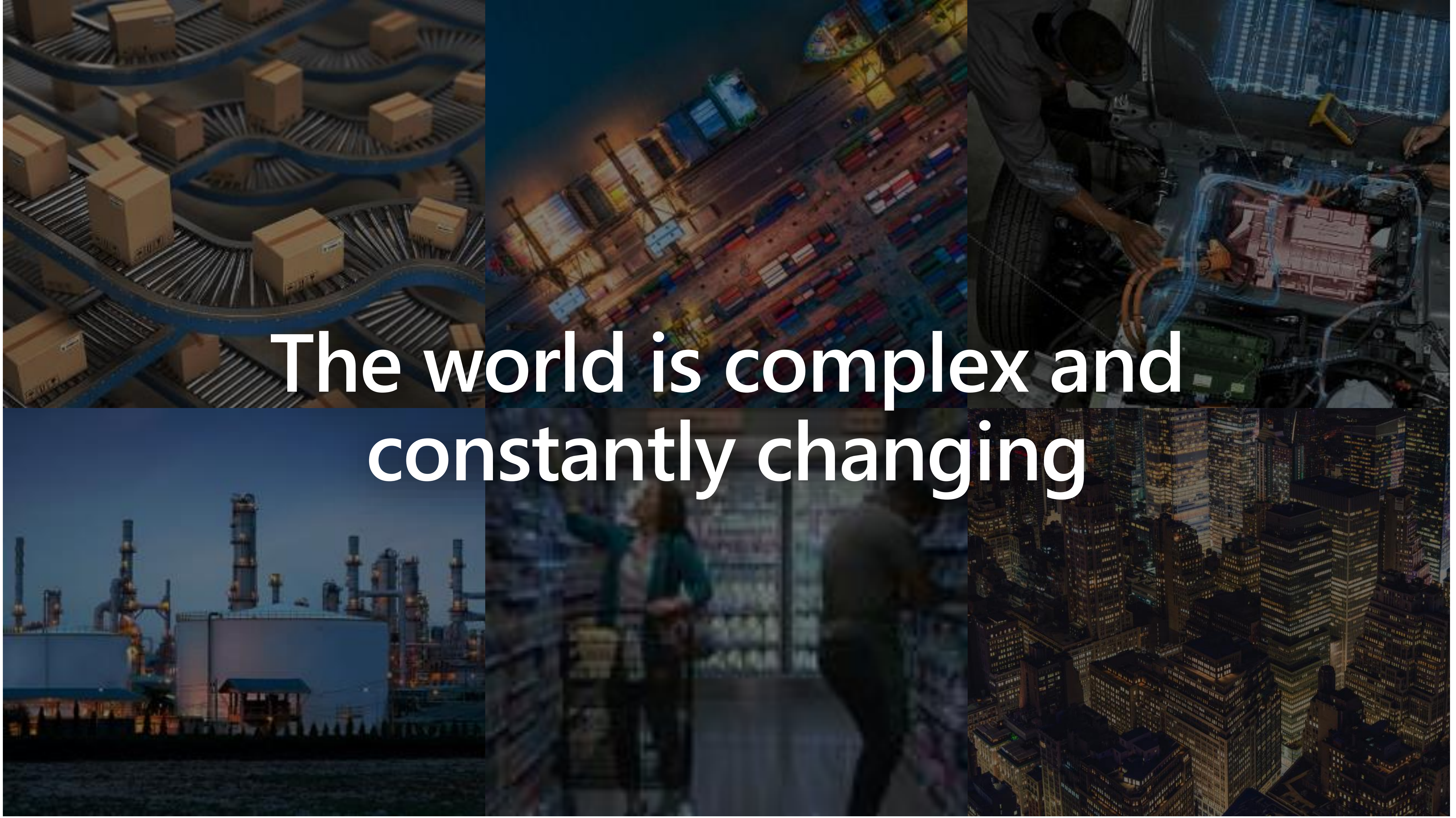
MTG

Optimisez vos architectures event-driven avec Drasi

— Christophe Gigax
Cloud Solution Architect @ Hager Group
MVP Microsoft

christophegigax.bsky.social



The world is complex and constantly changing

Change-driven scenarios are everywhere

Inventory & Supply Chain Management

System Health & Anomaly Detection

IoT and Smart Devices Alerting

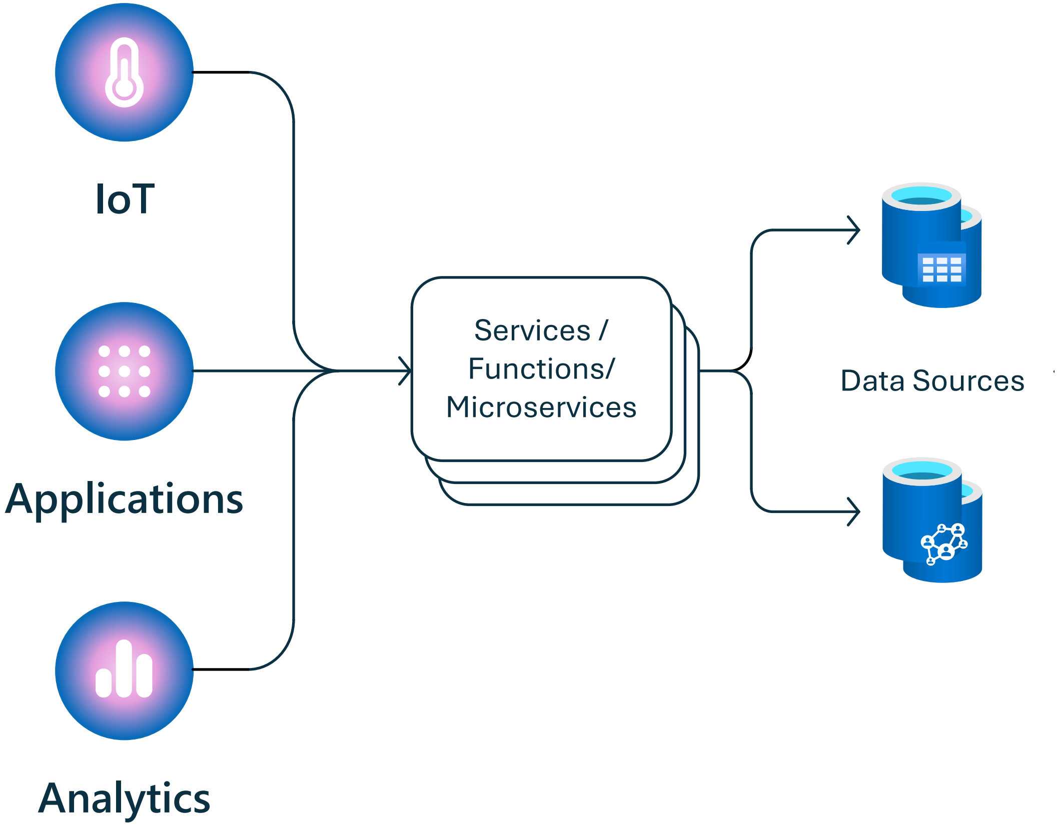
Business Ops

Compliance & Policy Enforcement

Streaming Service Quality with CDN

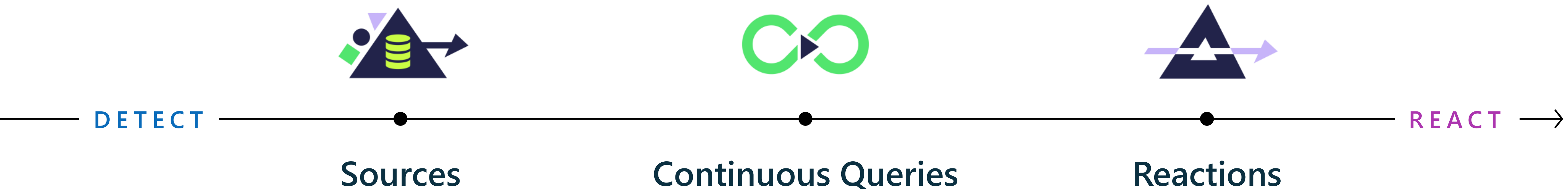
How do you detect change today?

Change-detection solutions today



What if you could detect *only* the changes you care about?

drasi



DETECT

Sources

Continuous Queries

Reactions

REACT

Open-source platform for **real-time data change** processing that turns **data updates** into **instant insights** and **actions**.

Azure Incubations

MISSION

Partner across Microsoft and the open-source community to explore and deliver industry-changing products

KEDA

GRADUATED

Aug 2023

dapr

GRADUATED

Nov 2024

drasi

SANDBOX

Jan 2025

copa

SANDBOX

Sep 2023

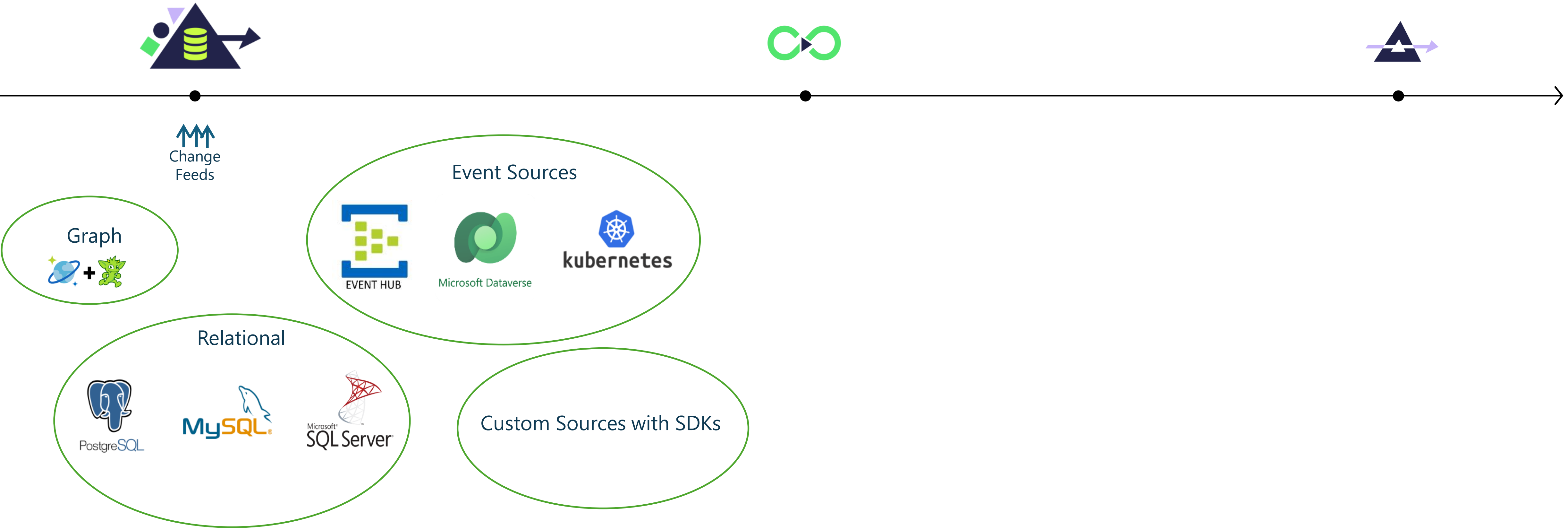
radius

SANDBOX

Apr 2024

CLOUD NATIVE
COMPUTING FOUNDATION

Sources



Change Feeds

Graph

Event Sources

Relational

Custom Sources with SDKs

drasi

Continuous Queries

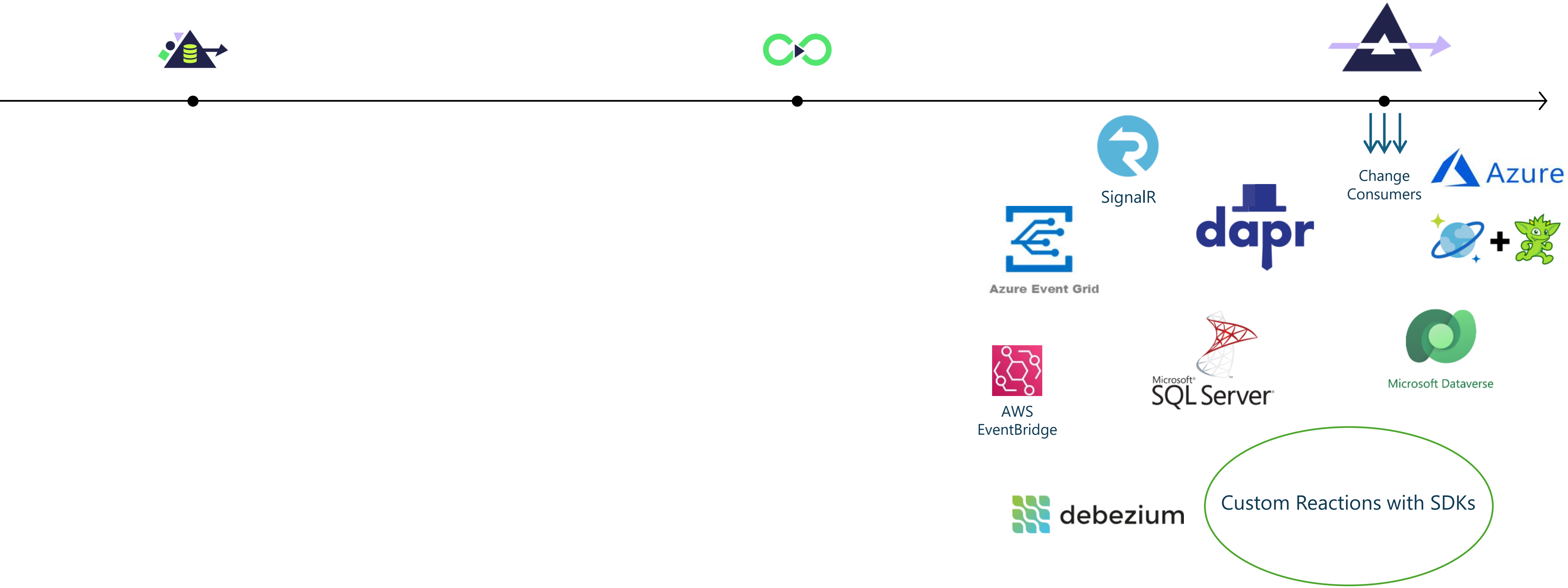
```
MATCH
 (o:orders)-[:PICKUP_BY]->(v:vehicles)
WHERE
o.status = 'ready'
AND v.location = 'Curbside’
```

```
RETURN
o.id AS orderId,
o.status AS orderStatus,
o.driver_name AS driverName,
v.location as vehicleLocation
```

OpenCypher Query Language
Or
Graph Query language (GQL)

Sample: Match orders that are ready with pickup drivers waiting curbside to deliver (OpenCypher)

Reactions



SignalR

Azure Event Grid

dapr

Change Consumers

Azure

AWS EventBridge

Microsoft SQL Server

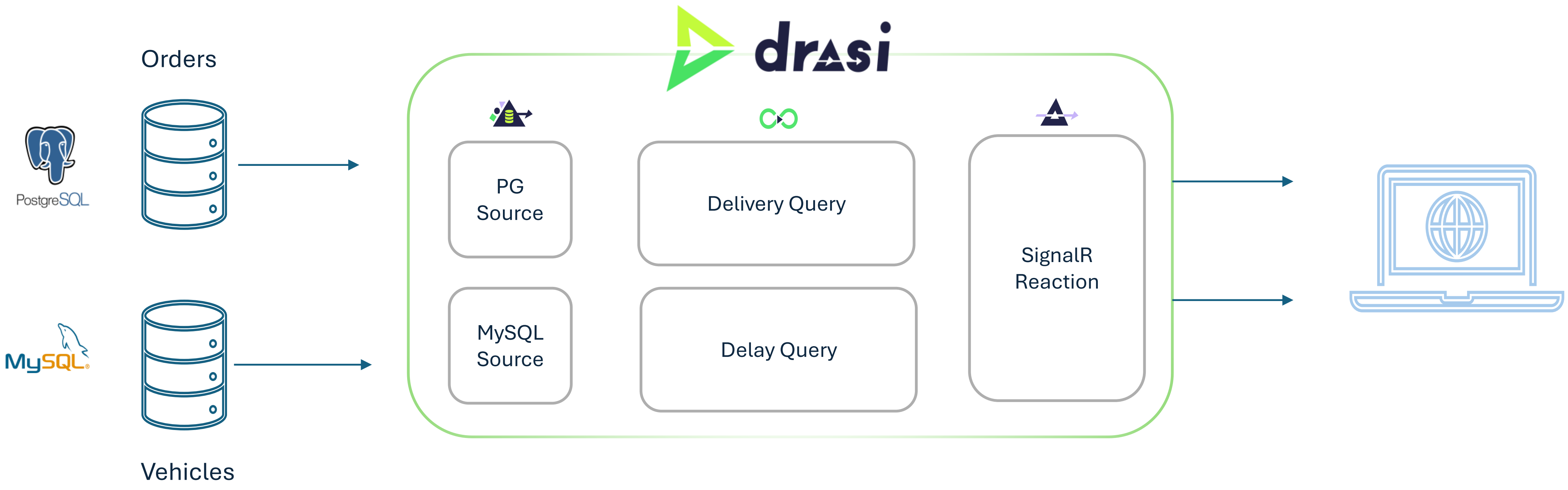
Microsoft Dataverse

debezium

Custom Reactions with SDKs

drasi

Curbside Pickup



https://drasi.io/tutorials/curbside-pickup/

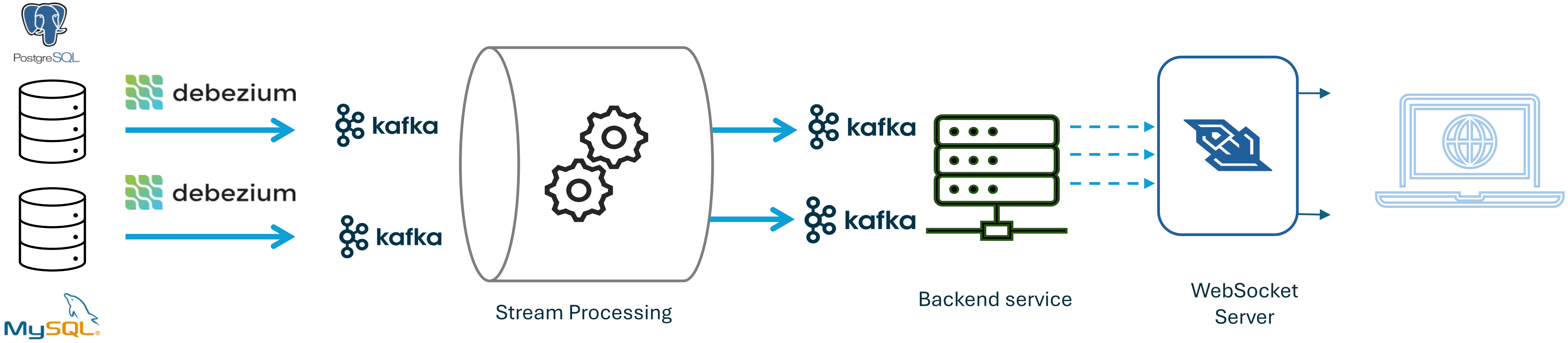
Demo

drasi

Sources, Reactions, and Continuous Queries

Detecting change with event-driven solutions

Scenario: App for curbside pickup of orders
Solution: Interactive dashboards for order status and pickup vehicle location



Stream Processing

Backend service

WebSocket
Server

Why Drasi

- ✓ Reduces engineering and operational overhead and complexity
- ✓ Precise, scalable, and declarative change detection
- ✓ Standardized framework for reacting to critical data changes

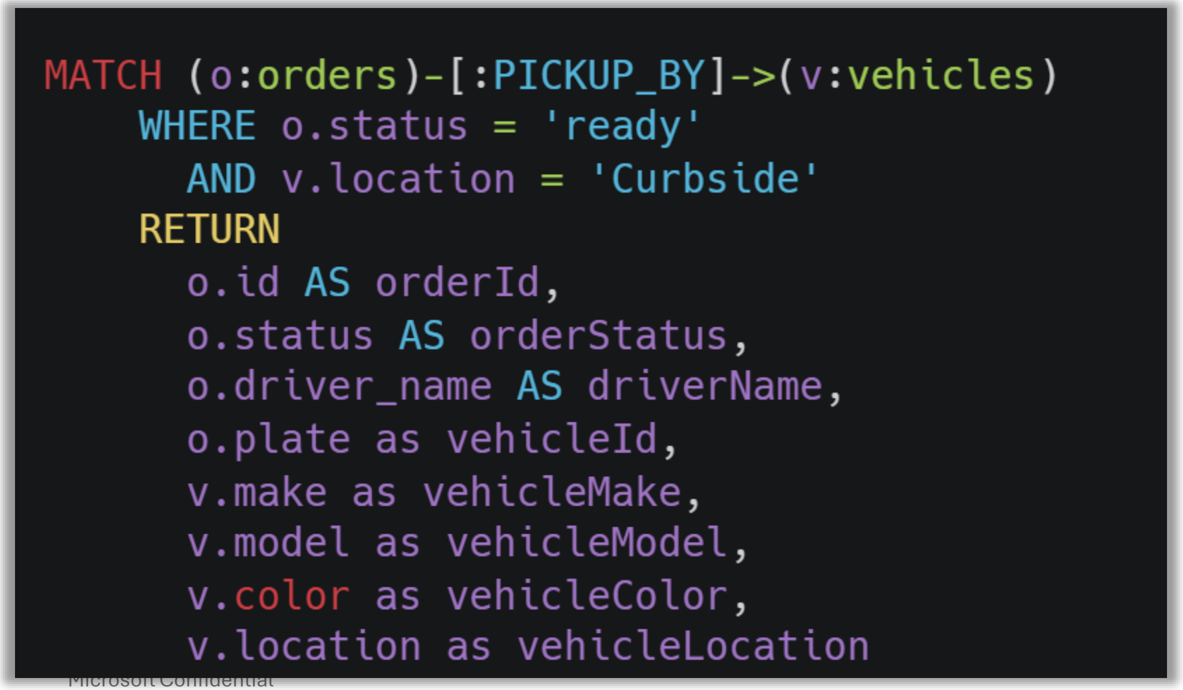
- x Resource waste querying unchanged data or building analytics systems
- x Complex logic to detect, parse, and filter generic events
- x Customized solutions to detect and react to events

Cypher vs GQL

🌐 **GQL** is the **ISO standard graph query language** unifying Cypher, PGQL, and GSQL.

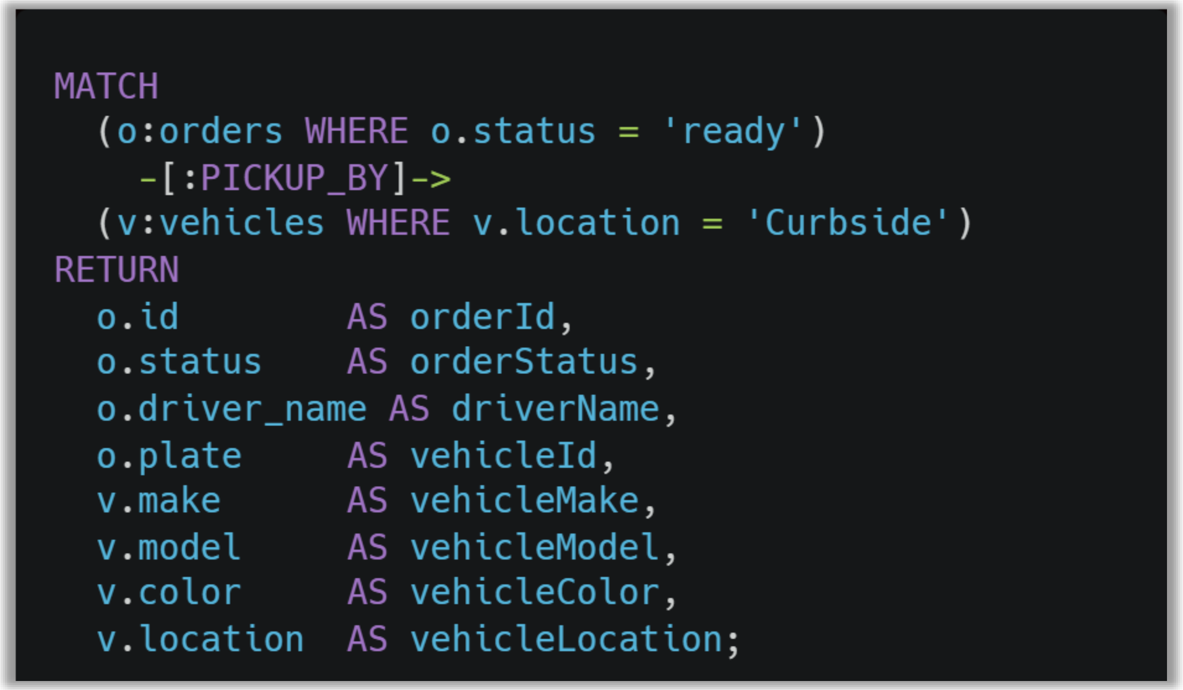
⚡ It enables **declarative creation and querying of graphs** with cross-platform interoperability.

Cypher



```
MATCH (o:orders)-[:PICKUP_BY]->(v:vehicles)
    WHERE o.status = 'ready'
      AND v.location = 'Curbside'
    RETURN
      o.id AS orderId,
      o.status AS orderStatus,
      o.driver_name AS driverName,
      o.plate as vehicleId,
      v.make as vehicleMake,
      v.model as vehicleModel,
      v.color as vehicleColor,
      v.location as vehicleLocation
```

GQL



```
MATCH
  (o:orders WHERE o.status = 'ready')
    -[:PICKUP_BY]->
  (v:vehicles WHERE v.location = 'Curbside')
RETURN
  o.id          AS orderId,
  o.status      AS orderStatus,
  o.driver_name AS driverName,
  o.plate       AS vehicleId,
  v.make        AS vehicleMake,
  v.model       AS vehicleModel,
  v.color       AS vehicleColor,
  v.location    AS vehicleLocation;
```

Installing Drasi

Installations Docs: **https://drasi.io/how-to-guides/installation/**

Install Drasi on Kubernetes

1. Install Kubernetes
2. Install drasi CLI
3. Run **drasi init**

Install Drasi in Docker

1. Install Docker
2. Install drasi CLI
3. Run **drasi init --docker**

Questions or feedbacks ?

Feedbacks

hager group

https://careers.hagergroup.com/

MTG Strasbourg

mtgstrasbourg

@MTG_Strasbourg

MTG Strasbourg