Data Saturday Holland

Is This Just Real Time or Is This Just Fantasy?







Thank you sponsors!

INSPARK

twintos

ilionx

InfoSupport
Solid Innovator

redgate

sopra steria

DATA GIANTS

BRANDER
company

SQL brander.nl
DATA brander.nl
CRM brander.nl
ERP brander.nl

Bull
Master your AI destiny

dbWatch

BBTG

Data Saturdays

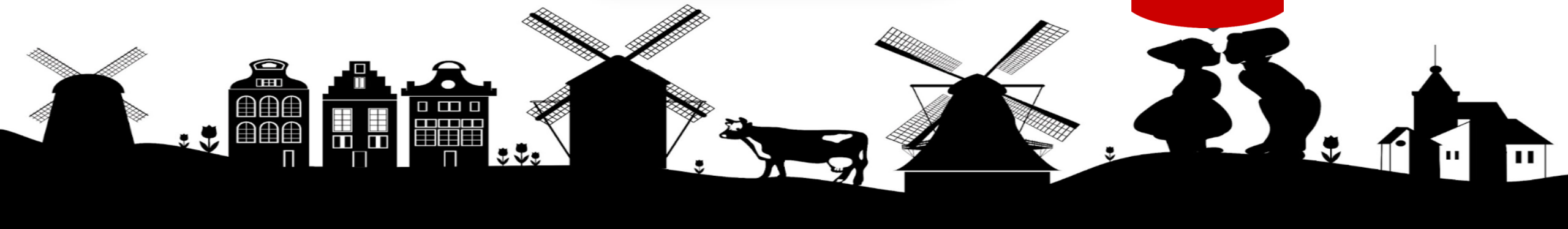
Who Am I

- Mission Critical Engineer
- Schuberg Philis
- DBA Turned MCE
- Junior SQL DBA for Next (UK)
- SQL DBA (Albelli)
- MCE (SBP)
- Azure Heroes Organizer NL
- Azure User Group NL



MVP Microsoft® Most Valuable Professional

redgate COMMUNITY AMBASSADOR



Content

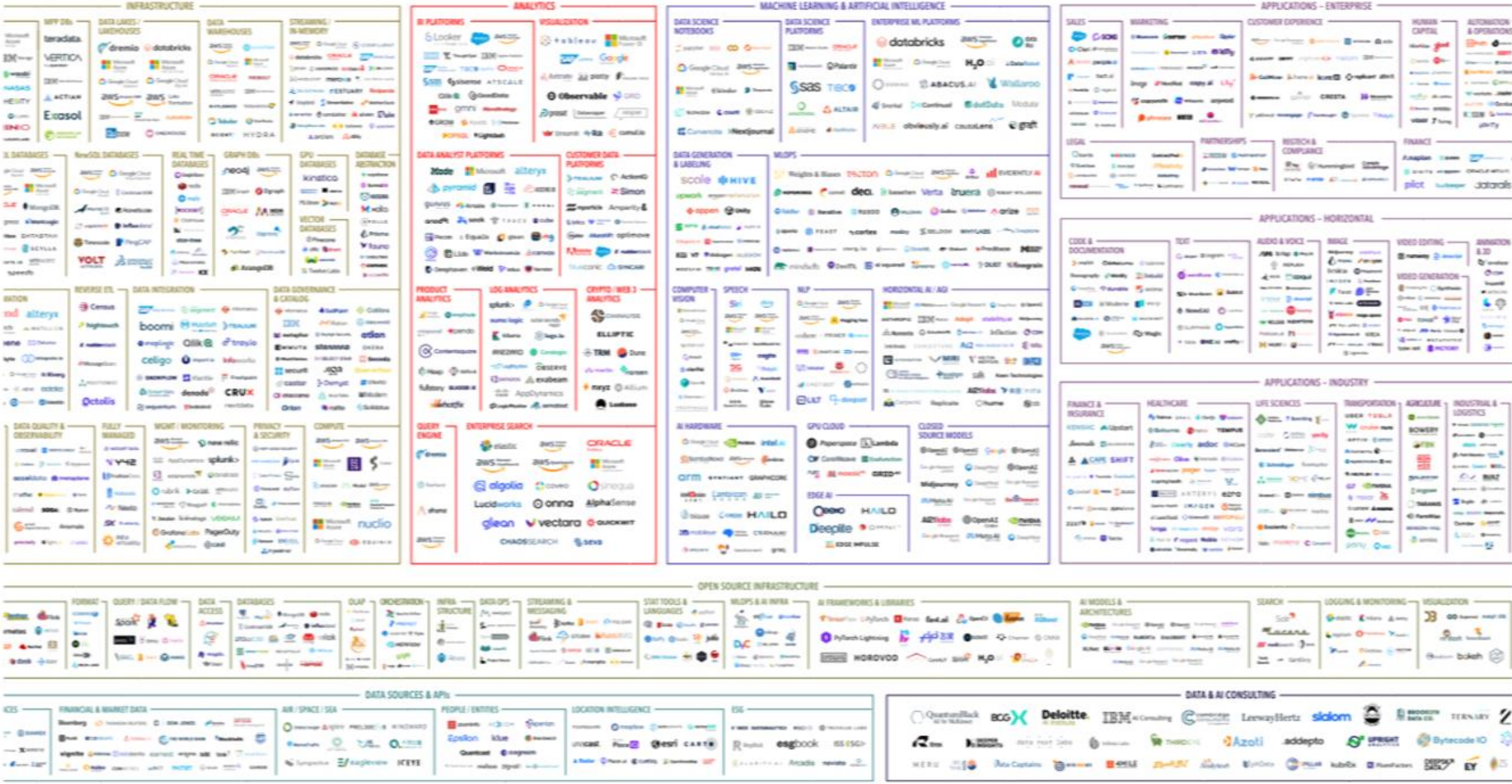
- What Does “Real-Time” Actually Mean?
- Deep Dive: Microsoft Fabric Real-Time Capabilities
- Demo
- Choosing the Right “Real-Time” for Your Business
- Q&A



Content

- But First what choices of Data tools are there?

THE 2023 MAD (MACHINE LEARNING, ARTIFICIAL INTELLIGENCE & DATA) LANDSCAPE



© Matt Turck (@mattturck), Kevin Zhang (@ykevinzhang) & FirstMark (@firstmarkcap)
Blog post: mattturck.com/MAD2023
Interactive version: MAD.firstmarkcap.com
Comments? Email MAD2023@firstmarkcap.com
FIRSTMAR
EARLY STAGE VENTURE CAP



Real Real Time

- What’s *real-time* for you?



“When your business says ‘real-time,’ what do they mean?”

Batch vs Micro-Batch vs Streaming

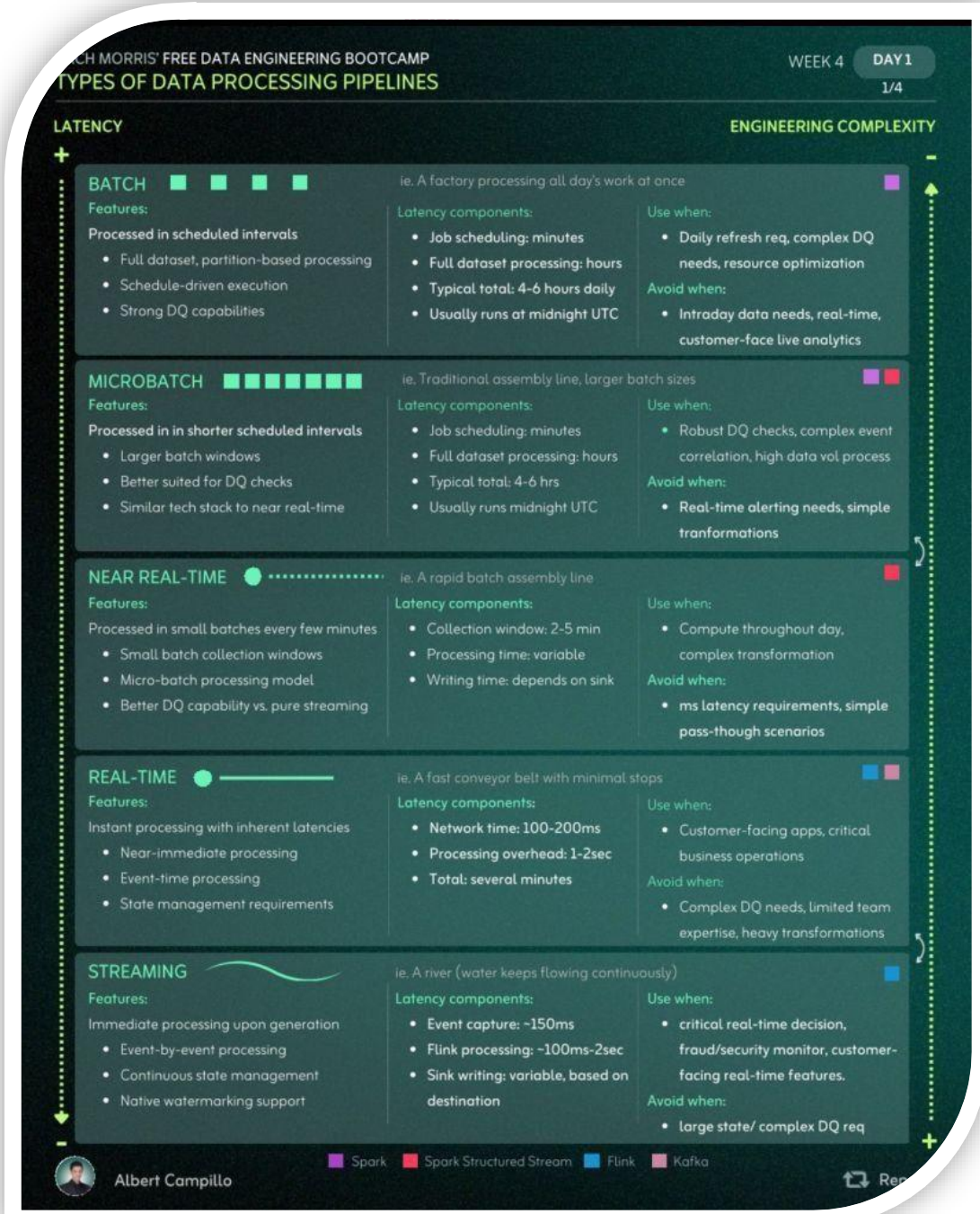
When to Use What and Why



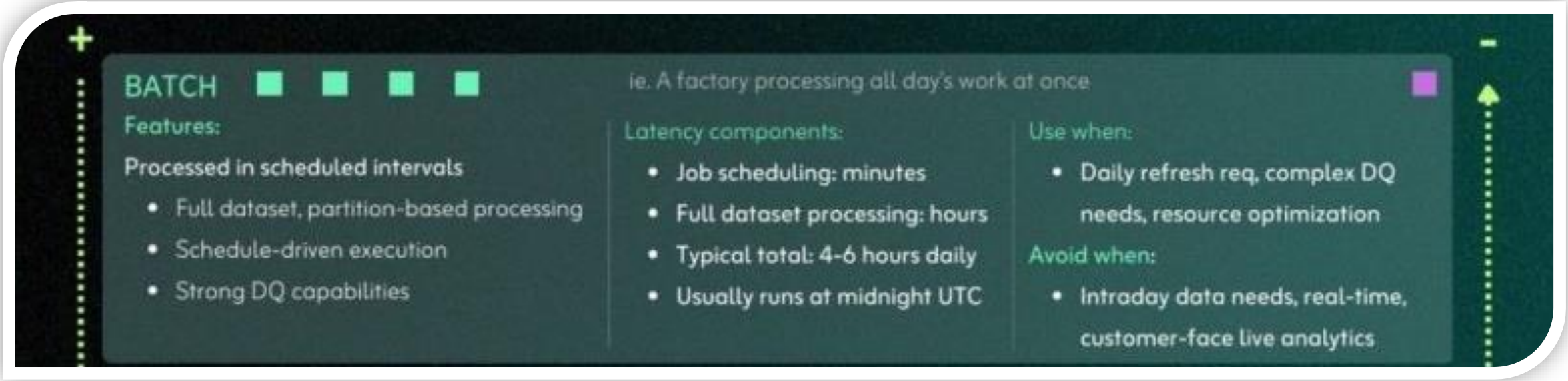




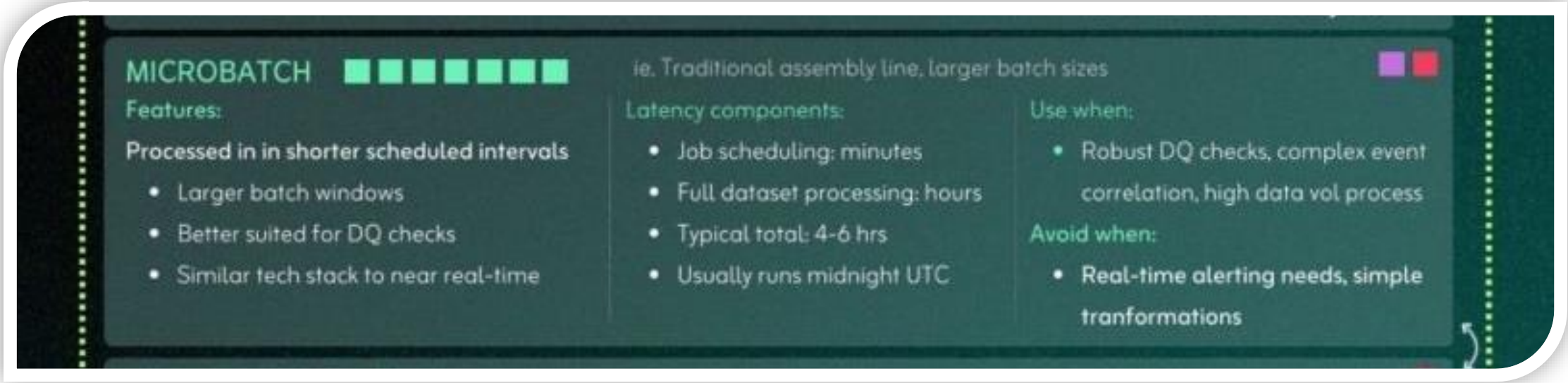
Batch	Micro-Batch	Streaming
Latency **Hours**	Latency **Minutes**	Latency **Seconds**
• BI reports • Historical loads	• Usage tracking • Metrics	• Fraud detection • Real-time UX


CH MORRIS' FREE DATA ENGINEERING BOOTCAMP
TYPES OF DATA PROCESSING PIPELINES
WEEK 4
DAY 1
1/4
LATENCY
ENGINEERING COMPLEXITY
BATCH
ie. A factory processing all day's work at once
Features:
Processed in scheduled intervals
Full dataset, partition-based processing
Schedule-driven execution
Strong DQ capabilities
Latency components:
Job scheduling: minutes
Full dataset processing: hours
Typical total: 4-6 hours daily
Usually runs at midnight UTC
Use when:
Daily refresh req, complex DQ needs, resource optimization
Avoid when:
Intraday data needs, real-time, customer-face live analytics
MICROBATCH
ie. Traditional assembly line, larger batch sizes
Features:
Processed in in shorter scheduled intervals
Larger batch windows
Better suited for DQ checks
Similar tech stack to near real-time
Latency components:
Job scheduling: minutes
Full dataset processing: hours
Typical total: 4-6 hrs
Usually runs midnight UTC
Use when:
Robust DQ checks, complex event correlation, high data vol process
Avoid when:
Real-time alerting needs, simple tranformations
NEAR REAL-TIME
ie. A rapid batch assembly line
Features:
Processed in small batches every few minutes
Small batch collection windows
Micro-batch processing model
Better DQ capability vs. pure streaming
Latency components:
Collection window: 2-5 min
Processing time: variable
Writing time: depends on sink
Use when:
Compute throughout day, complex transformation
Avoid when:
ms latency requirements, simple pass-though scenarios
REAL-TIME
ie. A fast conveyor belt with minimal stops
Features:
Instant processing with inherent latencies
Near-immediate processing
Event-time processing
State management requirements
Latency components:
Network time: 100-200ms
Processing overhead: 1-2sec
Total: several minutes
Use when:
Customer-facing apps, critical business operations
Avoid when:
Complex DQ needs, limited team expertise, heavy transformations
STREAMING
ie. A river (water keeps flowing continuously)
Features:
Immediate processing upon generation
Event-by-event processing
Continuous state management
Native watermarking support
Latency components:
Event capture: ~150ms
Flink processing: ~100ms-2sec
Sink writing: variable, based on destination
Use when:
critical real-time decision, fraud/security monitor, customer-facing real-time features.
Avoid when:
large state/ complex DQ req
Spark
Spark Structured Stream
Flink
Kafka
Albert Campillo

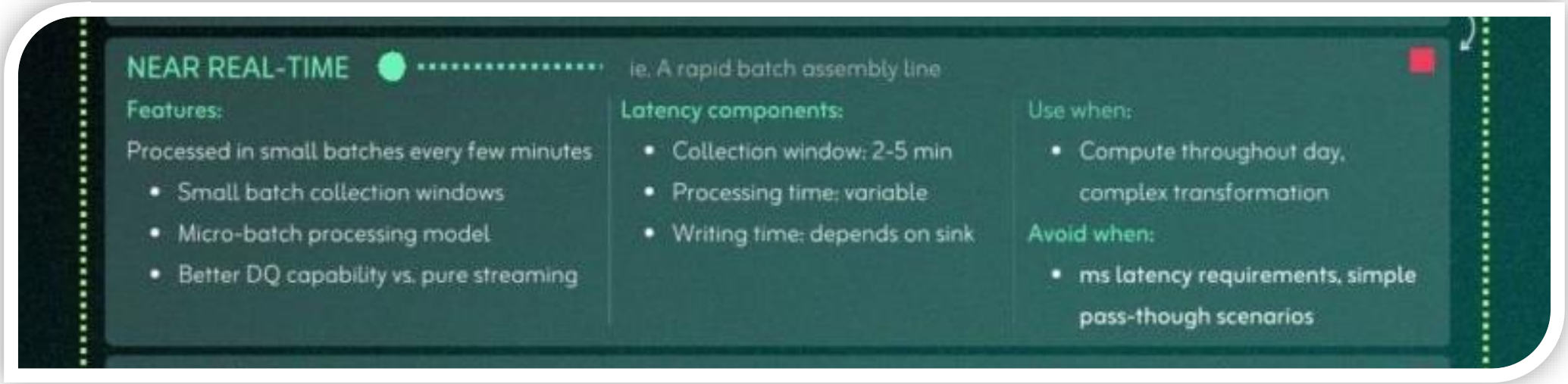
Batch


BATCH
ie. A factory processing all day's work at once
Features:
Processed in scheduled intervals
• Full dataset, partition-based processing
• Schedule-driven execution
• Strong DQ capabilities
Latency components:
• Job scheduling: minutes
• Full dataset processing: hours
• Typical total: 4-6 hours daily
• Usually runs at midnight UTC
Use when:
• Daily refresh req, complex DQ needs, resource optimization
Avoid when:
• Intraday data needs, real-time, customer-face live analytics

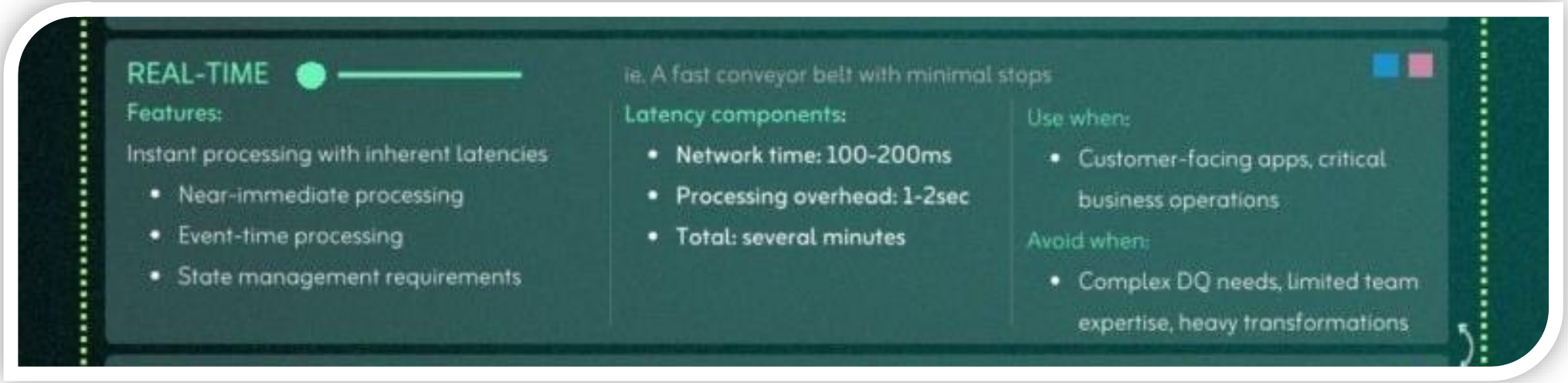
Microbatch


MICROBATCH
ie. Traditional assembly line, larger batch sizes
Features:
Processed in in shorter scheduled intervals
• Larger batch windows
• Better suited for DQ checks
• Similar tech stack to near real-time
Latency components:
• Job scheduling: minutes
• Full dataset processing: hours
• Typical total: 4-6 hrs
• Usually runs midnight UTC
Use when:
• Robust DQ checks, complex event correlation, high data vol process
Avoid when:
• Real-time alerting needs, simple tranformations

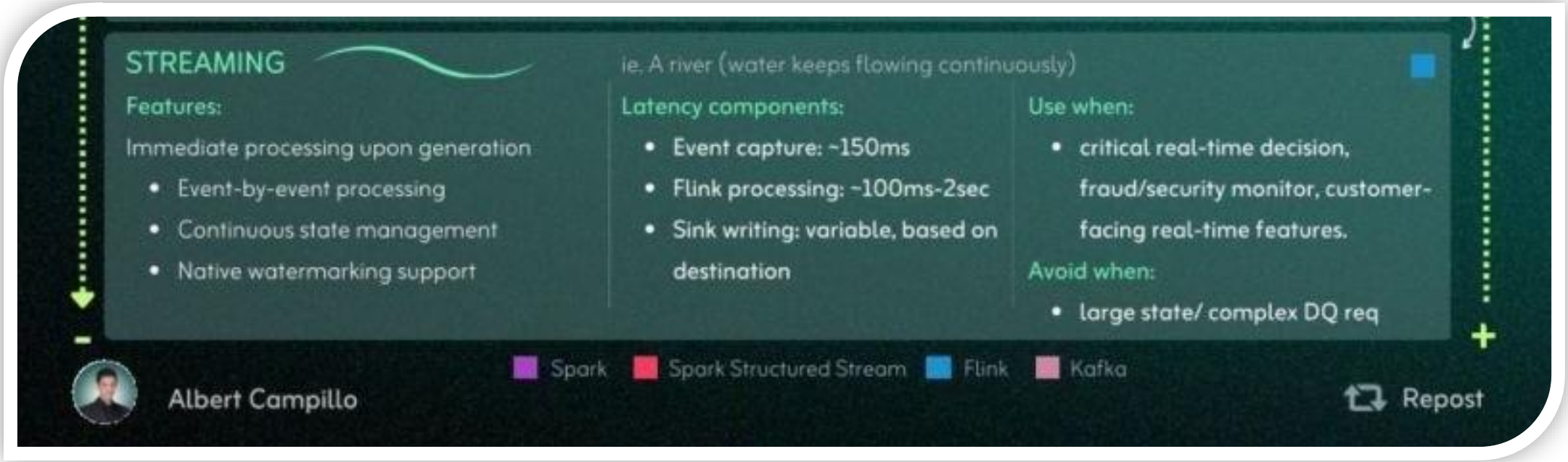
Near Real Time



Streaming


REAL-TIME
Features:
Instant processing with inherent latencies
• Near-immediate processing
• Event-time processing
• State management requirements
ie. A fast conveyor belt with minimal stops
Latency components:
• Network time: 100-200ms
• Processing overhead: 1-2sec
• Total: several minutes
Use when:
• Customer-facing apps, critical business operations
Avoid when:
• Complex DQ needs, limited team expertise, heavy transformations

Why This Choice Matters

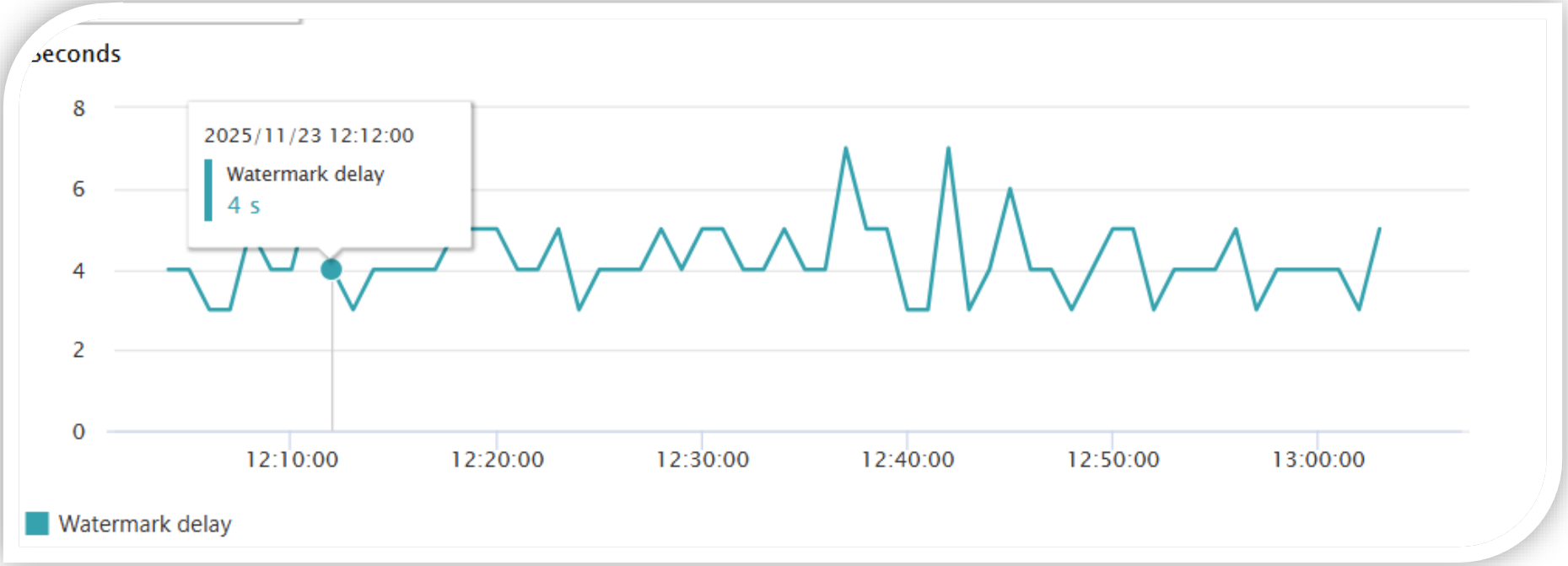




How Fast is Fast?

Node	Type	Throughput level	Approximate throughput (up-to)
Custom Endpoint	Source / Destination	Low	100 MB/s
		Medium	150 MB/s
		High	200 MB/s
Lakehouse	Destination	Low	40 MB/s
		Medium	120 MB/s
		High	200 MB/s
Eventhouse (Direct Ingestion)	Destination	Low	10 MB/s
		Medium	50 MB/s
		High	100 MB/s
Eventhouse (Event processing before ingestion)	Destination	Low	20 MB/s
		Medium	100 MB/s
		High	200 MB/s

How Fast is Fast?





LPF?

- Latency
- Processing Time
- Freshness

Latency

How long it takes for data to travel from the source to your system.

Think: **event leaves the device/app → lands in your platform**
(Fabric/Event Hub/Kafka/Storage).

Starts: when the event happens

Ends: when the event is available in your ingestion point (e.g., Eventstream/KQL DB input, landing file).

Latency is about **transport and arrival**, not about when the business *actually uses* the data.
Flight Time For example



Processing Time

- **How long your system spends transforming/handling the data once it has it.**

- Starts: when your job starts processing the data (e.g., pipeline kicked)

Examples:

- A Data Factory or Fabric pipeline that takes 2 minutes to copy + transform.
- A KQL query that runs in 200 ms on a streaming table.

- In **streaming**, processing tends to be continuous and low-latency (millis–seconds).
- In **batch**, processing can be heavier and happens at intervals
- The Suitcase Organising for collection

Freshness

- **How “up-to-date” the data is from the user’s perspective.**
- Think: **“How old is the data I’m seeing right now?**

- Starts: when the event happened.
- Ends: when the user/system **can query or see** it (dashboard, report, API).

- So:
- **Freshness = Now – Event time**

- Freshness is what the **business user actually feels** when they say:
- “The dashboard is 15 minutes behind.”
- “That number should update instantly!”
- End to End



Why This Matters for “Real-Time”

- When the business says “real-time,” you can ask:

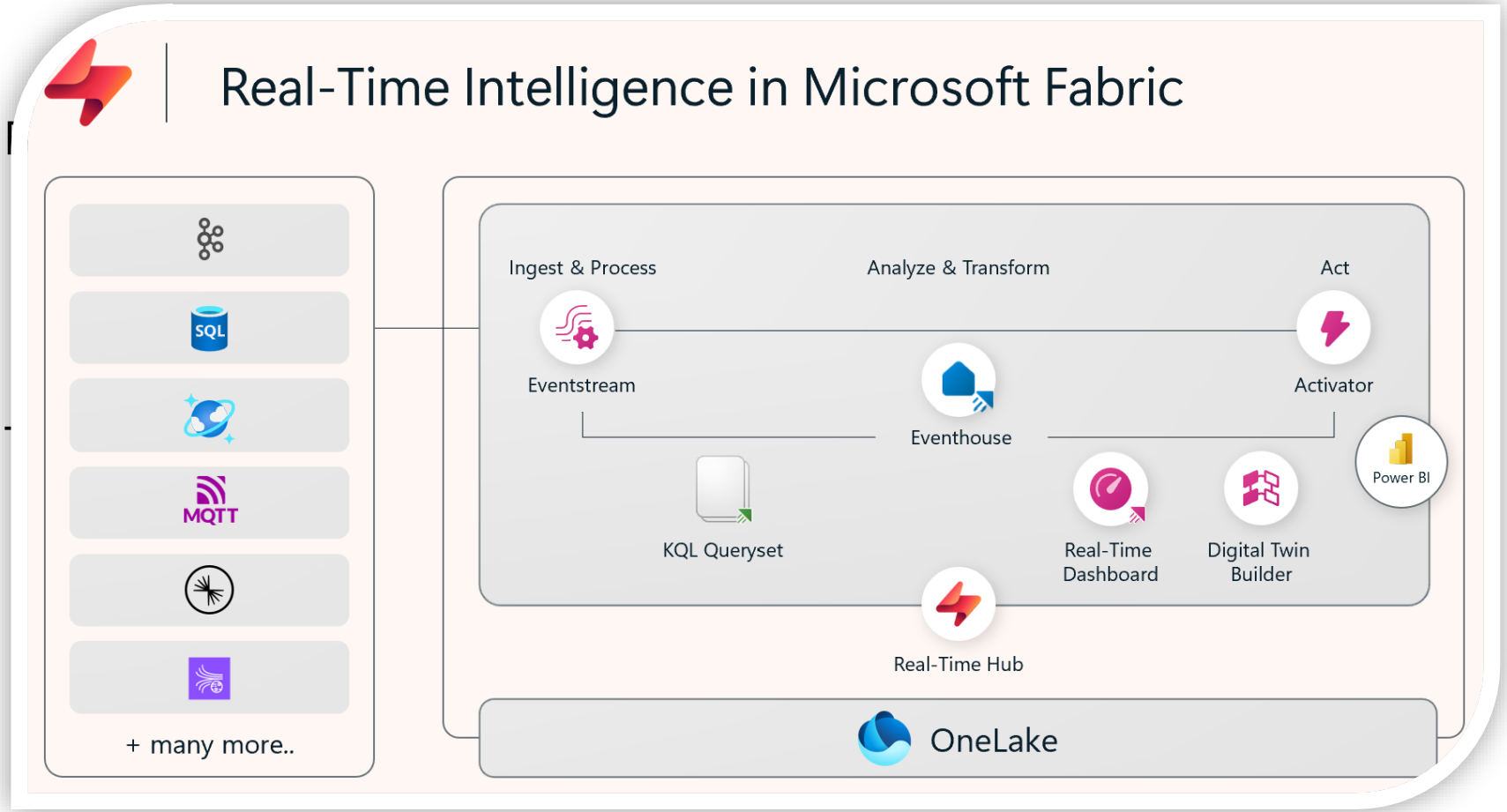
- **Latency** – How quickly must events reach Fabric?
  Seconds? Sub-second? Is a 30-second delay OK?

- **Processing time** – How complex are your transformations?
  Can they run continuously (streaming) or only in batches?

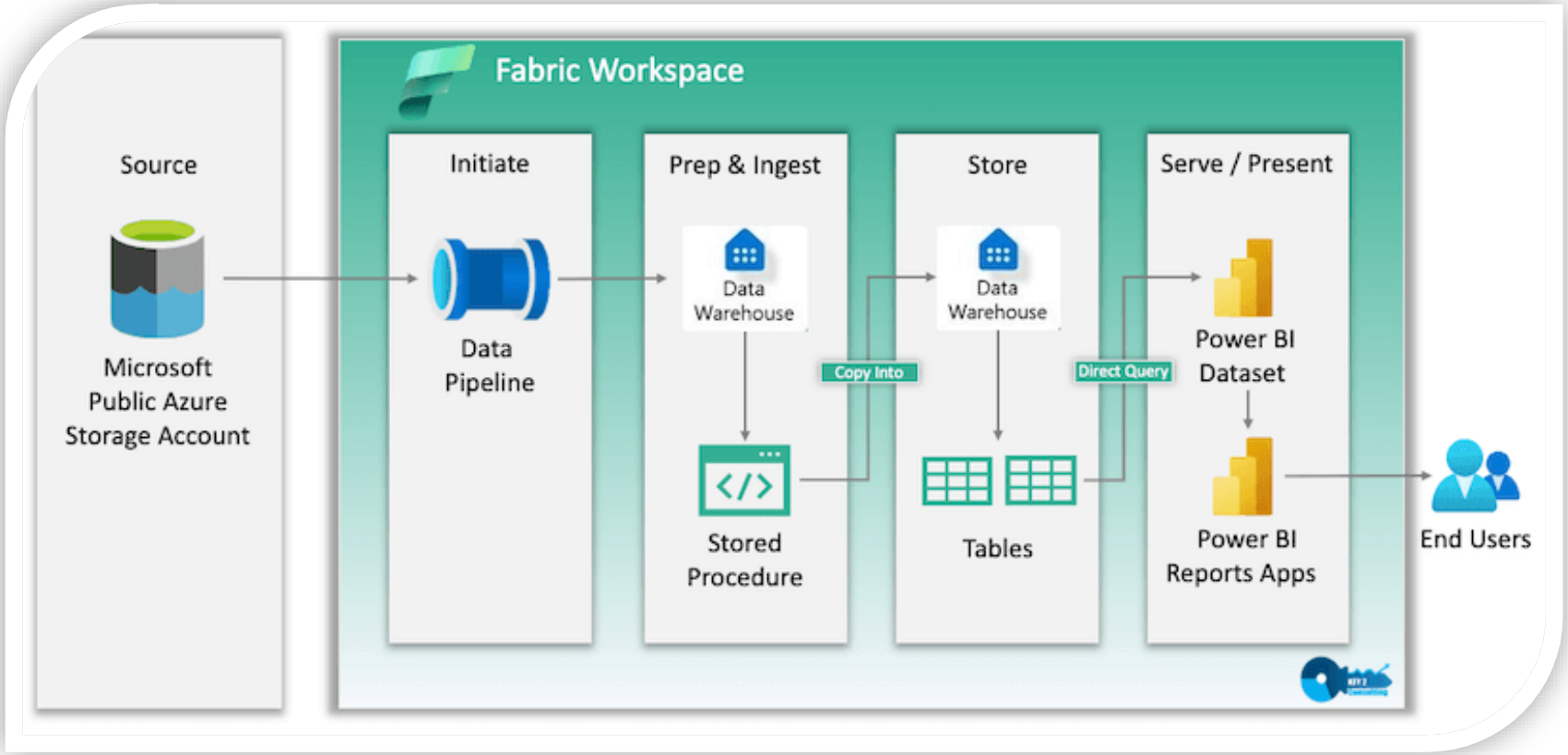
- **Freshness** – How old can the data on your dashboard be?
  “No more than 5 seconds behind live”? “Within 1 minute”?

- Requirement Requirements Requirements

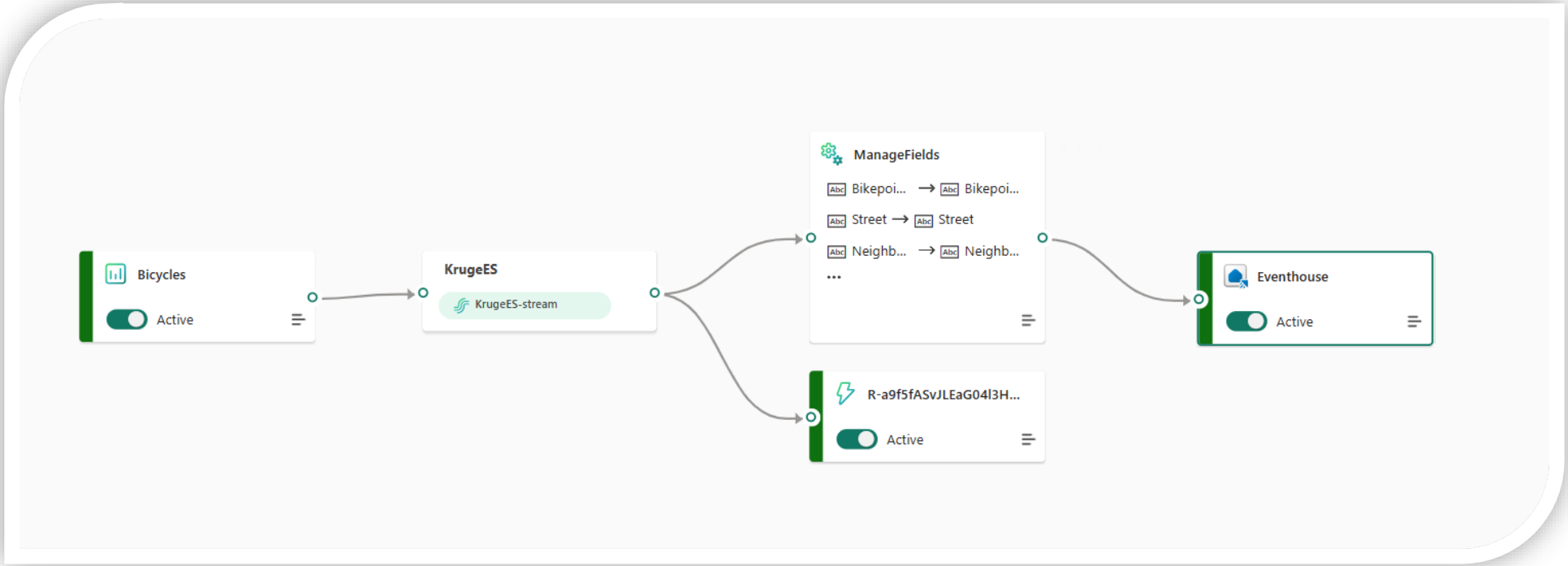
The CORE


Real-Time Intelligence in Microsoft Fabric
SQL
MQTT
+ many more..
Ingest & Process
Analyze & Transform
Act
Eventstream
Eventhouse
Activator
KQL Queryset
Real-Time Dashboard
Digital Twin Builder
Power BI
Real-Time Hub
OneLake

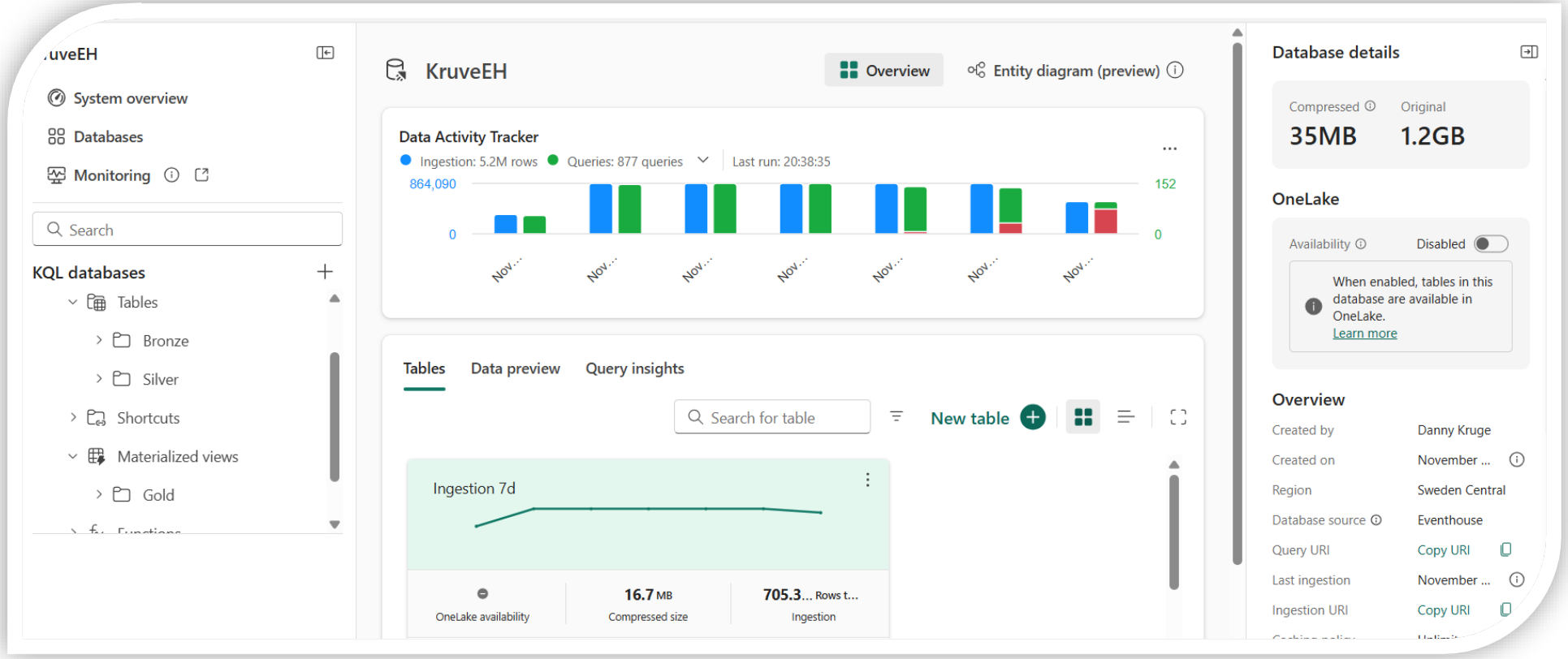
Data Factory Comparison



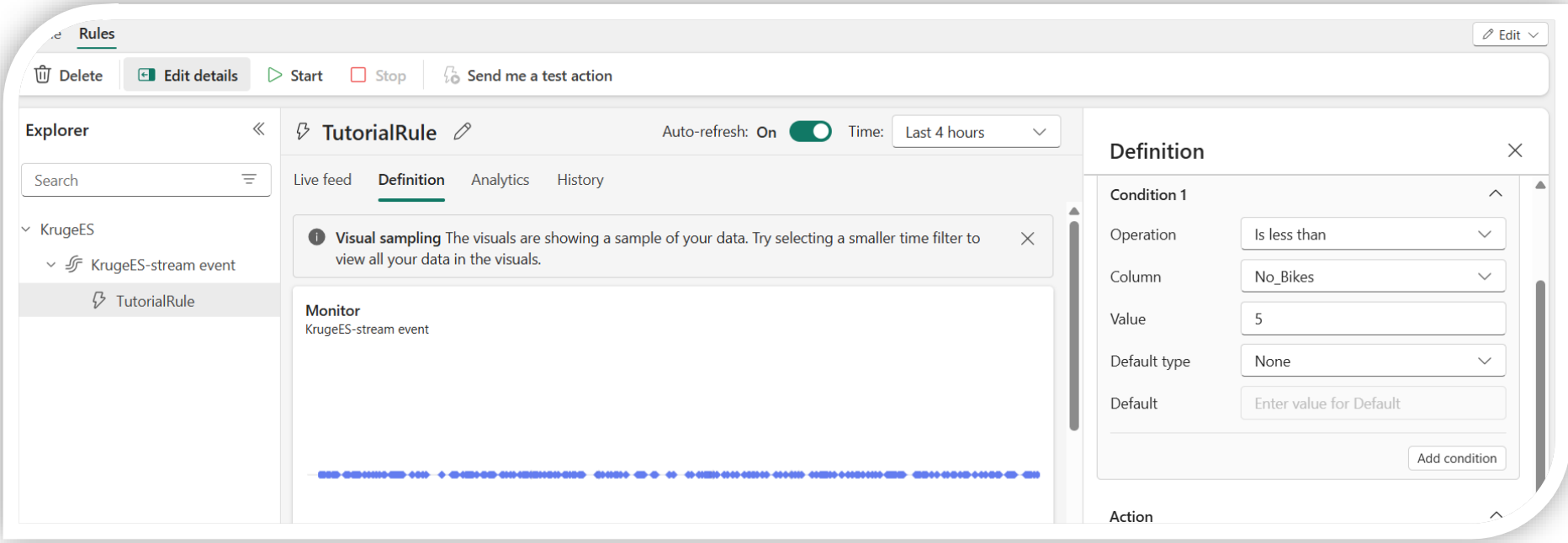
Demo Walkthrough
The Dutch love Bicylces



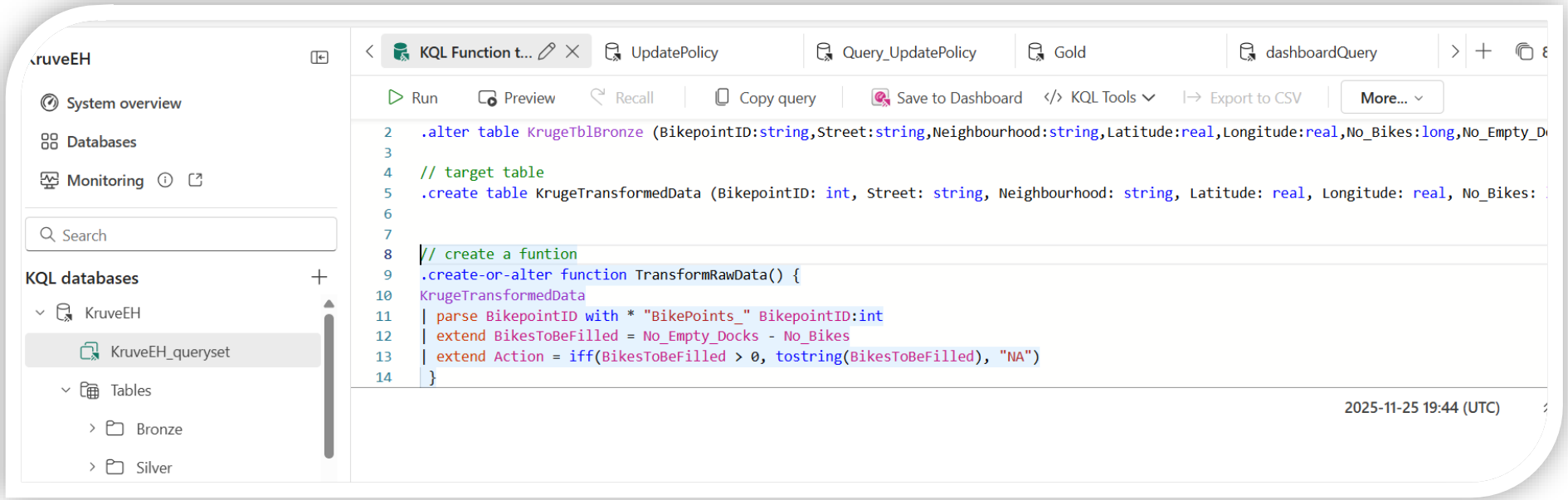
Demo Walkthrough
In the HOUSE



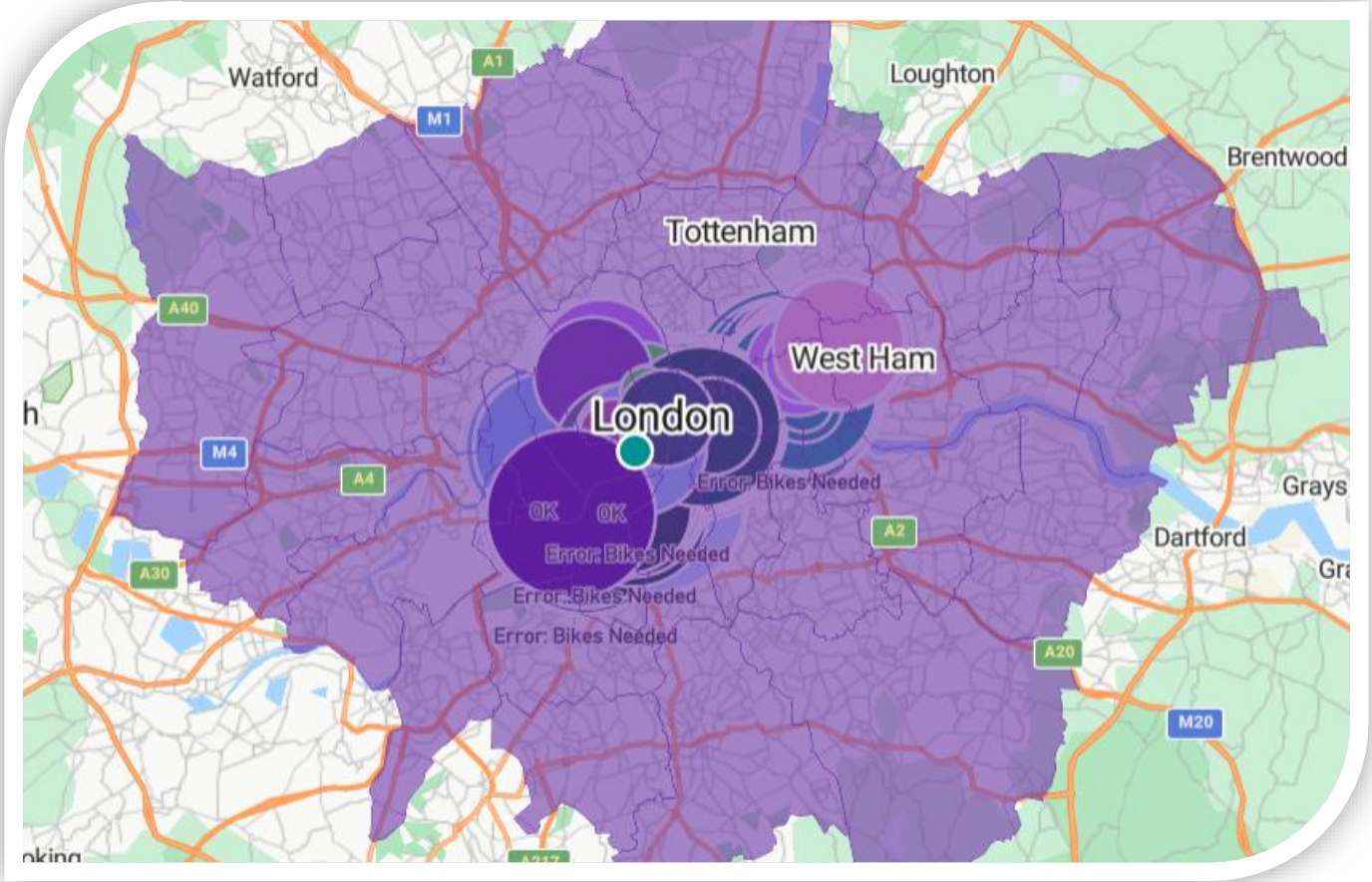
Demo Walkthrough
Rules are Rules



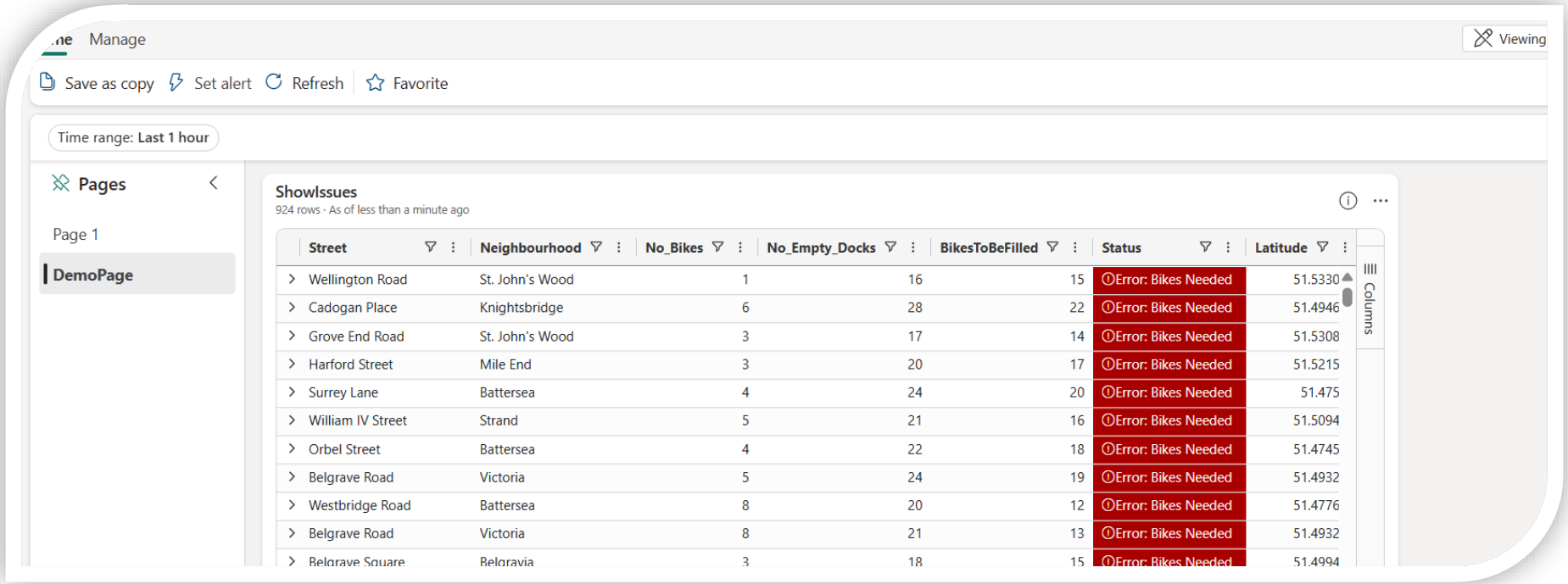
Demo Walkthrough
Gimme that Code BOY



Demo Walkthrough
Map it up Map it in... lemme Begin



Demo Walkthrough
Dash Bored



Street	Neighbourhood	No_Bikes	No_Empty_Docks	BikesToBeFilled	Status	Latitude
Wellington Road	St. John's Wood	1	16	15	Error: Bikes Needed	51.5330
Cadogan Place	Knightsbridge	6	28	22	Error: Bikes Needed	51.4946
Grove End Road	St. John's Wood	3	17	14	Error: Bikes Needed	51.5308
Harford Street	Mile End	3	20	17	Error: Bikes Needed	51.5215
Surrey Lane	Battersea	4	24	20	Error: Bikes Needed	51.475
William IV Street	Strand	5	21	16	Error: Bikes Needed	51.5094
Orbel Street	Battersea	4	22	18	Error: Bikes Needed	51.4745
Belgrave Road	Victoria	5	24	19	Error: Bikes Needed	51.4932
Westbridge Road	Battersea	8	20	12	Error: Bikes Needed	51.4776
Belgrave Road	Victoria	8	21	13	Error: Bikes Needed	51.4932
Belgrave Square	Belgravia	3	18	15	Error: Bikes Needed	51.4994

Stop ..... Its Demo Time



Or Sacrifice an MVP?



Stop ..... Its Demo Time

What was covered Real time?

- Get data in the Real-Time hub
- Transform events
- Use update policies to transform data in Eventhouse
- Create a KQL query
- Create an alert based on a KQL query
- Create a Real-Time dashboard
- Explore data visually in the Real-Time dashboard
- Set up anomaly detection on Eventhouse tables
- Create a map using geospatial data
- Set an alert on the eventstream

https://learn.microsoft.com/en-us/fabric/real-time-intelligence/tutorial-introduction

Stop ..... Its Demo Time



Or Sacrifice an MVP?



Stop ..... Its Demo Time

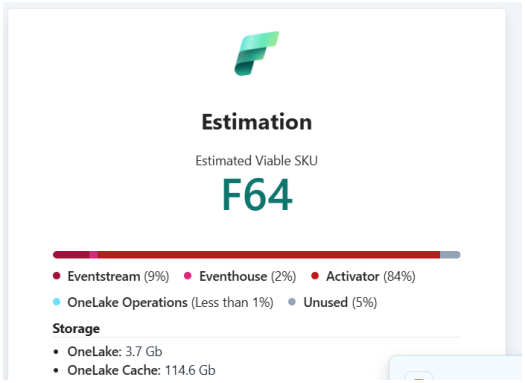
- What was covered Micro Batch / near real time?

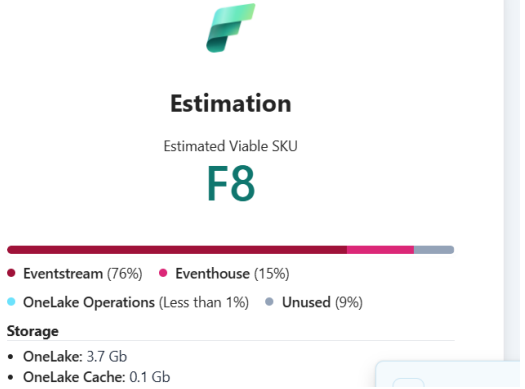
- Copy Activity
- Views
- Semantic Model
- Report

- https://learn.microsoft.com/en-us/fabric/real-time-intelligence/tutorial-introduction

Cost

- https://www.microsoft.com/en-us/microsoft-fabric/capacity-estimator
- Real time

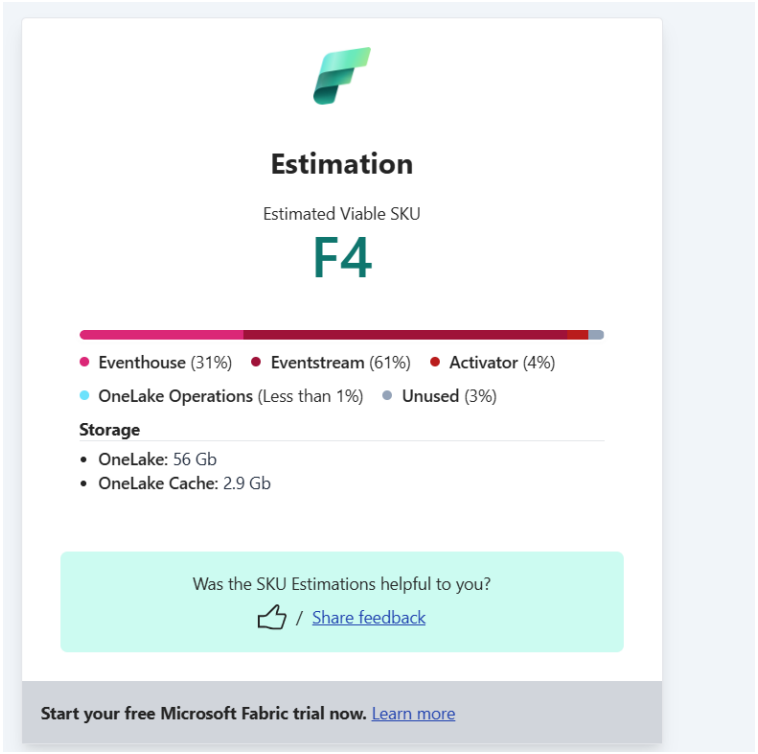




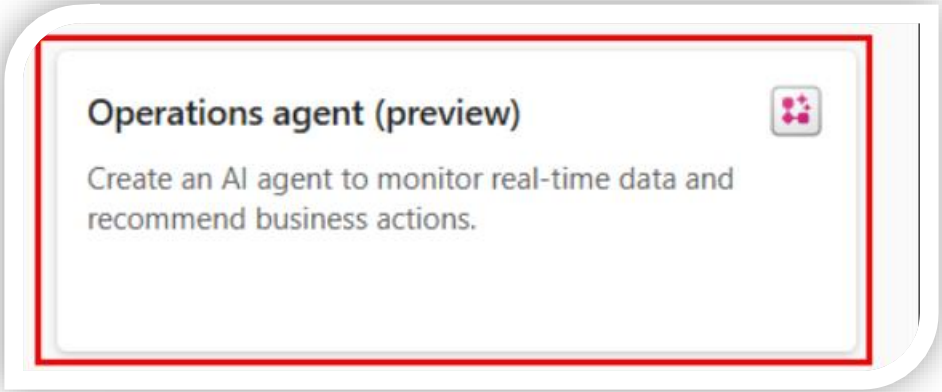
- around **€7.209,877**/Month **vs €901,235**/Month
- Data Factory

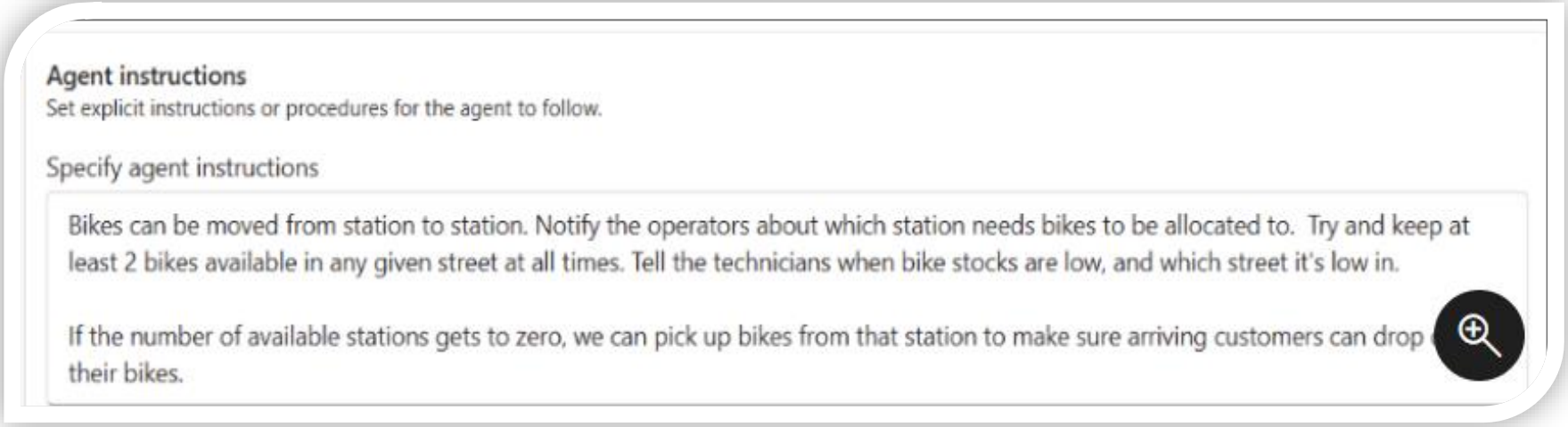
€225,309/Month

Cost a what who?

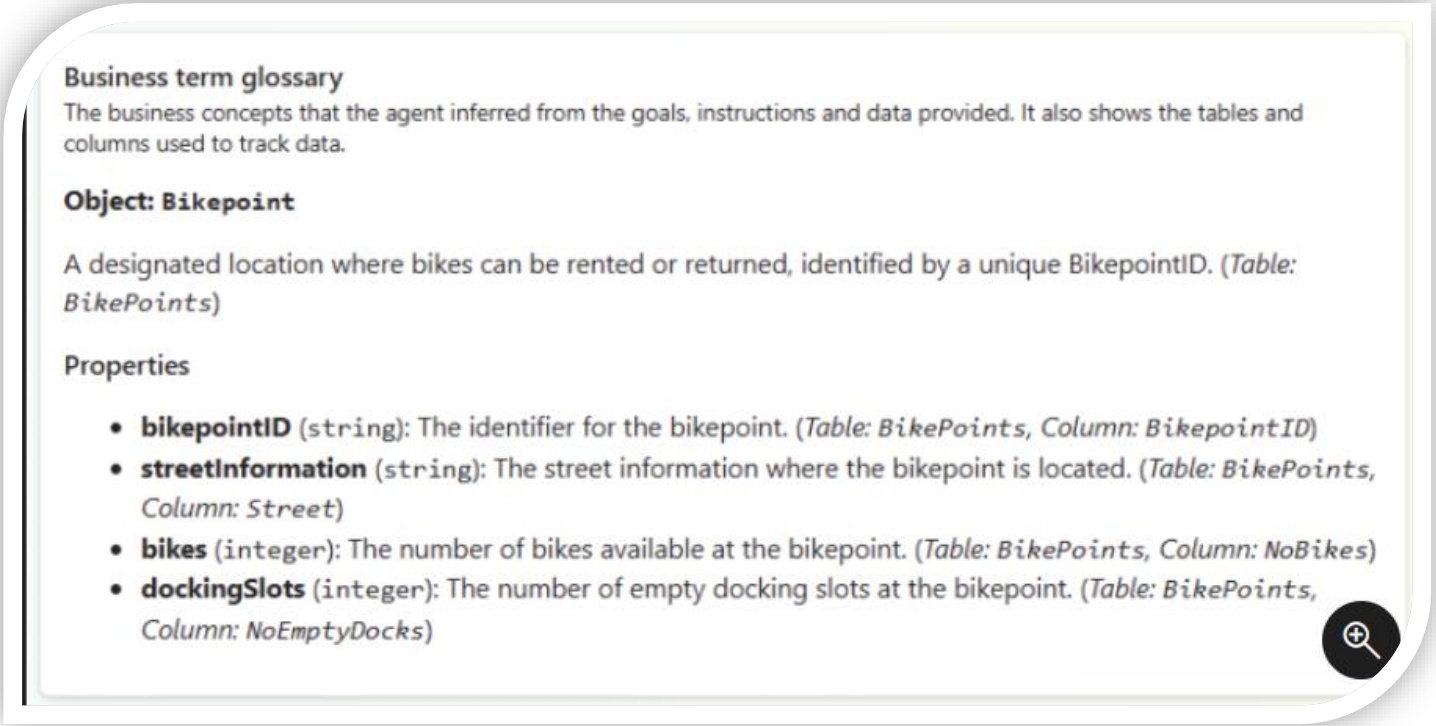


Bonus Slide


Operations agent (preview)
Create an AI agent to monitor real-time data and recommend business actions.


Agent instructions
Set explicit instructions or procedures for the agent to follow.
Specify agent instructions
Bikes can be moved from station to station. Notify the operators about which station needs bikes to be allocated to. Try and keep at least 2 bikes available in any given street at all times. Tell the technicians when bike stocks are low, and which street it's low in.
If the number of available stations gets to zero, we can pick up bikes from that station to make sure arriving customers can drop their bikes.

Bonus Slide



Business term glossary

The business concepts that the agent inferred from the goals, instructions and data provided. It also shows the tables and columns used to track data.

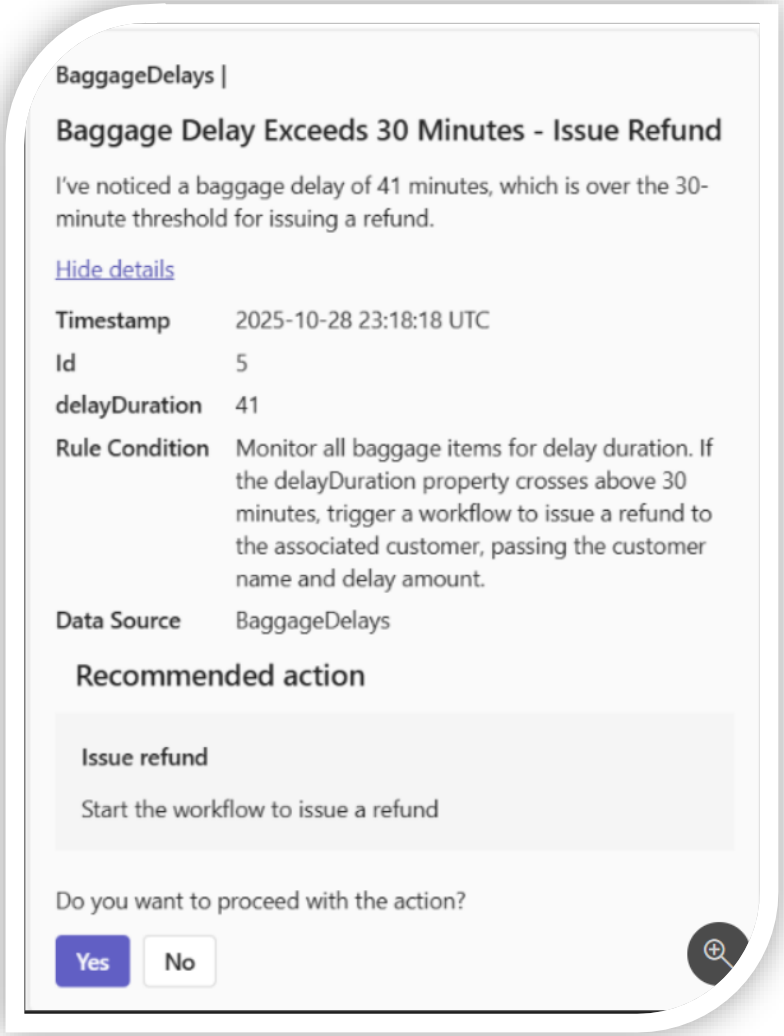
Object: `Bikepoint`

A designated location where bikes can be rented or returned, identified by a unique BikepointID. (*Table: `BikePoints`*)

Properties

- **bikepointID** (`string`): The identifier for the bikepoint. (*Table: `BikePoints`, Column: `BikepointID`*)
- **streetInformation** (`string`): The street information where the bikepoint is located. (*Table: `BikePoints`, Column: `Street`*)
- **bikes** (`integer`): The number of bikes available at the bikepoint. (*Table: `BikePoints`, Column: `NoBikes`*)
- **dockingSlots** (`integer`): The number of empty docking slots at the bikepoint. (*Table: `BikePoints`, Column: `NoEmptyDocks`*)

Bonus Slide


BaggageDelays |
Baggage Delay Exceeds 30 Minutes - Issue Refund
I've noticed a baggage delay of 41 minutes, which is over the 30-minute threshold for issuing a refund.
Hide details
Timestamp 2025-10-28 23:18:18 UTC
Id 5
delayDuration 41
Rule Condition Monitor all baggage items for delay duration. If the delayDuration property crosses above 30 minutes, trigger a workflow to issue a refund to the associated customer, passing the customer name and delay amount.
Data Source BaggageDelays
Recommended action
Issue refund
Start the workflow to issue a refund
Do you want to proceed with the action?
Yes
No



Wrap up

Choosing the Right “Real-Time” for Your Business

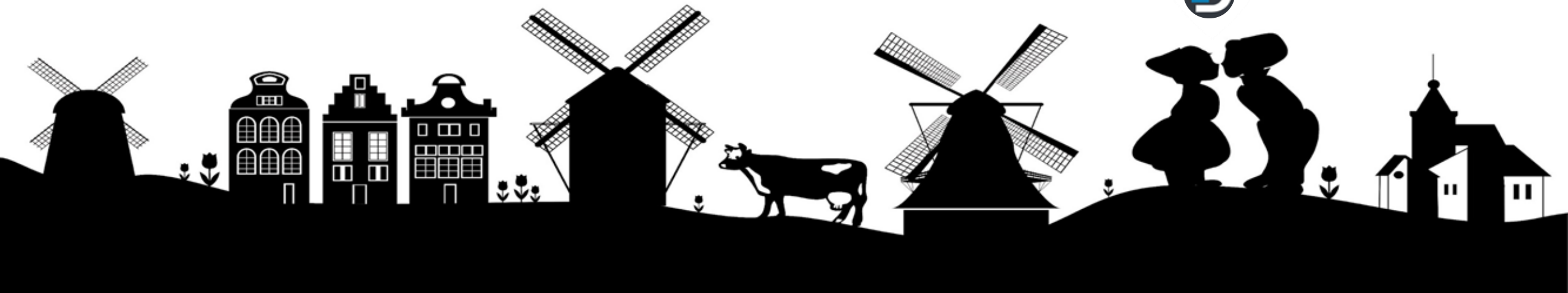
Choose your “real enough”

Questions you should ask

1. “How quickly do you *need* to see changes after they happen?”
2. “What decision or action depends on this data, and what happens if it’s late?”
3. “Who uses this data and how often do they actually look at it?”
4. “What is the value of having it faster, and who owns the budget?”
5. “What is the maximum acceptable delay you can live with?”

If you did not like the session
my name is **Tonie Huizer**

Is This Just Real-Time… or Is It Just Fabric?





Danny Kruge

Data Platform MVP 💪Mission Critical Engineer at Schuberg Philis 👩‍🚒 Azure Hero...



