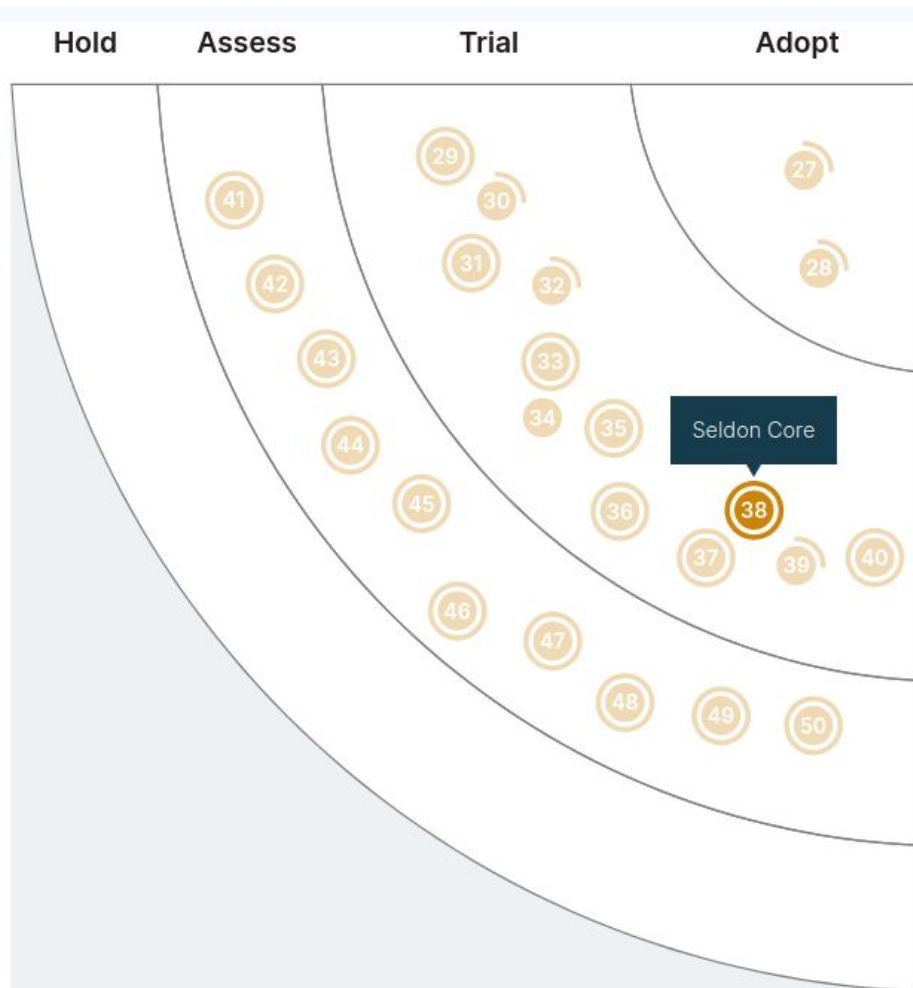# Seldon Core

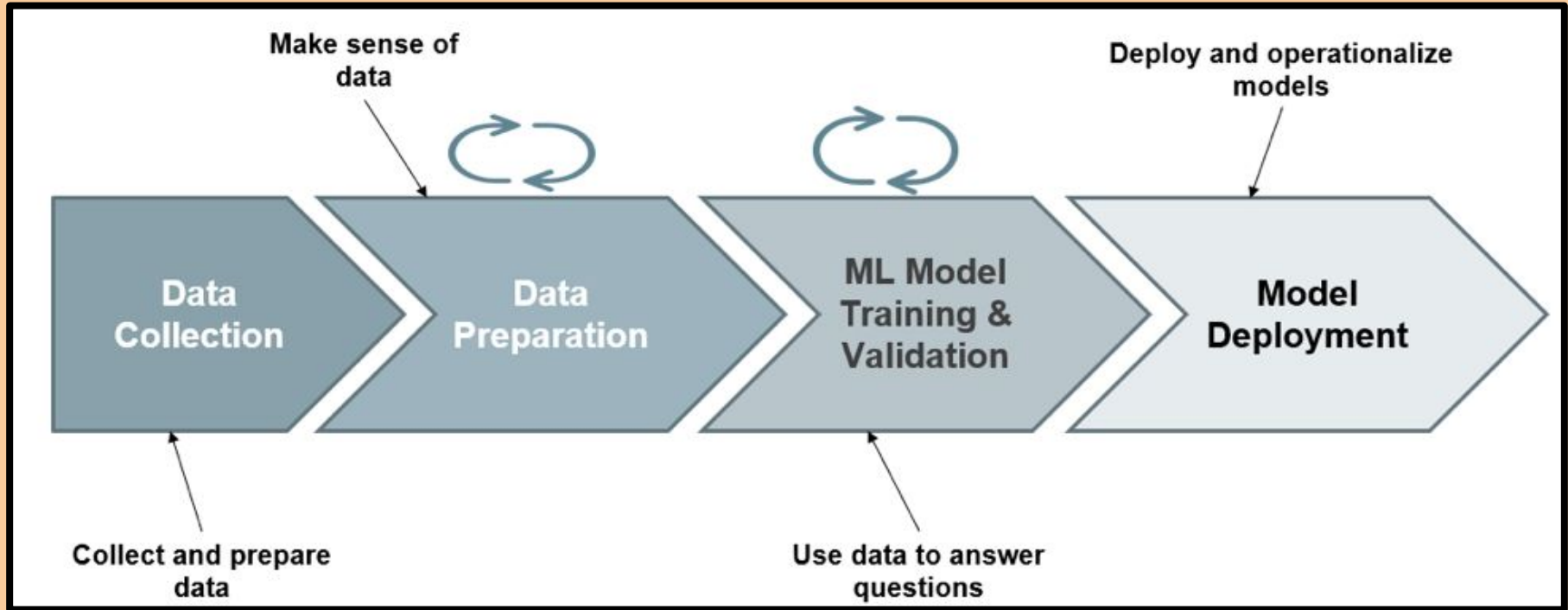## 38. Seldon Core

Seldon Core is an open-source platform to package, deploy, monitor and manage machine learning models in Kubernetes clusters. With out-of-the-box support for several machine-learning frameworks, you can easily containerize your models using prepackaged inference servers, custom inference servers or language wrappers. With distributed tracing through Jaeger and model explainability via Alibi, Seldon Core addresses several last-mile delivery challenges with machine learning deployments, and our data teams like it.

View blip history ›

Machine Learning overview

Make sense of data

Deploy and operationalize models

Data Collection

Data Preparation

ML Model Training & Validation

Model Deployment

Collect and prepare data

Use data to answer questions

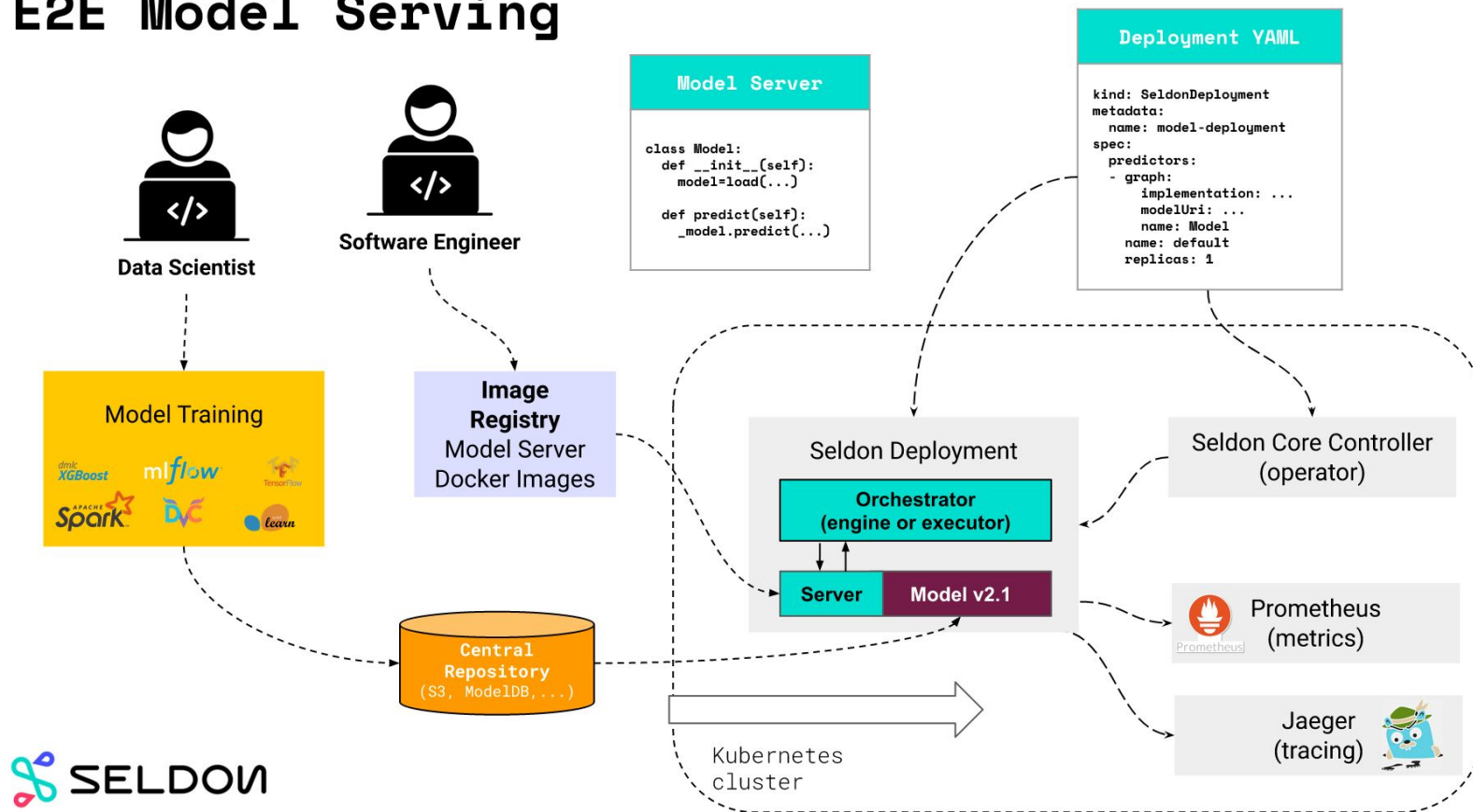Image source: https://www.snaplogic.com/blog/revolutionize-your-business-with-intelligent-integration

Kubernetes overview

- Gives you a way to centrally manage deploying containers to multiple nodes.
- You can tell the Kubernetes API that you want 3 objects:
  - A Deployment of a Node.js container with 2 replicas
  - A Service to map port 3000 on the containers to port 80 internally to other apps inside Kubernetes at http://pottery-barn-app
  - An Ingress to tell a load balancer that you'd like to expose your Service to the world at https://bobert-family-pottery-barn.com
- It's declarative — you tell it the definition of what you want, and the controllers inside Kubernetes make it happen.

Seldon Core overview

- Gives you a way to deploy your trained ML model, with extra features.
- Automatically generates:
    - OpenAPI docs (Swagger)
    - Prometheus metrics
    - Jaeger traces
- Provides premade servers for popular frameworks like scikit-learn, Tensorflow, and XGBoost.
    - And you then just provide the location of the model in blob storage (like s3).
- Includes the ability to set up pre-processors for the data before the model server receives it.

E2E Model Serving

Data Scientist

Software Engineer

Model Server

```
class Model:
    def __init__(self):
        model=load(...)

    def predict(self):
        _model.predict(...)
```

Deployment YAML

```
kind: SeldonDeployment
metadata:
  name: model-deployment
spec:
  predictors:
  - graph:
      implementation: ...
      modelUri: ...
      name: Model
    name: default
    replicas: 1
```

Model Training

dmlc **XGBoost** ml*flow* TensorFlow

APACHE **Spark** DVC learn

Image Registry
Model Server
Docker Images

Central Repository
(S3, ModelDB,...)

Seldon Deployment

Orchestrator
(engine or executor)

Server   Model v2.1

Seldon Core Controller
(operator)

Prometheus
(metrics)

Jaeger
(tracing)

Kubernetes cluster

SELDON

Image source: https://docs.seldon.io/projects/seldon-core/en/latest/workflow/overview.html

Questions?