectionBuilder()
42 | .WithUrl("https://notifications.trimblemaps.com/register?apikey=" + apiKey)
43 | .WithAutomaticReconnect(new [] {TimeSpan.FromSeconds(5)})
44 | .Build();
45
46 connection.On<dynamic>("notificationMessage", (message) =>
47 {
48     try
49     {
50         Console.WriteLine($"message: {message}");
51
52         // create a stream from the message
53         MemoryStream stream = new MemoryStream();
54         StreamWriter writer = new StreamWriter( stream );
55         stream.Position = 0;
56         // write the message to the stream
57         writer.Write( message );
58     }
59 }
```

Status Bar: main ↻ 0↓ 1↑ ⌂ 0 △ 0 ⓘ 2 [Azurite Queue Service] [Azurite Blob Service] ↺ Ln 87, Col 30 Spaces: 4 UTF-8 with BOM CRLF C# ⚙ 2 Spell ⚙

The screenshot shows a Microsoft Visual Studio interface with the following details:

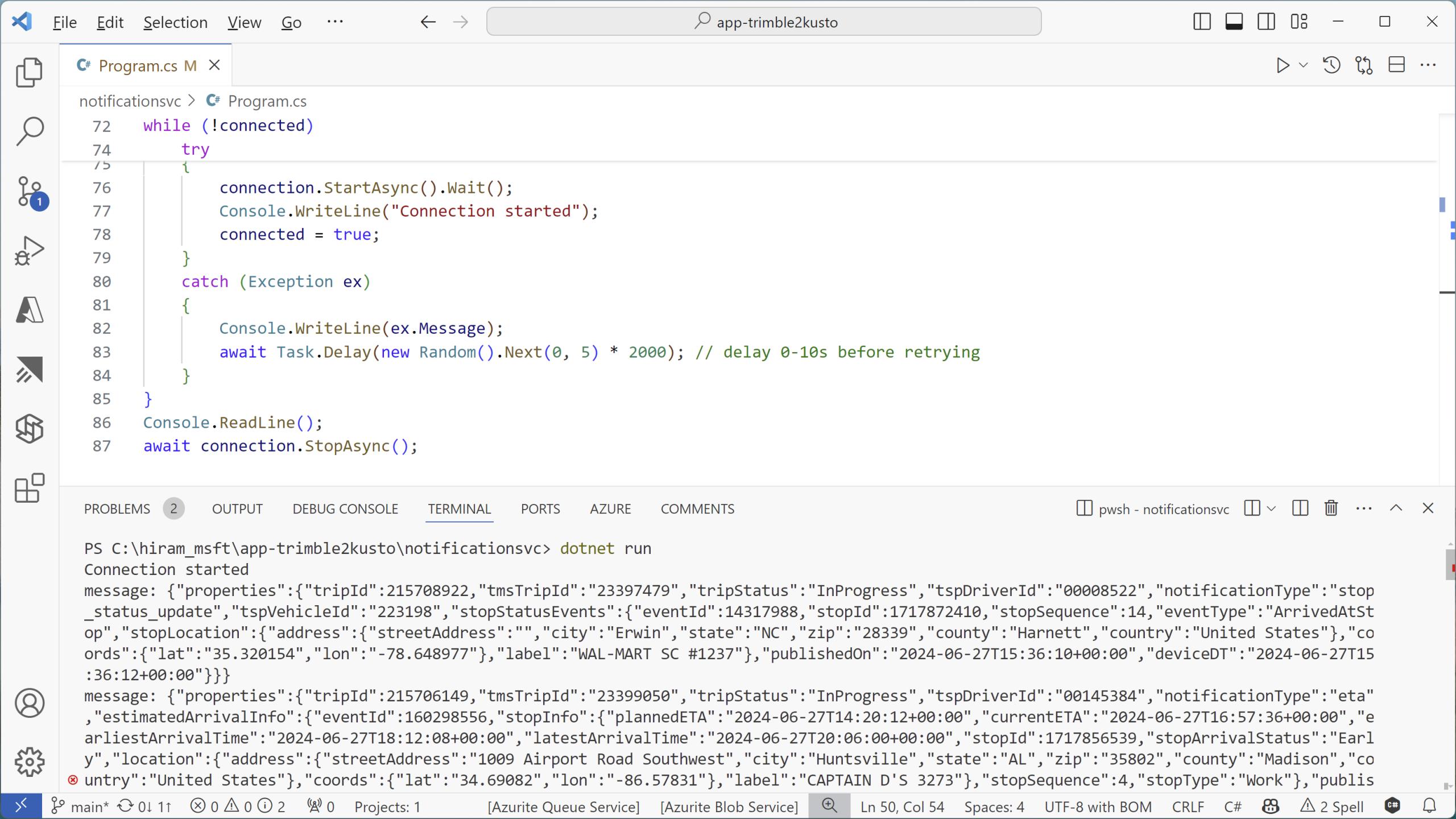
- Top Bar:** File, Edit, Selection, View, Go, ...
- Search Bar:** app-trimble2kusto
- Toolbar:** Standard icons for file operations like Open, Save, Print, etc.
- Code Editor:** The main window displays the 'Program.cs' file from a project named 'notificationsvc'. The code is written in C# and handles signalR connection to trimble notification service, message processing, and ingest to kusto.
- Solution Explorer:** Shows the project structure with 'Program.cs' selected.
- Toolbars and Icons:** On the left, there are various icons for navigation, search, and other development tasks.
- Status Bar:** Shows file path (main.cs), projects count (1), and other system information.

```
39
40 // signalR connection to trimble notification service
41 var connection = new HubConnectionBuilder()
42     .WithUrl("https://notifications.trimblemaps.com/register?apikey=" + apiKey)
43     .WithAutomaticReconnect(new [] {TimeSpan.FromSeconds(5)})
44     .Build();
45
46 connection.On<dynamic>"notificationMessage", (message) =>
47 {
48     try
49     {
50         Console.WriteLine($"message: {message}");
51
52         // create a stream from the message
53         MemoryStream stream = new MemoryStream();
54         StreamWriter writer = new StreamWriter( stream );
55         stream.Position = 0;
56         // write the message to the stream
57         writer.Write( message );
58         writer.Flush();
59         stream.Position = 0;
60
61         // ingest the stream to kusto
62         _= ingestClient.IngestFromStreamAsync(stream, ingestProps).Result;
63     }
64     catch (Exception ex)
65     {
66         Console.WriteLine($"Exception: {ex}");
67     }
}
```



C# Program.cs X

```
notificationsvc > C# Program.cs
46     connection.On<dynamic>("notificationMessage", (message) =>
47         try
48             {
49                 }
50             catch (Exception ex)
51             {
52                 Console.WriteLine($"Exception: {ex}");
53             }
54         });
55
56 // retry connecting to trimble notification service
57 bool connected = false;
58 while (!connected)
59 {
60     try
61     {
62         connection.StartAsync().Wait();
63         Console.WriteLine("Connection started");
64         connected = true;
65     }
66     catch (Exception ex)
67     {
68         Console.WriteLine(ex.Message);
69         await Task.Delay(new Random().Next(0, 5) * 2000); // delay 0-10s before retrying
70     }
71 }
72
73 Console.ReadLine();
74 await connection.StopAsync();
```



[Run](#) [Preview](#) [Recall](#)
[Copy query](#)
[Pin to dashboard](#)
[KQL Tools](#)
[Export to CSV](#)
[Build PowerBI report](#)

```

1 //base query _trimble
2 TrimbleTrips
3 | where retrievedOn =='2024-02-27T20:37:54Z'
4 | where tripId == 200631167 and etaEvents <> ''
5 mv-expand etaEvents
6 extend tostring(etaEvents.eventId), etaEvents.stopInfo, todatetime(etaEvents.publishedOn), etaEvents.gpsLocationInfo
7 extend todatetime(etaEvents_stopInfo.currentETA), etaEvents_stopInfo.location
8 extend etaEvents_stopInfo location.Coords
9 | extend stop_lat=toreal(etaEvents_stopInfo_location_Coords.Lat), stop_lon=toreal(etaEvents_stopInfo_location_Coords.Lon)
10 extend tostring(etaEvents_stopInfo_location.PlaceName), toint(etaEvents_stopInfo.stopSequence)
11 project-away tmsTripId, outOfCorridorEvents, etaEvents, outOfRouteEvent, weatherEvents, weatherV2Events
12 project-away stopStatusEvents, unplannedStopEvents, trafficIncidentEvents, approachingEvents, excessiveDwellEvents
13 project-away etaEvents_stopInfo, etaEvents_stopInfo_location, etaEvents_stopInfo_location_Coords
14 | extend device_lat=toreal(etaEvents_gpsLocationInfo.lat), device_lon=toreal(etaEvents_gpsLocationInfo.lon), todatetime(etaEvents_gpsLocationInfo.deviceDT)
15 project-away etaEvents_gpsLocationInfo
16 | sort by etaEvents_eventId asc
17 | extend tostring(tripId)
18

```

Table 1

Stats



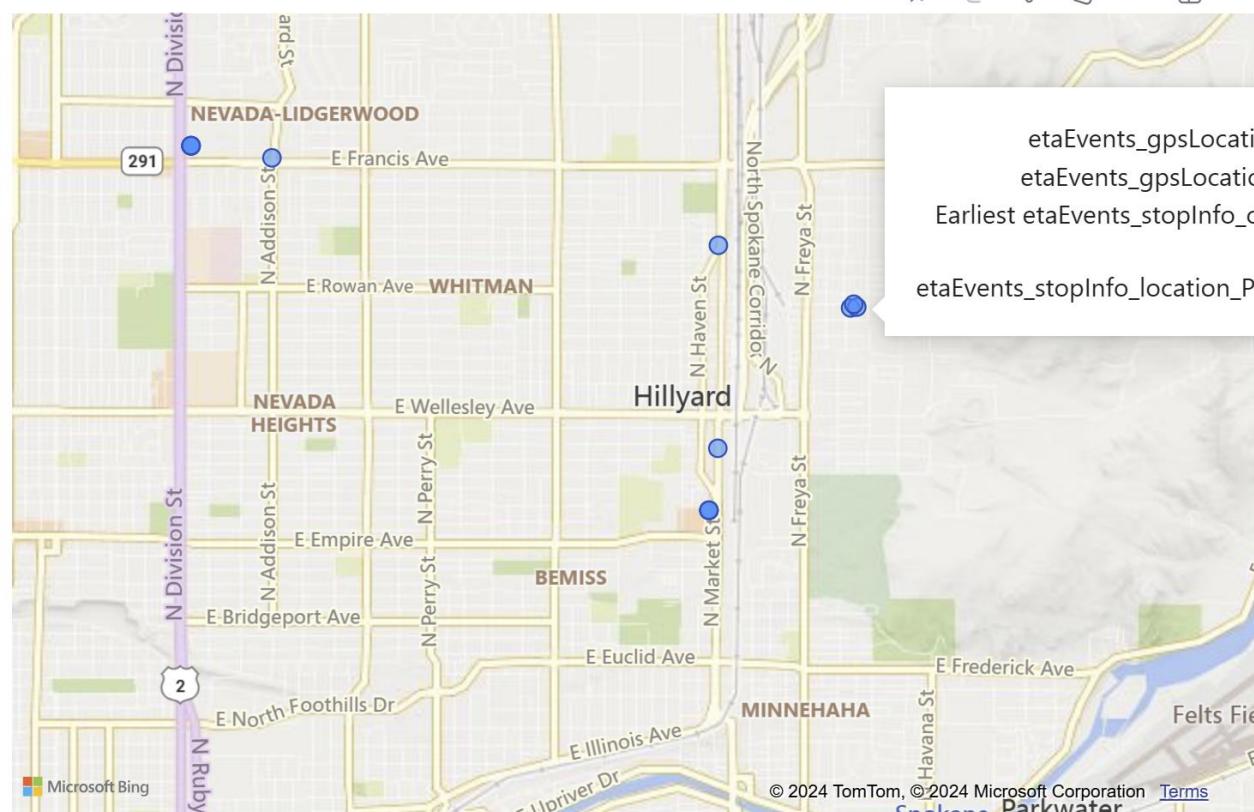
Search Done (0.154 s) 75 records



| tripId | retrievedOn | etaEvents_eventId | etaEvents_publishedOn | etaEvents_stopInfo_currentETA | stop_lat | stop_lon | etaEvents_stopInfo_location_PlaceName | etaEv |
|-------------|--------------------------|-------------------|--------------------------|-------------------------------|-----------|-------------|---------------------------------------|-------|
| > 200631167 | 2024-02-27 20:37:54.0000 | 27838209 | 2024-02-26 20:44:46.0000 | 2024-02-26 22:53:03.0000 | 47.6748 | -117.2814 | DENNY'S #6954 | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27838210 | 2024-02-26 20:44:46.0000 | 2024-02-26 21:19:59.0000 | 47.742778 | -117.408333 | N NEWPORT HWY & E HOERNER | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27838211 | 2024-02-26 20:44:46.0000 | 2024-02-27 05:38:13.0000 | 46.891389 | -114.039167 | MISSOULA FSU 4795 | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27838212 | 2024-02-26 20:44:46.0000 | 2024-02-27 09:24:34.0000 | 48.369785 | -114.23928 | Town Pump | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27838213 | 2024-02-26 20:44:46.0000 | 2024-02-27 08:24:41.0000 | 48.2385 | -114.3317 | KALISPELL FSU #3434 | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27838214 | 2024-02-26 20:44:46.0000 | 2024-02-27 13:19:40.0000 | 47.716388 | -116.894444 | JIB 6069 | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27839520 | 2024-02-26 20:45:47.0000 | 2024-02-26 20:54:27.0000 | 47.7068 | -117.3538 | D 657 SPOKANE | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27839526 | 2024-02-26 20:45:48.0000 | 2024-02-26 21:21:00.0000 | 47.742778 | -117.408333 | N NEWPORT HWY & E HOERNER | |
| > 200631167 | 2024-02-27 20:37:54.0000 | 27839527 | 2024-02-26 20:45:48.0000 | 2024-02-27 05:39:14.0000 | 46.891389 | -114.039167 | MISSOULA FSU 4795 | |

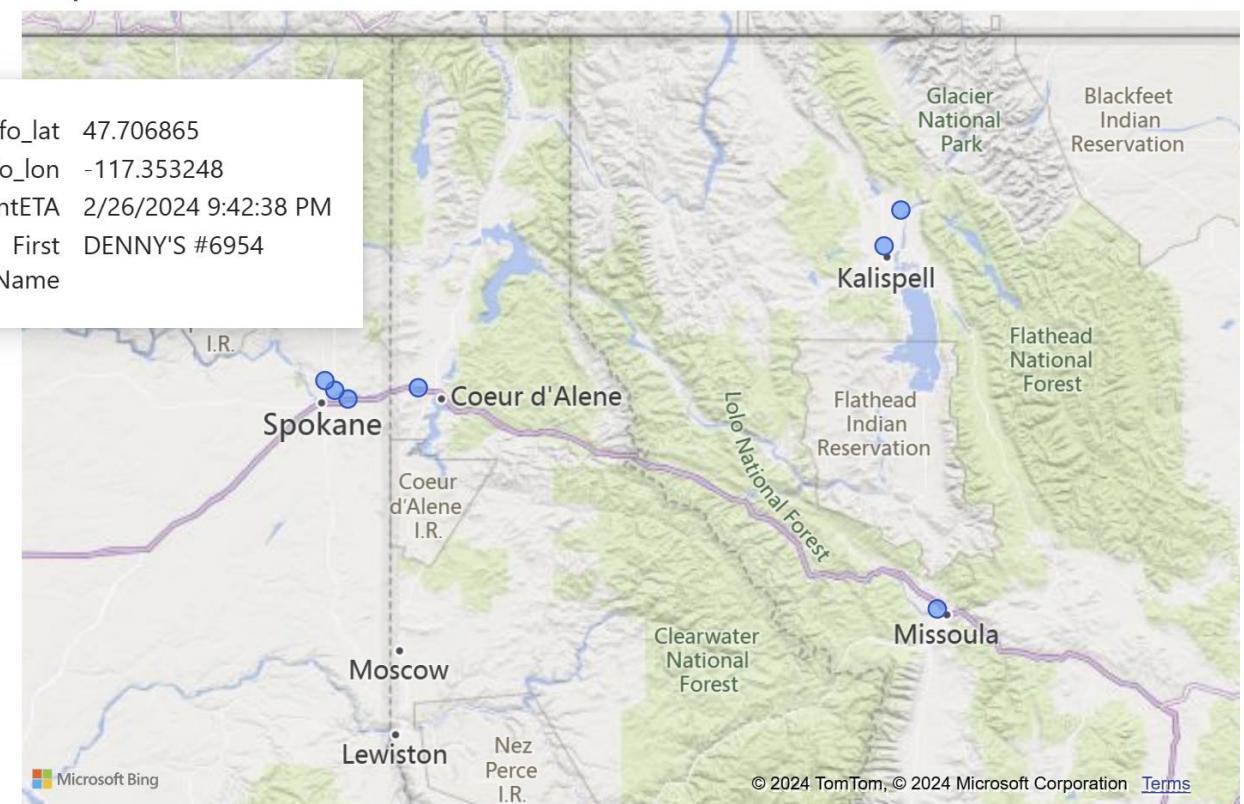
| tripId | etaEvents_eventId | etaEvents_gpsLocationInfo_deviceDT | etaEvents_gpsLocationInfo_lat | etaEvents_gpsLocationInfo_lon | etaEvents_publishedOn | etaEvents_stopInfo_currentETA | etaEvents_stopInfo_l |
|-----------|-------------------|------------------------------------|-------------------------------|-------------------------------|-----------------------|-------------------------------|----------------------|
| 200631167 | 27838209 | 2/26/2024 8:44:45 PM | | 47.72 | | -117.41 | 2/26/2024 8:44:46 PM |
| 200631167 | 27838210 | 2/26/2024 8:44:45 PM | | 47.72 | | -117.41 | 2/26/2024 8:44:46 PM |
| 200631167 | 27838211 | 2/26/2024 8:44:45 PM | | 47.72 | | -117.41 | 2/26/2024 8:44:46 PM |
| 200631167 | 27838212 | 2/26/2024 8:44:45 PM | | 47.72 | | -117.41 | 2/26/2024 8:44:46 PM |
| 200631167 | 27838213 | 2/26/2024 8:44:45 PM | | 47.72 | | -117.41 | 2/26/2024 8:44:46 PM |
| 200631167 | 27838214 | 2/26/2024 8:44:45 PM | | 47.72 | | -117.41 | 2/26/2024 8:44:46 PM |
| 200631167 | 27839520 | 2/26/2024 8:45:46 PM | | 47.72 | | -117.41 | 2/26/2024 8:45:47 PM |
| 200631167 | 27839526 | 2/26/2024 8:45:46 PM | | 47.72 | | -117.41 | 2/26/2024 8:45:48 PM |
| 200631167 | 27839527 | 2/26/2024 8:45:46 PM | | 47.72 | | -117.41 | 2/26/2024 8:45:48 PM |
| | | | | | | | 2/27/2024 5:39:14 AM |

GPS



Stops

etaEvents_gpsLocationInfo_lat 47.706865
 etaEvents_gpsLocationInfo_lon -117.353248
 Earliest etaEvents_stopInfo_currentETA 2/26/2024 9:42:38 PM
 First DENNY'S #6954
 etaEvents_stopInfo_location_PlaceName



Database

```
21 // serilog
22 // https://learn.microsoft.com/en-us/azure/data-explorer/serilog-sink?tabs=windows
23 // https://techcommunity.microsoft.com/t5/azure-data-explorer-blog/getting-started-with-serilog-and-azure-data-
24 // https://github.com/Azure/serilog-sinks-azuredataexplorer
25 // https://aka.ms/adx.connectors
26 // .create table Serilogs (Timestamp: datetime, Level: string, Message: string, Exception: string, Properties:
27
28
29 .create table TrimbleNotificationRaw (message:dynamic)
30
31 TrimbleNotificationRaw
32 | take 10
33
34 TrimbleNotificationRaw
35 | count
36
37 .clear table TrimbleNotificationRaw data
```

JPath: /message Right ...

```
1 < {  
2   "properties": {  
3     "tripId": 215708922,  
4     "tmsTripId": "23397479",  
5     "tripStatus": "InProgress",  
6     "tspDriverId": "00008522",  
7     "notificationType":  
8       "stop_status_update",  
9     "tspVehicleId": "223198",  
10    "stopStatusEvents": {  
11      "eventId": 14317988,  
12      "stopId": 1717872410,  
13      "stopSequence": 14,  
14      "eventType":  
15        "ArrivedAtStop",  
16      "stopLocation": {  
17        "address": {  
18          "streetAddress": "",  
19          "city": "Erwin",  
20          "state": "NC",  
21          "zip": "28339",  
22          "county": "Harnett",  
23          "country": "United  
24          States"  
25        },  
26        "coords": {  
27          "lat": "35.320154",  
28          "lon": "-78.648977"  
29        },  
30        "label": "WAL-MART SC  
31          #1237"  
32      },  
33      "publishedOn":  
34    }  
35  }  
36 }
```

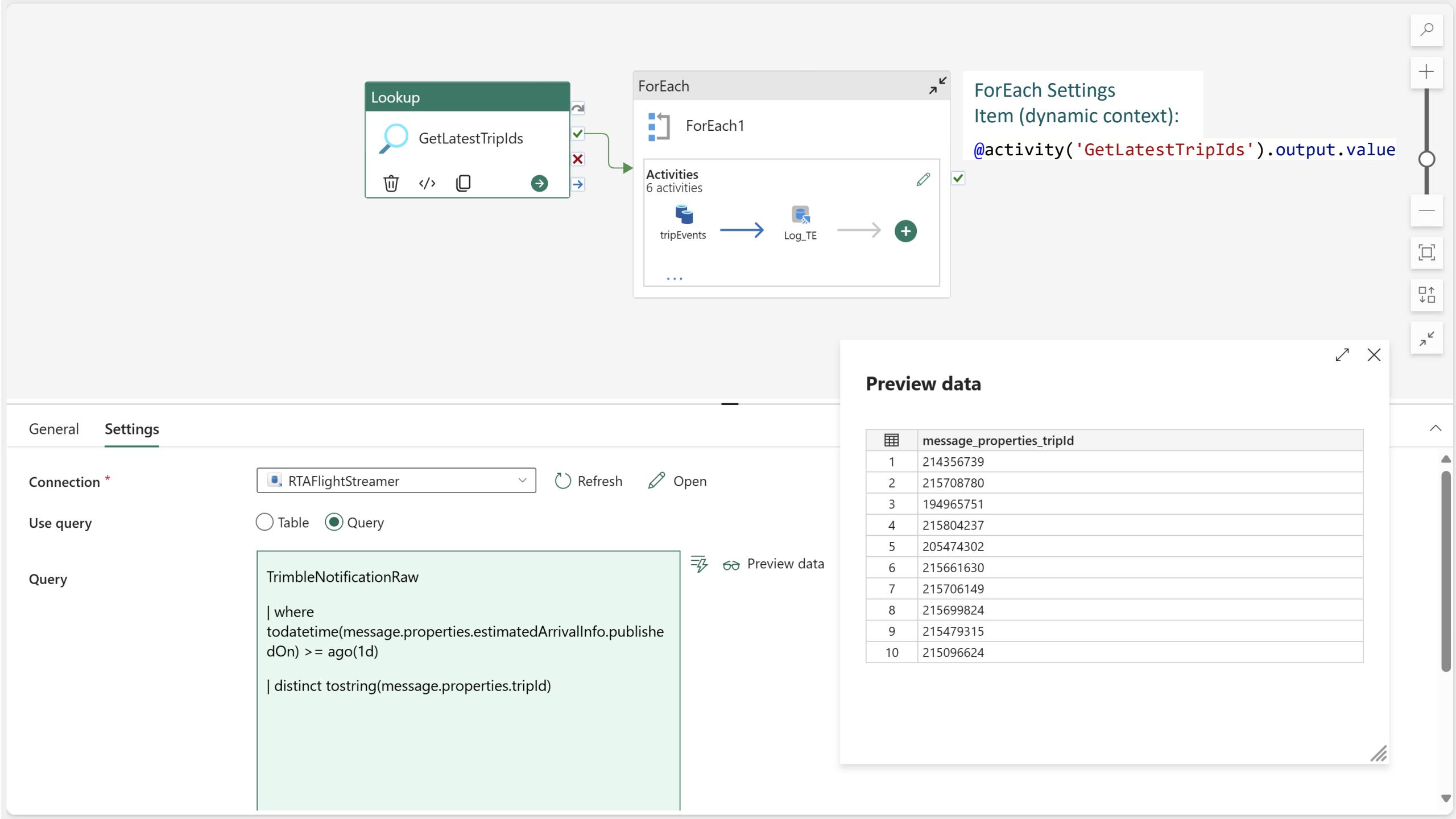
» Database

```
32 | take 10
33
34 TrimbleNotificationRaw
35 | count
36
37 .clear table TrimbleNotificationRaw data
38
39 .delete table TrimbleNotificationRaw records <| TrimbleNotificationRaw | where message == ''
40
41 TrimbleNotificationRaw
42 | take 1000
43
44 TrimbleNotificationRaw
45 | summarize max(ingestion_time())
46
47 let i=TrimbleNotificationRaw
48 | where ingestion_time() > ago(2s)
49 | top 5 by ingestion_time() desc
```

| Table 1 | Stats | Search | Done (0.223 s) | 1,000 records | Columns |
|---------|--|--------|----------------|---------------|---------|
| message | | | | | |
| | {"properties": {"tripId": 200437695, "tmsTripId": "18909748", "tripStatus": "InProgress", "tspDriverId": "00050302", "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}}, {"properties": {"tripId": 200437695, "tmsTripId": "18909748", "tripStatus": "InProgress", "tspDriverId": "00050302", "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}}, {"properties": {"tripId": 201574624, "tmsTripId": "19604949", "tripStatus": "InProgress", "tspDriverId": "00134428", "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}}, {"properties": {"tripId": 200437695, "tmsTripId": "18909748", "tripStatus": "InProgress", "tspDriverId": "00050302", "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}}, {"properties": {"tripId": "194149646", "tmsTripId": "15847985", "notificationType": "weather_notification_v2", "publishedOn": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, {"properties": {"tripId": 200437695, "tmsTripId": "18909748", "tripStatus": "InProgress", "tspDriverId": "00050302", "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}}, {"properties": {"tripId": 185133391, "tmsTripId": "510996083", "tripStatus": "InProgress", "tspDriverId": null, "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}}, {"properties": {"tripId": 200437695, "tmsTripId": "18909748", "tripStatus": "InProgress", "tspDriverId": "00050302", "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}}, {"properties": {"tripId": 201541159, "tmsTripId": "19553273", "tripStatus": "InProgress", "tspDriverId": "00015634", "notificationType": "eta", "estimatedArrivalInfo": {"eventId": 123, "eta": "2024-03-15T13:46:04.000000Z", "ownerId": 428}, "publishedOn": "2024-03-15T13:46:04.000000Z"}]} | | | | |

JPath: /message/properties/tripURL

```
1 1 {  
2 2 "properties": {  
3 3     "tripId": "194149646",  
4 4     "tmsTripId": "15847985",  
5 5     "notificationType":  
6 6         "weather_notification_v2"  
7 7         ",  
8 8     "publishedOn":  
9 9         "2024-03-15T13:46:04.  
0000000Z",  
10 10    "ownerId": 500572002,  
11 11    "accountId": 100003498,  
12 12    "tripURL": "https://tripmanagement.alk.com/trip-viewer/token/a547612lp2vv48k7mbzm5senhsn9s9momcy21shxa/details/194149646?region=NA",  
13 13    "distanceUOM": "Miles",  
14 14    "precipitationUOM": "Inches",  
15 15    "temperatureUOM": "Fahrenheit",  
16 16    "weatherNotificationId":  
17 17        "dc260697-d085-464f-87c3  
-c839e2fcfa160",  
18 18    "slowdownSummary": 0.25,  
19 19    "weatherSummary": [],  
20 20    "riskSummary": {  
21 21        "maxRoadRisk": "Moderate",  
22 22        "maxRoadIndex": 5.1,  
23 23        "maxCrashRisk": "Moderate",  
24 24        "maxCrashIndex": "5",  
25 25        "maxDisruptionRisk": "Low"
```



Main canvas > ForEach1

```

graph TD
    tripEvents1[tripEvents] --> LogTE1[KQL Log_TE]
    tripEvents2[tripEvents] --> LogGC1[KQL Log_GC]
    tripEvents3[tripEvents] --> LogT1[KQL Log_T]
    gpsCoords1[gpsCoords] --> LogGC1
    trip1[trip] --> LogT1

```

General Source Destination Mapping Settings

Source

Connection *

Relative URL

Advanced

Request method

Request timeout

Request interval (ms)

Additional headers

| | Value |
|--------------------------|----------------------|
| <input type="checkbox"/> | Authorization |
| <input type="checkbox"/> | @variables('apiKey') |

https://developer.trimblemaps.com/restful-apis/trip-management/api-documentation/get-trip/

Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```
@concat('/api/trip/','item().message_properties_tripId')
```

Clear contents

ForEach iterator

Activity outputs

Parameters

System variables

Trigger param

Search

ForEach1
Current item

GET /trip/{tripId} #

Resource URL #

https://tripmanagement.alk.com/api/trip/{tripId}

OK

Cancel

Home Activities Run View

Validate Cancel Schedule Trigger (preview) View run history

Main canvas > ForEach1

The screenshot shows a pipeline named 'ForEach1' on the main canvas. It consists of three parallel 'Copy data' activities:

- The first activity copies 'tripEvents' from a source to 'Log_TE' using KQL.
- The second activity copies 'gpsCoords' from a source to 'Log_GC' using KQL.
- The third activity copies 'trip' from a source to 'Log_T' using KQL.

Below the canvas, the 'Settings' tab is selected in the 'General' section. The 'Activity outputs' pane displays the following KQL code:

```
'Succeeded'  
,  
concat(  
    '.set-or-append PipelineLogs with(folder="Logs") <|  
    print tripId="",  
    string(item().message_properties_tripId),  
    "", ErrorMessage=''',  
    '''', FullOutput=todynamic(''',  
    string(activity('trip').output),  
    ''')',  
    ' , FullActivity=todynamic(''',  
    string(activity('trip')),  
    ''')'  
,  
concat(  
    '.set-or-append PipelineLogs with(folder="Logs") <|  
    print tripId="",  
    string(item().message_properties_tripId),  
    "", ErrorMessage=''',  
    activity('trip').output.errors[0].Message,  
    '''', FullOutput=todynamic(''',  
    string(activity('trip').output), '''),  
    ' , FullActivity=todynamic(''',  
    string(activity('trip')),  
    ''')'  
)  
)
```

A 'View in expression builder' link is located at the bottom of the settings pane.

Command timeout: 00:01:00

Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```
@if (  
    equals(  
        activity('trip').status,  
        'Succeeded'  
,  
    concat(  
        '.set-or-append PipelineLogs with(folder="Logs") <|  
        print tripId="",  
        string(item().message_properties_tripId),  
        "", ErrorMessage=''',  
        '''', FullOutput=todynamic(''',  
        string(activity('trip').output),  
        ''')',  
        ' , FullActivity=todynamic(''',  
        string(activity('trip')),  
        ''')'  
,  
    concat(  
        '.set-or-append PipelineLogs with(folder="Logs") <|  
        print tripId="",  
        string(item().message_properties_tripId),  
        "", ErrorMessage=''',  
        activity('trip').output.errors[0].Message,  
        '''', FullOutput=todynamic(''',  
        string(activity('trip').output), '''),  
        ' , FullActivity=todynamic(''',  
        string(activity('trip')),  
        ''')'  
)  
)  
)
```

Clear contents

ForEach iterator Activity outputs Parameters System variables Trigger para

OK Cancel

```

80
81 TrimbleTrip
82 | top 10 by actualStartDate desc
83 | project-reorder alkTripId, url, *
84 | project-away accountName
85

```

Table 1

Stats

Search

Done (0.146 s)

10 records



alkTripId url

| | |
|-------------|---|
| > 216066226 | https://tripmanagement.alk.com/trip-viewer/token/12ny15cynze8phvtu85qlb9a9piseahhpto5badq74/details/216066226?region=NA |
| > 216066224 | https://tripmanagement.alk.com/trip-viewer/token/hm85m1riq2sqdizhd9v13x7mqypxuc4gmc0axdo158/details/216066224?region=NA |
| > 216066241 | https://tripmanagement.alk.com/trip-viewer/token/iz8c1tyfmcqoqhbz58t850cxu5xdks0rkx527wqot/details/216066241?region=NA |
| > 216066218 | https://tripmanagement.alk.com/trip-viewer/token/22ub933v6ufxr1s8ua040u8p0rbn357yfhhatqh6u/details/216066218?region=NA |
| > 216066171 | https://tripmanagement.alk.com/trip-viewer/token/g1tgar44n279j4z5ojkx8tkushz8x9elp1t2m7oevx/details/216066171?region=NA |
| > 216066209 | https://tripmanagement.alk.com/trip-viewer/token/p4yvdl3j1d0xo1hxtoe7pt68cjzivkxbfvlsclz7/details/216066209?region=NA |
| > 216066170 | https://tripmanagement.alk.com/trip-viewer/token/lr392dpg98uys773x5tp0zgdjdy2vj5rs3hy1vkhb6/details/216066170?region=NA |
| > 216057218 | https://tripmanagement.alk.com/trip-viewer/token/5d2o7p82p7hkrq4yo14zzroj07szmtetbdn52v61cm/details/216057218?region=NA |
| > 216047291 | https://tripmanagement.alk.com/trip-viewer/token/5di3v7vikpodlcc55e2jkqrfrm15nd4v15qg0xz15/details/216047291?region=NA |
| > 216052193 | https://tripmanagement.alk.com/trip-viewer/token/t91afasxw4njwlwu60ci00dl5pm1eth94dt7cm5iyd/details/216052193?region=NA |

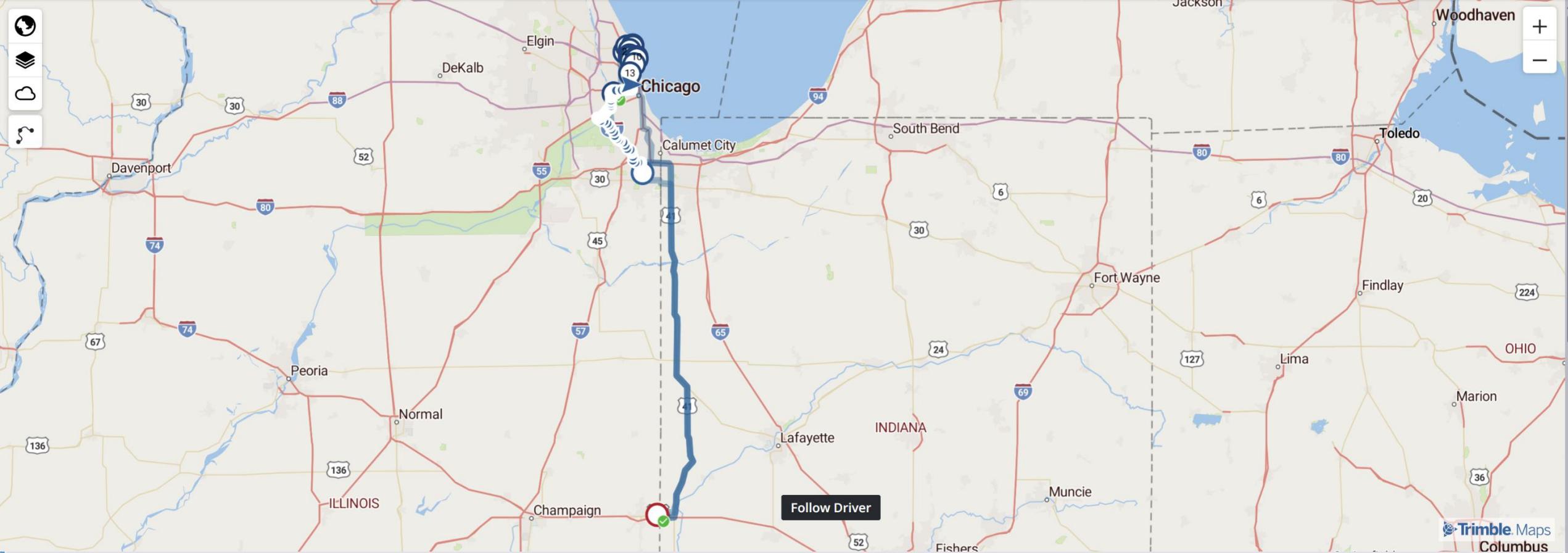
JPath: /alkTripId Right ...

```

1 "alkTripId": 216066226,
2 "url": https://tripmanagement.alk.com/
         trip-viewer/token/
         12ny15cynze8phvtu85qlb9a9piseahh
         to5badq74/details/216066226?
         region=NA,
3 "actualStartDate": 2024-06-27T21:20:23Z,
4 "tripStatus": InProgress,
5 "modifiedOn": 2024-06-27T21:34:18Z,
6 "tripDistance": 509.834,
7 "tripDriveDuration": 564.384,
8 "tripDuration": 993.017,
9 "hoursOfServiceRemaining": {
10     "enabled": true,
11     "remainingDriveTimeUntilBreak": 23040,
12     "remainingDriveTime": 14340,
13     "remainingOnDutyTime": 600,
14     "hoSRuleType": "USFed607LH",
15     "remainingCycleDutyTime": 186420,
16     "cycleTimeGains": [
17         0,
18         0,
19         0,
20         37380,
21         0,
22         26760
23     ],
24     "lastCycleResetDateTime": null,
25     "last24HourBreakDateTime": null,
26     "remainingCanadaOnDutyTime": 600,

```

Trip Viewer



Trimble Maps
Columbus

InProgress

Trip Name: TMW Trip:23436987
Source TripID: (23436987)

VehicleID: 220632
DriverID: 00144239

350 MIDWEST

40.1319, -87.5468

350 MIDWEST

40.1319, -87.5468



ETD 06/27/24 16:19:43 CDT

ETA 06/28/24 05:42:47 CDT

Actual Departure 06/27/24 16:20:08 CDT

Current Arrival 06/28/24 06:56:23 CDT

Next Stop: 7-11 30107 CORP ETA 06/27/24 19:31:21 CDT Remaining 38 mins

Trip Stats: Trip Ends : 06/28/24 06:56:23 CDT

Time To Dest. 12 hours 3 mins

Total Duration: 15 hours 24 mins

Total Distance: 347.487 mi

Projected Trip Cost: \$399.33

350 MIDWEST

40.1319, -87.5468

350 MIDWEST

40.1319, -87.5468



ETD 06/27/24 16:19:43 CDT

ETA 06/28/24 05:42:47 CDT

Actual Departure 06/27/24 16:20:08 CDT

Current Arrival 06/28/24 06:56:23 CDT

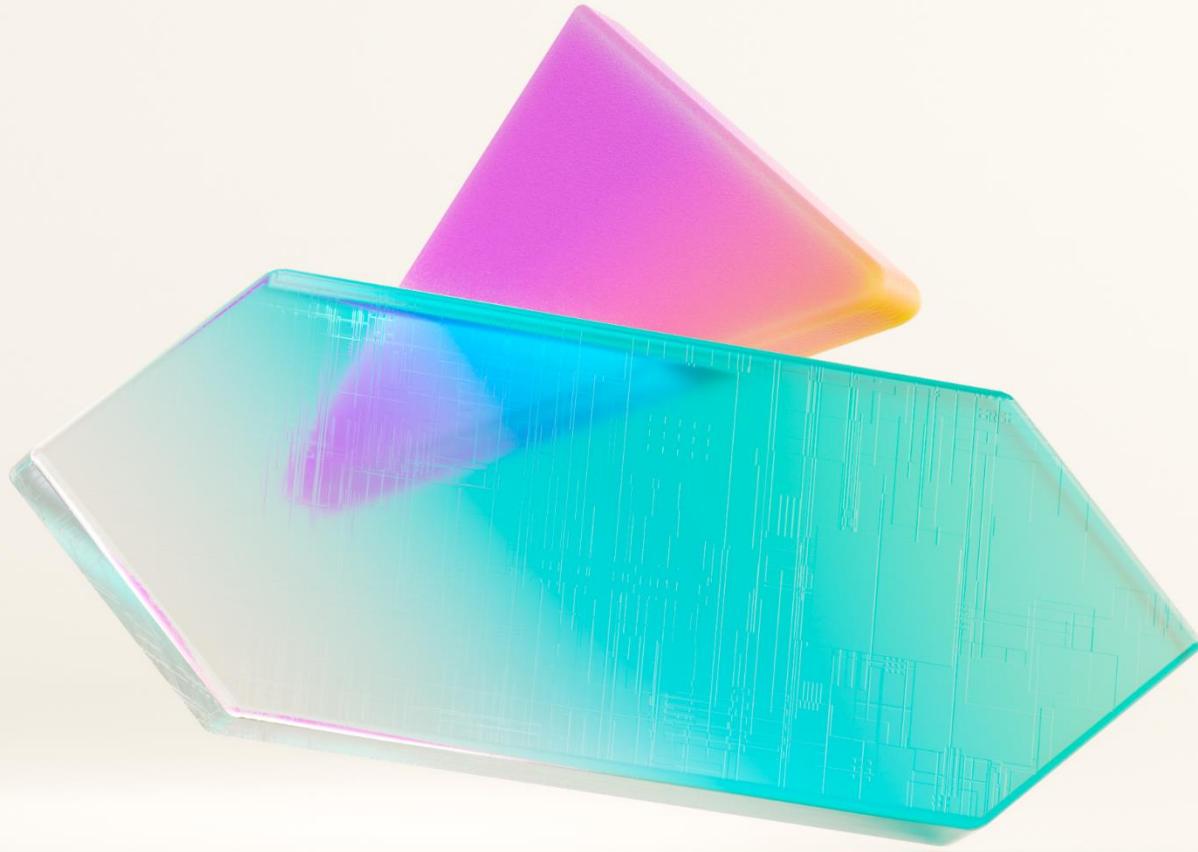
Next Stop: 7-11 30107 CORP ETA 06/27/24 19:31:21 CDT Remaining 38 mins

Trip Stats: Trip Ends : 06/28/24 06:56:23 CDT Time To Dest. 12 hours 3 mins Total Duration: 15 hours 24 mins Total Distance: 347.487 mi Projected Trip Cost: \$399.33

Events

ETA events 42 Stop Events: 2

| | Address | Status | Arrival Time | Arrival Time Window | Dept. Time | Stop Duration |
|---|--|---|--|--|--|-------------------------------------|
| 0 | Planned Start Location, Glenwood, IL, 60425 | -- | -- | -- | 06/27/24 16:20:08 CDT | -- |
| 1 | 350 MIDWEST 40.1319, -87.5468 | Completed ! Late | Est: 06/27/24 16:30:34 CDT Act: 06/27/24 16:30:34 CDT | 06/27/24 13:00:01 CDT to 06/27/24 15:15:01 CDT | Est: 06/27/24 16:45:34 CDT Act: 06/27/24 16:30:34 CDT | Estimated: 15 mins |
| 2 | 7-11 27599 41.8582, -87.78904 | Completed ✓ OnTime | Est: 06/27/24 19:31:53 CDT Act: 06/27/24 17:33:39 CDT | 06/27/24 16:20:00 CDT to 06/27/24 19:20:00 CDT | Est: 06/27/24 20:04:53 CDT Act: 06/27/24 17:34:40 CDT | Estimated: 33 mins Actual: 1 min |
| 3 | 7-11 30107 CORP 42.026095, -87.728748 | Open ✓ OnTime | Current ETA 06/27/24 19:31:21 CDT | 06/27/24 17:39:32 CDT to 06/27/24 20:39:32 CDT | Current ETD 06/27/24 20:04:21 CDT | Estimated: 33 mins |
| 4 | 7-11 16979 42.047855, -87.714846 | Open ✓ OnTime | Current ETA 06/27/24 20:09:16 CDT | 06/27/24 18:21:52 CDT to 06/27/24 21:21:52 CDT | Current ETD 06/27/24 20:42:16 CDT | Estimated: 33 mins |
| 5 | 7-11 33888 42.05251, -87.6827 | Open ✓ OnTime | Current ETA 06/27/24 20:47:50 CDT | 06/27/24 19:07:12 CDT to 06/27/24 22:07:12 CDT | Current ETD 06/27/24 21:20:50 CDT | Estimated: 33 mins |
| 6 | 7-11 33961 42.0472, -87.68298 | Open ✓ OnTime | Current ETA 06/27/24 21:22:59 CDT | 06/27/24 19:45:39 CDT to 06/27/24 22:45:39 CDT | Current ETD 06/27/24 21:55:59 CDT | Estimated: 33 mins |
| 7 | 7-11 27070 | Open ✓ | Current ETA | 06/27/24 20:29:11 CDT to | Current ETD | Estimated: 33 mins |



The opportunity
with streaming data



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

Leveraging real-time intelligence can help companies accelerate speed and precision of business



Operational efficiency

Streamline processes and **make data-driven decisions** with accurate, up-to-date information



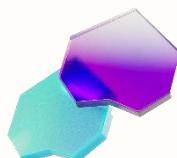
End-to-end visibility

Gain a **holistic understanding of business health** and discover actionable insights for timely action



Competitive advantage

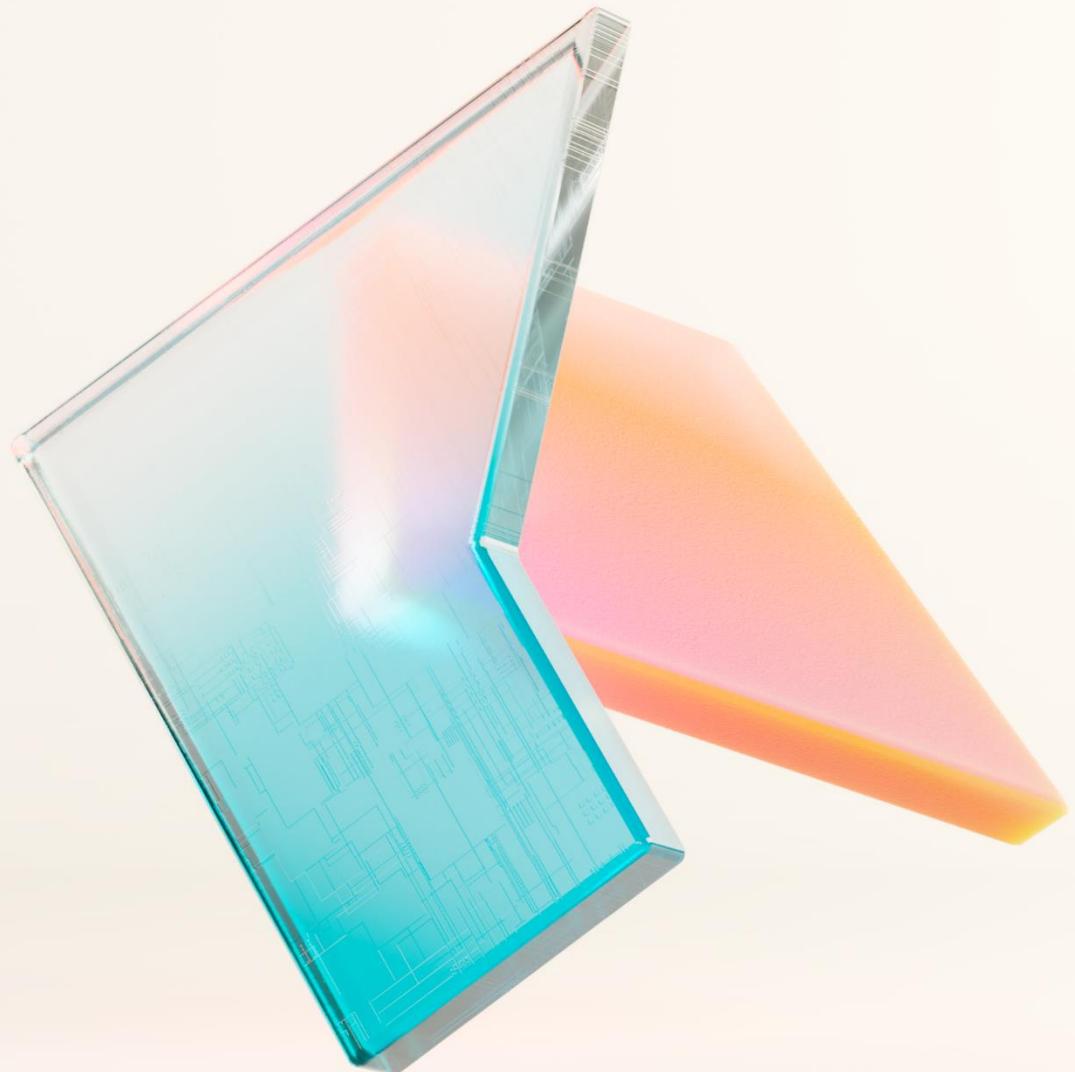
Quickly react to shifting market trends, **identify opportunities and mitigate risk** in real time



Opportunities abound across industries when real-time data value is realized



| Automotive | Manufacturing | Logistics | Finance & Insurance | Energy & Utilities | Retail | Healthcare |
|---|--|---|---|--|--|--|
| <ul style="list-style-type: none">Connected fleet applicationsAutonomous drivingManufacturing R&D | <ul style="list-style-type: none">Quality and throughput improvementPredictive maintenancePredictive inventory | <ul style="list-style-type: none">Delivery tracking and routingWarehouse managementSupply & demand operations | <ul style="list-style-type: none">Finance automationFraud detectionOperational efficiency | <ul style="list-style-type: none">Station monitoring, energy leakage detectionEquipment maintenance & monitoringFailure monitoring | <ul style="list-style-type: none">Inventory trackingPromotions and buying experiencesSupply chain management | <ul style="list-style-type: none">Emergency responseRemote patient monitoringClinical decision supportSurgical assistanceHospital operations |



Real-Time
Intelligence in
Microsoft Fabric



Microsoft Fabric

The data platform for the era of AI

Unify your analytics on a complete, governed platform

Give your data teams all the tools they need in a unified, governed, and secure experience that reduces the cost and effort of integration.

Establish a trusted data foundation

Integrate data from anywhere into a single, multi-cloud data lake for the entire organization and work from the same copy of data across analytics engines.

Empower every business user

Empower everyone to uncover insights with the data they need, easy-to-use tools, and visuals embedded in the Microsoft 365 apps they use everyday.

Fuel your AI innovation

Adopt a data platform that's infused with AI at every layer to help you get more done, faster.



Microsoft Fabric

Intelligent data foundation



Data
Factory



Data
Engineering



Data
Warehouse



Data
Science



Real-Time
Intelligence



Power
BI



Industry
Solutions



Powered by AI with Copilot in Microsoft Fabric



Catalog for data in motion

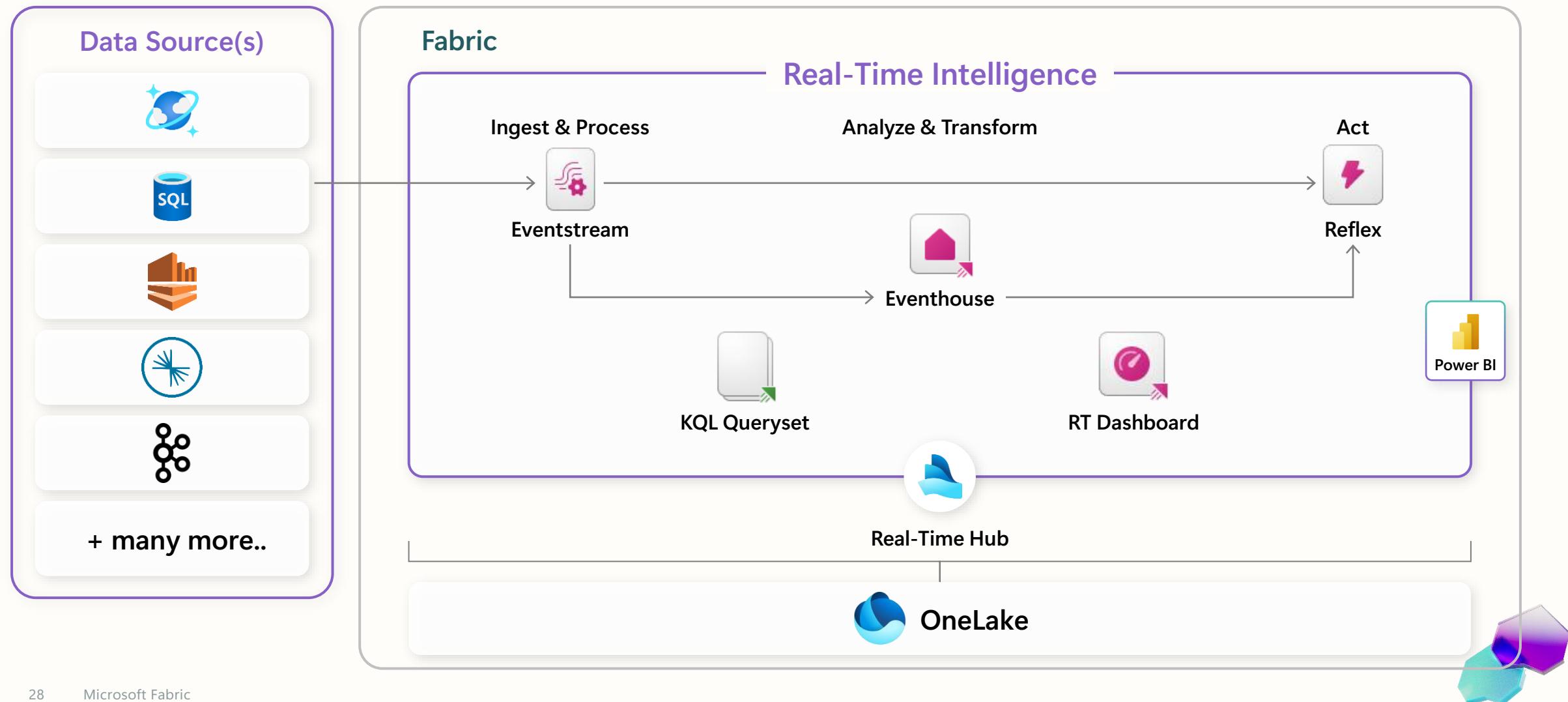
Real-Time Hub

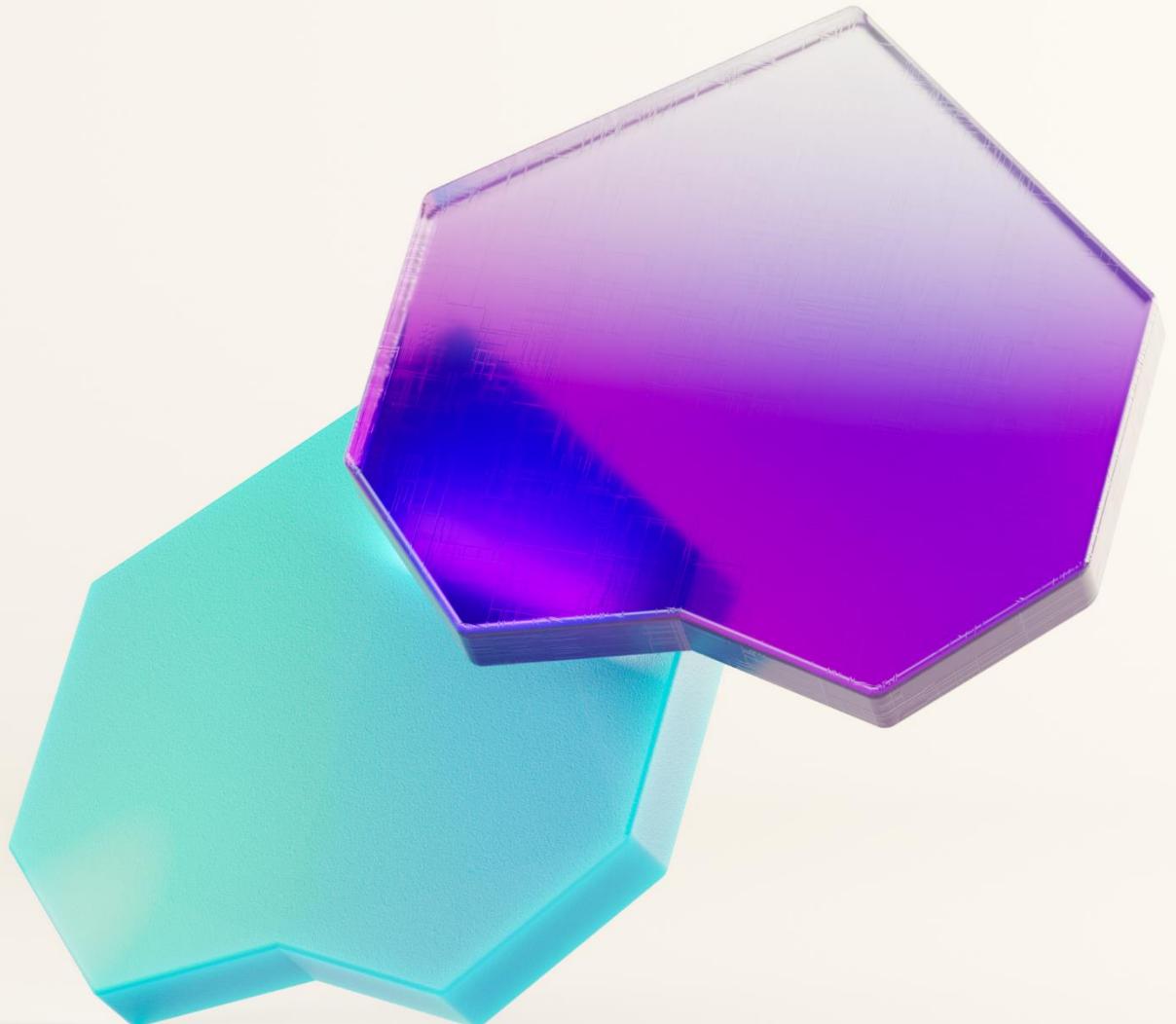


Unified data foundation

OneLake

Components of Fabric's Real-Time Intelligence





A complete SaaS
solution



A complete SaaS solution

Choose from a range of end-to-end capabilities to act quickly on time-oriented data



Ingest & process all event sources

Connect to a diverse set of streaming sources and leverage no-code and low-code experiences to process and route quickly



Analyze data event streams

Analyze and transform data streams, using queries and visual exploration to discover insights in real-time



Act quickly on top of data

Create triggers and set alerts on changing data to respond automatically and set action when specific conditions are detected

Ingest & process all data sources

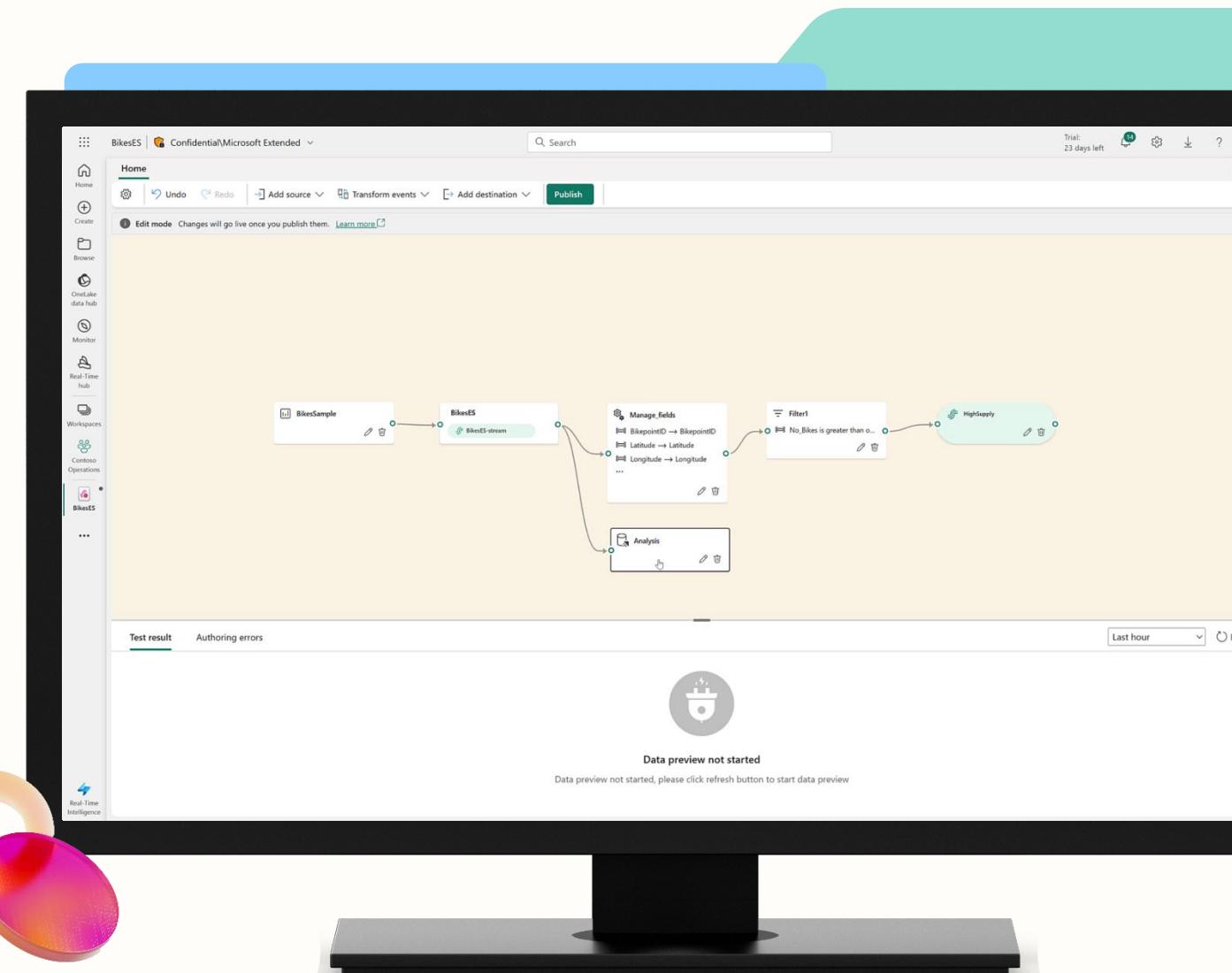
Ingest data from any source, in any data format

Capture, transform and route event data without writing any code

Access out-of-the-box connectors for streaming and event data sources

Process data and enrich real-time events

Route events to other Fabric and 3rd party entities



Analyze data event streams

Manage an unlimited amount of data, from gigabytes to petabytes, with unlimited scale

Use event houses to handle and analyze large volumes of real-time data streams

Monitor and manage multiple databases at once

Create KQL databases and querysets to run, view, and customize queries directly on raw data

Enhance organizational BI reports with enriched data



The screenshot displays the Microsoft Fabric web interface. On the left, a sidebar shows navigation links like Home, Create, Browse, OneLake data hub, Monitor, Real-Time hub, Workspaces, Contoso Operations, Manufacturing_evh, Manufacturing_kdb, and BikeES. The main area has tabs for Database, Manage, and Eventhouse. Under Database, there's a 'Database: Manufacturing_kdb' section showing details such as 'Created by', 'Region', 'Created on', and 'Last ingestion'. Below this is a 'Top tables' section listing 'Deliveries', 'DeliveryStream', 'Bikes', 'DeliveryZones', and 'Trucks' with their respective sizes. A 'Recently updated functions' section shows two entries for 'Deliveries'. To the right, a 'Explore your data' panel contains a code editor with KQL queries, a 'Run' button, and a preview table titled 'Table 1' showing data for bike points. The table includes columns for BikePointID, Street, Neighbourhood, Latitude, Longitude, No_Bikes, and No_Empty_Docks, with 100 records listed.

| BikePointID | Street | Neighbourhood | Latitude | Longitude | No_Bikes | No_Empty_Docks |
|----------------|------------------------------|-----------------|------------|-----------|----------|----------------|
| BikePoints_357 | Howland Street | Fitzrovia | 51.5209923 | -0.139016 | 24 | 6 |
| BikePoints_624 | Courland Grove | Wandsworth Road | 51.47292 | -0.132102 | 7 | 33 |
| BikePoints_460 | Burdett Road | Mile End | 51.5161972 | -0.029130 | 28 | 5 |
| BikePoints_602 | Union Grove | Wandsworth Road | 51.4729919 | -0.133972 | 37 | 8 |
| BikePoints_491 | Queen Mary's | Mile End | 51.5225067 | -0.041378 | 7 | 38 |
| BikePoints_492 | Maplin Street | Mile End | 51.5255 | -0.03267 | 10 | 29 |
| BikePoints_714 | Stewart's Road | Wandsworth Road | 51.4731178 | -0.137235 | 15 | 13 |
| BikePoints_522 | Clinton Road | Mile End | 51.52594 | -0.03607 | 24 | 12 |
| BikePoints_531 | Twig Folly Bridge | Mile End | 51.5303268 | -0.042744 | 0 | 21 |
| BikePoints_518 | Antill Road | Mile End | 51.5282249 | -0.037471 | 7 | 23 |
| BikePoints_550 | Harford Street | Mile End | 51.5215645 | -0.039264 | 4 | 19 |
| BikePoints_712 | Mile End Stadium | Mile End | 51.51854 | -0.034903 | 13 | 9 |
| BikePoints_763 | Mile End Park Leisure Centre | Mile End | 51.5205956 | -0.032566 | 24 | 23 |

Real-time AI insights – and more to come!

Leverage Copilot to automate routine tasks and act as an interactive aide

Easily generate KQL queries with copilot experience

Monitor data and automatically drive alerts when anomalies are detected

Use real-time insights to build and scale advanced ML models in Azure Machine Learning

Create generative AI experiences on top of your time-oriented data with Azure AI Studio



The screenshot shows the Microsoft Fabric AI Studio interface. On the left, there's a sidebar with various data sources like BikesQS, Manufacturing_kdb, and BikeES. The main area has a KQL query editor with the following code:

```
// what is the total number of bikes by neighborhood and street
// 
Bikes
| summarize totalNoBikes=sum(No_Bikes) by Neighbourhood, Street
// 
// what is the total number of bikes and total number of docks by neighborhood
// 
Bikes
| summarize totalBikes=sum(No_Bikes), totalDocks=sum(No_Empty_Docks) by Neighbourhood
```

Below the query editor is a table titled "Table 1" with the following data:

| Neighbourhood | totalBikes | totalDocks |
|-----------------|------------|------------|
| Battersea | 767,151 | 1,811,492 |
| Chelsea | 1,025,125 | 453,591 |
| Strand | 1,178,044 | 1,056,621 |
| Mile End | 836,128 | 1,228,694 |
| Fitzrovia | 863,769 | 399,663 |
| Knightsbridge | 661,864 | 790,599 |
| Victoria | 540,081 | 627,186 |
| West Chelsea | 575,657 | 523,774 |
| St. John's Wood | 536,185 | 493,946 |
| Bankside | 623,707 | 836,793 |
| Sands End | 744,302 | 607,420 |
| West End | 466,617 | 271,655 |
| Belgravia | 283,397 | 457,025 |

To the right, there's a "Copilot" sidebar with a question input field and a list of previous questions:

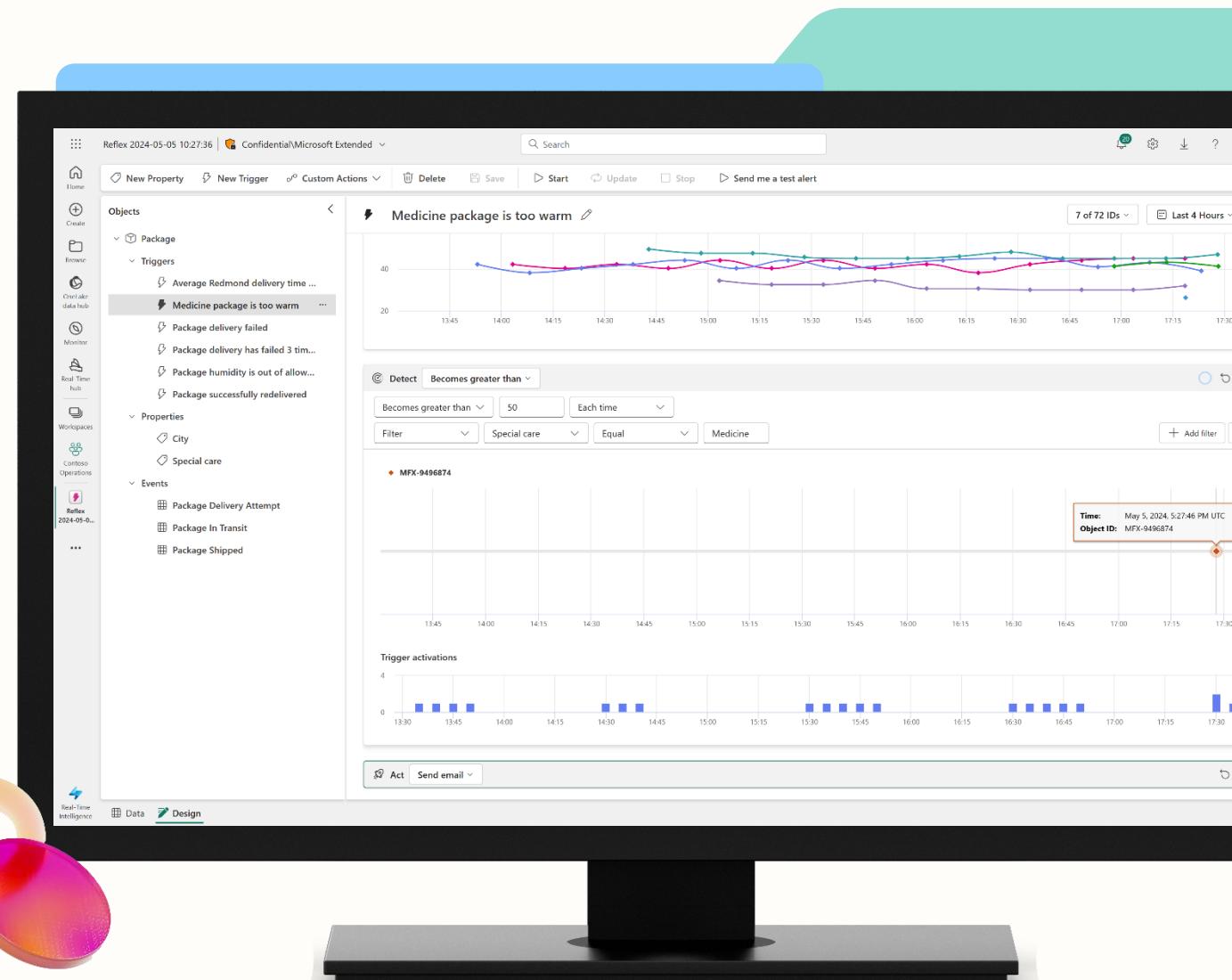
- What is the total number of bikes by neighborhood and street
- What is the total number of bikes by neighborhood

Act quickly on top of data

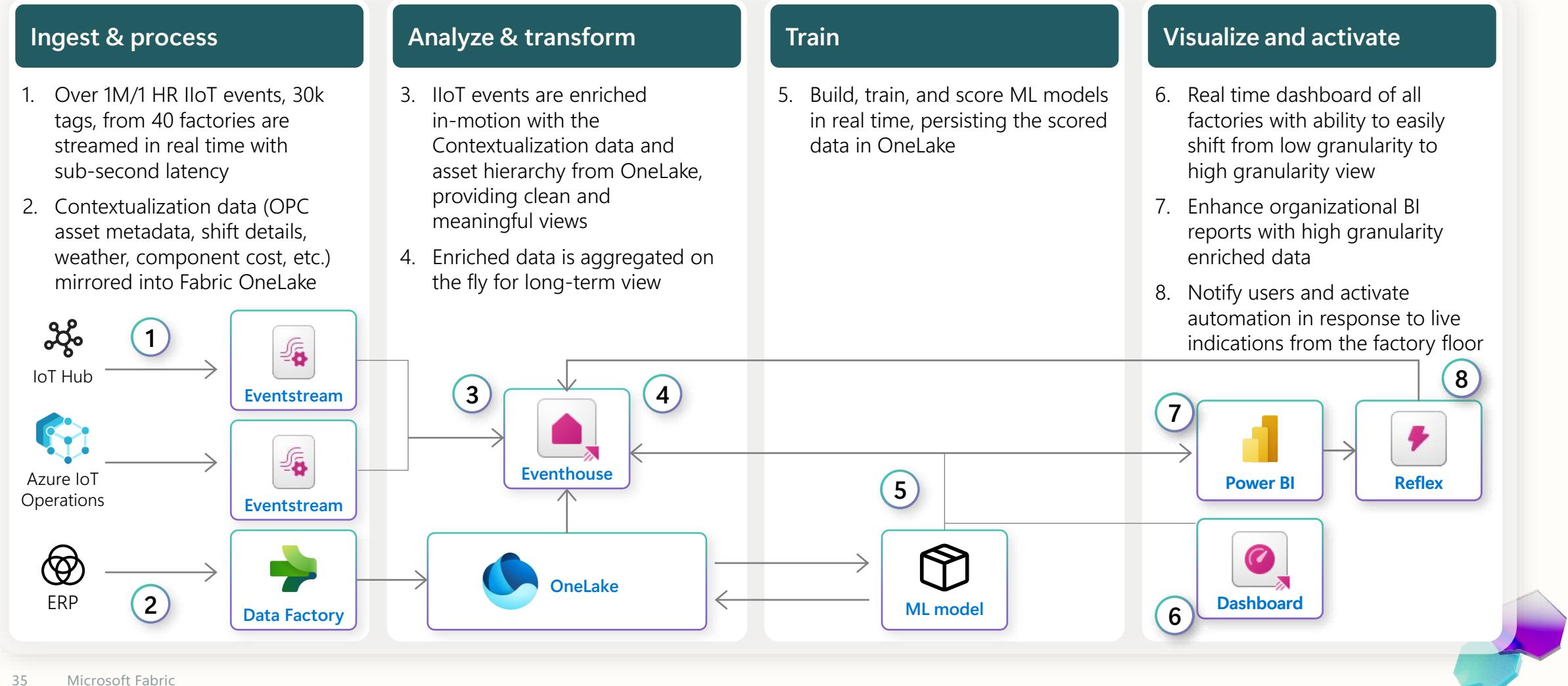
Automatically take actions when patterns are detected in changing data

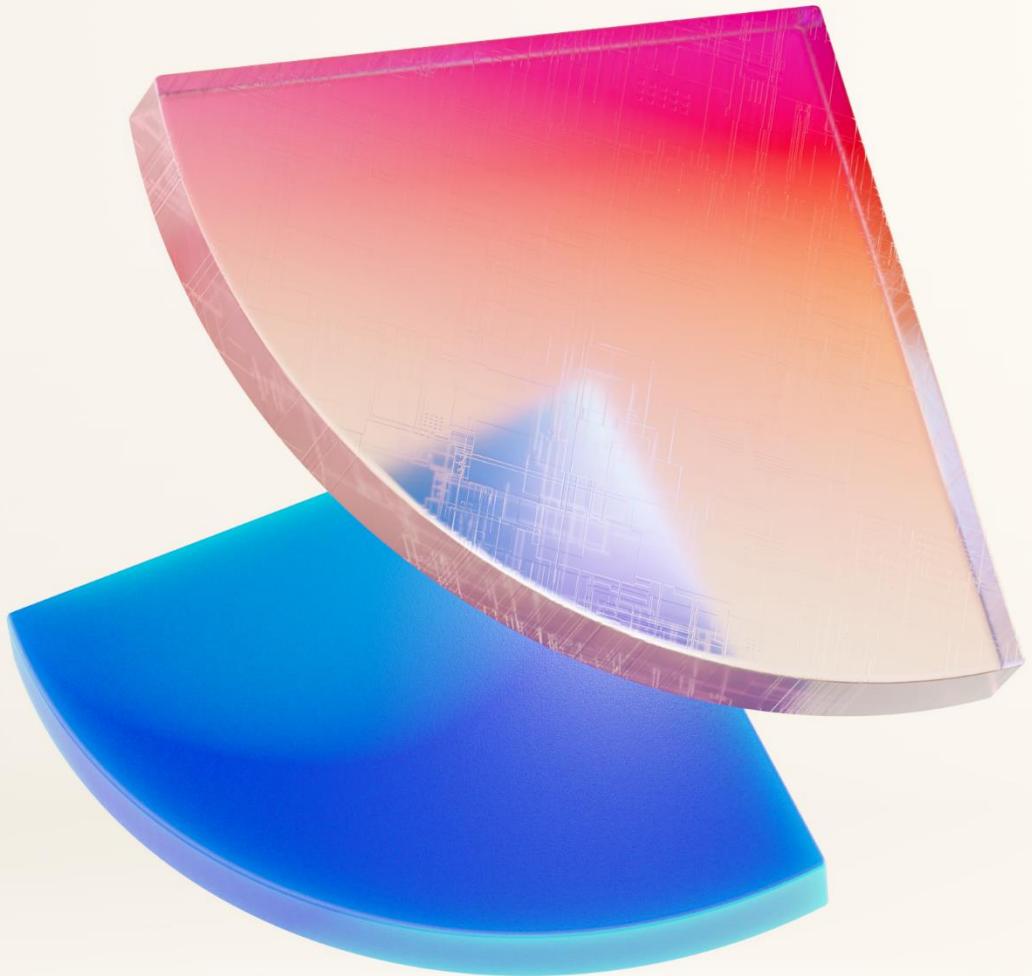
Drive actions on a per instance state that evolves over time

Act on data without needing a deep schema and semantic modeling



An end-to-end Real-Time Intelligence experience – Connected Factory





A single place for
data in motion

Access Real-Time Hub

Find and consume event sources in Real-Time Hub, a single catalog of all streaming sources

Use Get Events and all streaming connectors to easily bring data in

Subscribe to internal and external discrete events

Quickly connect experiences for Microsoft streaming sources like IoT Hub

Access system events emitted by Fabric and Azure storage

Ingest data streams from all clouds (e.g. AWS, Kinesis, Google Pub-Sub, etc.)

| Name | Item | Owner | Location | Endorsement | Sensitivity |
|----------------------------|---------------------|------------------------|--|--------------|--------------------|
| CosmosDBCDC-goods-stream | CosmosDBCDC-goods | Tessa Kloster (PALMER) | Contoso Operations | — | Microsoft Extended |
| new_event_stream-stream | new_event_stream | Chami Rupasinghe | SQL DB Native Bug Bash | — | Microsoft Extended |
| CosmosGoods-stream | CosmosGoods | Tessa Kloster (PALMER) | Contoso Operations | — | Microsoft Extended |
| CustomAppStream-stream | CustomAppStream | Shiv Kumar | DreamDemoCopilot | — | Microsoft Extended |
| ThermostatStream-stream | ThermostatStream | Shiv Kumar | DreamDemoCopilot | — | Microsoft Extended |
| Thermostat | Retail_Eventhouse | — | DreamDemoCopilot | — | Microsoft Extended |
| ThermostatData | Retail_Eventhouse | — | DreamDemoCopilot | — | Microsoft Extended |
| Sensors-stream | Sensors | Tessa Kloster (PALMER) | Contoso Operations | — | Microsoft Extended |
| fabric_event_stream-stream | fabric_event_stream | Reeham Johnson | Power BI Admin and Governance (Fabric) | — | Microsoft Extended |
| CustomerSensors-stream | CustomerSensors | Tessa Kloster (PALMER) | Contoso Operations | — | Microsoft Extended |
| SQLCDC-customer-stream | SQLCDC-customer | Tessa Kloster (PALMER) | Contoso Operations | — | Microsoft Extended |
| Shipments-stream | Shipments | Tessa Kloster (PALMER) | Contoso Operations | (○ Promoted) | Microsoft Extended |
| HighDemand | Shipments | Tessa Kloster (PALMER) | Contoso Operations | (○ Promoted) | Microsoft Extended |
| HighSupply | Shipments | Tessa Kloster (PALMER) | Contoso Operations | (○ Promoted) | Microsoft Extended |
| LowSupply | Shipments | Tessa Kloster (PALMER) | Contoso Operations | (○ Promoted) | Microsoft Extended |
| OneRiverEvent-stream | OneRiverEvent | Bert Cotton | Power BI Admin and Governance (Fabric) | — | Microsoft Extended |
| fabric_event_stream-stream | fabric_event_stream | Bert Cotton | Power BI Admin and Governance | — | Microsoft Extended |

Consume data from anywhere

Find time-oriented data-in-motion readily available in minutes for use in OneLake

Discover events and seamlessly consume them from across organization

Simplify integration of stream processing frameworks

Expose and use well-established open-source APIs, standards, protocols and connectors

Maintain data ownership – data is not trapped in Microsoft's proprietary formats

Integrate seamlessly with other experiences in Microsoft Fabric



The screenshot shows the Microsoft Fabric Real-Time hub (preview) interface. The left sidebar includes options like Home, Create, Browse, OneLake data hub, Monitor, Real-Time hubs, Workspaces, Contoso Operations, and more. The main area is titled "Real-Time hub (preview)" and "Select a data source". It features a search bar and tabs for "Microsoft sources", "Database CDC", and "Fabric events". A "Recommended" section lists connectors such as Azure Event Hubs, Azure IoT Hub, Confluent Cloud Kafka, and Fabric Workspace Item events. Below this is a larger "All" section listing many more connectors, including Amazon Kinesis Data Streams, Azure Cosmos DB (CDC), Azure Event Hubs, Azure IoT Hub, Azure SQL DB (CDC), Azure Blob Storage events, Confluent Cloud Kafka, Fabric Workspace Item events, Google Cloud Pub/Sub (CDC), and Power BI Admin and Governance (Fabric). The interface is designed for easy consumption and integration of real-time data from various sources.

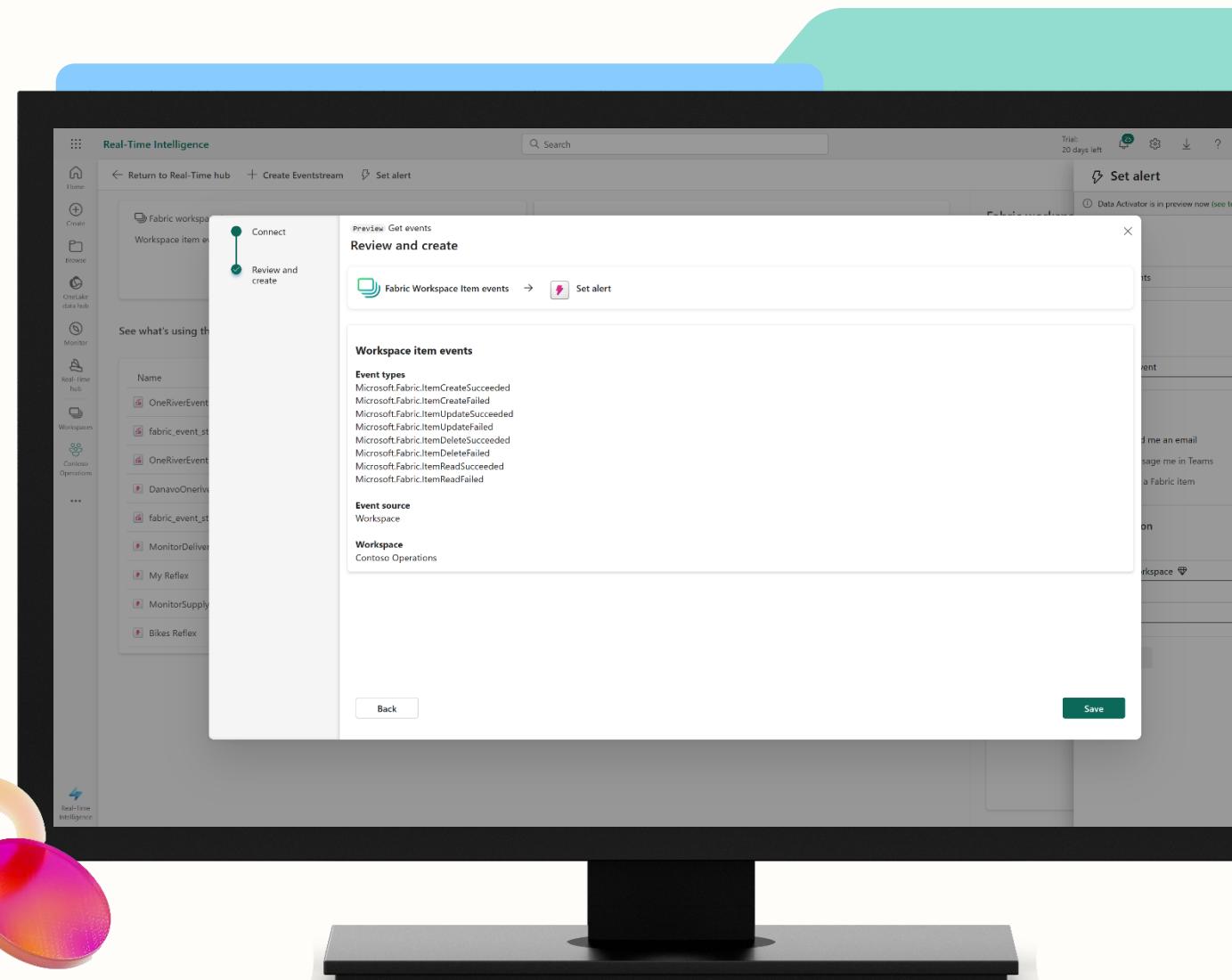
Rapid solution development – no-, low-, and pro-code experiences

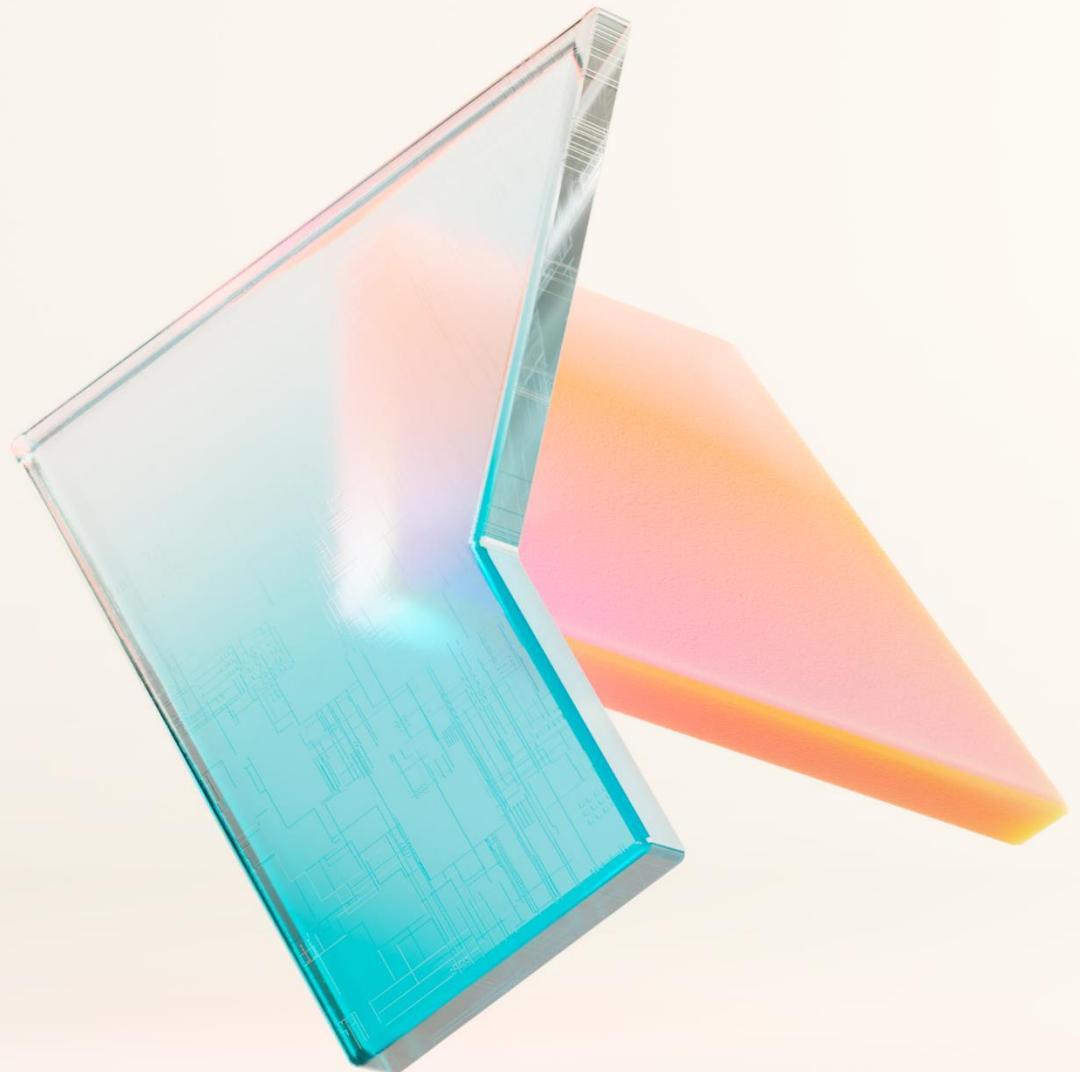
Access wide range of experiences to quickly build production solutions

Leverage custom endpoints and APIs to integrate directly into existing systems

Empower developers with CI/CD tools within their workflows

Gain proven performance at scale across ingestion, querying and storage





What we're
hearing from
customers



One NZ democratizes data access, enabling customer support with Microsoft Fabric Real-Time Intelligence

“

Previously, you needed to be a data engineer or scientist to access and understand customer information. Now we're making it user-friendly, so anyone can easily make data-driven decisions.

– Strathan Campbell, Channel Environment Technology Lead, One NZ

Situation

Seeking to improve customer experience, New Zealand telecom giant, One NZ, wanted to take its performance to the next level. Increasing data volumes were leading to delayed refresh rates and One NZ needed a solution with real-time data and analytics capabilities they could easily implement with their existing systems.

Solution

One NZ selected Microsoft Fabric's Real-Time Intelligence (RTI) and went from concept to delivered product in just two weeks. Using event streams to ingest data from external data sources and run analytics on top of its real-time streaming data, One NZ was able to generate immediate actions on their data for quicker decision-making.

Impact

With Microsoft Fabric's RTI, One NZ has improved overall customer experience. All teams have access to high quality data that can be used in ML models, plus, dashboards are updated 6x faster than before, so agents can identify customer behavior patterns and respond to customers more quickly than ever.





Dener Motorsport gains real-time race and car insights with Microsoft Fabric

PORSCHE
CARRERA CUP
BRASIL

“

Before we used Microsoft Fabric and real-time analytics, it was probably 30 minutes before the engineers who knew that something was wrong with a car could get the data, analyze it, and have a result. Today the process is done within minutes.

– Dener Pires, CEO, Dener Motorsport

Situation

The Porsche Carrera Cup Brasil is a premier racing event, produced annually by Dener Motorsport. Given the prestige of the Porsche brand, expectations are high, but race information and reporting did not live up to the brand image, with manual solutions that prevented the sharing of time-critical information.

Solution

Dener Motorsport partnered with Microsoft Fabric, leveraging its streaming analytics capabilities, as well as improved data storage and reporting. Using Fabric, Dener Motorsport created a new architecture able to ingest all their data streams, analyze them and distribute findings to team managers and race participants.

Impact

With this new, unified architecture, Dener Motorsport can better maintain car health, using Fabric's real-time analytics for faster identification of vehicular problems, having reduced the time to insight from half an hour to a matter of minutes. This keeps drivers safer, sustains optimal car performance, and keeps the cars in good condition.



What other customers are saying

"This cloud solution has empowered us to easily understand and act on high-volume, high-granularity events in real-time with fewer resources."

"Our data now informs immediate actions—reallocating resources or preemptively solving customer issues."

"We will now have more accurate information at our fingertips. Higher-quality data also means greater potential for machine learning, IoT, and AI-powered workloads. We expect to drive a lot of innovation."

"Our clients have built out reports tracking IoT data. Water utility services use high-frequency sensors to keep track of water quality. Now users can ensure quality with near real-time analytics that save time in both generating reports and chasing down data."

"With IoT data and AI insights, we can quickly identify geographic areas where our customers are buying fewer replacement parts from us, for instance. We can then take the needed steps to win back that business."

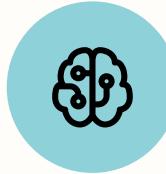


Get started today



Set up your free Microsoft Fabric trial

aka.ms/fabric-try



Learn more about Real-Time Intelligence

aka.ms/realtimeskill



Watch Build 2024 sessions

youtu.be/4h6Wnc294gA?si=Dw--Ppj4Dm4LX1ZB



Check out the Fabric RTI Blog

aka.ms/rti-blog



Learn more with Mechanics

aka.ms/RTFabricMechanics



Thank you

Hiram Fleitas

Senior Cloud Solution Architect, Microsoft Data & AI

<https://aka.ms/hiram>