Microsoft Azure

Project Batcomputer

A working DevOps implementation for Machine Learning

Ben Coleman **@BenCodeGeek**
Phil Harvey **@CodeBeard**

v0.0.3 (Beta)

Background

Motivation

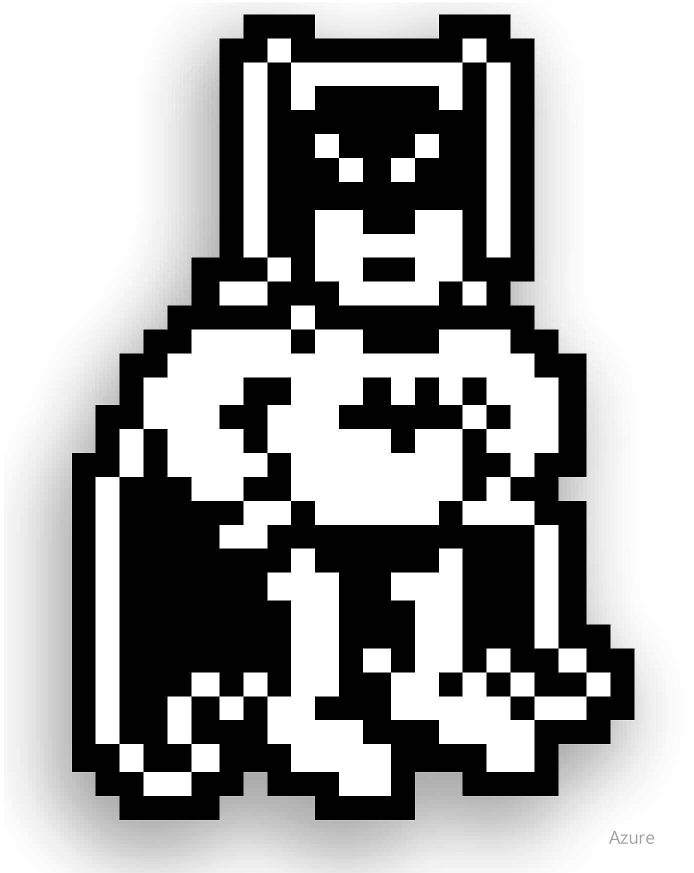
- Understand challenges in operationalisation of ML models
- Existing processes approaches deemed problematic
- “DevOps for AI”

Why Batcomputer?

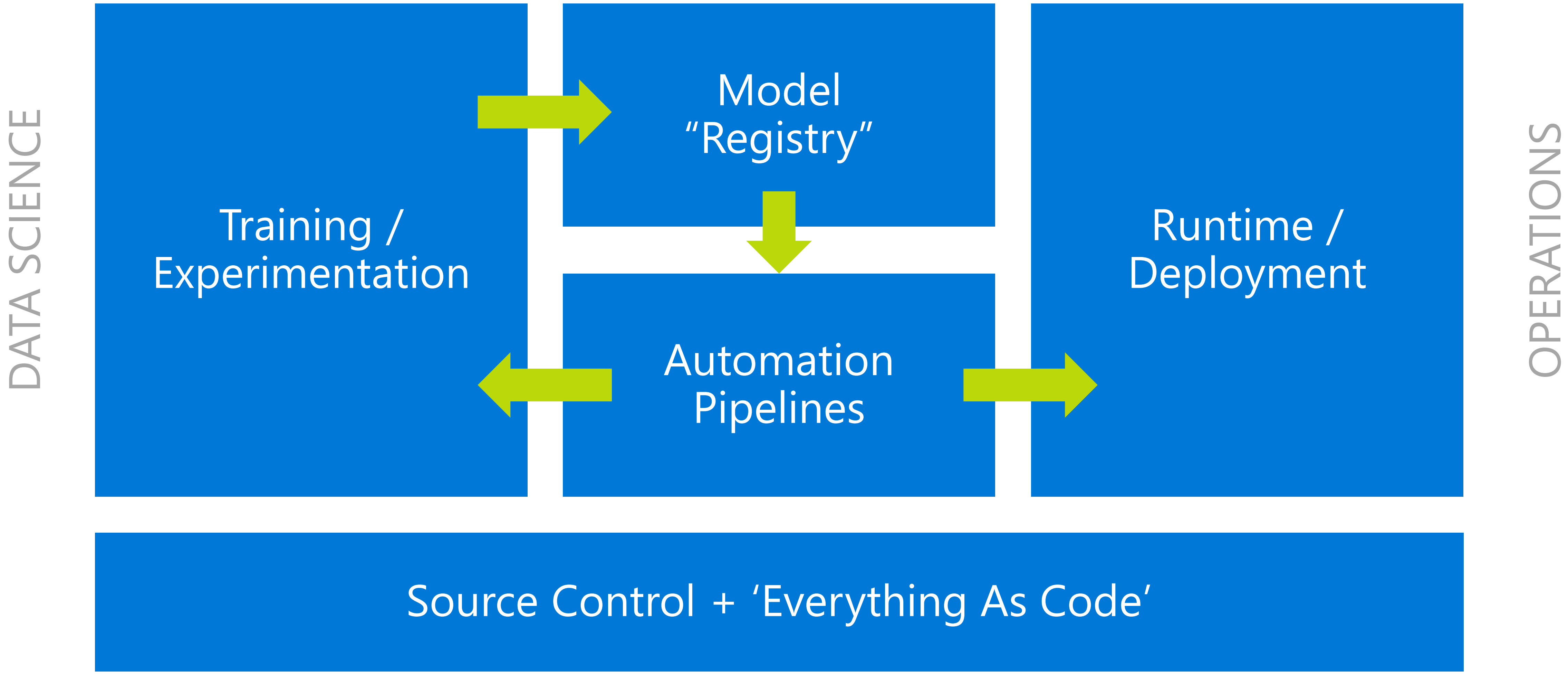
- Police recorded crime and outcomes data
- Source data as CSV - https://data.police.uk/data
- Build model of a given crime and region to predict – “Would you get caught?”

Core Principals & Benefits

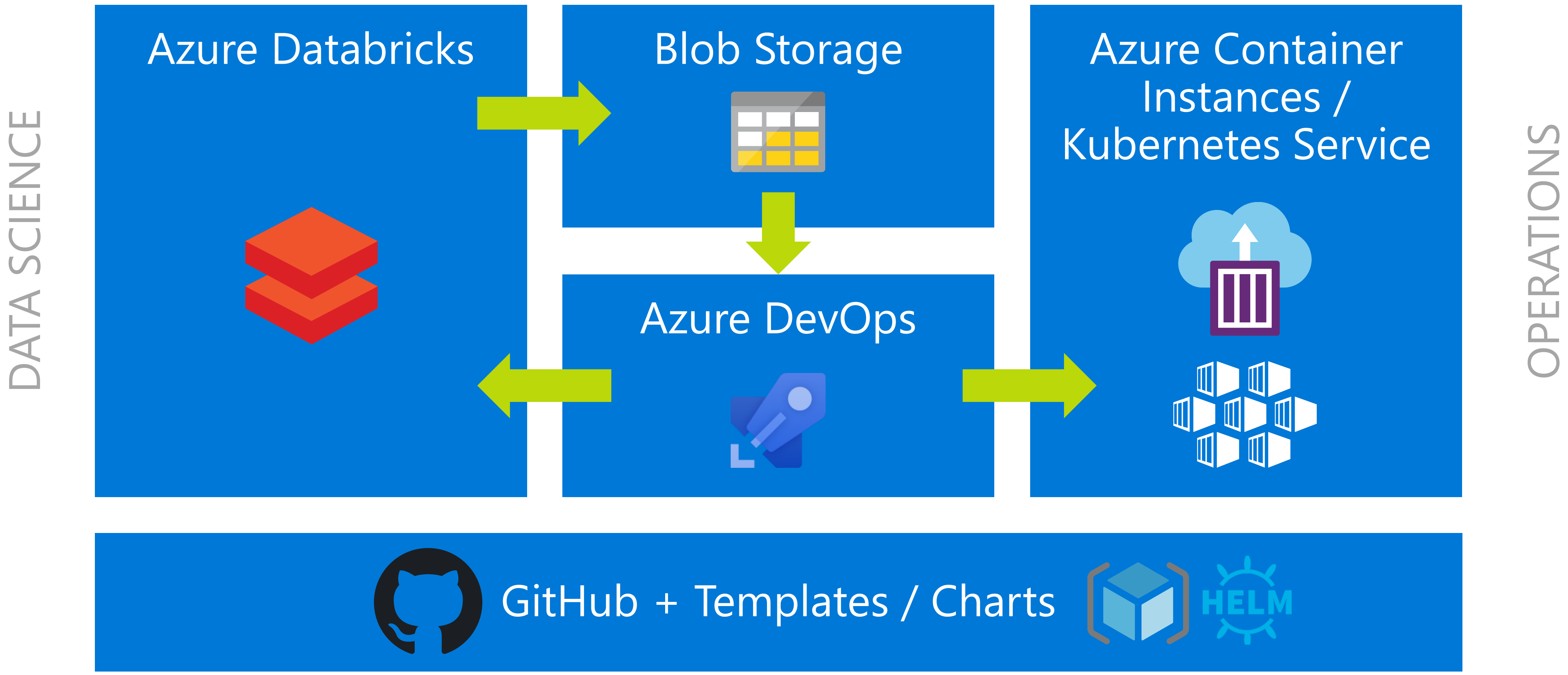
- Decouple model training experiments from operations/runtime
- Automated training, API builds & deployment
- Versioned models and API
- Config & infra as code
- Traceability



Conceptual Building Blocks



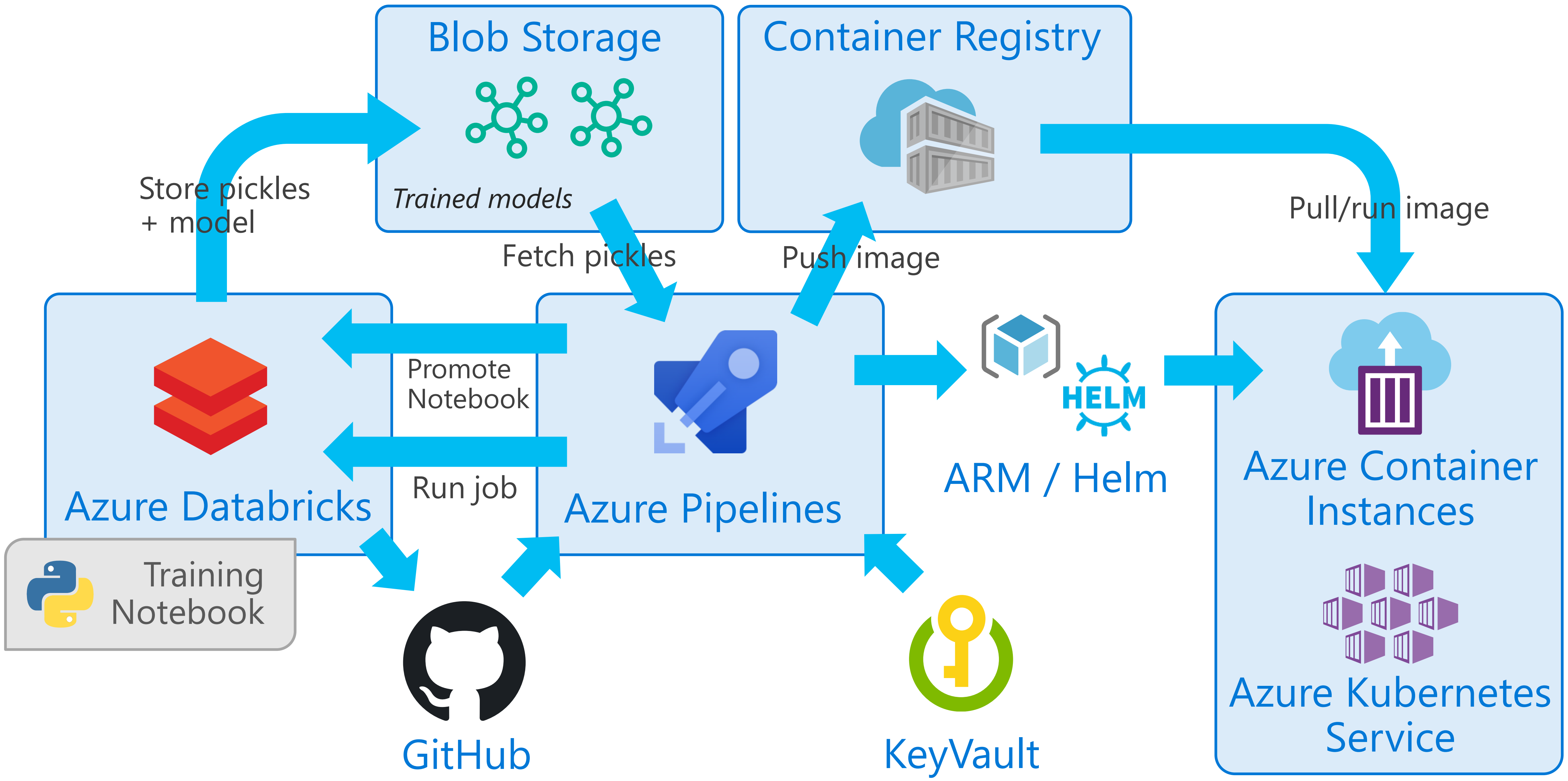
Conceptual Building Blocks – Project Batcomputer


DATA SCIENCE
Azure Databricks
Blob Storage
Azure DevOps
Azure Container Instances / Kubernetes Service
OPERATIONS
GitHub + Templates / Charts
HELM

Low Level Technology Stack

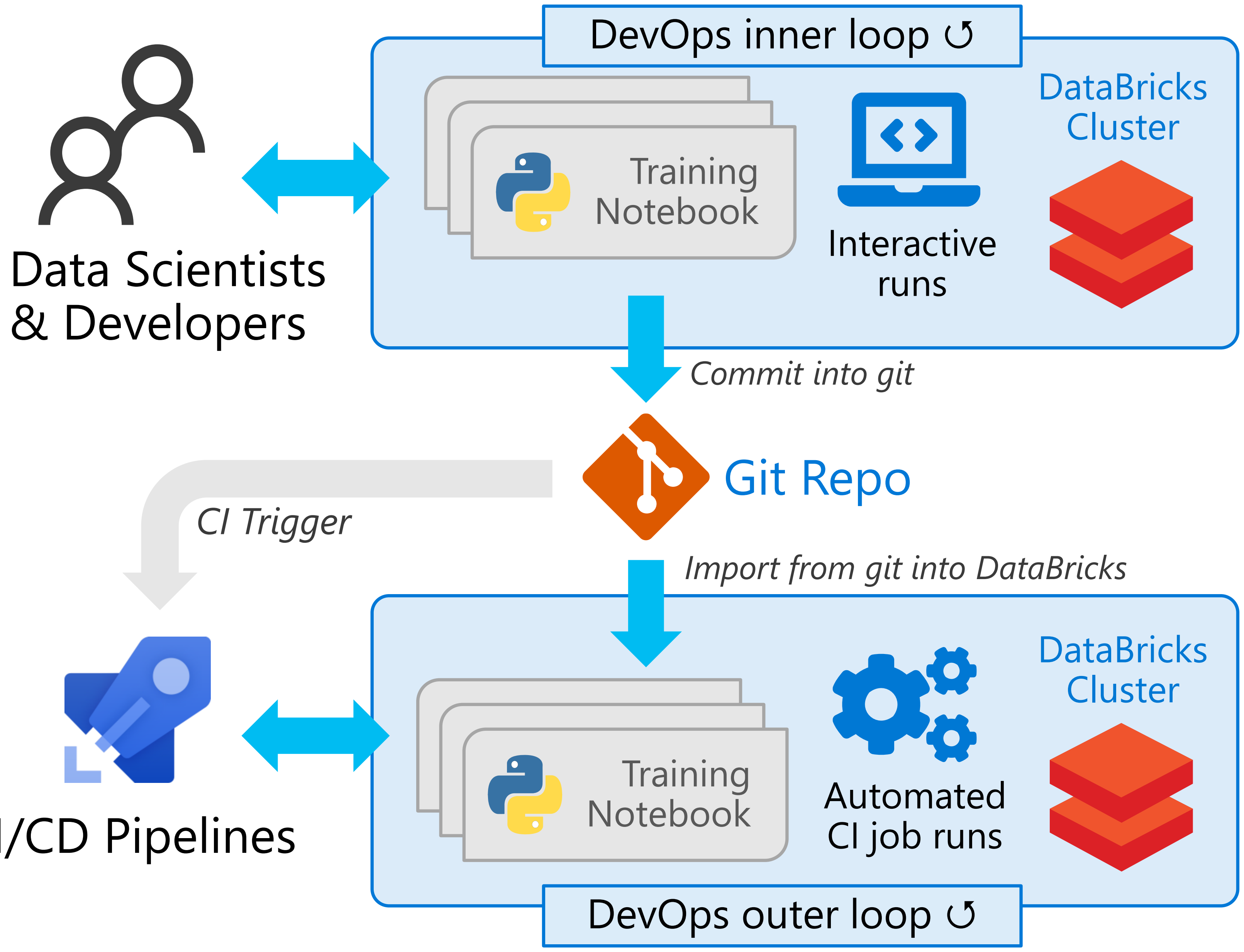
Swagger	← API niceness
Gunicorn	← HTTP Server
Flask	← Web framework
Pickle	← Serialisation
Scikit-Learn	← Main ML framework
Python	← Core language
Docker	← Container Runtime

Training & Deployment – End To End Flow



Core DevOps Practice - Continuous Integration

- Development & experimentation
- Central shared git repo
- CI triggered training & testing job runs



Pickling – Not Just The Model

scikit learn

```
{
“gender”: {
  “male”: 0,
  “female”: 1
}
```

```
[
  “conviction”,
  “dropped”,
  “settled”
]
```

model.pkl	lookup.pkl	flags.pkl
Scikit-learn model/classifier	Python dictionary of dictionaries	Python array
Standard object rehydration, version sensitive	Mapping parameters/strings to num for predict function	Maps output of prediction function to human readable strings or labels

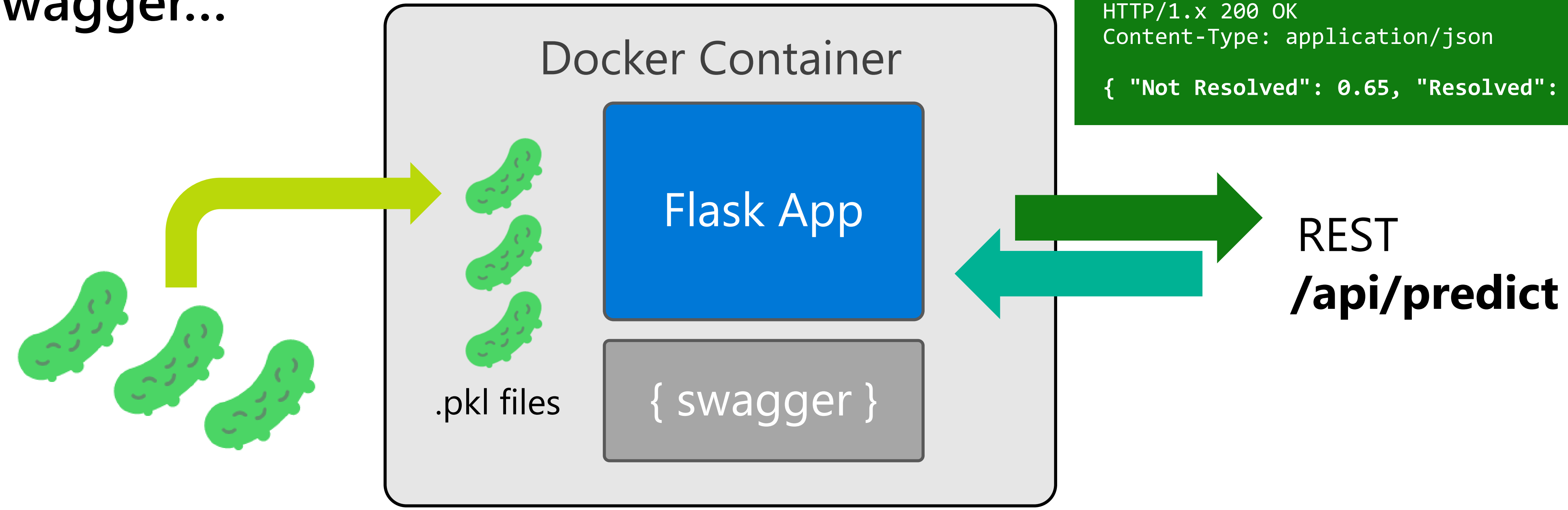
Wrapper App – Flask API

- Uses Flask web framework + Gunicorn
- Creates RESTful API for model parameters
- Consumes .pkl files
- Swagger...

```
POST /api/predict
{
  "force": "Thames Valley Police",
  "crime": "Bicycle theft",
  "month": 10
}
```

```
HTTP/1.x 200 OK
Content-Type: application/json

{ "Not Resolved": 0.65, "Resolved": 0.35 }
```



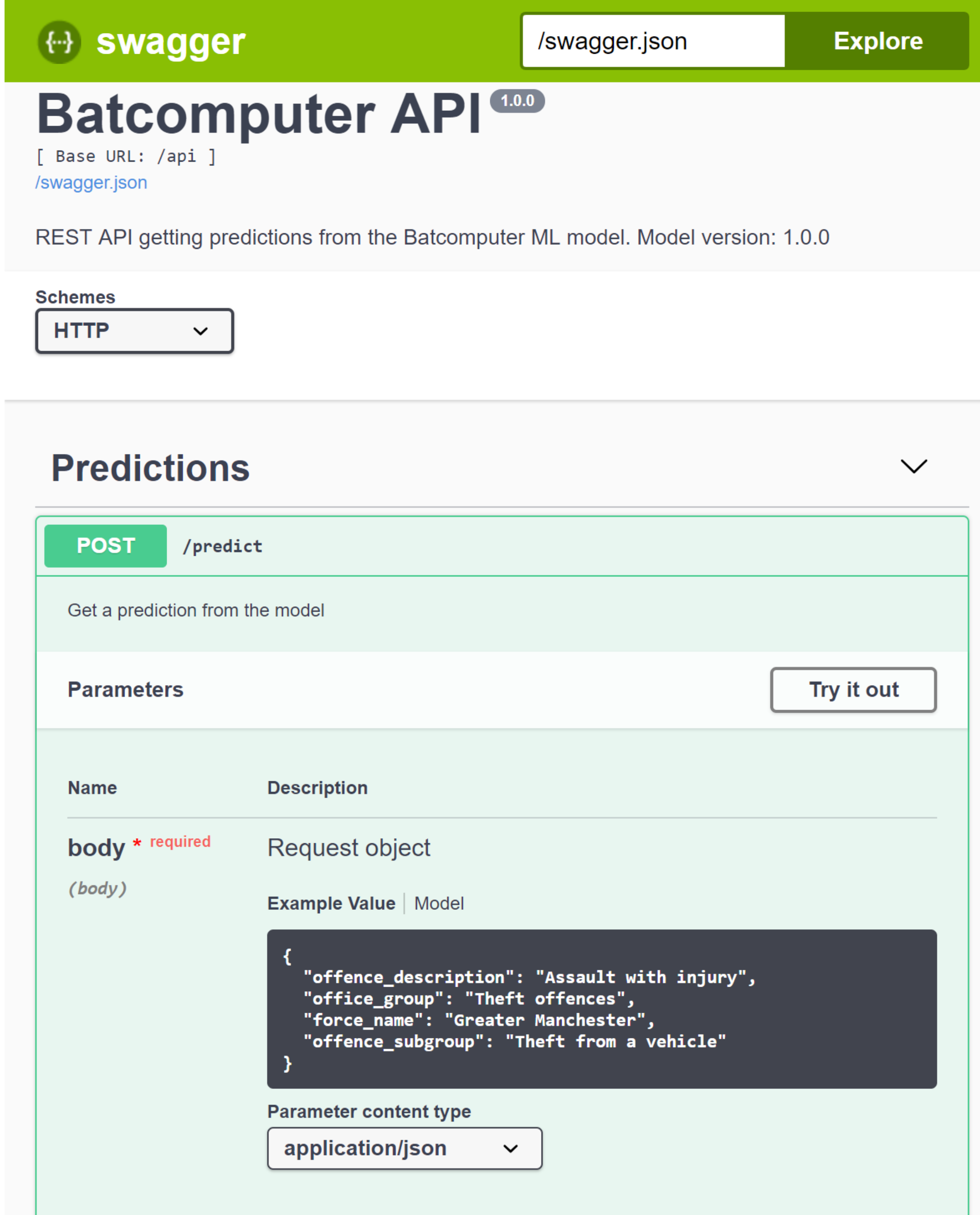
Some Decision Points

- Include model in container image or fetch at runtime?
- Wrapper app – make generic or tied into model?
- What are my API parameters?
- Convention based model registry – good enough?
- Use more robust web server than Flask?
- Automate training?

Swagger

- We want to be RESTful
- Dynamic
  - Generated from lookup & flags pickles at runtime
- Swagger UI
  - For testing & eye candy

OPENAPI INITIATIVE



Batcomputer API 1.0.0

[Base URL: /api]

/swagger.json

REST API getting predictions from the Batcomputer ML model. Model version: 1.0.0

Schemes: HTTP

Predictions

POST /predict

Get a prediction from the model

Parameters

Name	Description
body * required (body)	Request object

Example Value | Model

```
{
  "offence_description": "Assault with injury",
  "office_group": "Theft offences",
  "force_name": "Greater Manchester",
  "offence_subgroup": "Theft from a vehicle"
}
```

Parameter content type: application/json

Semantic Versioning

MAJOR Version

- Incompatible API or other breaking changes
- **Scoring inputs / feature changes i.e. API change**

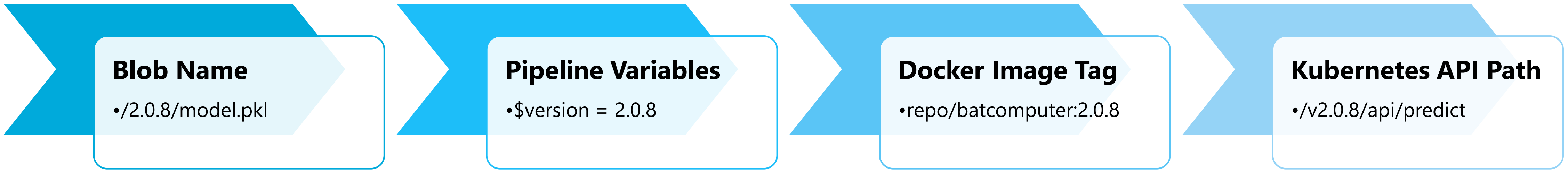
MINOR version

- Add functionality in a backwards-compatible manner
- **Trained using different parameters/classifiers, but API same**

PATCH version

- Backwards-compatible bug fixes
- **Trained the model on different data**

Versioning – Touches Everything



Also…

- Resource names in Azure controlled via ARM templates
- ACI DNS names & prefixes,
  e.g. `batcomputer-2-0-8.westeurope.azurecontainer.io`
- Object names in Kubernetes (pods, services), controlled via Helm chart

Some Learnings

- Pickled Scikit-learn models are version sensitive
- Keep Python version in sync with DataBricks
- Installing numpy, scipy and scikit-learn is SLOW, pre-build base image
- Version number is the key parameter for the whole deployment process
- Writing your own wrapper isn’t hard
- DataBricks has a great CLI & API
- DataBricks can be used with CI but it’s not obvious

Microsoft Azure