FLANDERS MAKE

DRIVING INNOVATION IN MANUFACTURING

KDM Connectivity and Cybersecurity

Nuno Ruas, Robbert Hofman, Valentijn de Leeuw

Content

Industrial IoT architecture

Connectivity and data transfer

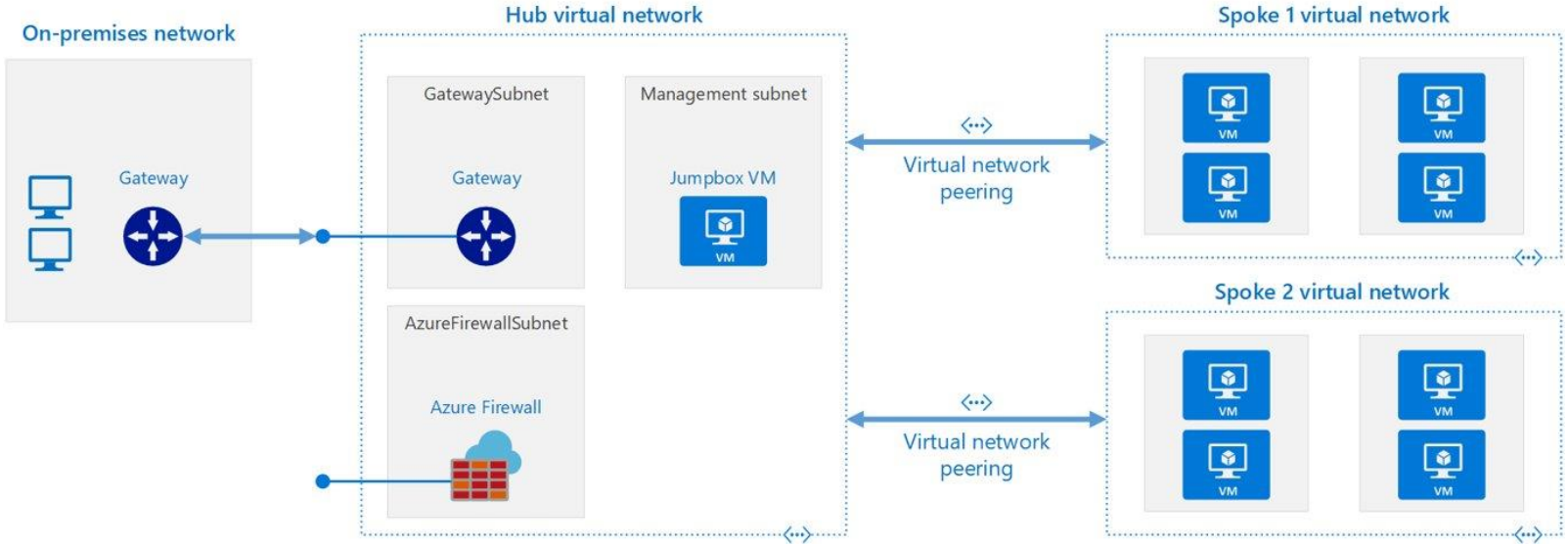
Data processing

Industrial IoT Architecture

Commonly used pattern: Hub and spoke architecture

Labs,
Remote users

Data exchange with lab only via Hub



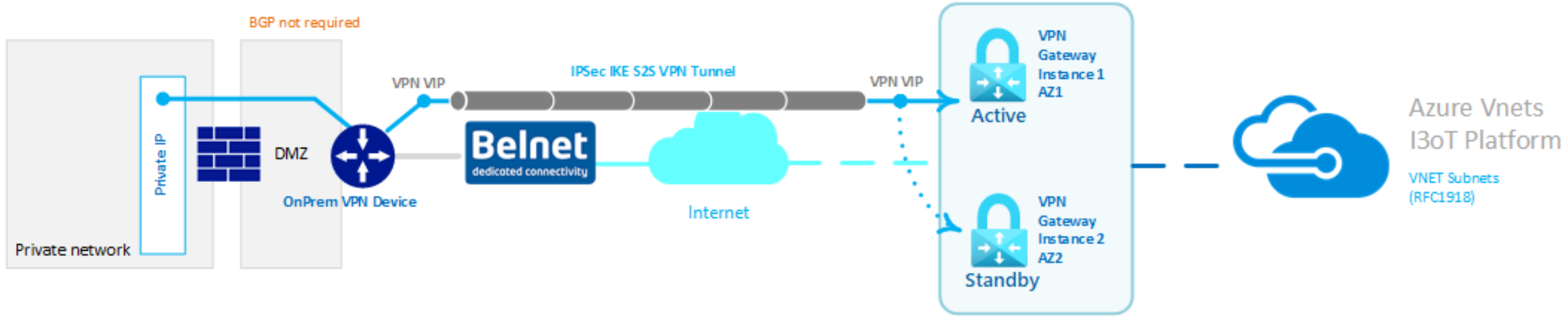
I3oT ”Hub” with firewall, gateway traffic filtering and security

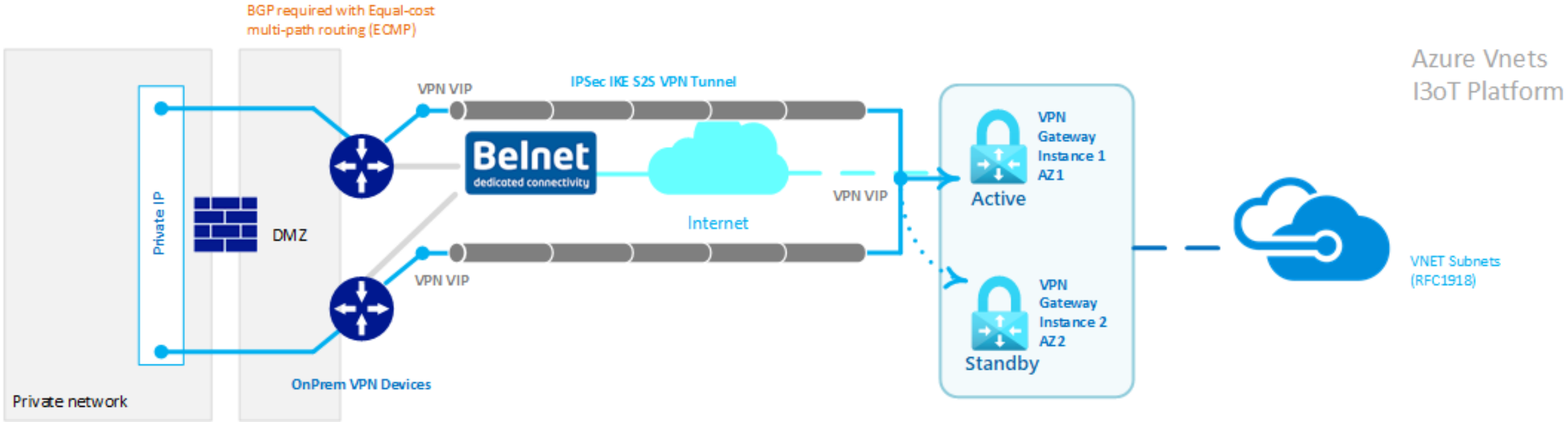
Fully isolated project spaces,

Built upon software provisioned “Landing Zones”,

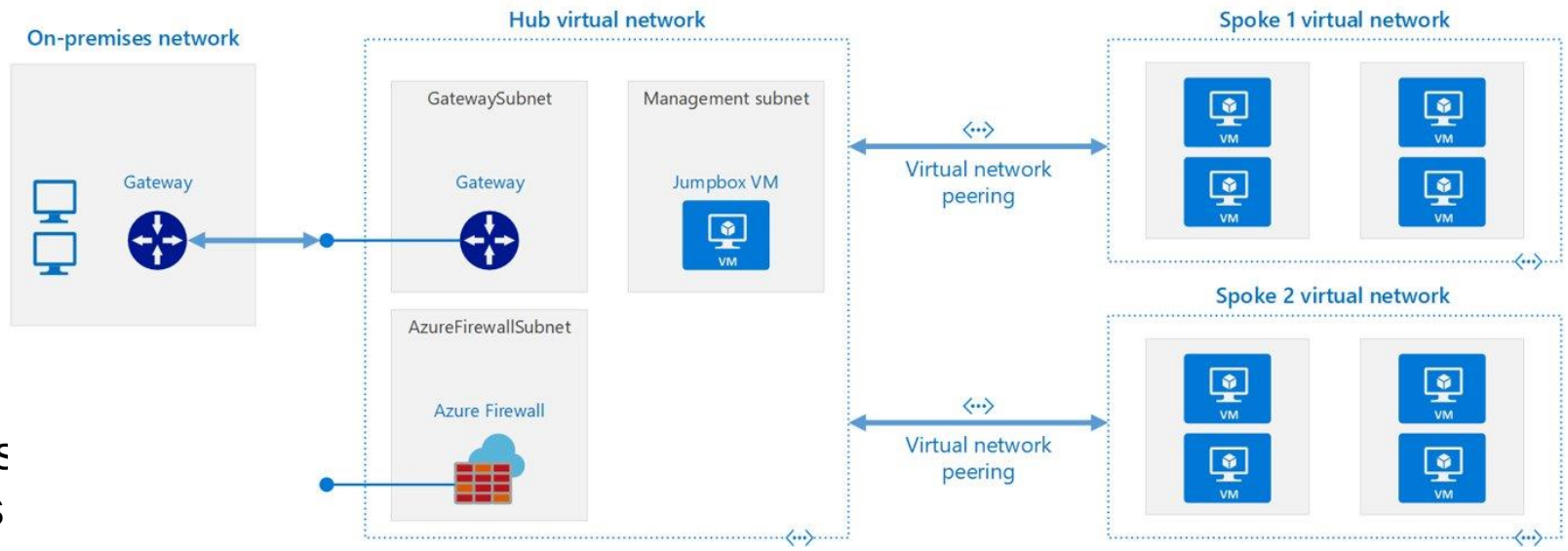
Connectivity - physical

VPN tunnel, single or redundant, site-to-site (exceptionally point-to-site)





Use case or project template: hub and spoke approach



Data exchange with lab only via Hub

Labs,
On premises
With access
to specific
spoke(s)

I3oT ”Hub” with firewall, gateways, traffic filtering and security

Completely isolated project spaces

Connectivity and data transfer

Connectivity – Streaming from an AGV

The device is not (always) on premises, not always accessible via secured network

- The device cannot use VPN

The device is connected directly to the landing zone

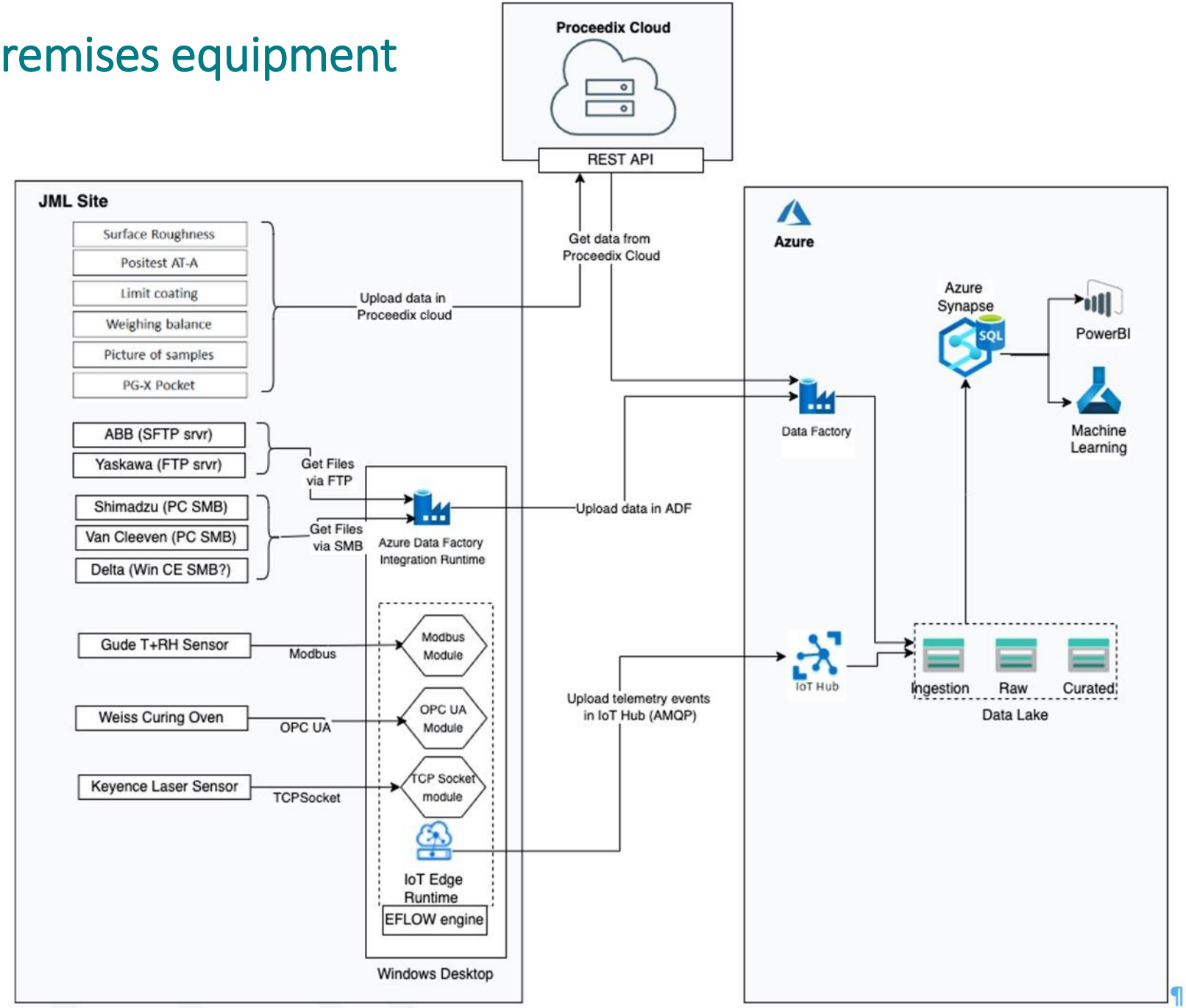
- Azure IoT Hub is used to connect the drone
- The Python Azure IoT SDK is used to capture data and connect
- The IoT Hub generates a connection string used in the device code to connect
- IoT Hub routes the data to a storage account
  - Azure Eventhub is an intermediary between IoT Hub and the storage account to authenticate the drone

Connectivity modes on premises equipment

File transfer

Streaming

API retrieval



1/3 Connectivity - File transfer

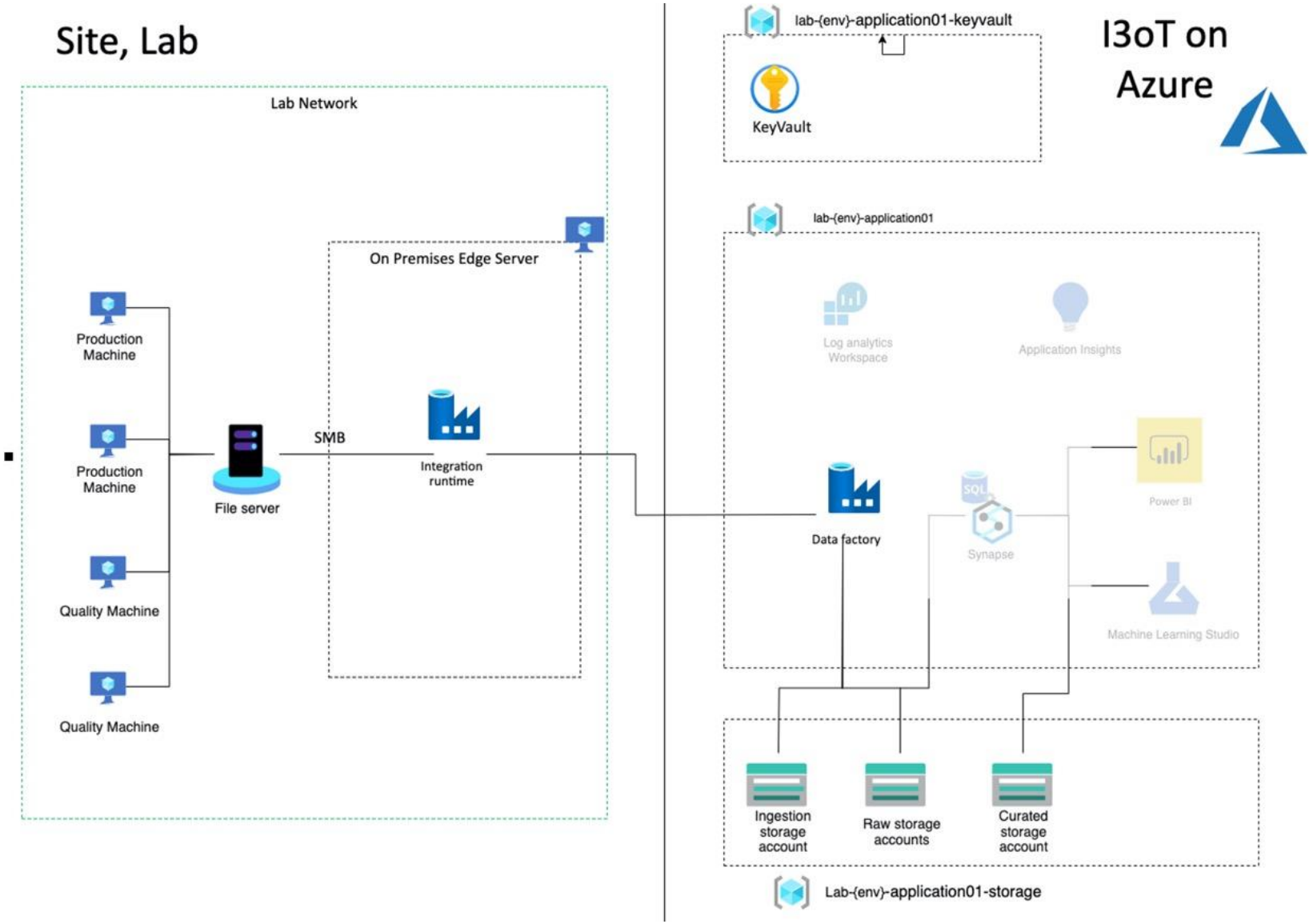
Edge, Lab, Site

ADF integration runtime

Cloud

Azure Data Factory

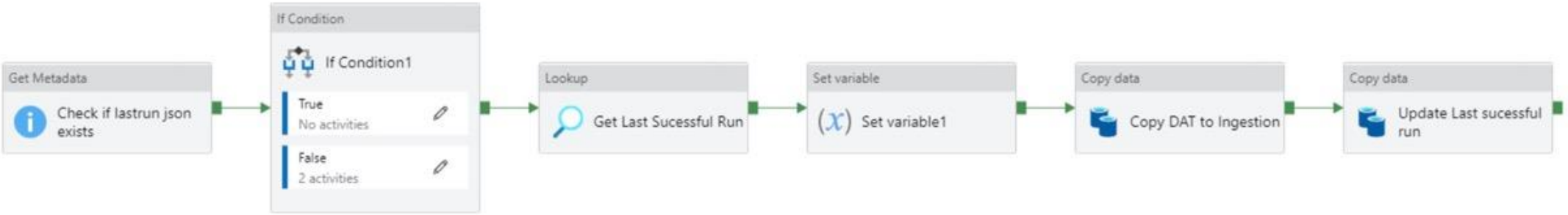
Files can be transferred through an automated ”pipeline”



1/3 Connectivity - File transfer pipeline and data transformation

Pipelines are (conditional) activities in sequence

- Check with set frequence: are there new data?
- If yes copy to “Ingestion” storage account
- From “ingestion” data can be moved into directories in the “raw” storage account



Alternative

- Ingestion done with Azure Data Factory
- Transformation with Azure Synapse (the analytics environment)

1/3 Azure Data Factory

Microsoft Azure | Data Factory ▸ adf-bphot-prd-we-application01

Search

master branch | Validate all | Save all | Publish

IngestionCopyAppli... | IngestionCopyDat

Saved | Save as template | Validate | Validate copy runtime | Debug | Trigger (1)

Activities

Search activities

- Move & transform
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

If Condition1

Lookup: Get Last Sucessful Run

Set variable: Set variable1

Copy data: Copy DAT to Ingestion

Copy data: Update Last sucessful run

General | **Source** | Sink | Mapping | Settings | User properties

Source dataset *: OnPremDatDataSet | Open | New | Learn more

File path type: File path in dataset | File filter | Wildcard file path | List of files

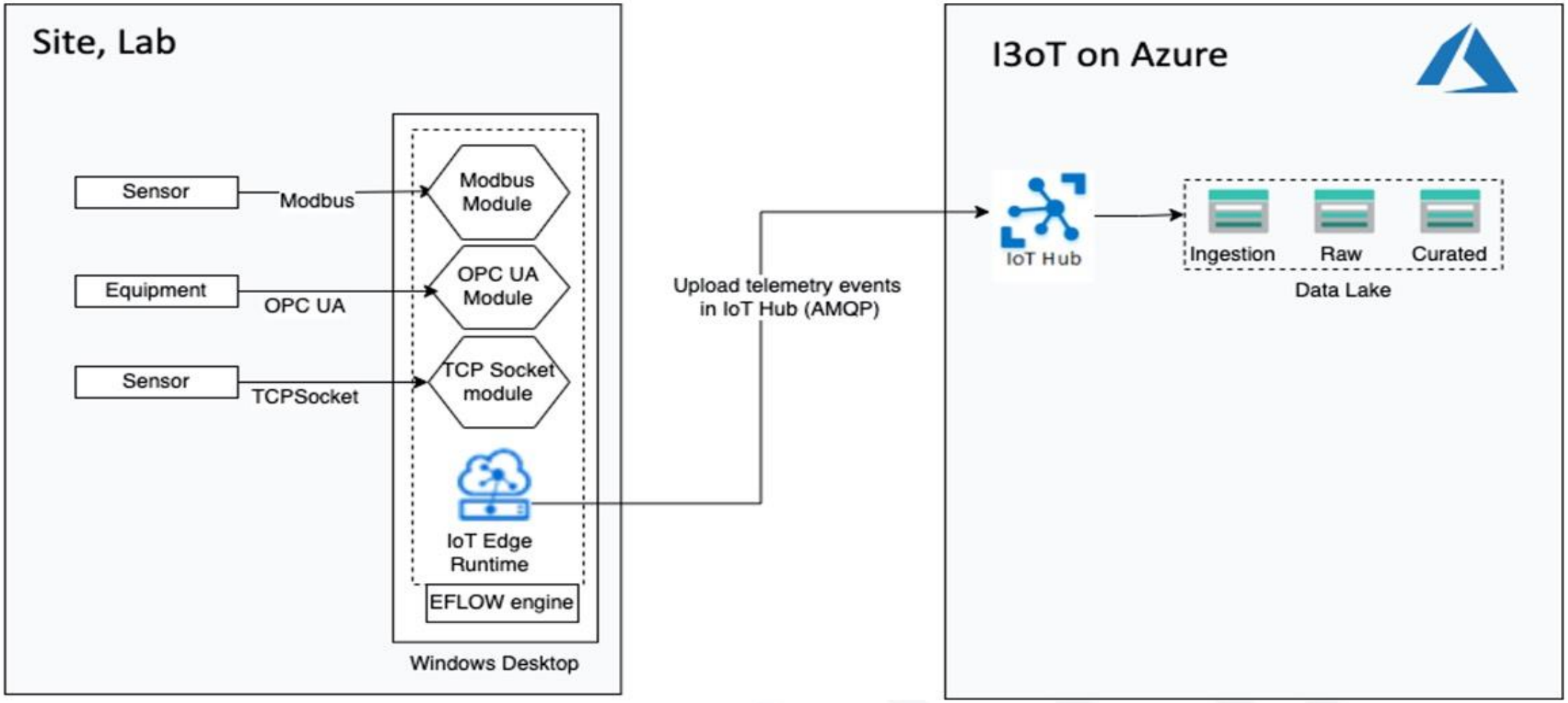
File filter: *.dat

Filter by last modified: Start time (UTC): @variables('LastSuccessfulRun') | End time (UTC): @utcnow()

Recursively: ☑

Delete files after completion: ☐

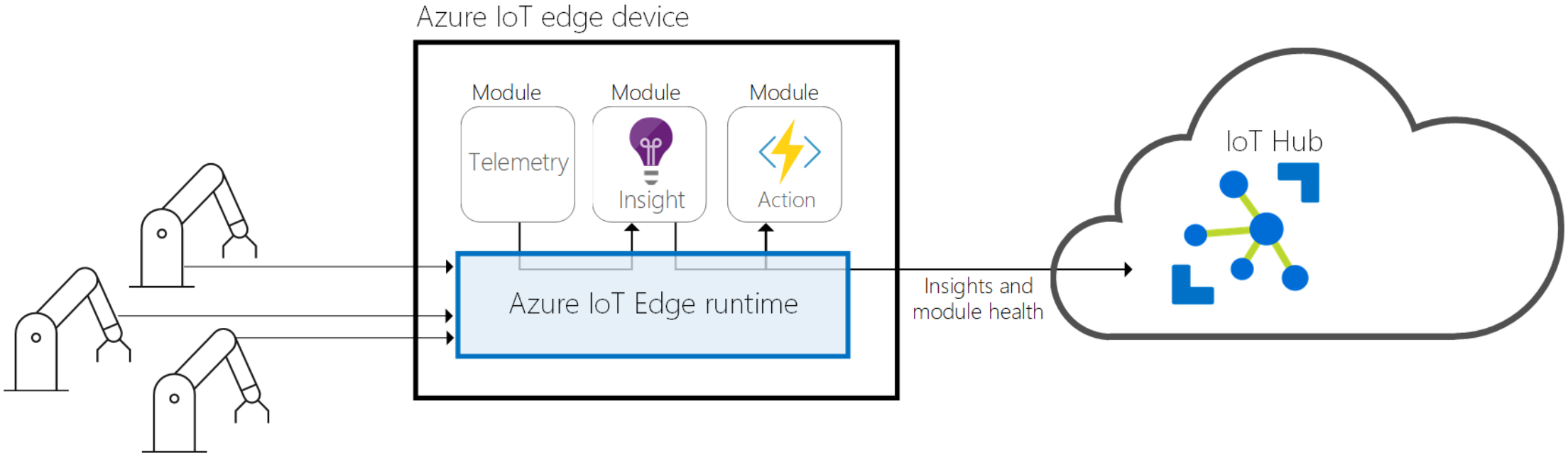
2/3 Connectivity – Streaming from a site



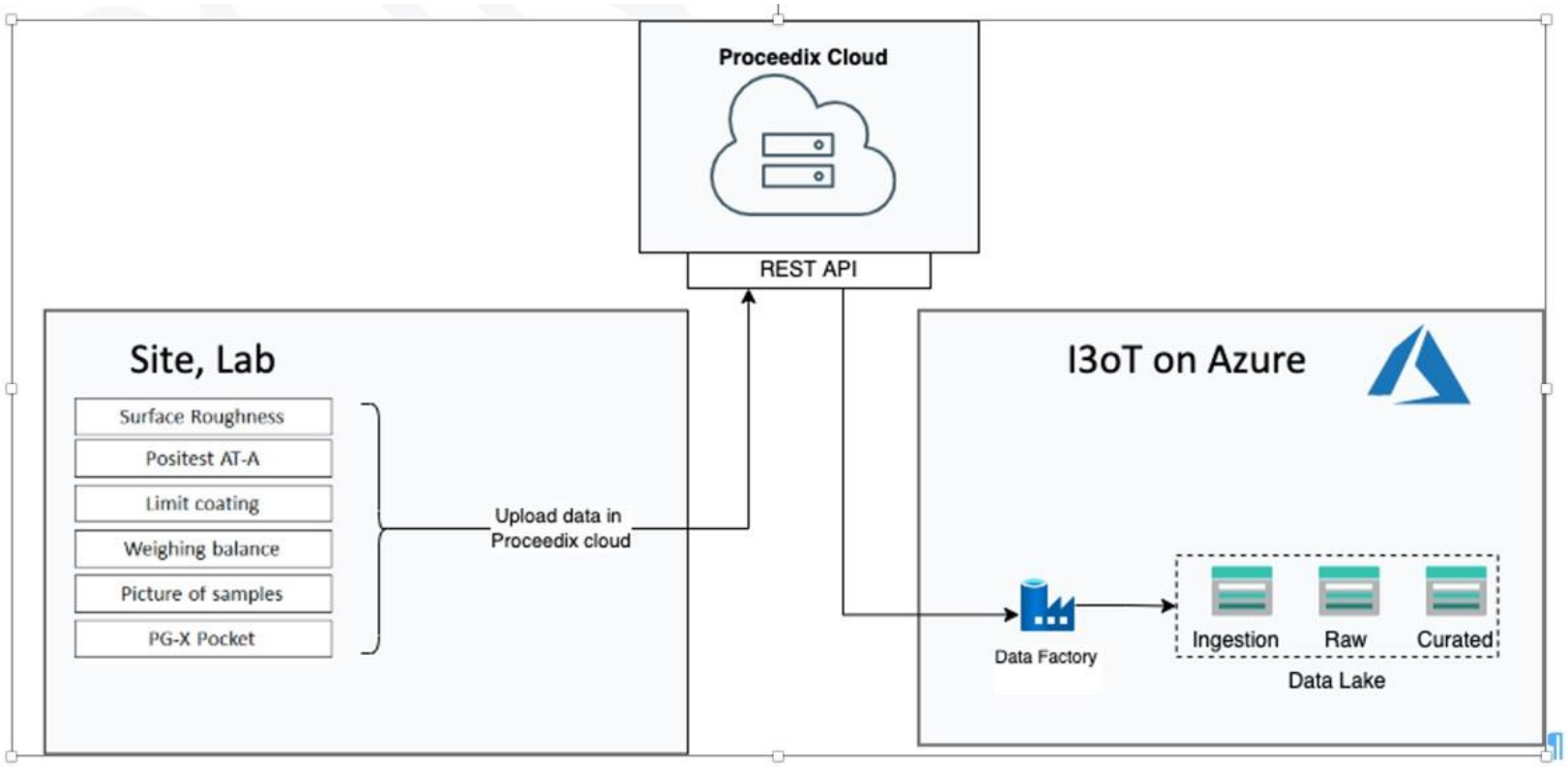
The data transformation pipelines have been implemented in Synapse in the JML case

2/3 Streaming from a site: IoT Edge

- Container runtime (~docker) with some built-in telemetry & library of azure "modules" (=containers) (https://azuremarketplace.microsoft.com/en-us/marketplace/apps/category/internet-of-things?page=1&subcategories=iot-edge-modules)



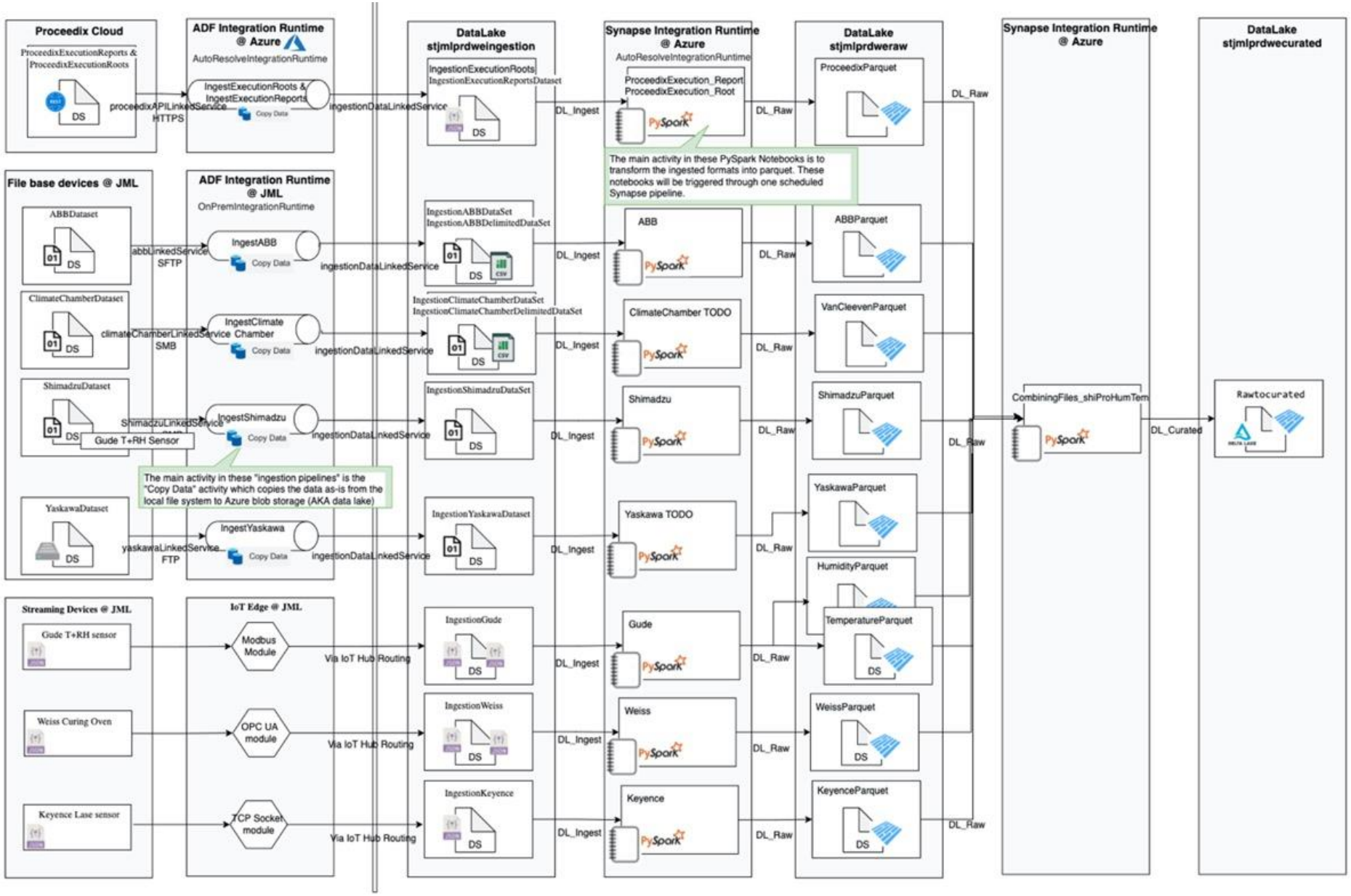
3/3 Retrieval through API



Azure Data Factory accepts “API” as one of the possible data sources (and data sinks)

- In JML, Proceedix contains
  - context data on the experiments
  - certain experimental data

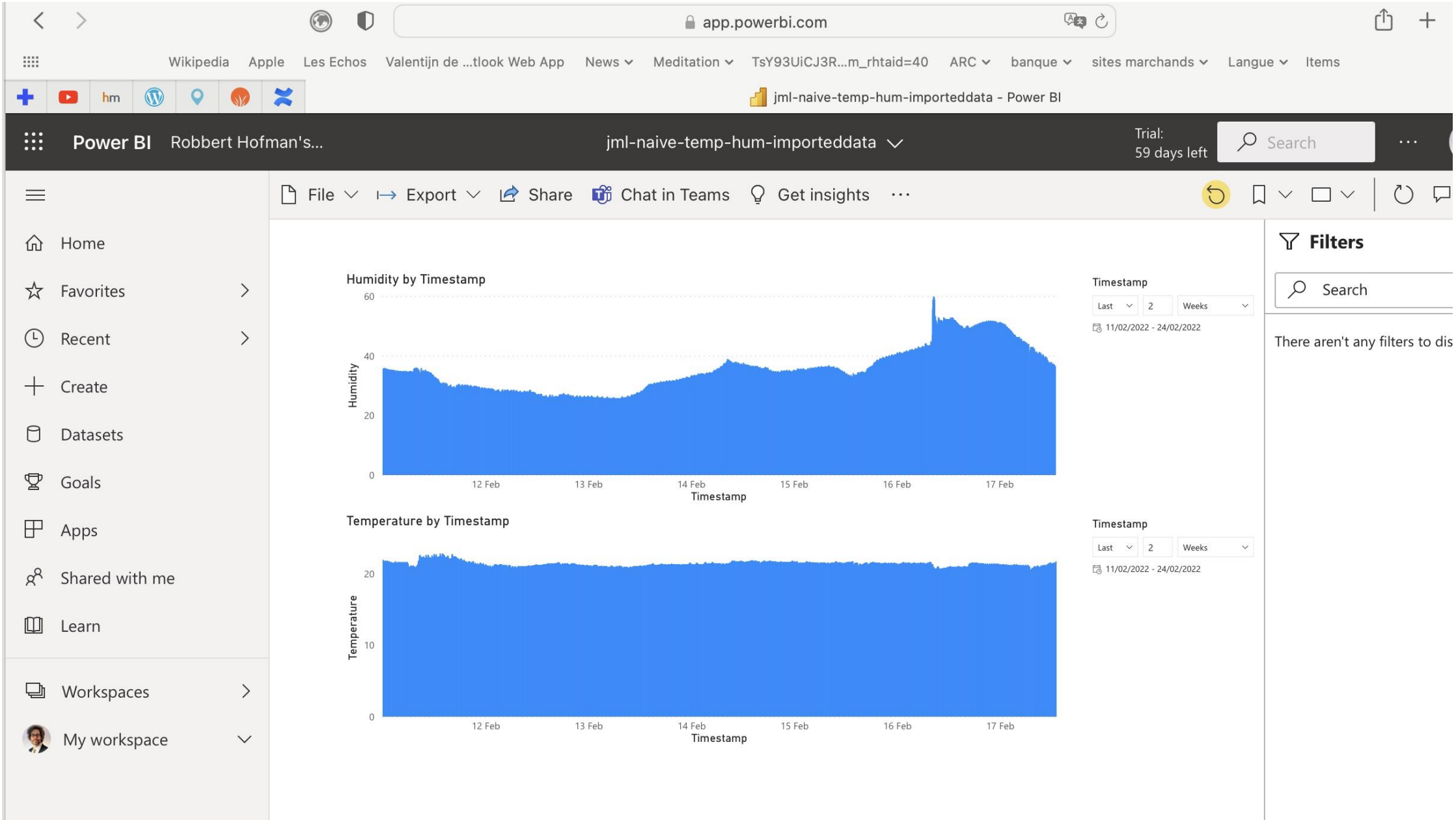
JML – Overview of data pipelines for all connectivity options



Data processing

Visualization and analysis in Power BI

https://services.flandersmake.be/confluence/display/intranet/Dashboards#



Azure Synapse Analytics

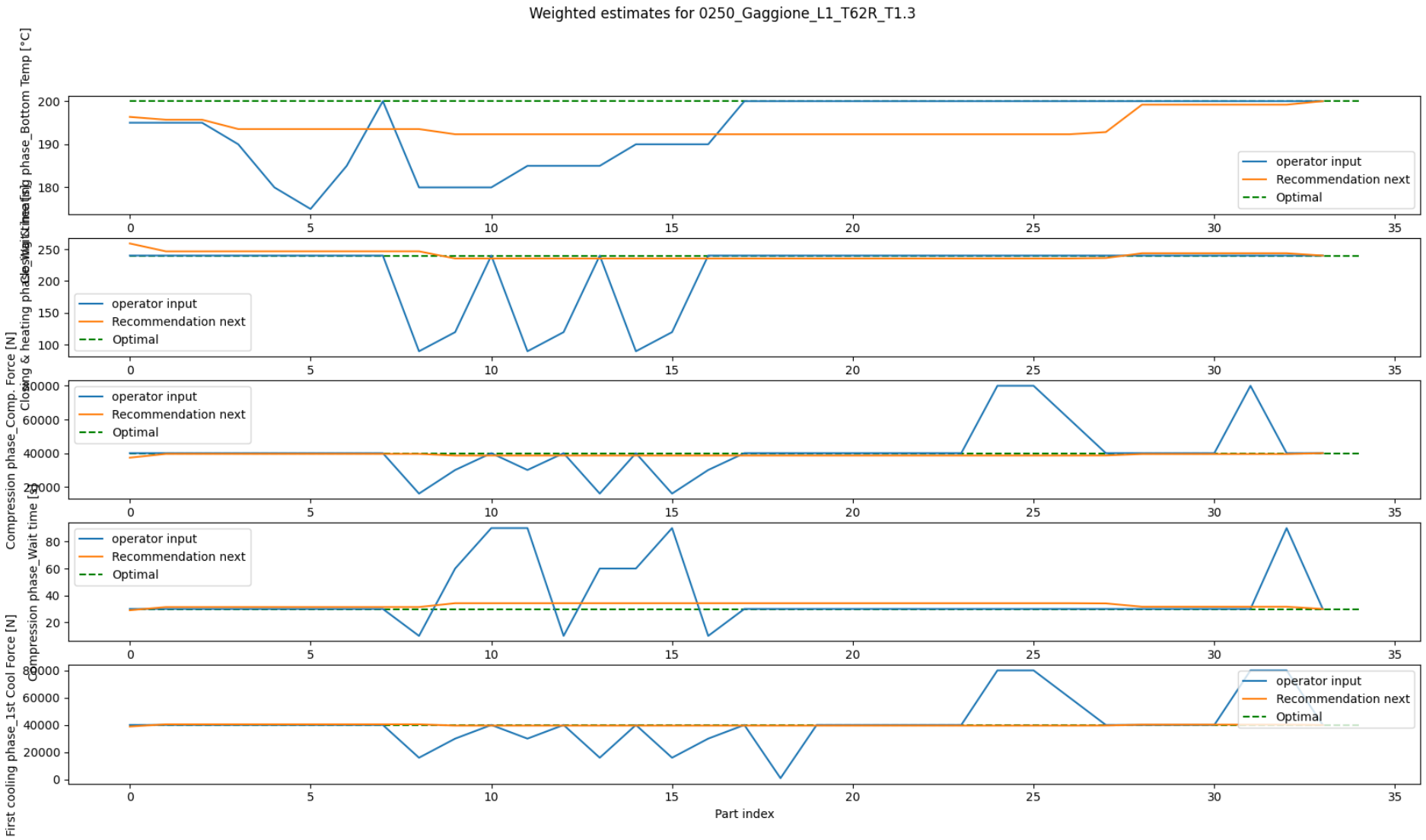
PaaS solution, that comes as a package

- SQL database
- Apache Spark for big data analytics
- Data explorer

Example application

- Operator makes manual parameter changes to come to optimal result
- ML derives a model from several of the iterations and the final result
- Model predicts the final parameter set faster

Azure Synapse Analytics: example application



lessons learnt

Lessons learned 1/3

Set up (alerts for) infrastructure monitoring ASAP

- e.g. using heartbeat: external service checking every x time to see if service is up
- In JML, we had a gateway that went down often. Monitoring helps in debugging faster.

For cloud: set up cost monitoring and limiting ASAP

- In JML, a pipeline errored out and was restarted automatically. Cost a lot of money producing the same error.

Find the right hardware for the right job

- In JML, gateway should've been server-grade hardware that makes remotely (re)booting easy

Lessons learned 2/3 - Connecting brownfield devices is hard

Contacting OEM takes a long time

Retrofitting with additional sensor is OK but not always possible

Reverse-engineering (websocket at Weiss) turned out not to be stable for us, might work with other machines / protocols

Lots of devices have old / insecure protocols (no SFTP on Yaskawa, no file size readout available)

Takeaway: at the time of purchase, ask for and purchase connectivity options

Lessons learned 3/3

- Cost saving: sometimes it is possible to do processing on the edge, taking advantage of available hardware
  - Sometimes even hybrid solutions (e.g. Azure integration runtime allows to run limited workloads on edge)
  - Some resources/applications need to be switched off after usage for major cost saving
- A single cloud provider can have multiple solutions with overlapping functionality
  - E.g. Azure Data Factory <-> Azure Synapse
  - Get inspired by Azure Architectures (https://learn.microsoft.com/en-us/azure/architecture/browse/)
- No-code can be limiting (scale or capabilities)
  - E.g. Azure Data Factory, copying files, using Function Apps