



**M-Files**®

*The Smarter Way to Work.*

The background features a dark blue diagonal shape on the left and a white background on the right. On the right side, there are two overlapping, tilted rectangular shapes: a light blue one on top and a teal one on the bottom.

# **BUILDING M-FILES OBSERVABILITY PLATFORM WITH MICROSOFT FABRIC REAL-TIME INTELLIGENCE**

Jari Turkia  
Senior Cloud Architect  
17<sup>th</sup> Oct 2024

- Intro
- The Setup
- Observability
  - Demo
- The Build
  - With Azure
  - With Fabric
  - Demo
- The Punchline
  - We got this far ...
  - Future

# WHO AM I AND WHAT I DO

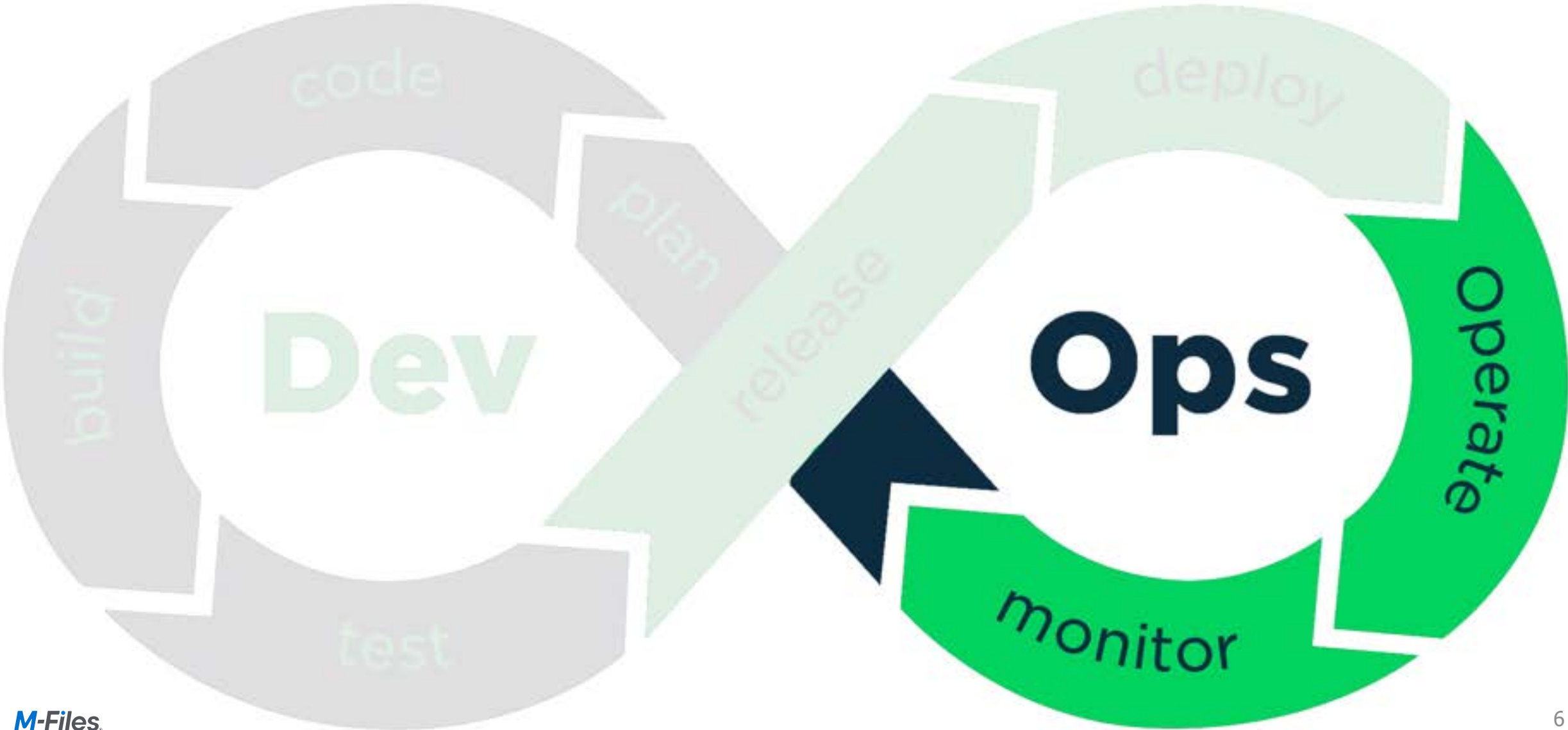
- At work:
  - "*Jack of all trades – master of none*"
  - Fullest stack architect you've seen for a while
  - Software: Design / implement / run in production
  - Data
- At home:
  - Nerd / geek / coder
    - Open-source aficionado
    - Hacker
    - Blogger
  - Fixer-upper
  - Wrangler of many things



# SETTING UP THE SCENE

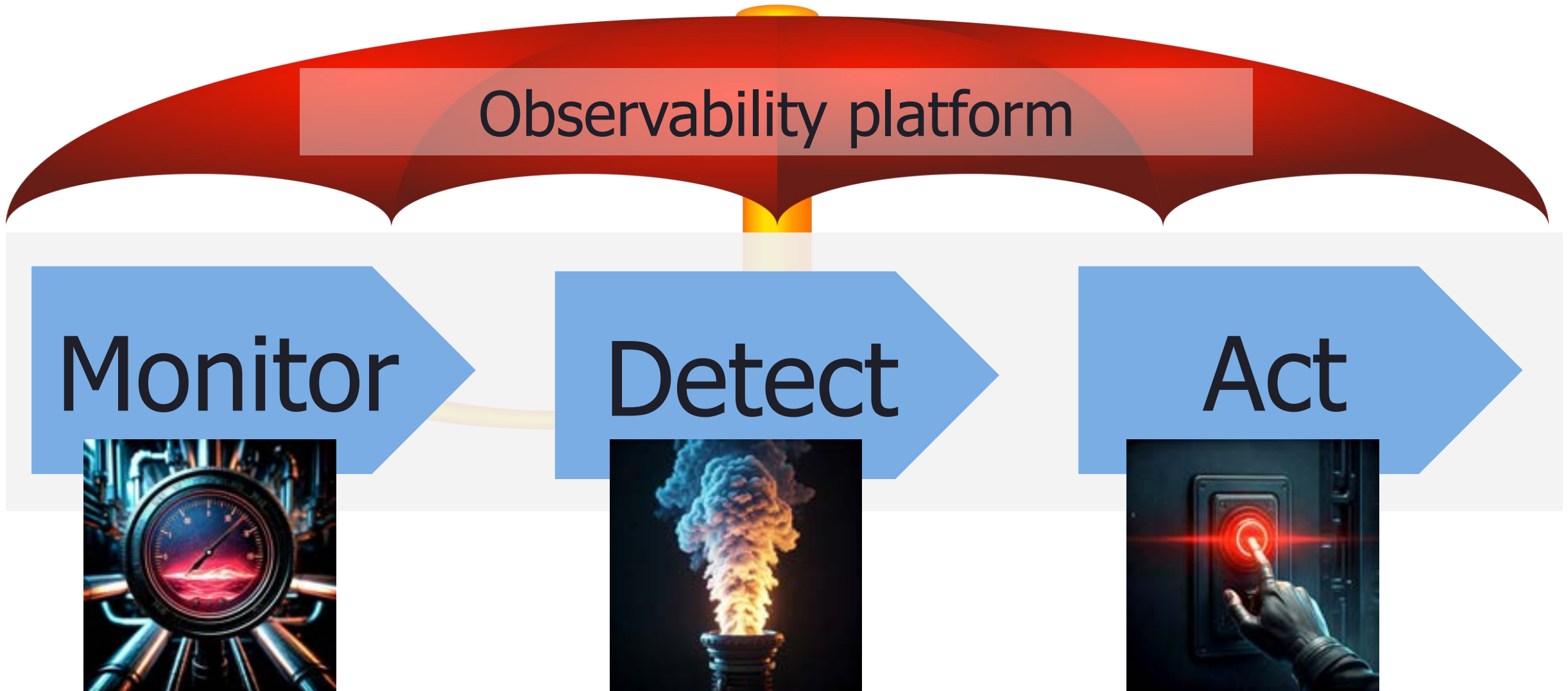
About observability

# IN FIELD OF SOFTWARE DEVELOPMENT, WHERE IS OBSERVABILITY LOCATED AT



# WHAT IS AN OBSERVABILITY PLATFORM?

# PRINCIPAL TASKS OF AN OBSERVABILITY PLATFORM



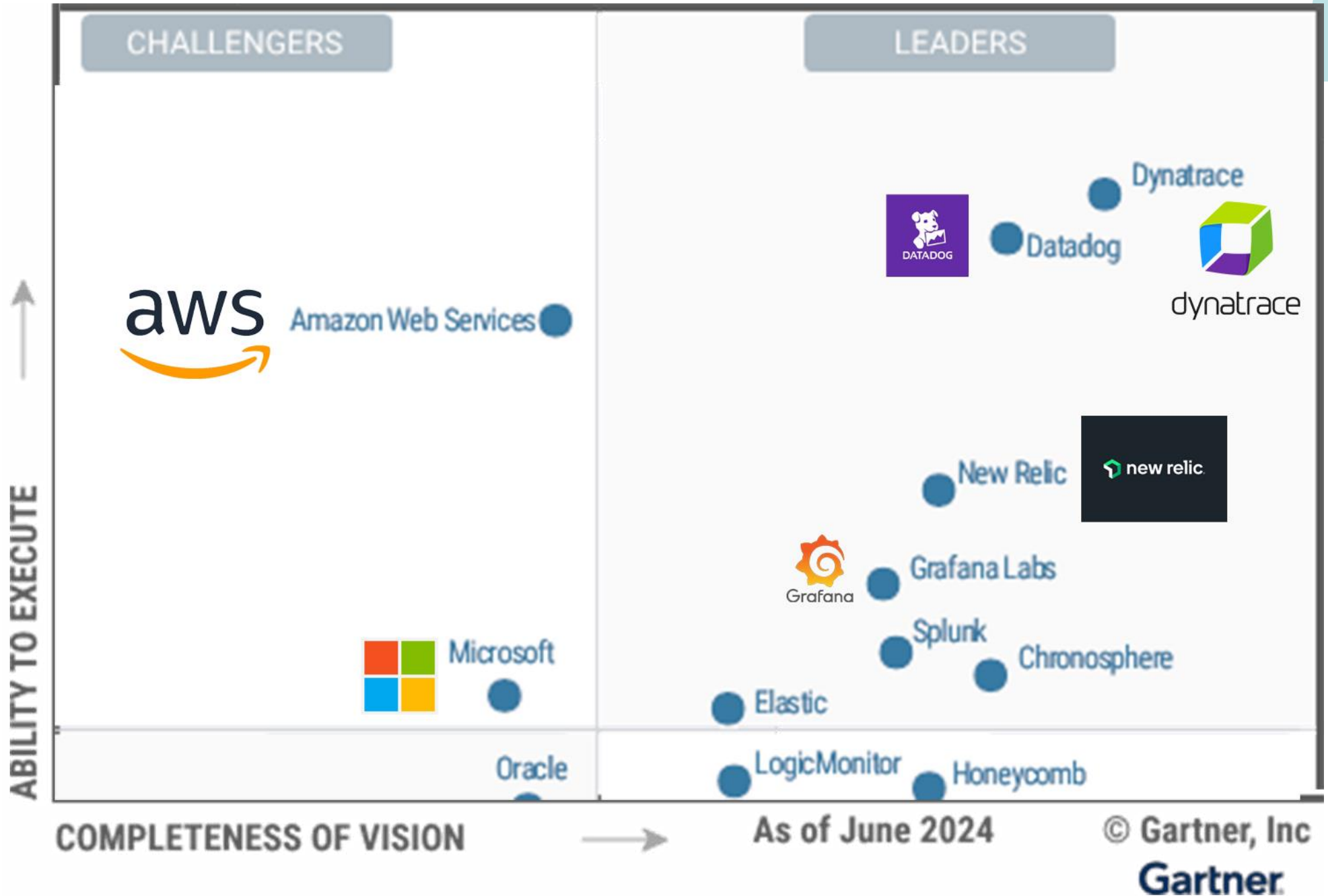


# MAJOR PLAYERS

[Magic Quadrant for Observability Platforms](#)

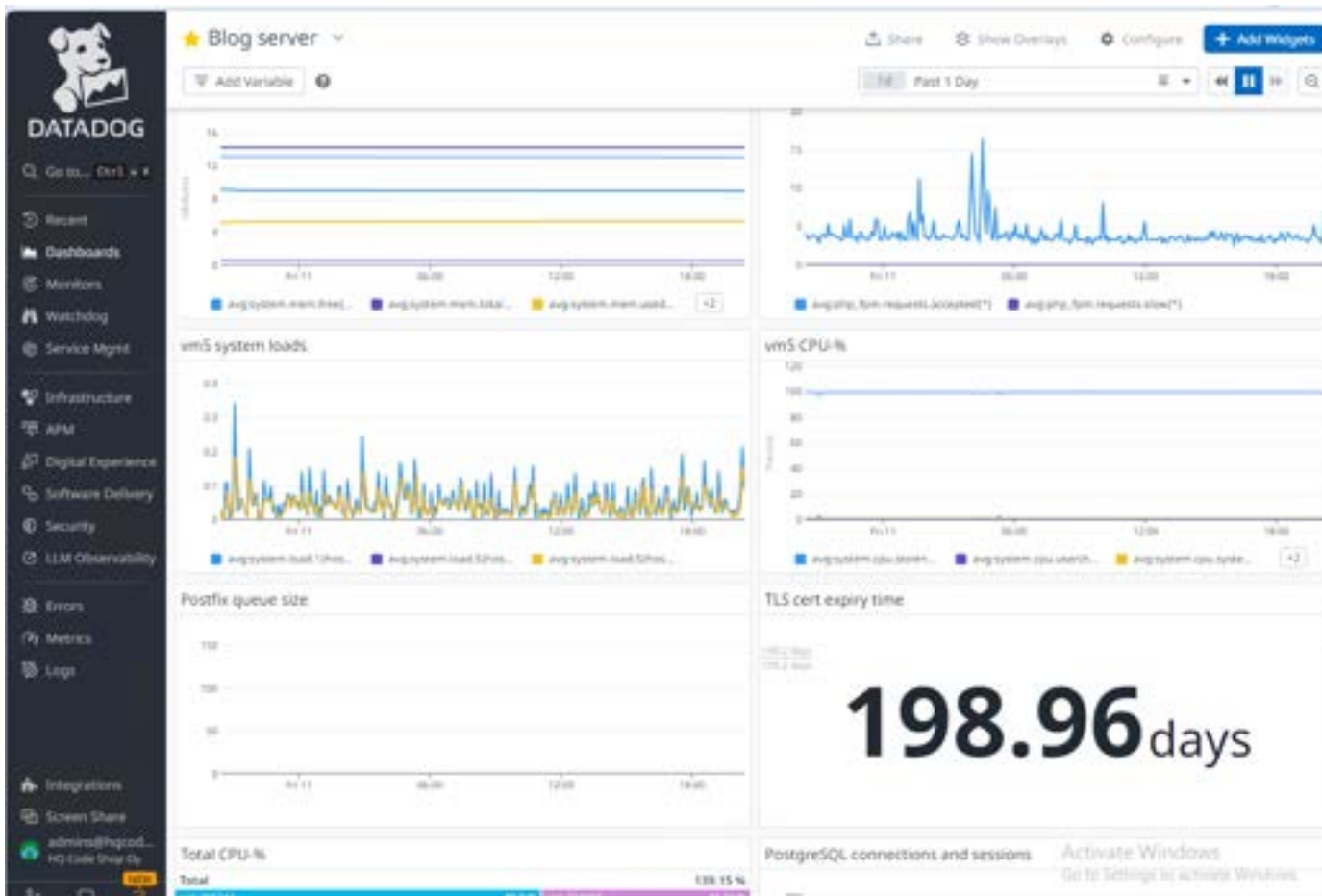
12 August 2024

M-Files



# DEMO – TAKING A LOOKSIE ON AN OBSERVABILITY PLATFORM WEB GUI

- A randomly selected SaaS product:



# BUILD OR PURCHASE



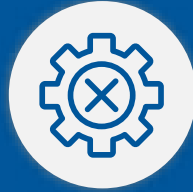
VS





## INFLEXIBILITY

The old monitoring system was not adaptable to changing needs



## UNDERUTILIZATION

Pre-built features did not align with specific organizational needs, leading to their underutilization



## MANUAL PROCESSES

Many monitoring tasks, including scaling, were manual



## COST

The monitoring tool was expensive without justifying its value

# THE BUILD

# DEVELOPMENT OBJECTIVES



**SCALABILITY**



**CONFIGURABILITY**



**EASY ACCESS  
TO DATA**



**REAL TIME  
INSIGHTS INTO  
IMMEDIATE  
ACTIONS**

# THE BUILD – COLLECTING TELEMETRY

# OPEN TELEMETRY COLLECTOR



- Aka. OTelCol
- <https://opentelemetry.io/docs/collector/>
- <https://github.com/open-telemetry/opentelemetry-collector-releases/>

The screenshot shows the OpenTelemetry Collector documentation page. The header includes the OpenTelemetry logo and navigation links for Docs, Ecosystem, Status, Community, and Blog. A left sidebar lists various documentation topics under 'Docs', including 'What is OpenTelemetry?', 'Getting Started', 'Concepts', 'Demo', 'Language APIs & SDKs', 'Zero-code Instrumentation', and 'Collector'. The main content area is titled 'Collector' and describes it as a 'Vendor-agnostic way to receive, process and export telemetry data.' Below the text is a diagram of the collector architecture, showing data flow from 'Sources' through 'Batch' and 'Process' components to 'Sinks'. The text 'OTEL COLLECTOR' is prominently displayed at the bottom of the diagram.

The screenshot shows the GitHub repository page for 'open-telemetry/opentelemetry-collector-releases'. The repository name is at the top, along with navigation links for Product, Solutions, Resources, Open Source, Ecosystem, and Pricing. Below the repository name, there are tabs for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. A table of releases is displayed, listing various distributions like 'github', 'linux', 'darwin', 'windows', 'darwin-amd64', 'darwin-arm64', 'linux-amd64', 'linux-arm64', 'linux-ppc64le', 'linux-s390x', 'windows-amd64', and 'windows-arm64'. Each row includes a description of the release and its date. On the right side, there are sections for 'About' (Official Release), 'Downloads', and 'Releases'.



# THE BUILD WITH AZURE SERVICES

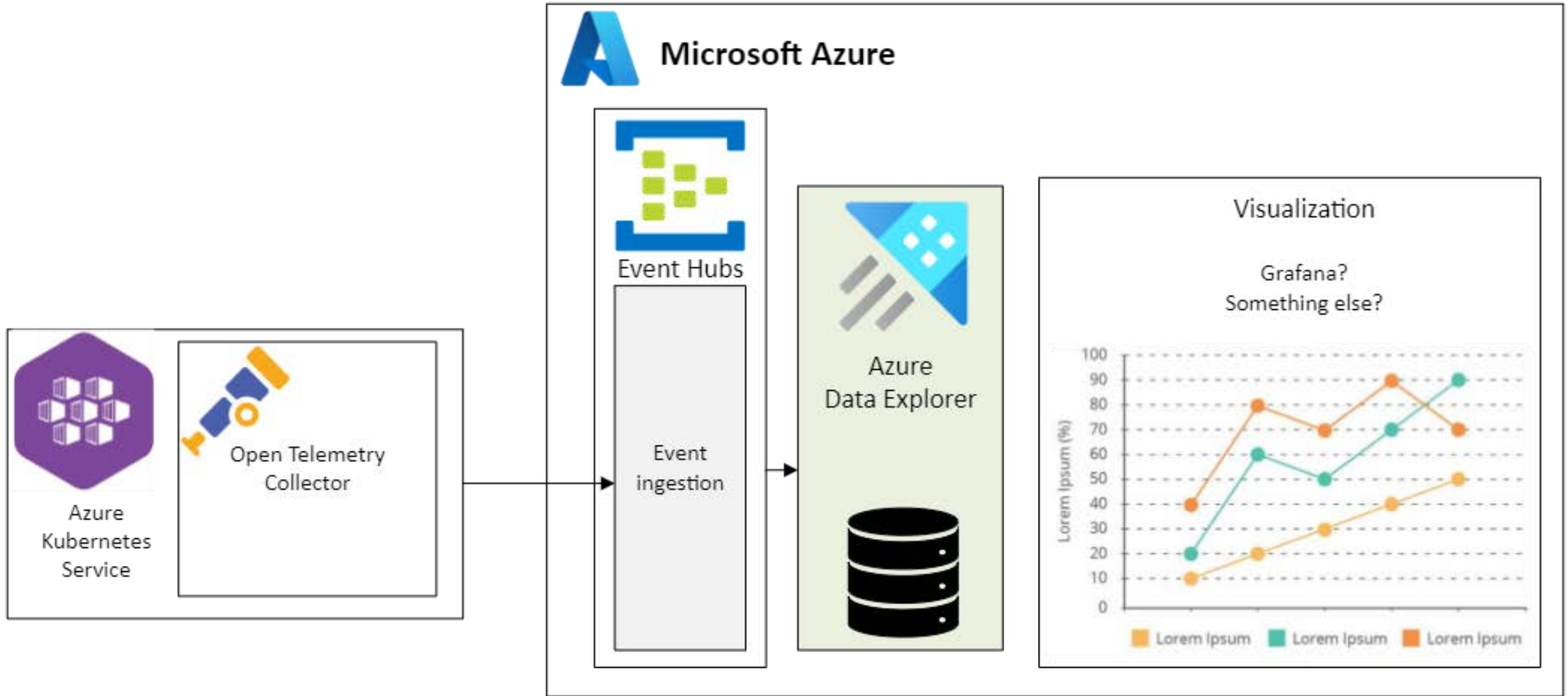
# AZURE SERVICES FOR MONITORING AND ALERTING



- Azure Monitor
  - Monitor Workspaces
  - Linked Grafana Workspaces
- Azure Monitor pipelines (preview)
- Azure Application/Container/VM/Network Insights
- Azure Log Analytics
  - Integral part of Insights
- Azure Native New Relic Service
  - Managed New Relic, integrates easily, single bill
- Azure AI Metrics Advisor
  - Monitor to monitor the monitors
- Azure IoT Hub
  - Analyze with Stream Analytics
  - Visualize with PowerBI / Grafana

- Ingesting Kafka-style
  - Event Hubs (Kafka v1.1)
  - Event Grid (HTTP / MQTT)
  - Service Bus
- Applicable Storage
  - Azure Data Explorer
  - Azure Synapse Analytics
- Analytics
  - Azure Stream Analytics

# M-FILES PROOF-OF-CONCEPT



# POC RESULT



# THE BUILD WITH MICROSOFT FABRIC

# FABRIC?



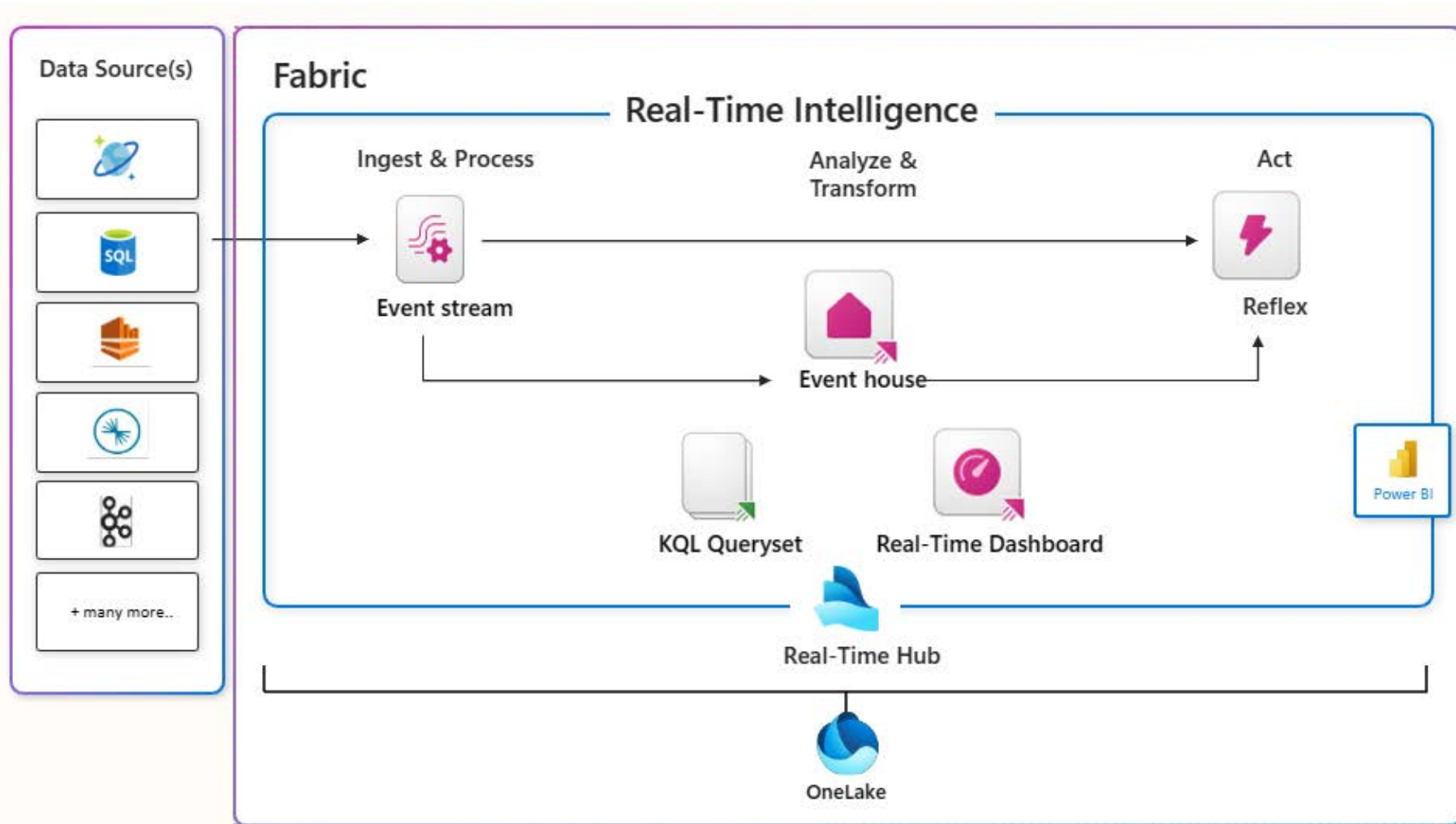
VS



# TIMELINE – HOW FABRIC RTI BECAME TO EXIST

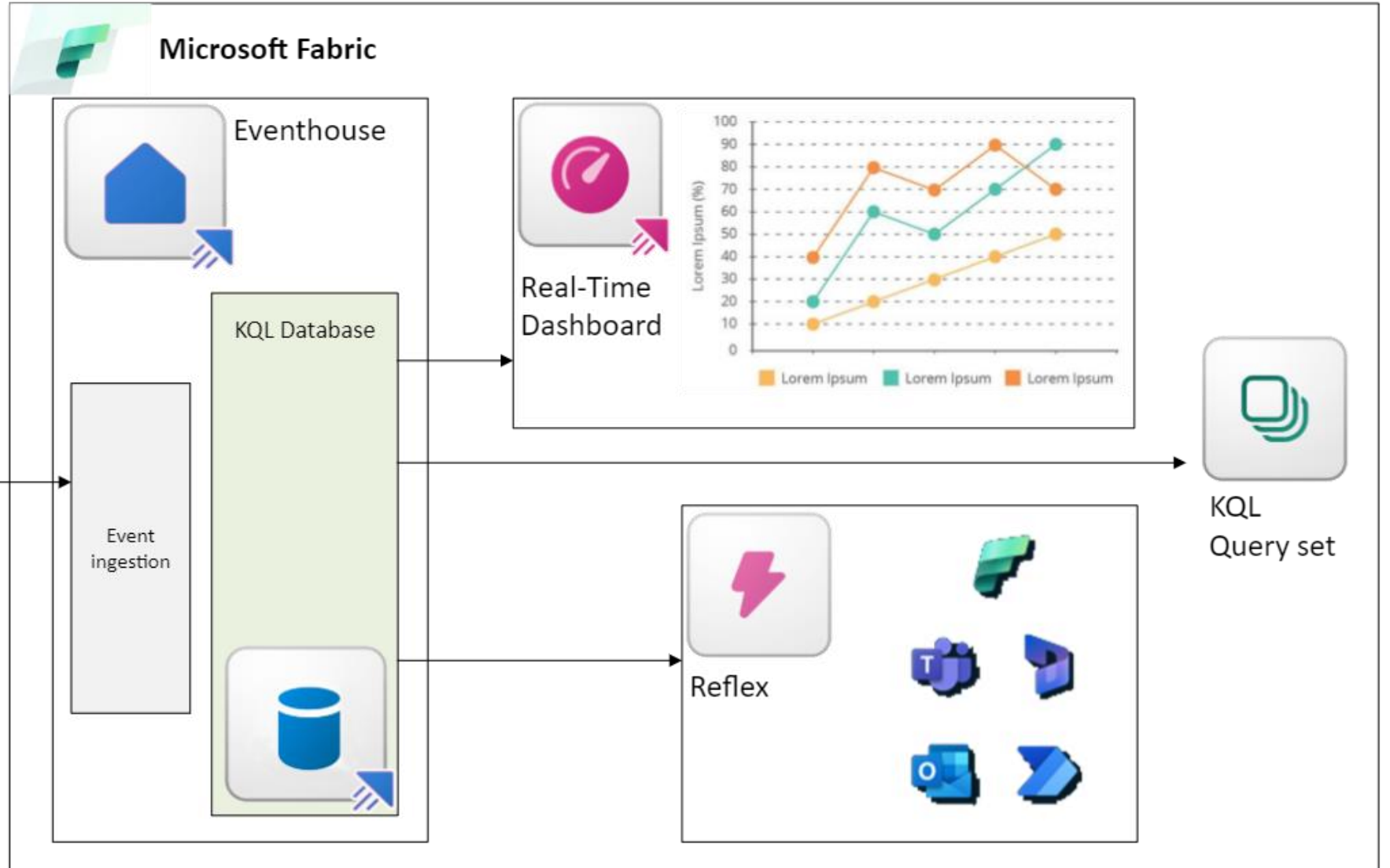
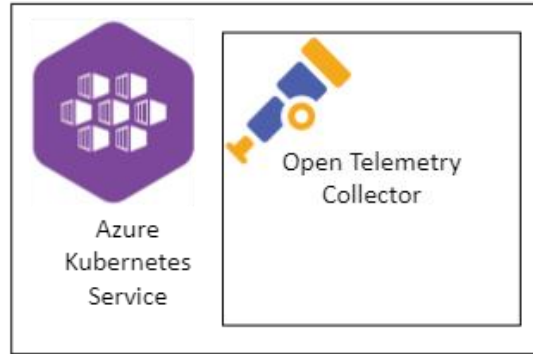
- -14: Snowflake GA
- -15: Databricks GA
- Nov -16: Azure Application Insights GA
  - Kusto aka. KQL GA
- Feb -19: Azure Data Explorer GA
- May -23: Fabric public preview
- Nov -23: Fabric GA
- May -24: Fabric Real-Time Intelligence  GA
- Oct -24: Data Activator  public preview
- ?? -??: Data Activator GA

# FABRIC REAL-TIME INTELLIGENCE REFERENCE ARCHITECTURE

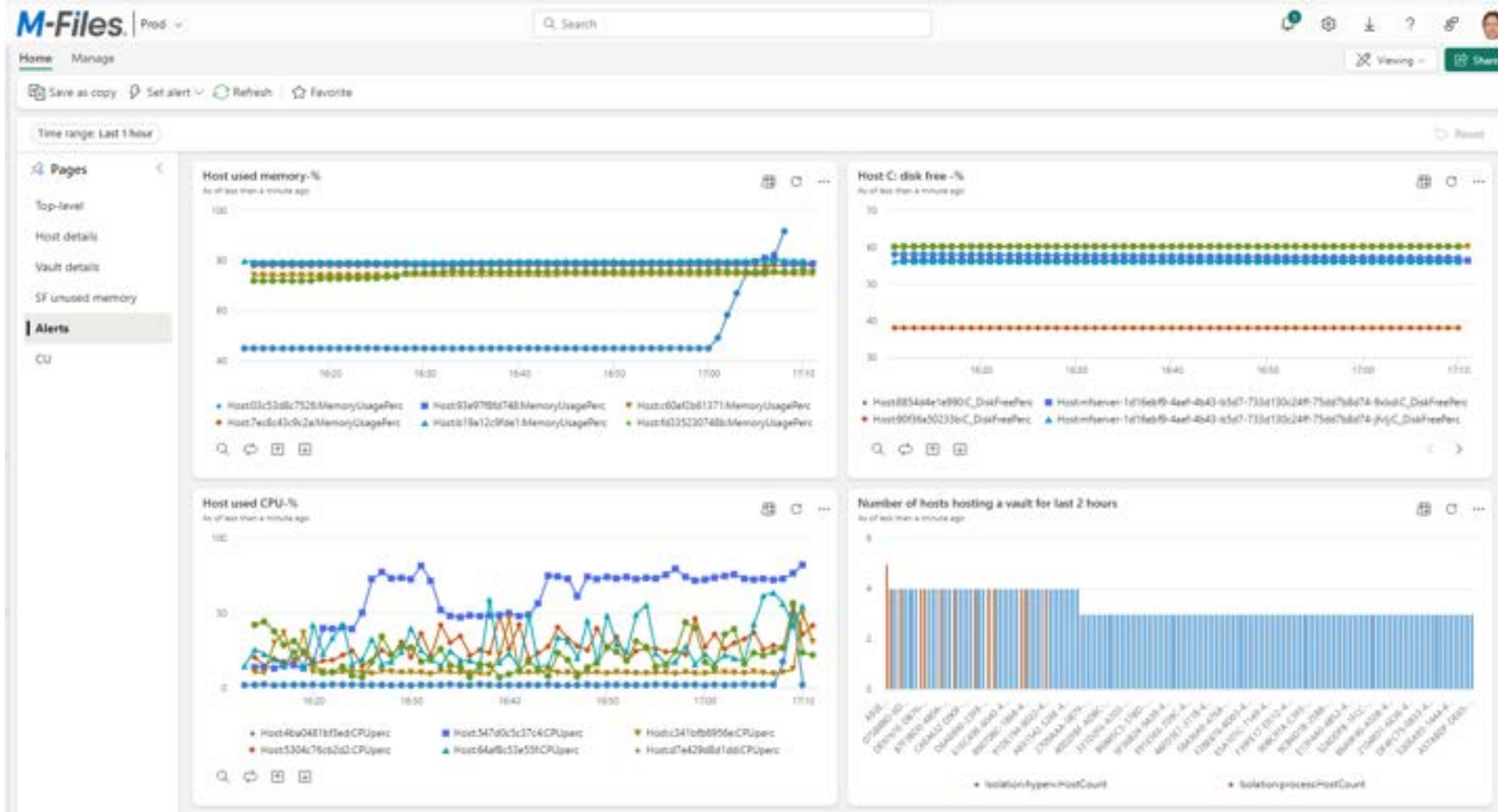




# M-FILES OBSERVABILITY ARCHITECTURE



# FABRIC REAL-TIME DASHBOARD DEMO



# PUNCHLINE – HOW DID IT GO?

# CURRENT STATUS - DEVELOPMENT OBJECTIVES



- Scalability
- Configurability
- Easy access to data
- Real-time Insights into immediate actions
  
- Operating cost

# CURRENT STATUS - CONS



- Fabric is new – really, really new
- Level of knowledge, understanding and support needed
  - Getting Data Activator to work
  - Boosting KQL performance
- Fabric is not really an observability platform
  - Lacks basic features present in market leaders
- Operating cost
  - Consumption of Fabric Capacity Units is a lottery
  - Based on max. requirement, rest goes to "No Value Club"

# FUTURE – WHAT NOW?

- Maintenance mode
- Support stakeholders
- Keep improving and experimenting with Data Activator



# KEY TAKEAWAYS

1. **Don't** build your own observability platform  
– unless you have to!
2. For any secondary data:  
**Fabric-first,**  
others if Fabric doesn't cut it for you
3. **Give it a go!**  
Fabric PoC yields quick results

**THANK YOU!**

Questions?





# M-Files®

*The Smarter Way to Work.*

