DotNet 2021

ONLINE TECH CONFERENCE

#DotNet2021

22nd June 2021

Model Serving

A bridge between Data Scientists and Engineers

www.dotnet2021.com

DotNet 2021

ORGANIZATION

plain concepts

IN COOPERATION WITH

FUNDACIÓN GOMAESPUMA

"Educando con una sonrisa."

SPONSORS

Microsoft

intelequia

Devs DNA™

My Public Inbox®

#DotNet2021

Daniela Solis

AI Team Lead

@danysolism
dsolis@plainconcepts.com

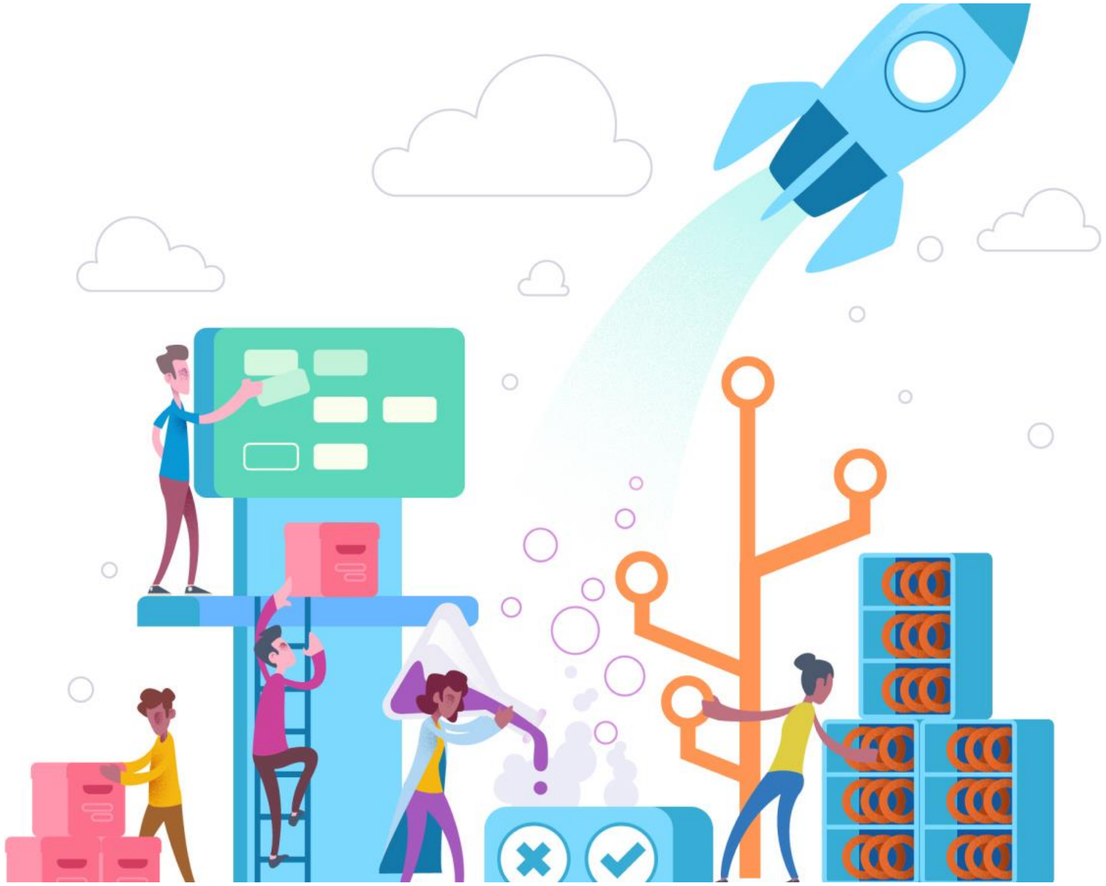
Rodrigo Cabello

AI Technical Lead

@mrcabellom
mrcabello@plainconcepts.com

Only a model that is running in production can bring value

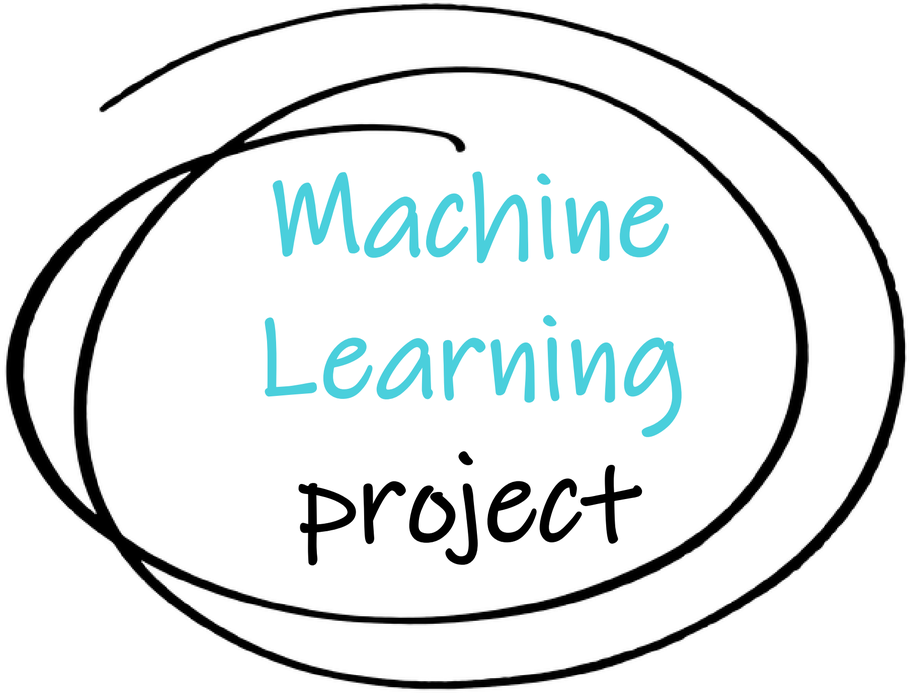
Agenda



- Machine Learning projects
- MLOps Workflow
- Productionalization
- Model Serving

Data Storage

Operations



Data Security

High availability

Access Control

Resource management

Machine Learning Roles

Data Scientist

Data engineer

Machine Learning Engineer

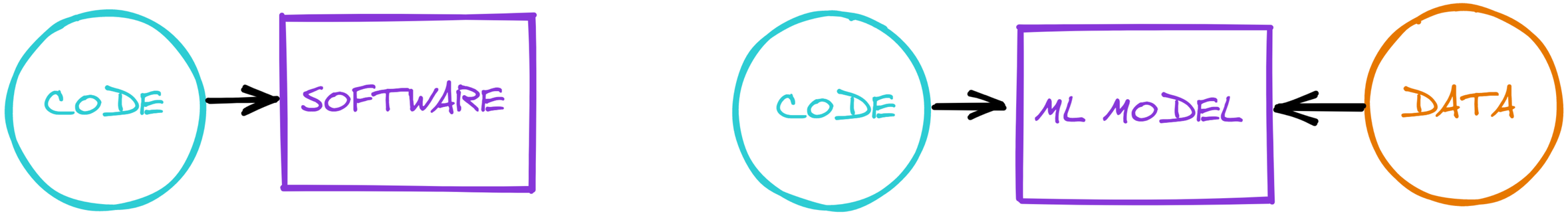
DevOps Engineer

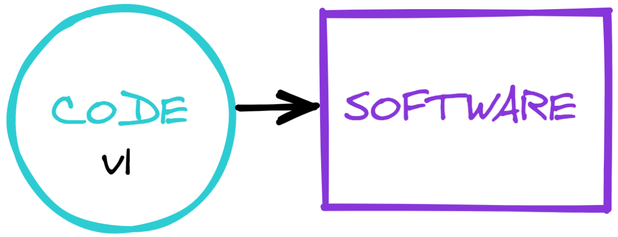
IT

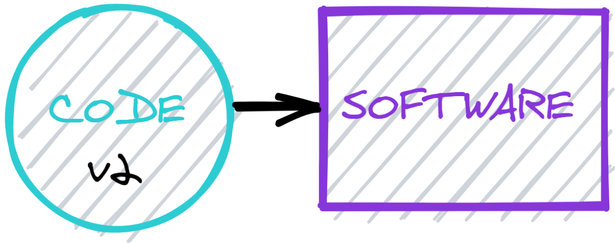
Business Owner

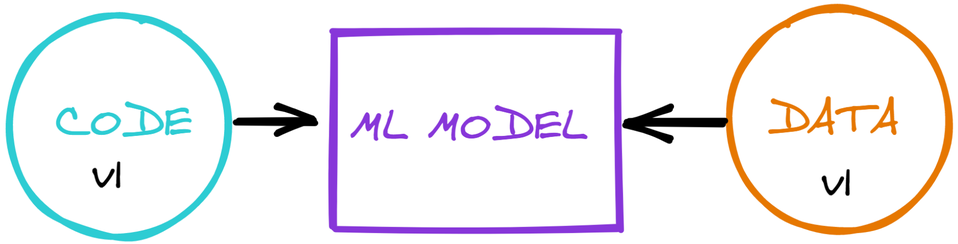
Manager

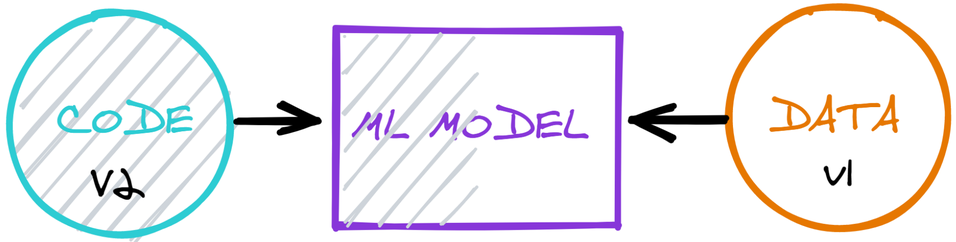
How is ML different from traditional software development?



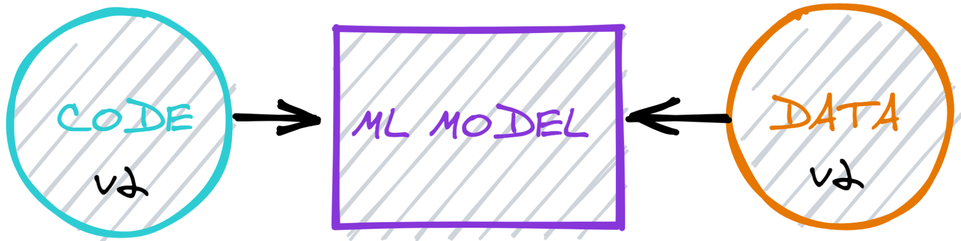

CODE
v1
SOFTWARE


CODE
v2
SOFTWARE

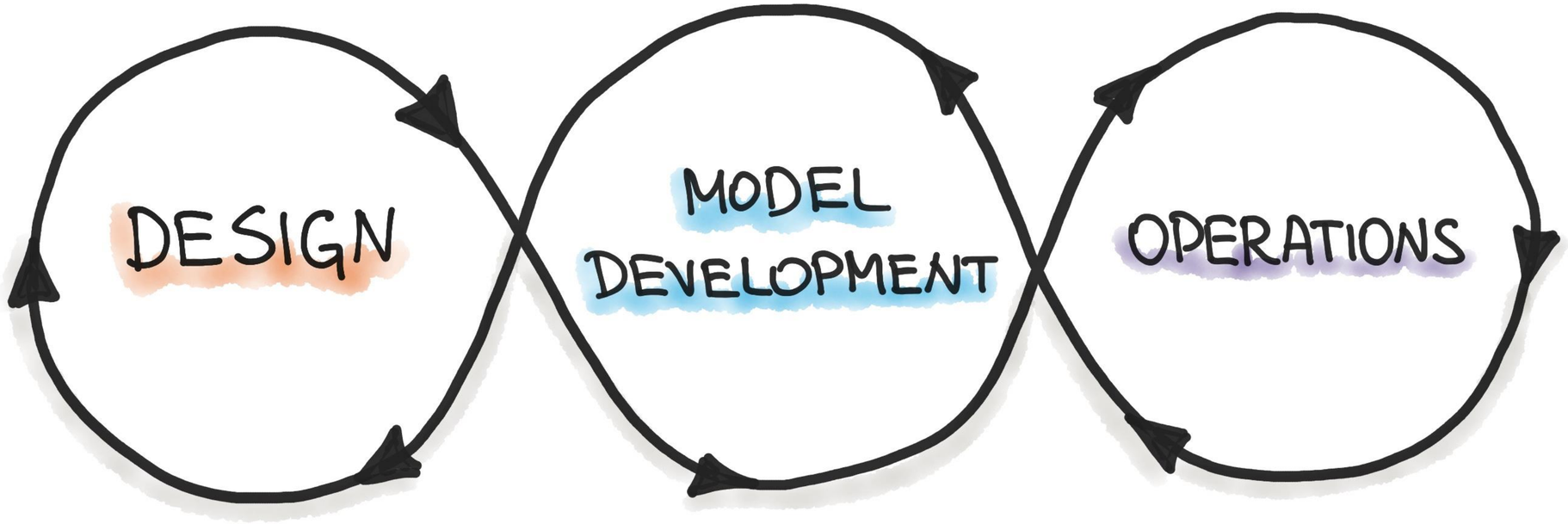

CODE
v1
ML MODEL
DATA
v1


CODE
v2
ML MODEL
DATA
v1

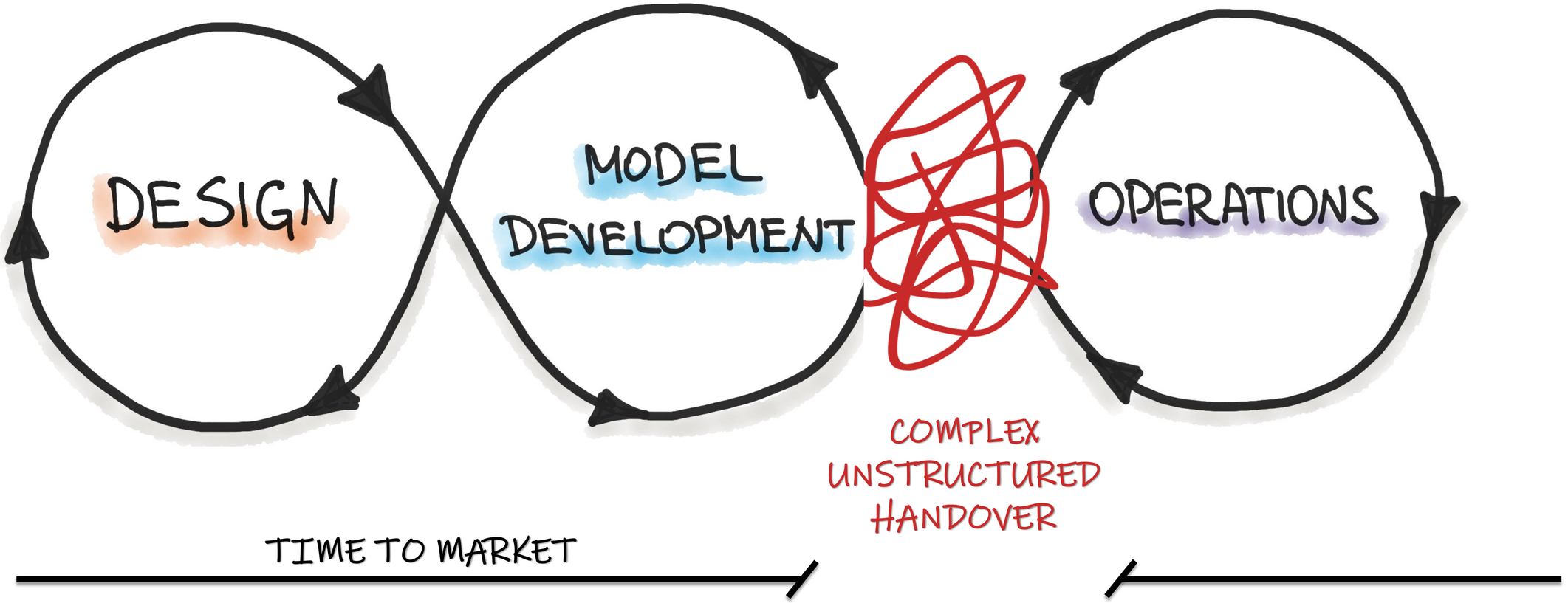

CODE
v1
ML MODEL
DATA
v2


CODE
v2
ML MODEL
DATA
v2

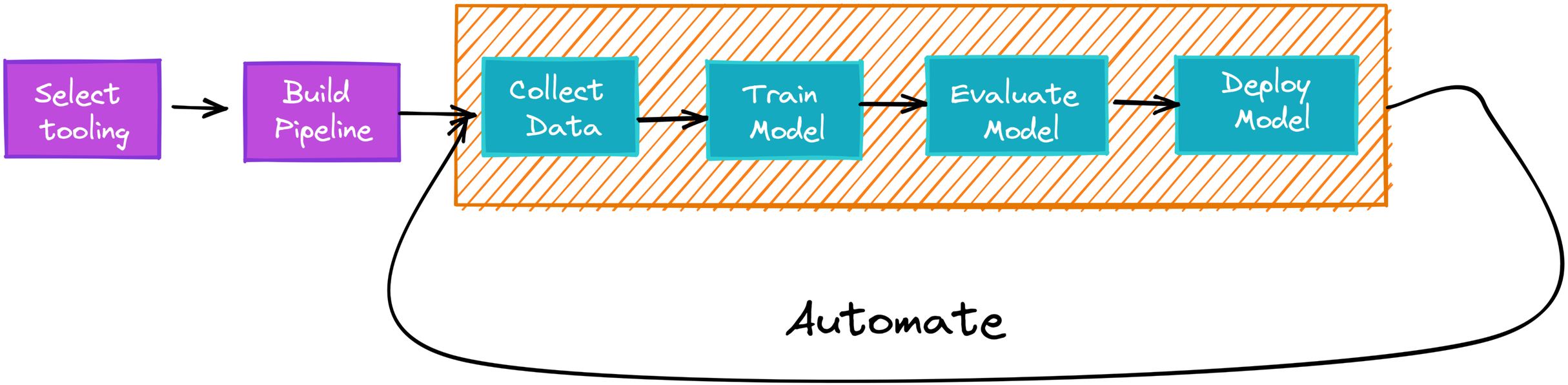
MLOps Workflow



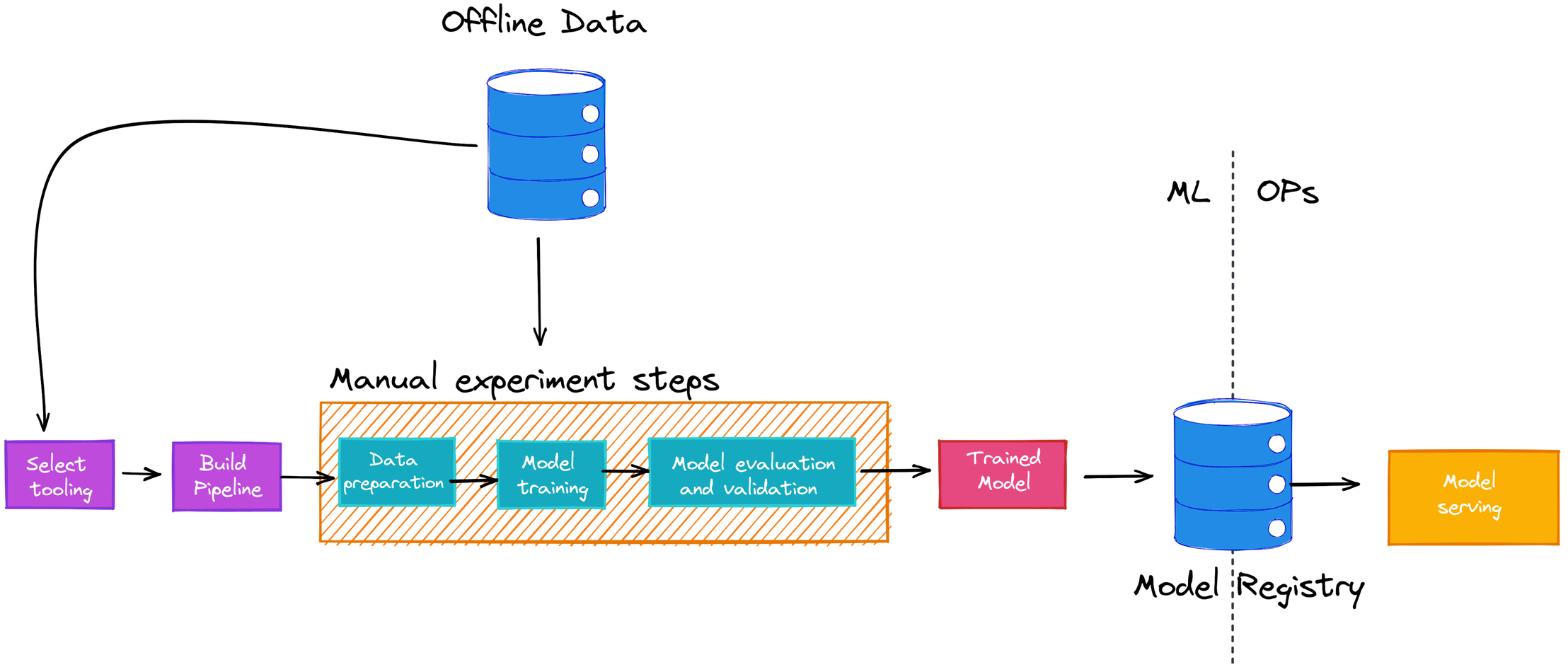
MLOps Workflow



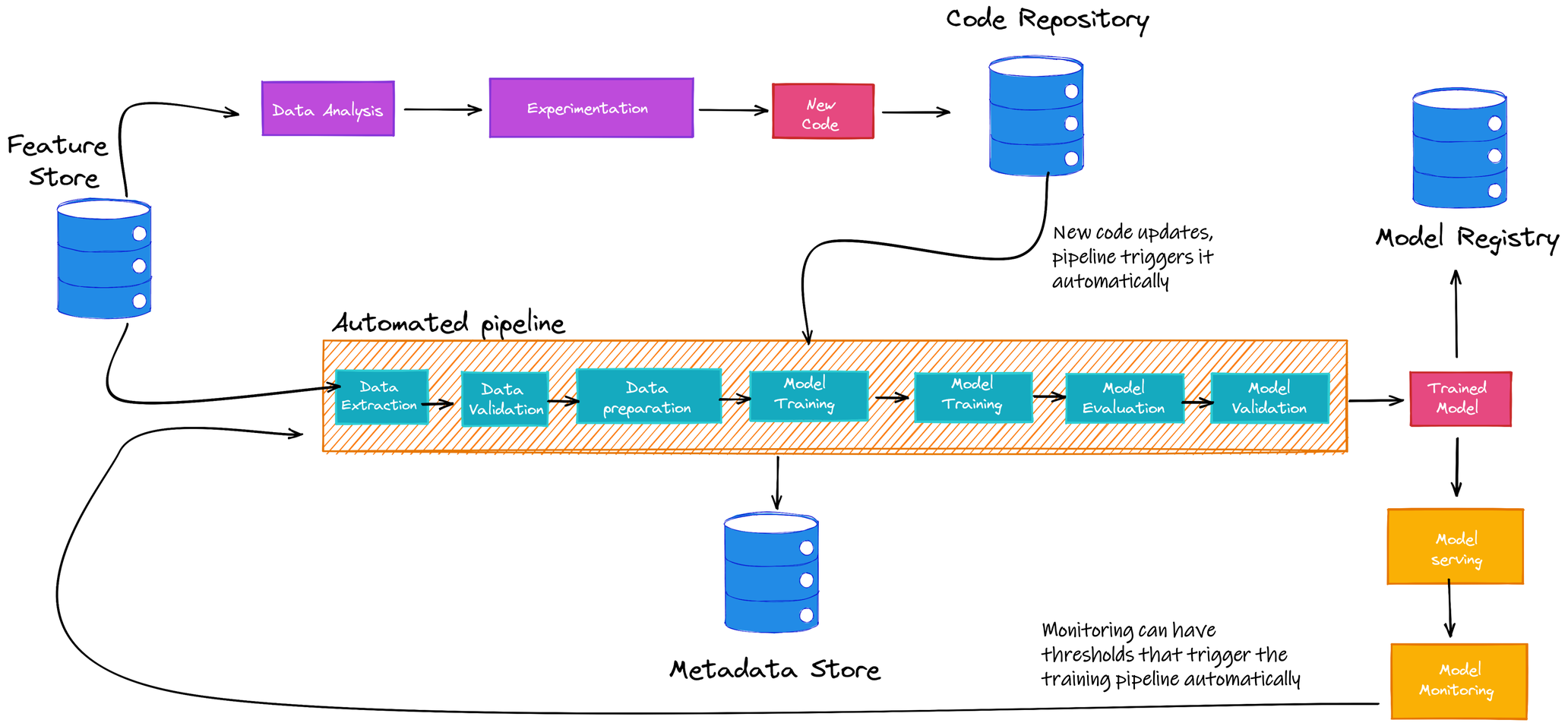
Machine Learining in production



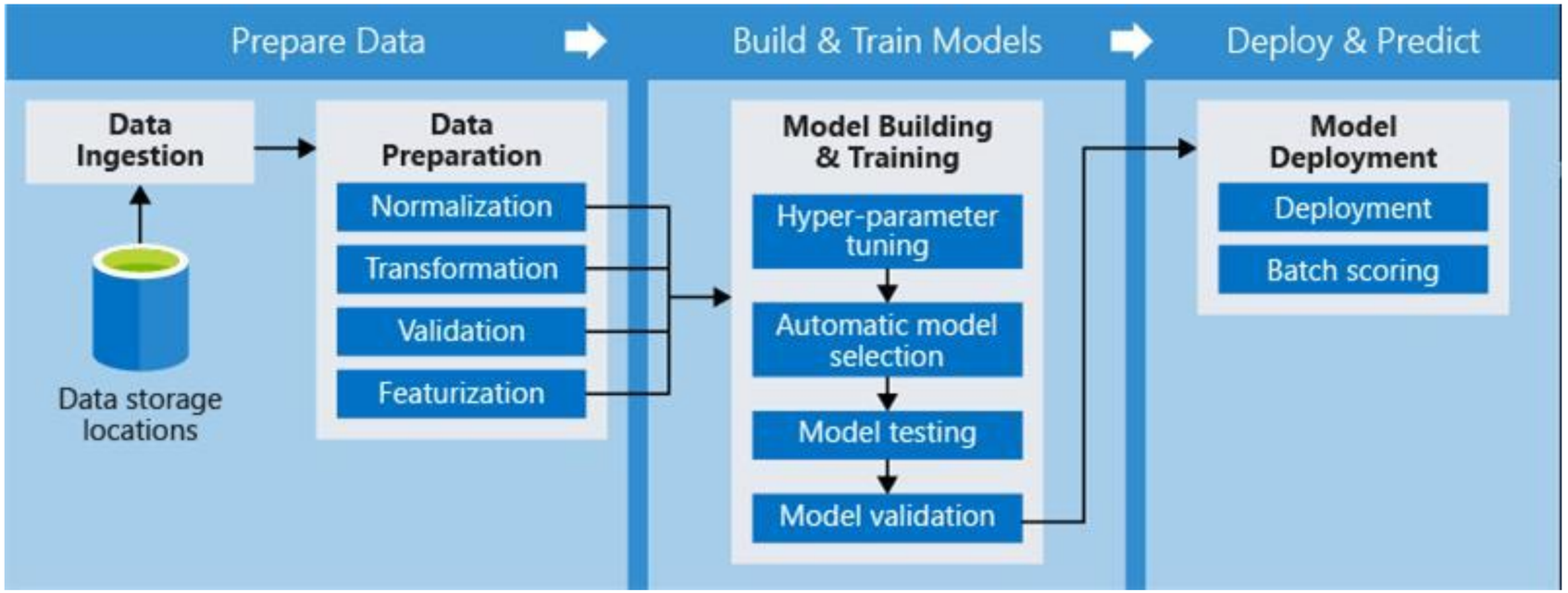
Productionalization- Manual Cycle



Productionalization- Automated Cycle



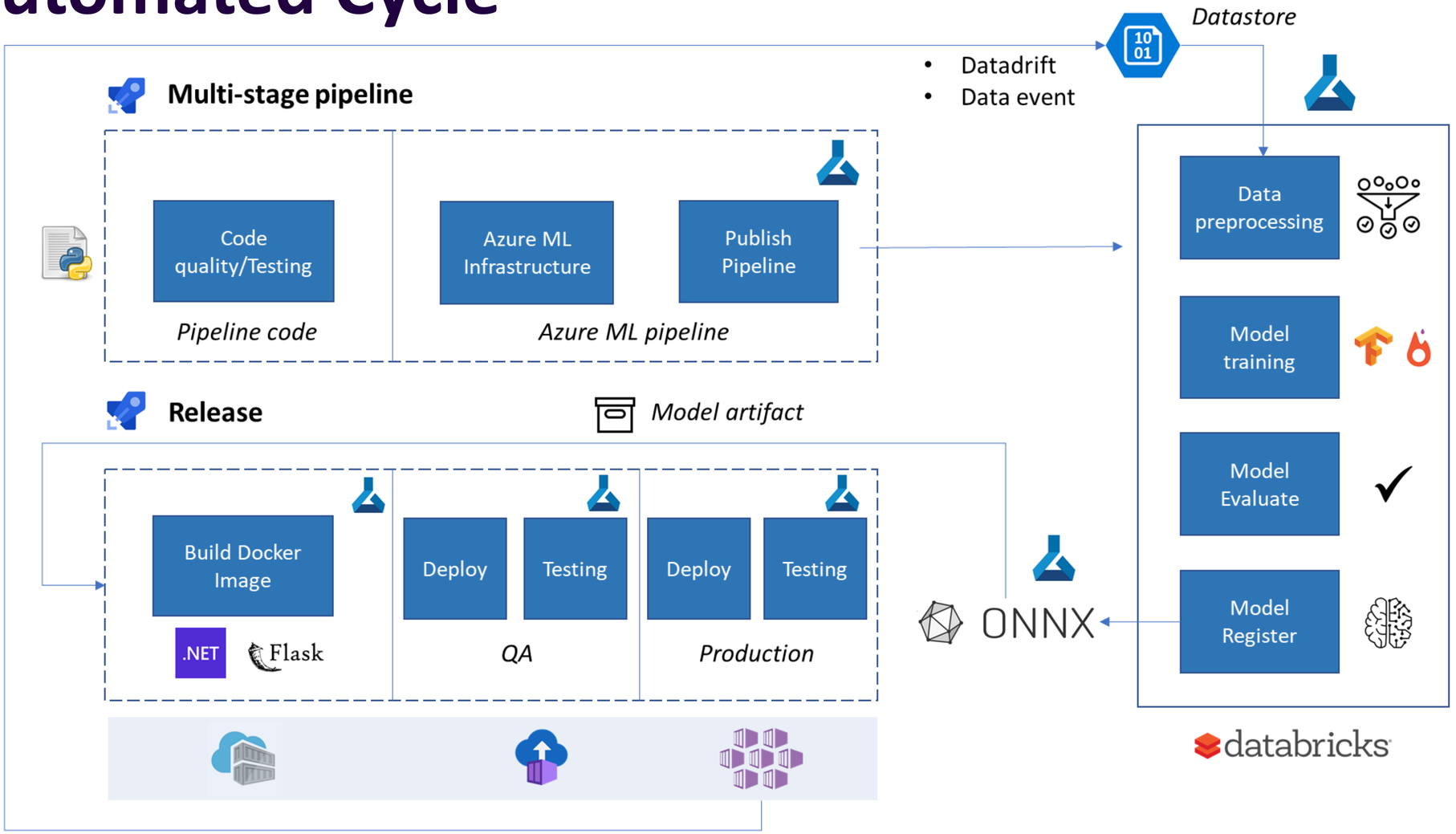
Machine Learning Pipeline - Azure

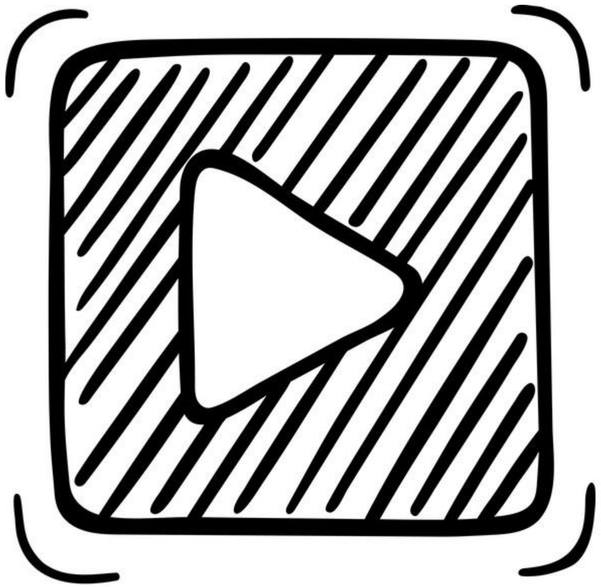


databricks

Azure Machine Learning compute

Our Automated Cycle





DEMO

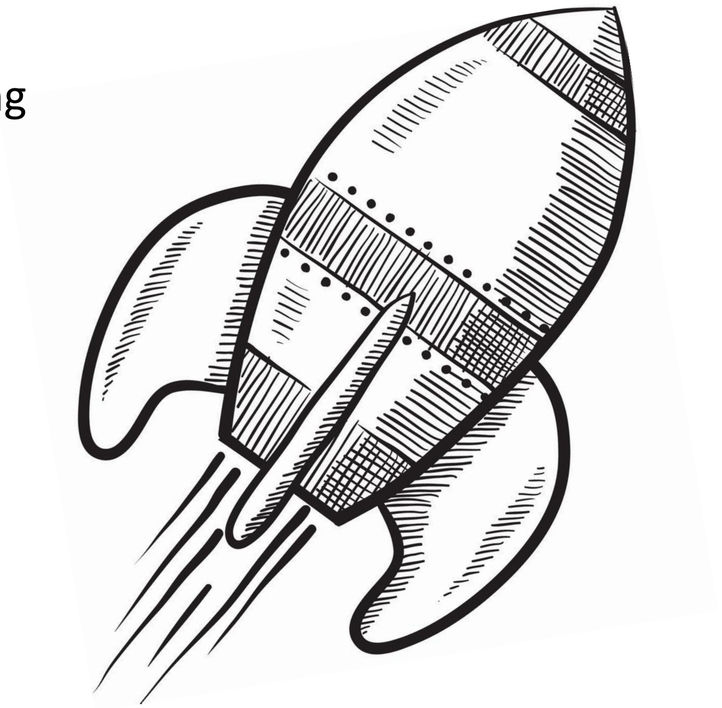
Model Serving

Deployment → Serving a ML modelo via API, application, or otherwise.

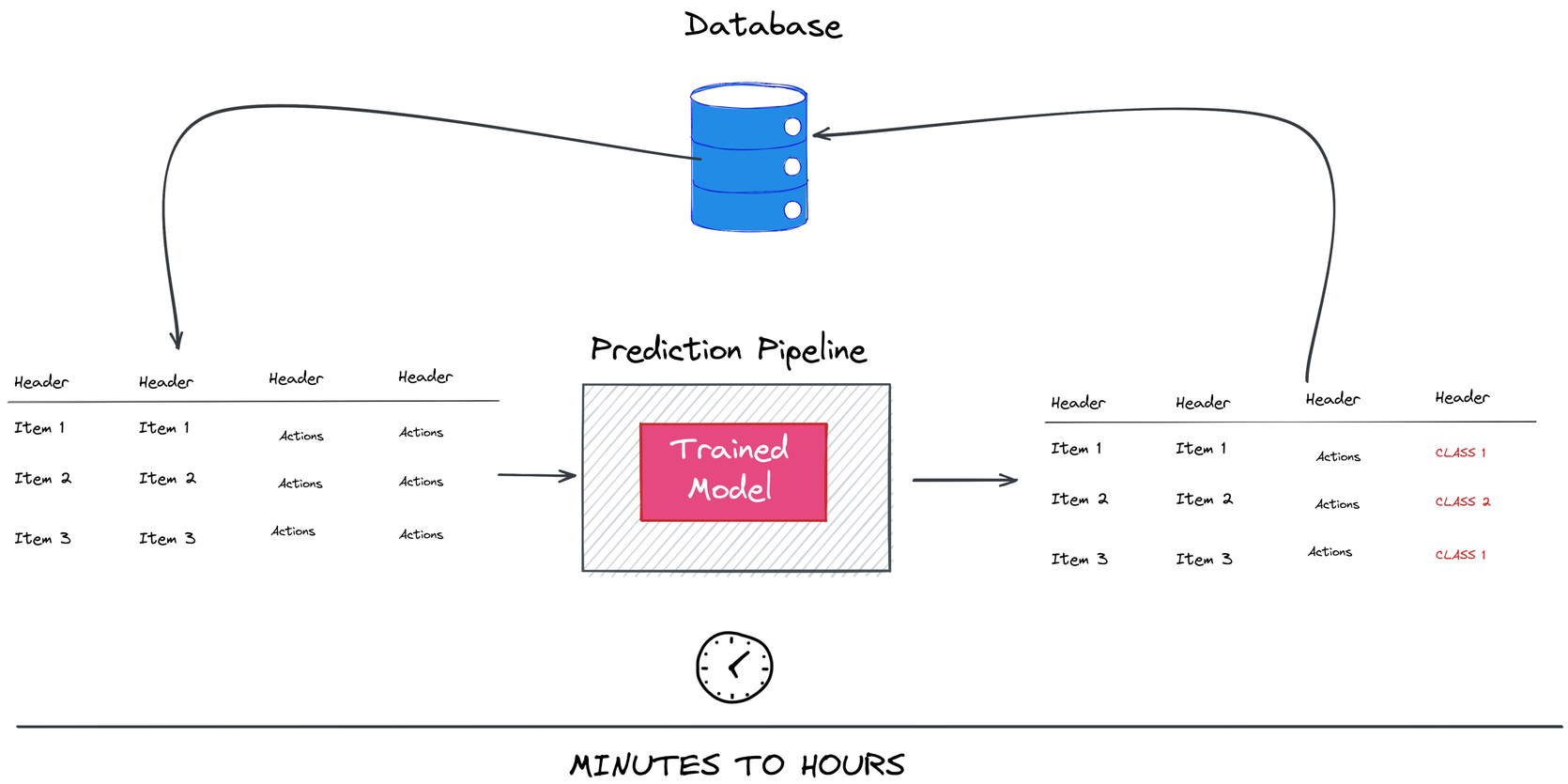
Inference → what the model does, once it is deployed. Whether it is making predictions, classifying input, or clustering data, it is always referred to as inference.

Types of model serving:

- **Batch Inference**
- **Online Inference**
- **Edge Inference**



Model Deployment – Batch Inference



Batch Serving

Suits any scenario where latency is not an issue, whenever predictions can be generated asynchronously on a batch of input samples. Especially when predictions are needed on intervals longer than an hour.

Some examples:

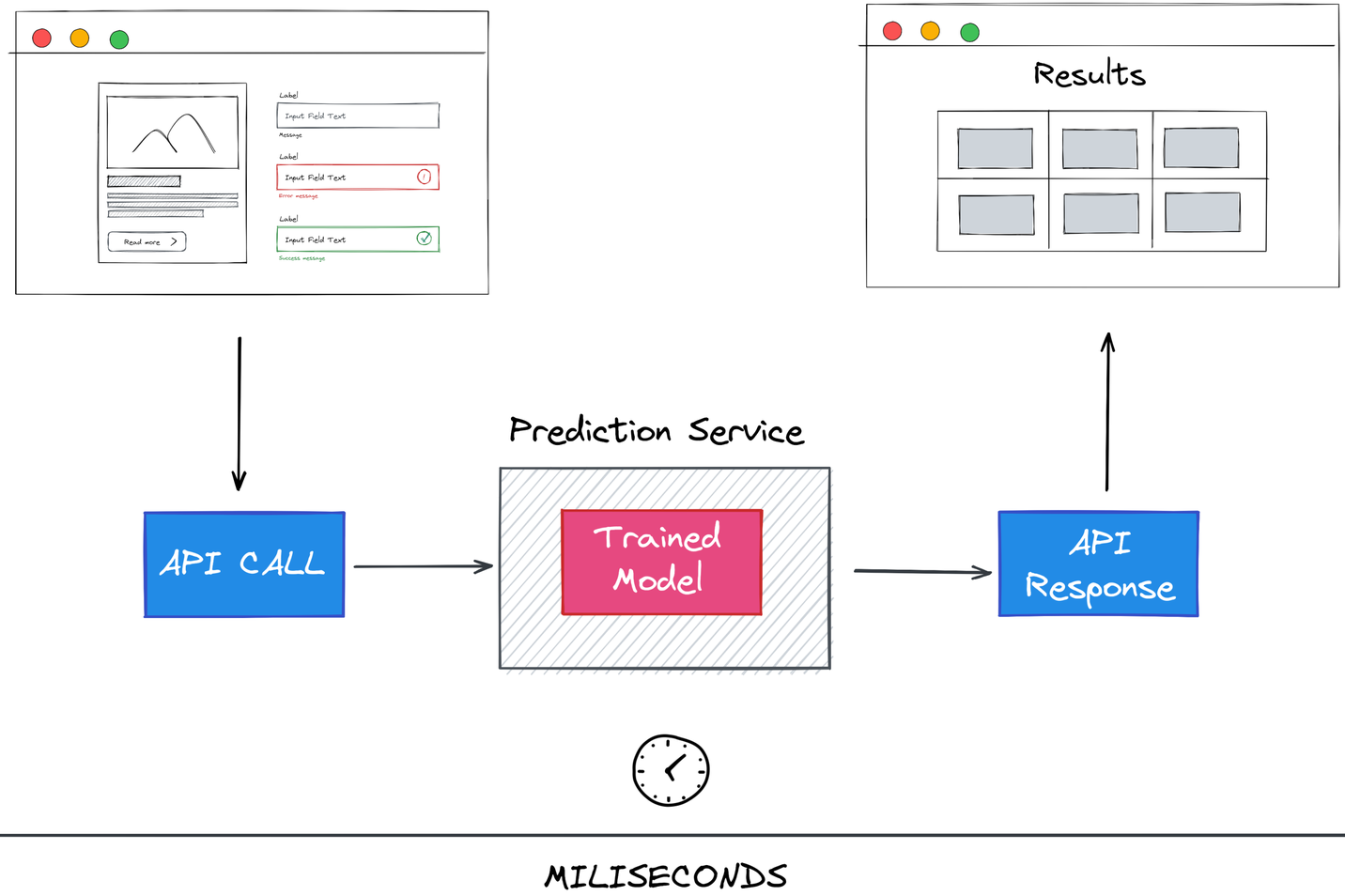
Databricks jobs→ create a Databricks job to run a notebook or JAR either immediately or on a scheduled basis.

MLFlow Serving → batch inference on Apache Spark using an online single node

Spark ML→ create a Job to do the inference on a scheduled basis. Using spark to read and do the inference.

Azure ML Pipeline →Azure Machine Learning provides a type of pipeline step specifically for performing parallel batch inferencing. Using the ParallelRunStep

Model Deployment – Online Inference

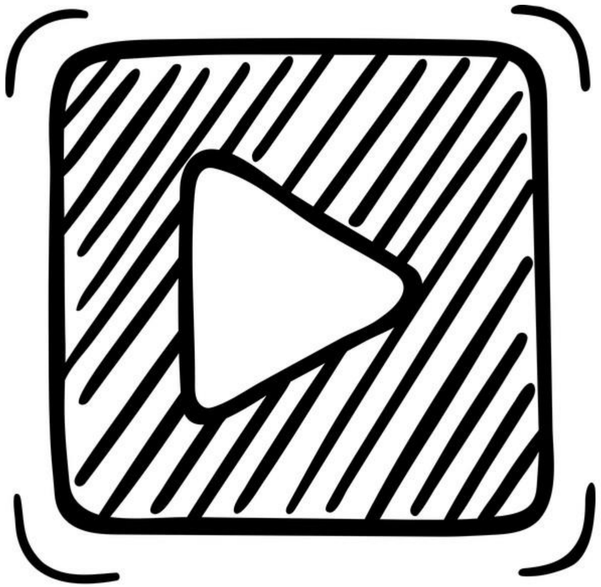


Online Serving

Direct embedding → Directly call the model as part of a larger program. This isn't just for apps; usually this is how robotics and dedicated devices work as well.

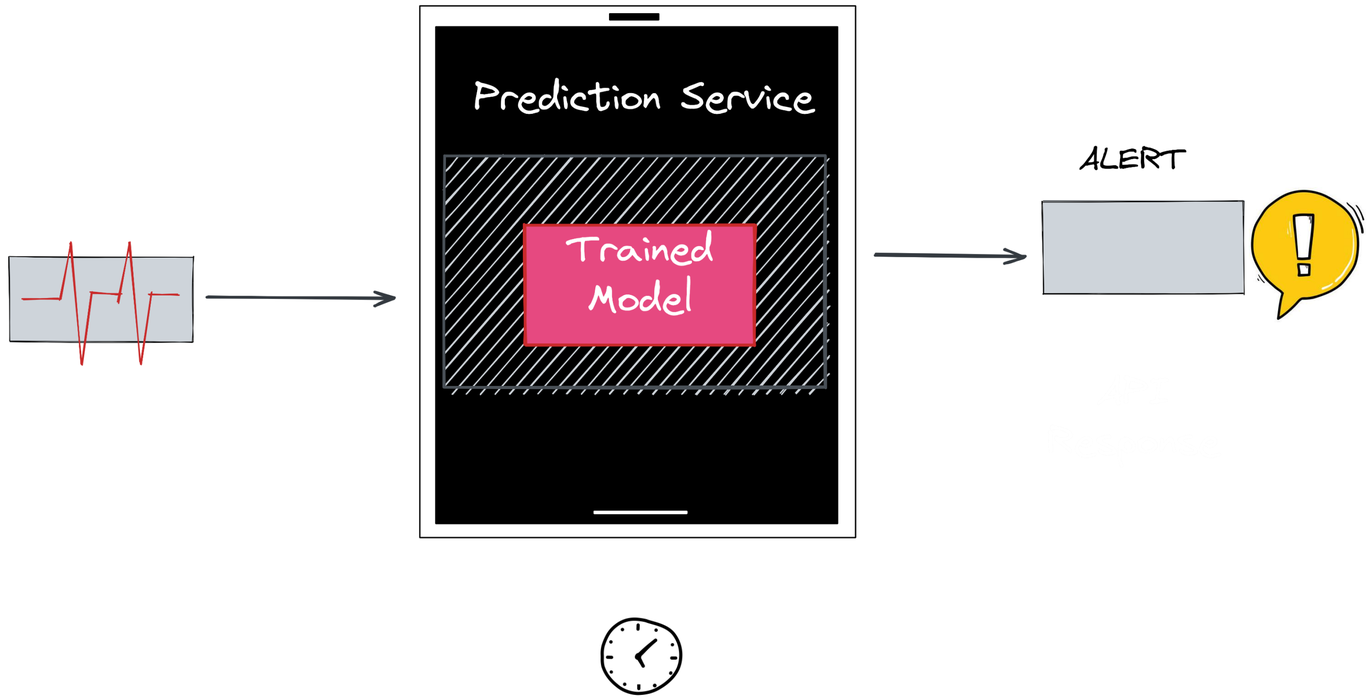
Model microservices → Model in a server side context. Treat each individual model (or each individual model version) as a separate service, usually using some sort of packaging mechanism like a Docker container


Model Server → An application built to manage and serve models. It allows you to upload multiple models and get distinct prediction endpoints for each of them.



DEMO

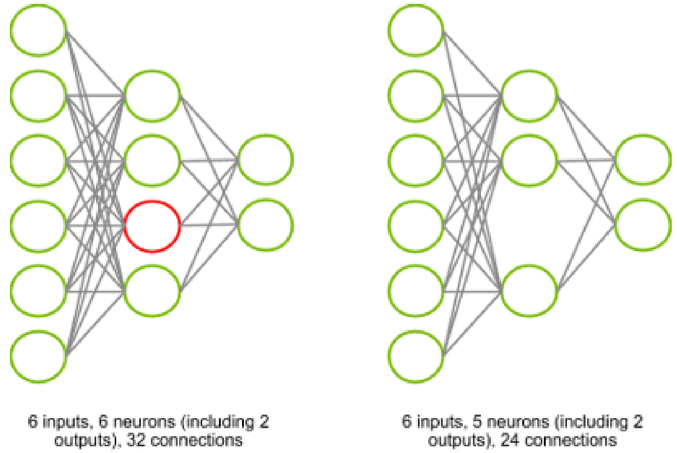
Model Deployment – Edge Inference



Deep Dive Edge

The goal of model compression is to achieve a model that is simplified from the original one without significantly diminished accuracy (size/latency)

Pruning.
Quantization.
Low-rank approximation and sparsity.
Knowledge distillation.
Neural Architecture Search (NAS).



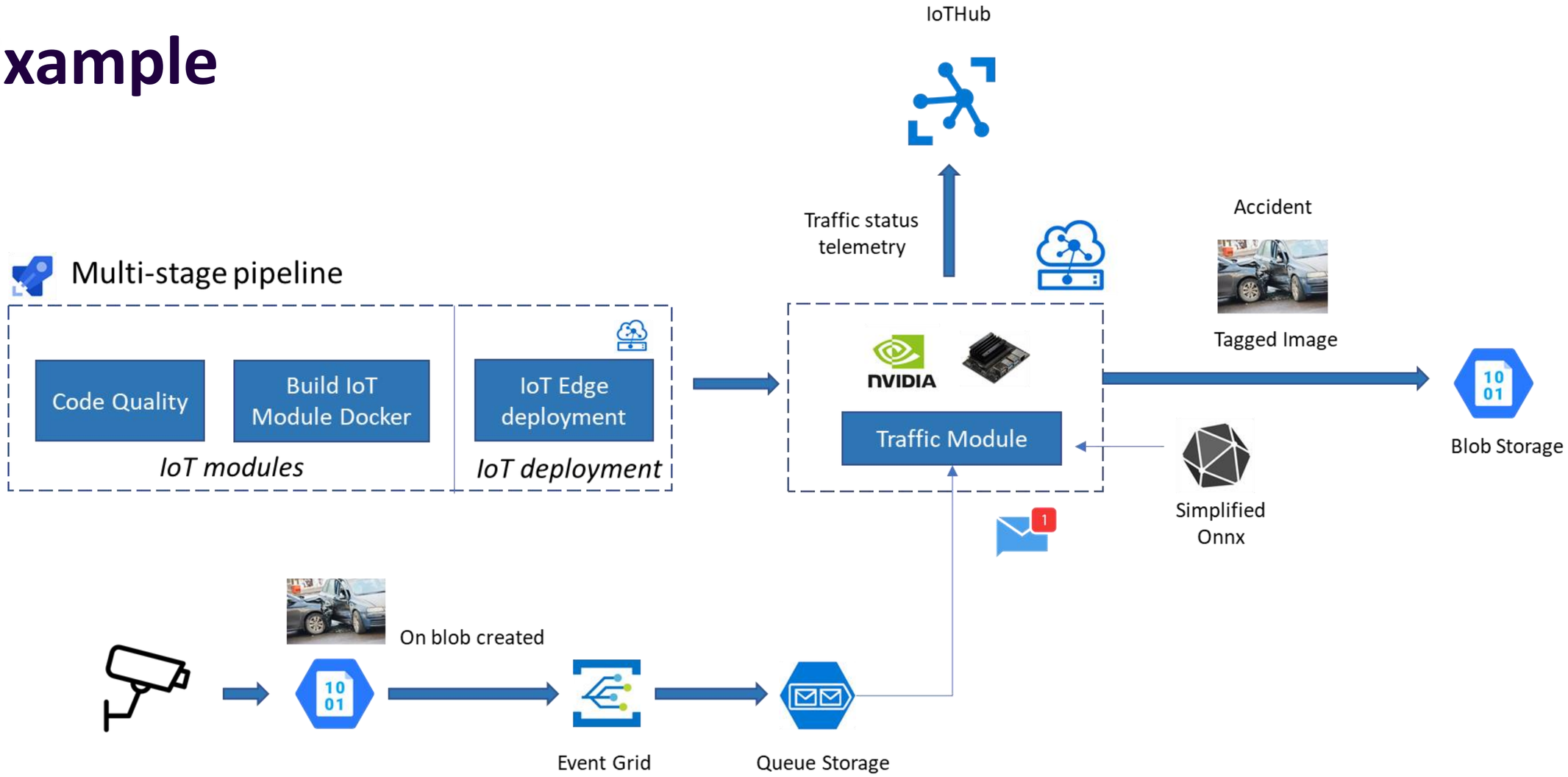
Onnx

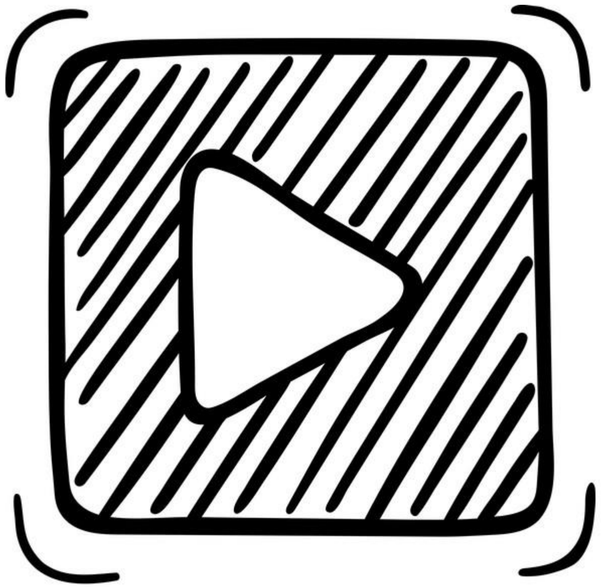
NVIDIA TensorRT

TensorFlow Lite

Core ML

Example


IoTHub
Traffic status telemetry
Accident
Multi-stage pipeline
Code Quality
Build IoT Module Docker
IoT Edge deployment
IoT modules
IoT deployment
NVIDIA
Tagged Image
Traffic Module
Blob Storage
Simplified Onnx
1
On blob created
Event Grid
Queue Storage



DEMO

DotNet 2021

ONLINE TECH CONFERENCE

www.dotnet2021.com

#DotNet2021

Thanks and ...
See you soon!

Thanks also to the sponsors. Without whom this would not have been posible.