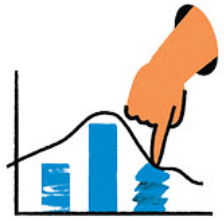
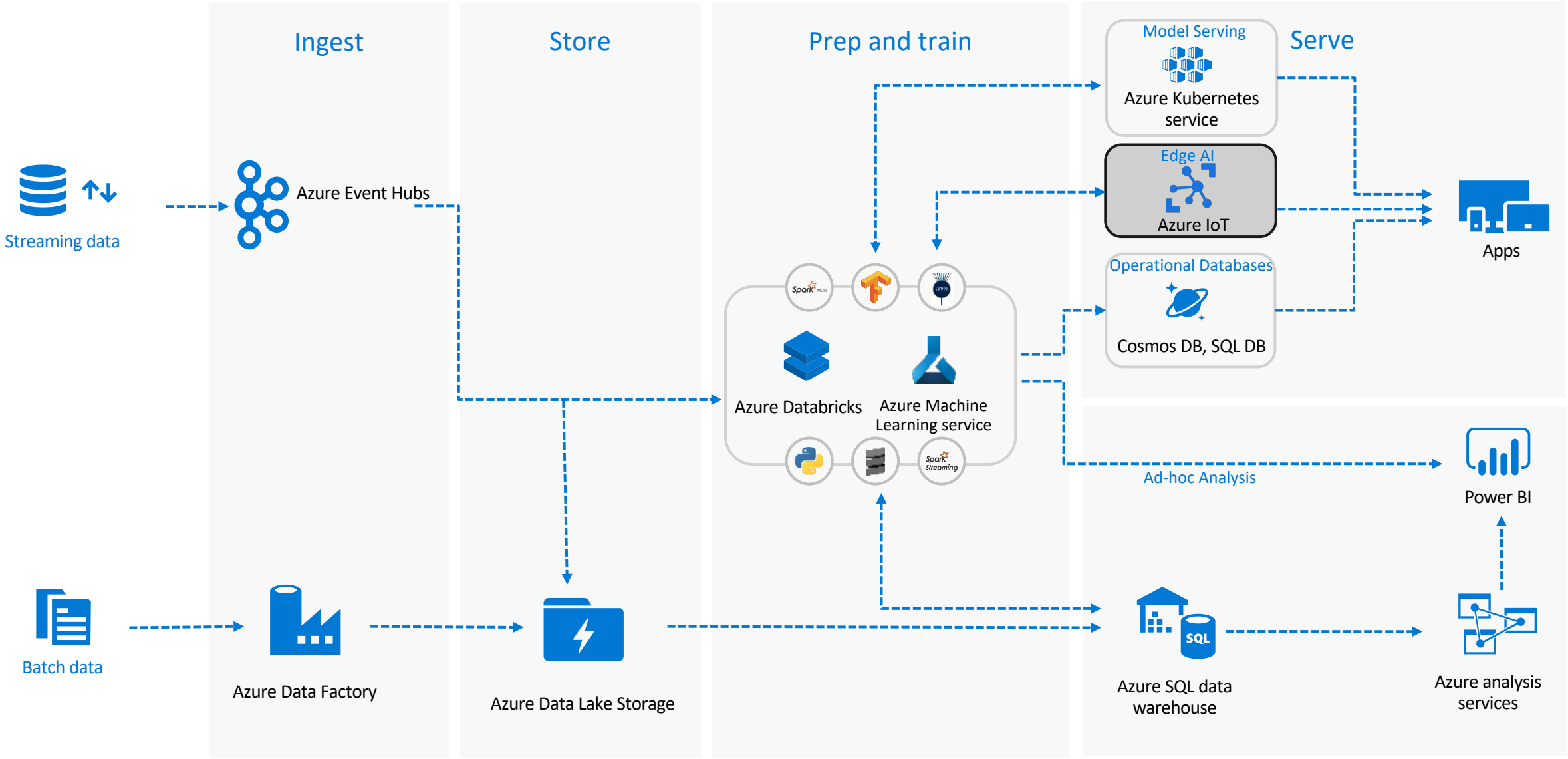
Microsoft



Azure AI & IOT

Part 2: Computer Vision on Edge

Architecture to build e2e ML solutions with Azure services



Transformative Solutions for Manufacturing

Get started easily with flexible, scalable solutions to transform your business function

Quality Assurance

What are the root causes of recurring problems?

(so I can redesign product, process, training)

Demand Forecasting

What sales or usage volume can I expect?

(so I can optimize my staffing and asset utilization)

Operational Efficiency

What factories, lines, teams, machines, people…are most effective ?

(so I can identify best practices and allocate resources)

Anomaly Detection

What machines are not performing properly?

(so I can immediately investigate)

Predictive Maintenance

What is the likelihood this part will fail ?

(so I can prevent it from happening)

Supply Chain and Inventory Optimization

What is right level and supplier for a part?

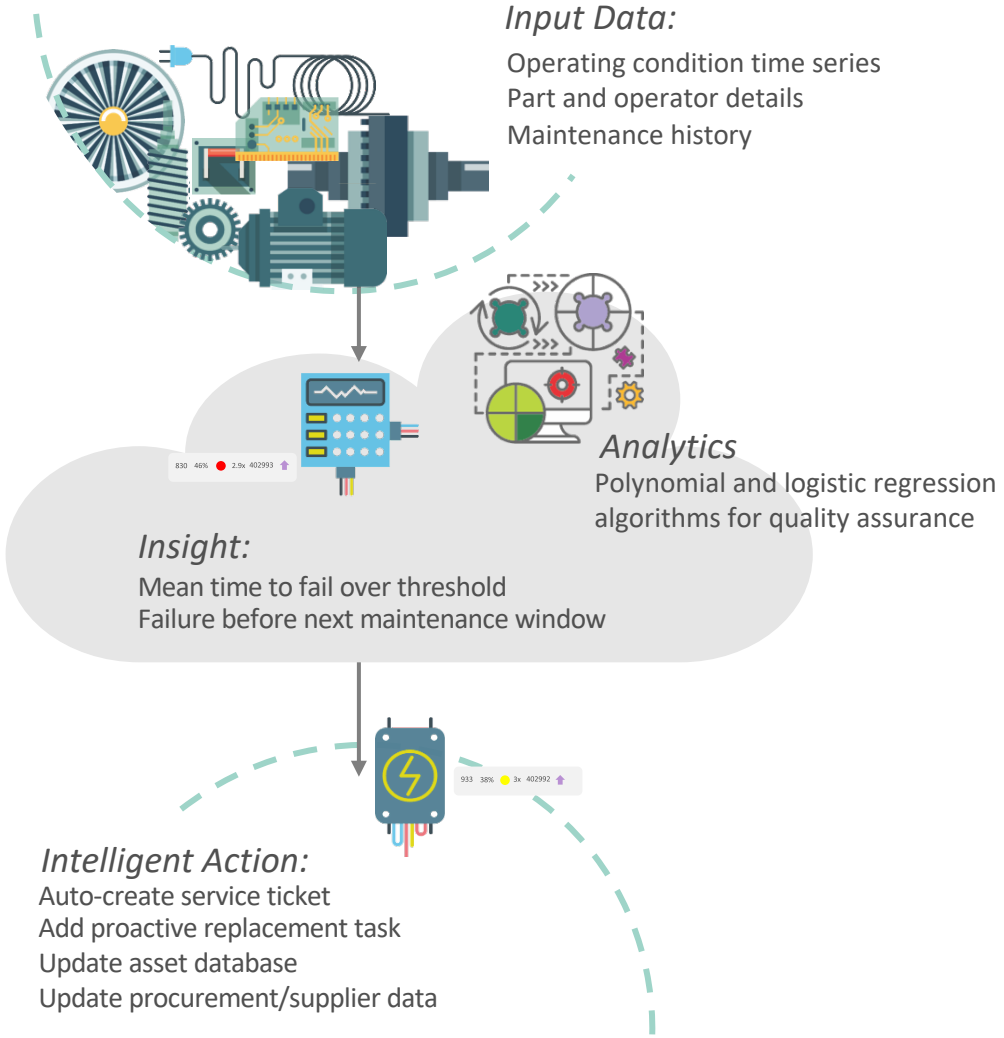
(so I can simplify and optimize procurement)

Connected Devices

Are my buildings, plant and equipment working at peak efficiency

(so I can reduce unnecessary costs)

Building a Quality Assurance solution



JABIL

*"Since deploying the Microsoft predictive analytics solutions we have seen at least an **80 percent accuracy rate** in the prediction of machine processes that will slow down or fail, contributing to a scrap and rework savings of 17 percent,"*

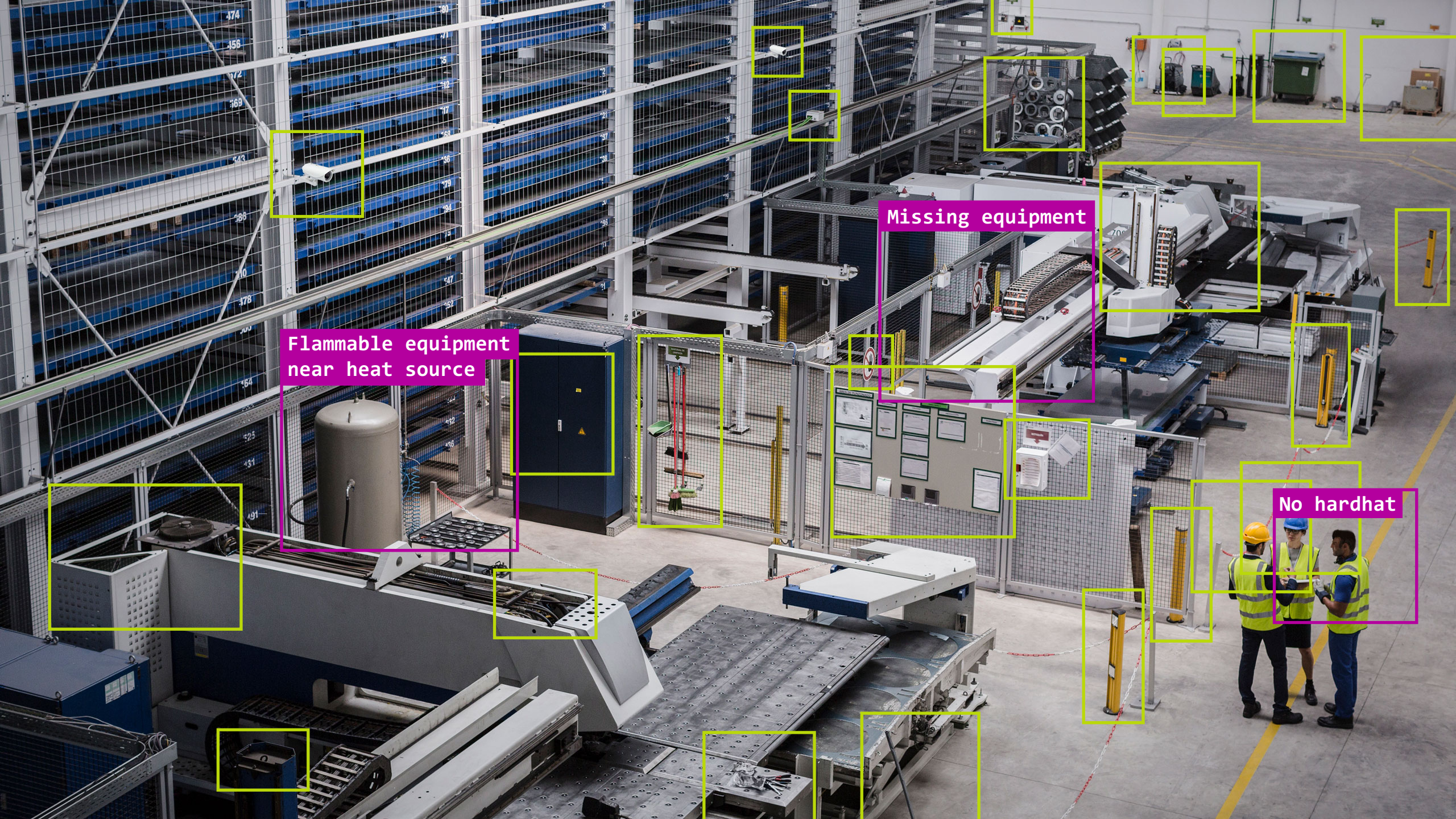
Clint Belinsky, Vice President of Global Quality, Jabil.

Challenges Jabil was facing…

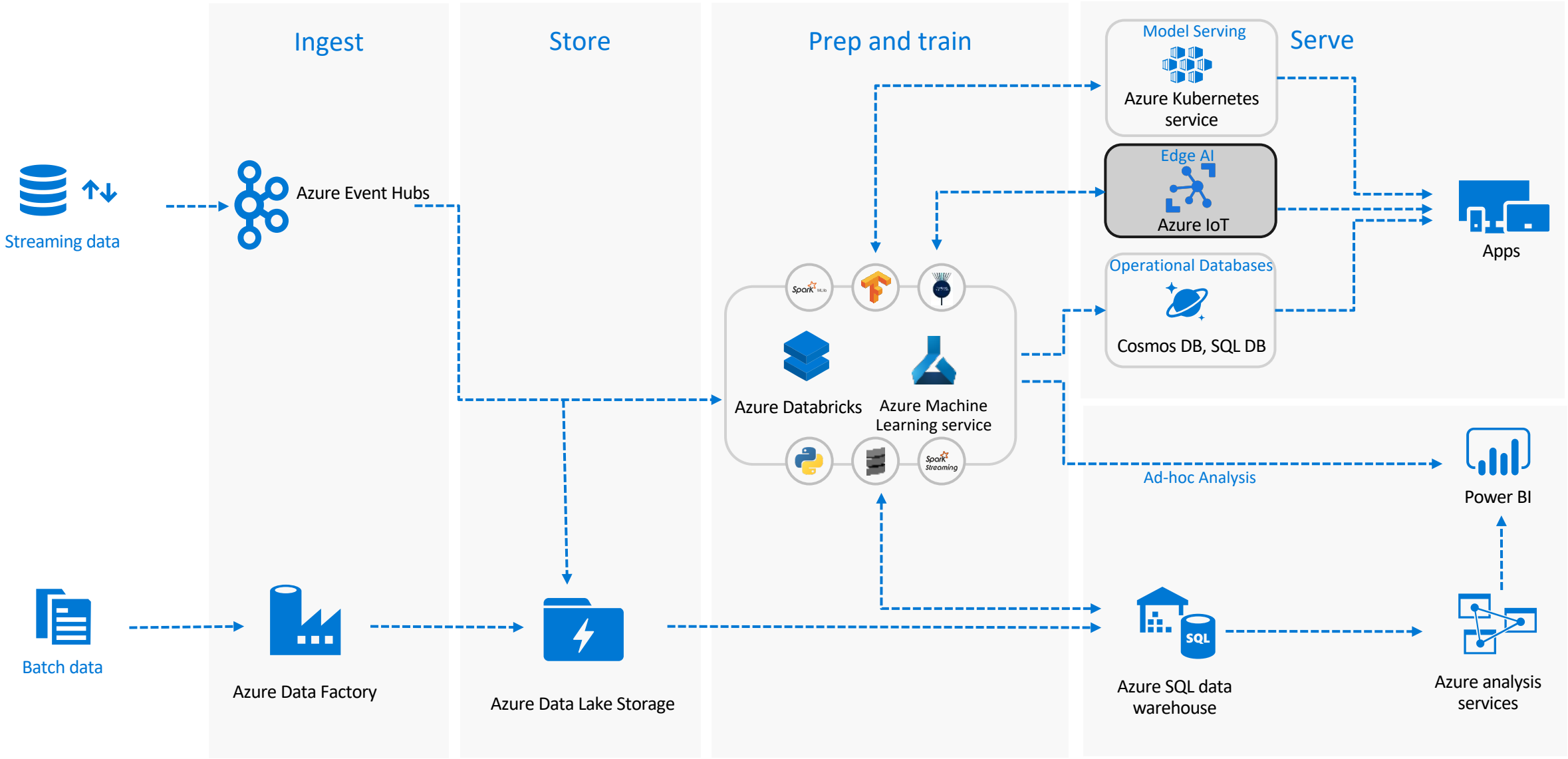
- Continuous requirement to increase yield and reduce the amount of re-work
- Traditional inspection techniques for ensuring quality quickly becoming outdated with more one-off production runs
- Adding more equipment and people to existing manufacturing processes would not have significant impact on increasing throughput

And their impact on the business

- Reduced manufacturing cycle time
- Higher cost of wasted materials, time and resources
- Inability to address customers' critical requirement for speed to market


Missing equipment
Flammable equipment near heat source
No hardhat

Architecture to build e2e ML solutions with Azure services



Azure Machine Learning service

Bring AI to everyone with an end-to-end, scalable, trusted platform

Boost your data science productivity

Increase your rate of experimentation

Deploy and manage your models everywhere

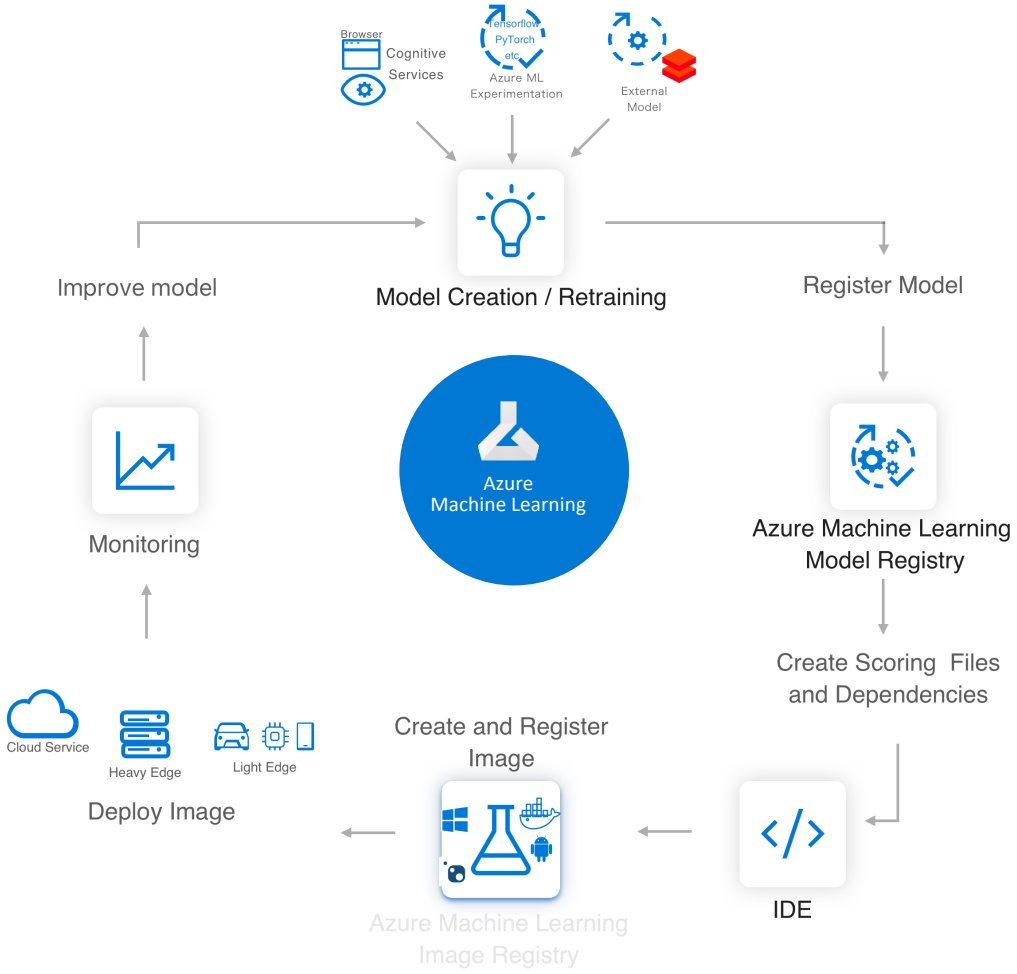
Built with your needs in mind

- Automated machine learning
- Managed compute
- DevOps for machine learning
- Simple deployment
- Tool agnostic Python SDK
- Support for open source frameworks

Seamlessly integrated with the Azure Portfolio

Azure ML service

Lets you easily implement this AI/ML Lifecycle



Workflow Steps

Develop machine learning training scripts in Python.

Create and configure a compute target.

Submit the scripts to the configured compute target to run in that environment. During training, the compute target stores run records to a datastore. There the records are saved to an experiment.

Query the experiment for logged metrics from the current and past runs. If the metrics do not indicate a desired outcome, loop back to step 1 and iterate on your scripts.

Once a satisfactory run is found, register the persisted model in the model registry.

Develop a scoring script.

Create an Image and register it in the image registry.

Deploy the image as a web service in Azure.

Deploy to IoT Edge

Deploying to IoT Edge Devices

Prerequisites

IoT Hub
IoT Edge Device with the IoT Egde runtime installed.
Docker Image based on the ML model and image configuration stored in the container registry. This can be done as follows

```
from azureml.core.image import Image, ContainerImage

#Image configuration
image_config = ContainerImage.image_configuration (
                                    runtime = "python", execution_script ="score.py",
                                    conda_file = "myenv.yml",
                                    tags = {"attributes", "classification"},
                                    description = "Image with my model")

image = ContainerImage.create (name = "myimage",
                               models = [model], #this is the model object
                               image_config = image_config, workspace = ws )
```

Deploying to the IoT Edge

Steps

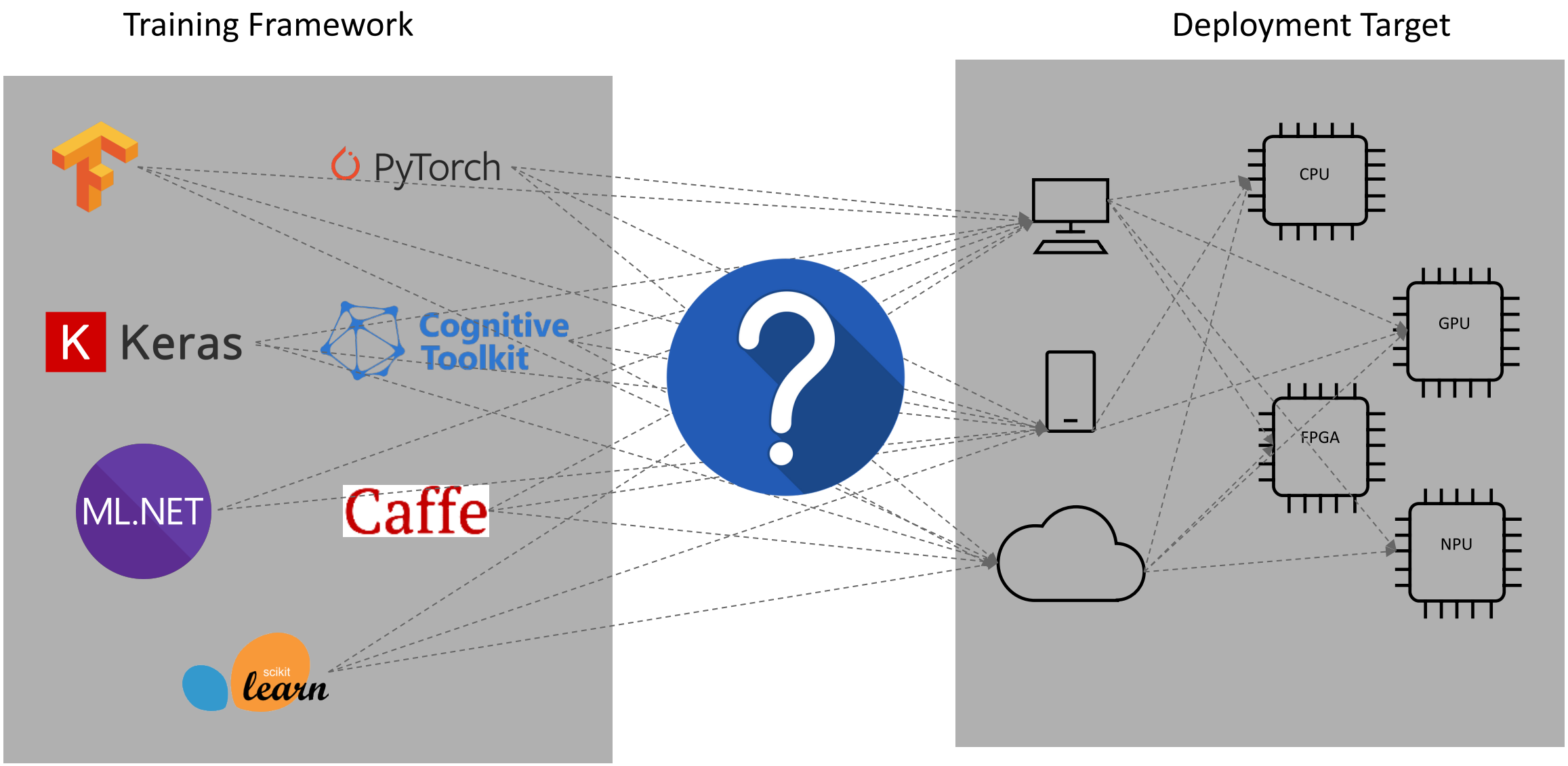
1. **Get the container registry credentials**: Azure IoT needs the credentials for the container registry that Azure Machine Learning service stores docker images in. You can get via the Azure Portal
2. Configure deployment manifest, a JSON document that describes which modules to deploy, how data flows between the modules, and desired properties of the module twins. You can use wizard in the Azure Portal to create this. The Wizard has 4 steps:
    1. Add Modules
    2. Specify Routes
    3. Review Deployment
    4. Submit
3. **View Modules on device:** Once you've deployed modules to your device, you can view all of them in the Device details page of the portal. This page displays the name of each deployed module, as well as useful information like the deployment status and exit code.

Edge Deployment

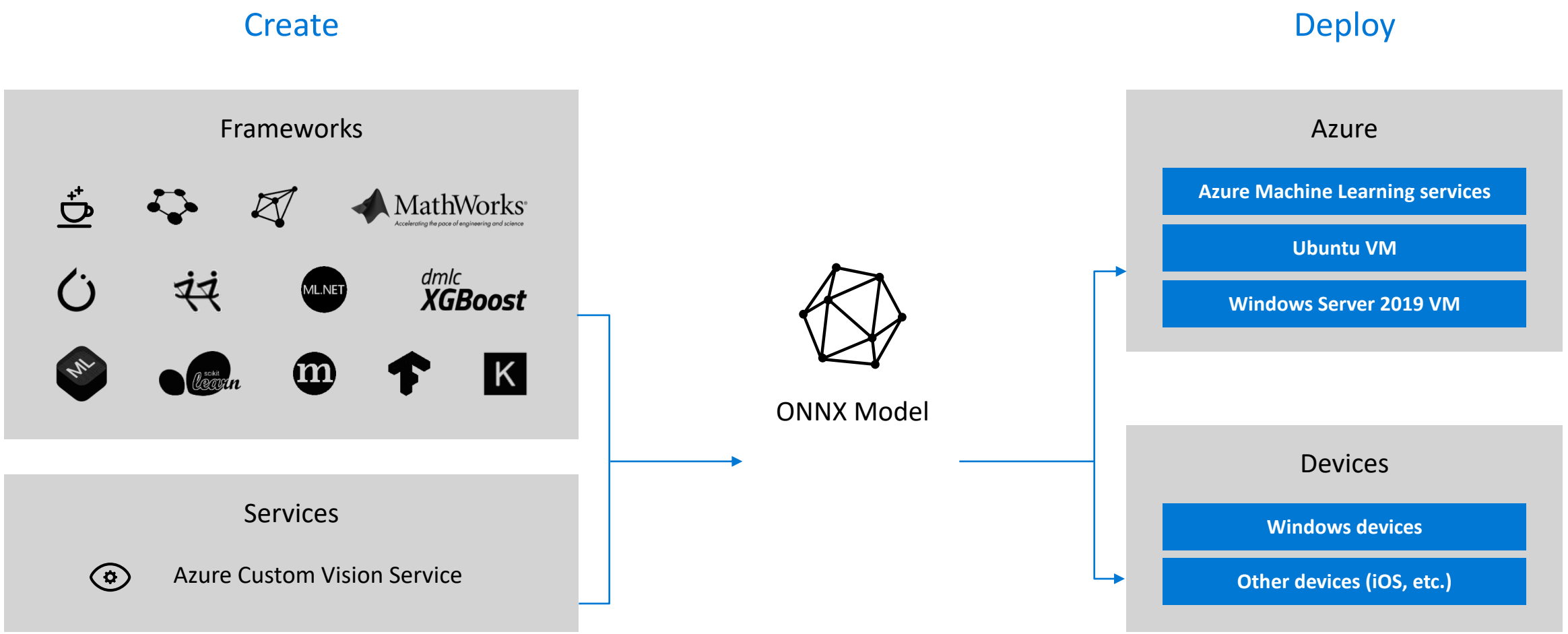
Light and Heavy

		Heavy Edge				Light Edge		
Description	An Azure host that spans from CPU to GPU and FPGA VMs	A server with slots to insert CPUs, GPUs, and FPGAs or a X64 or ARM system that needs to be plugged in to work **Cloud Consistent Hybrid Server**		**PC class devices**	**Gateway**	A Sensor with a SOC (ARM CPU, NNA, MCU) and memory that can operate on batteries **Smart Sensors + Ambient AI**		**Sensors**
Example	DSVM / ACI / AKS / Batch AI	- DataBox Edge - HPE - Azure Stack	- DataBox Edge	- Industrial PC	-Video Gateway -DVR	-Mobile Phones -VAIDK	-Mobile Phones -IP Cameras	-Azure Sphere - Appliances
What runs the model	CPU,GPU or FPGA	CPU,GPU or FPGA	CPU, GPU	x64 CPU	Multi-ARM CPU	Hw accelerated NNA	CPU/GPU	MCU

Deployment Challenge



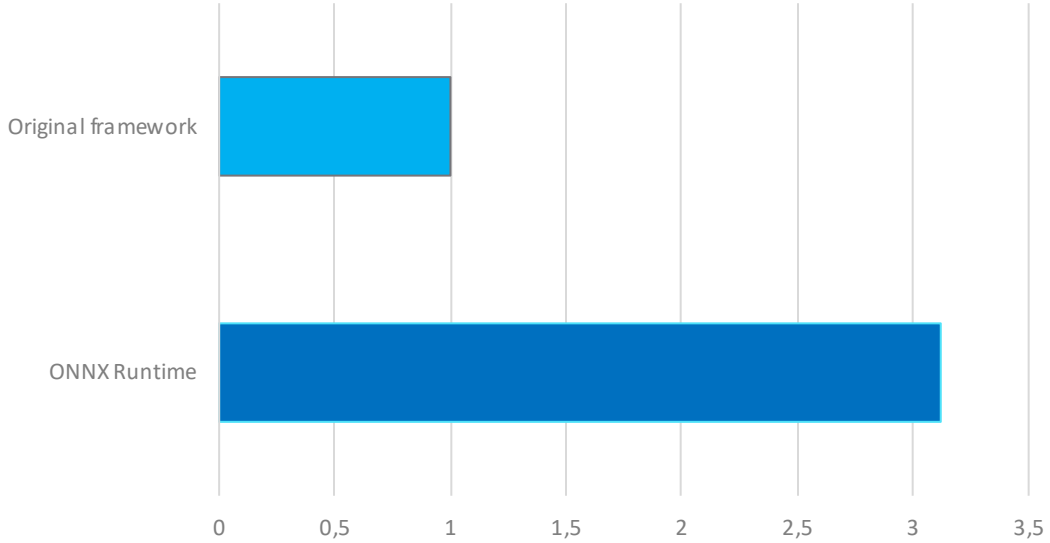
ONNX is the new open ecosystem for AI models



Cognitive Service – Optical Character Recognition

Detect text in an image and extract the recognized words into a machine-readable character stream

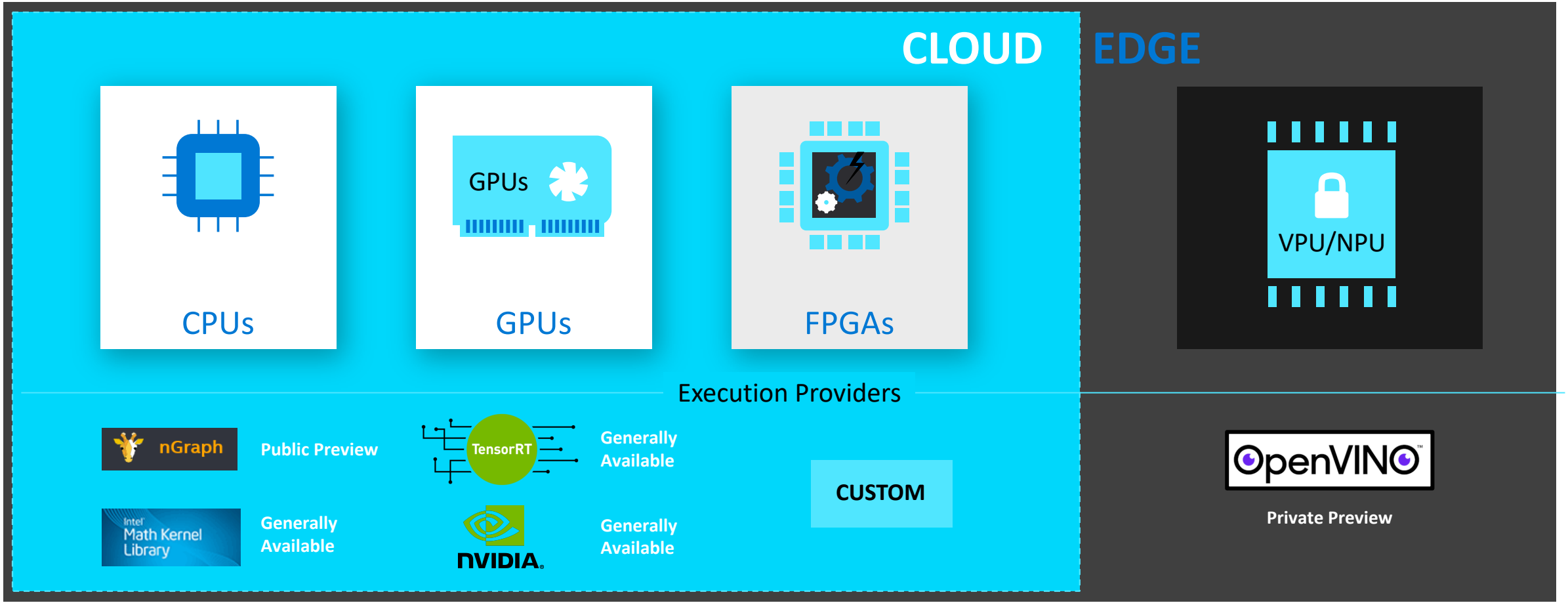
~3.5x performance gain by using ONNX and ONNX Runtime



Inferencing with ONNX Runtime

Silicon Alternatives : FPGAs and ASIC (CPU, GPU, NPUs)

FLEXIBILITY **EFFICIENCY**



Vision AI Development kit

System Architecture