FROM CONCEPT TO DEPLOYMENT:

THE JOURNEY OF ML OPS IN MODERN MACHINE LEARNING

Digital Liechtenstein Webinar
Jonas Bokstaller
11th of March 2025

UNIVERSITÄT LIECHTENSTEIN

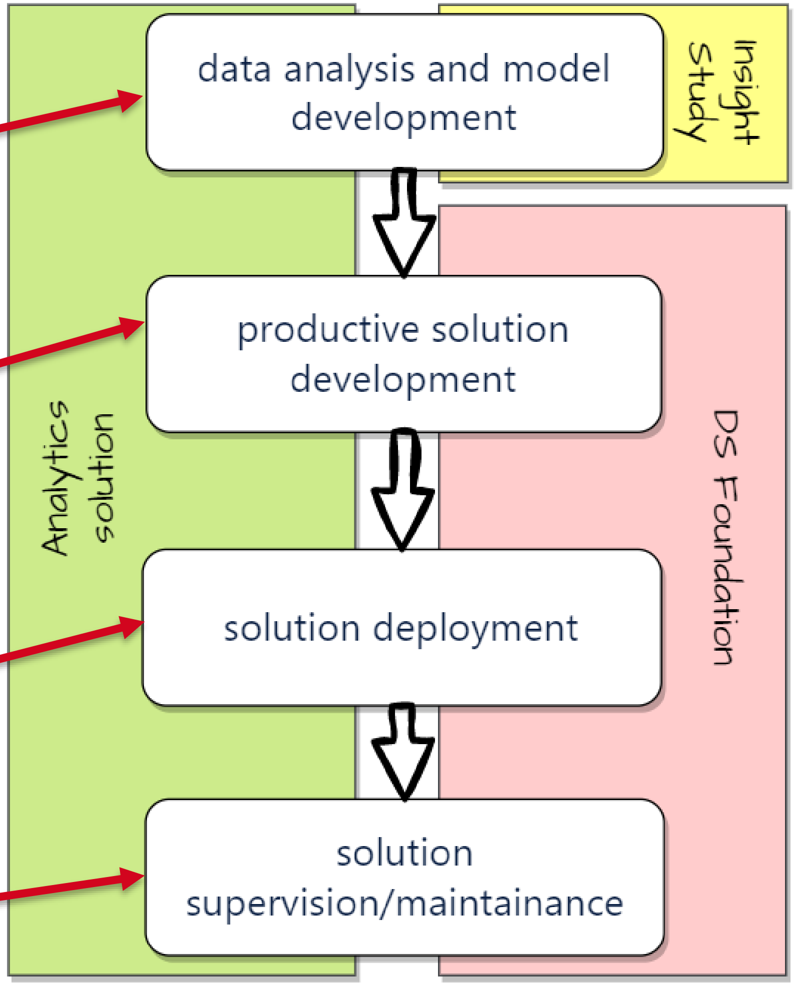
DIFFERENT STAGES OF DATA SCIENCE PROJECTS

Try things out

Make things more professional

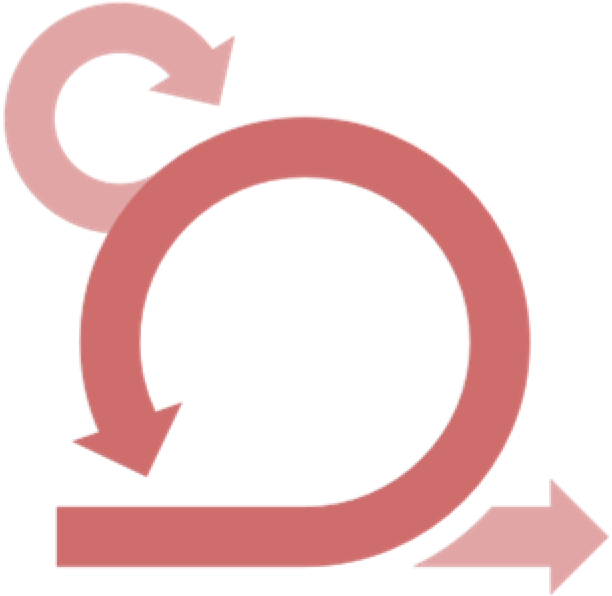
Deploy solution

Monitor solution

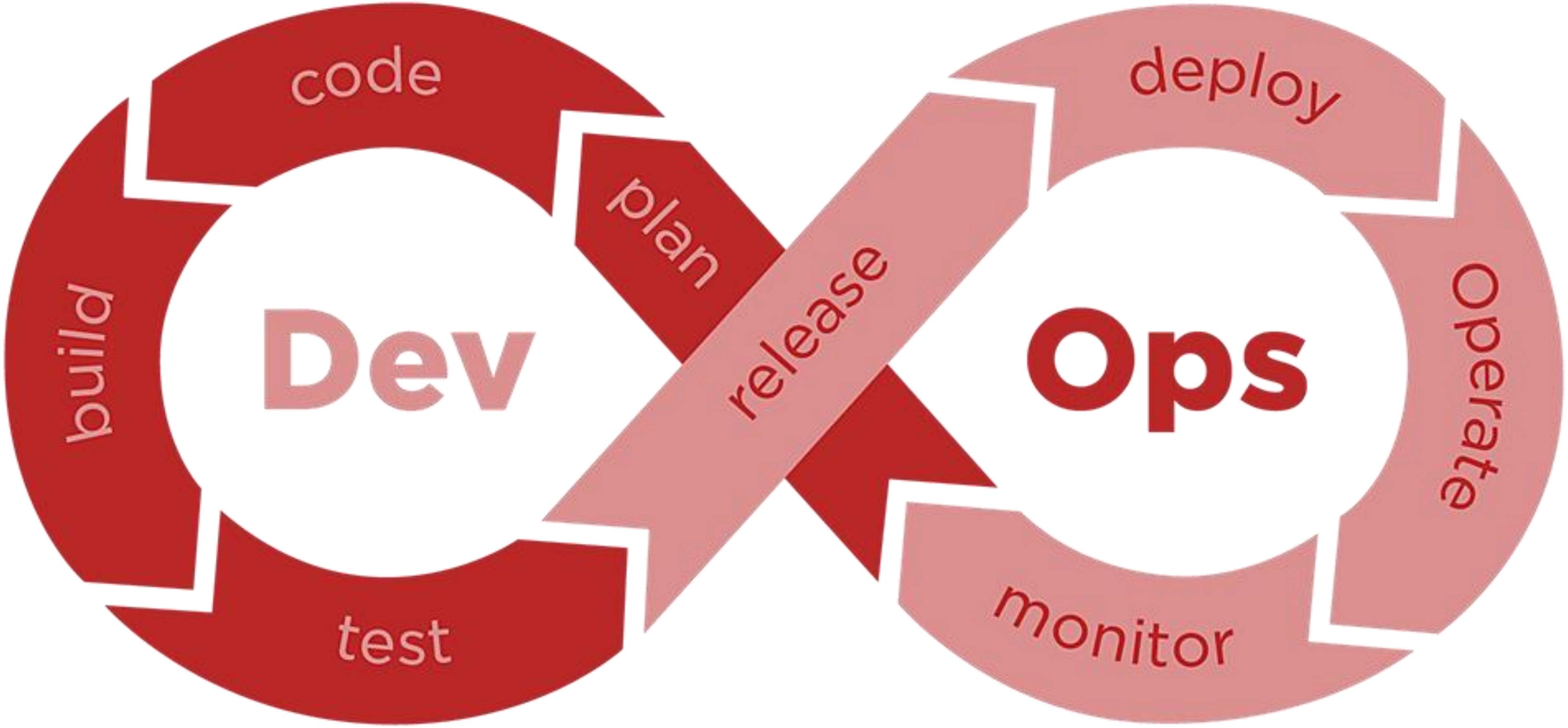


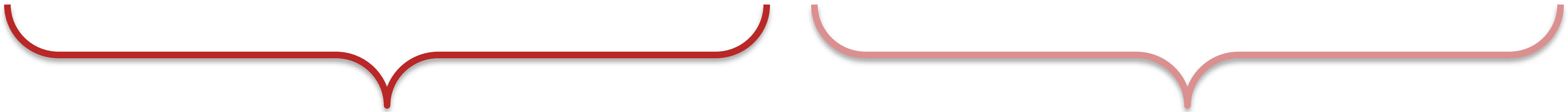
AGILE METHODOLOGIES FOR EFFICIENT WORKFLOW

- Agile methodologies proven to be effective in software development
- Can also be applied to Data Science projects
- Most popular framework within Agile approach is Scrum
- Scrum provides structured and iterative approach to project management



DEVOPS LIFECYCLE SIMILAR TO MLOPS LIFECYCLE





Develop the solution

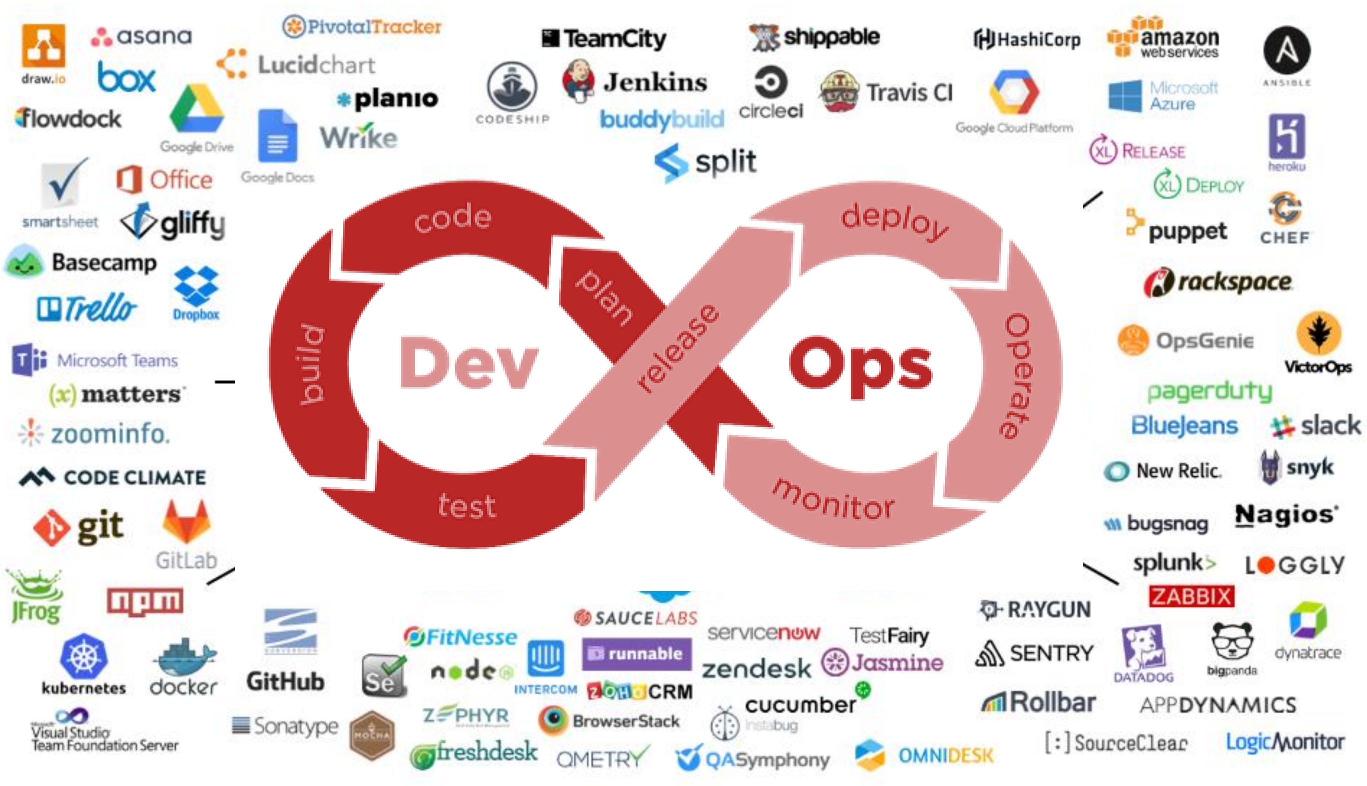
Put solution into production

MULTIPLE TOOLS SUPPORTING DEVOPS

Collaboration & Knowledge Sharing

Source Code Management

Build Process

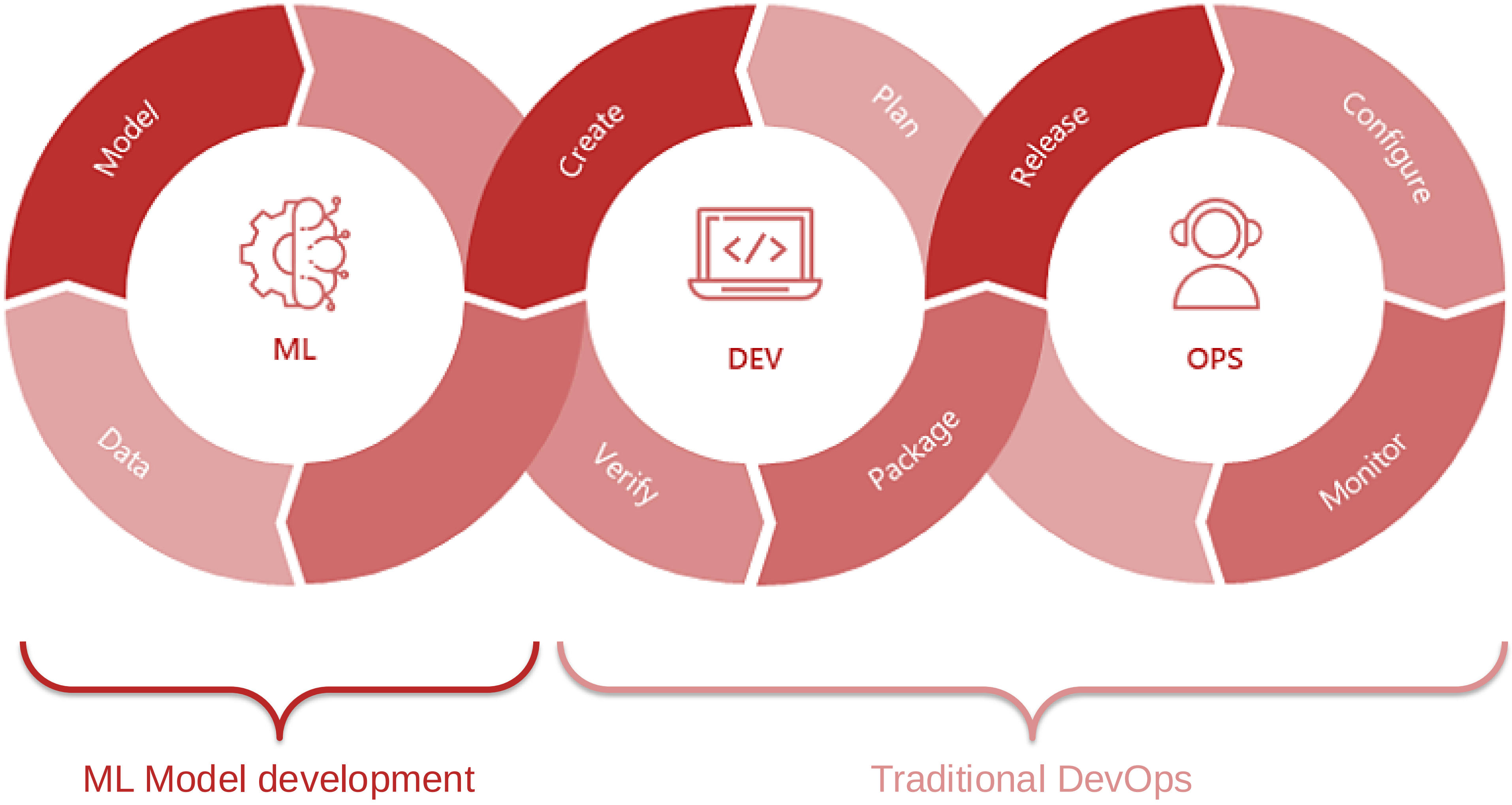


Continuous Integration

Deployment Automation

Monitoring & Logging

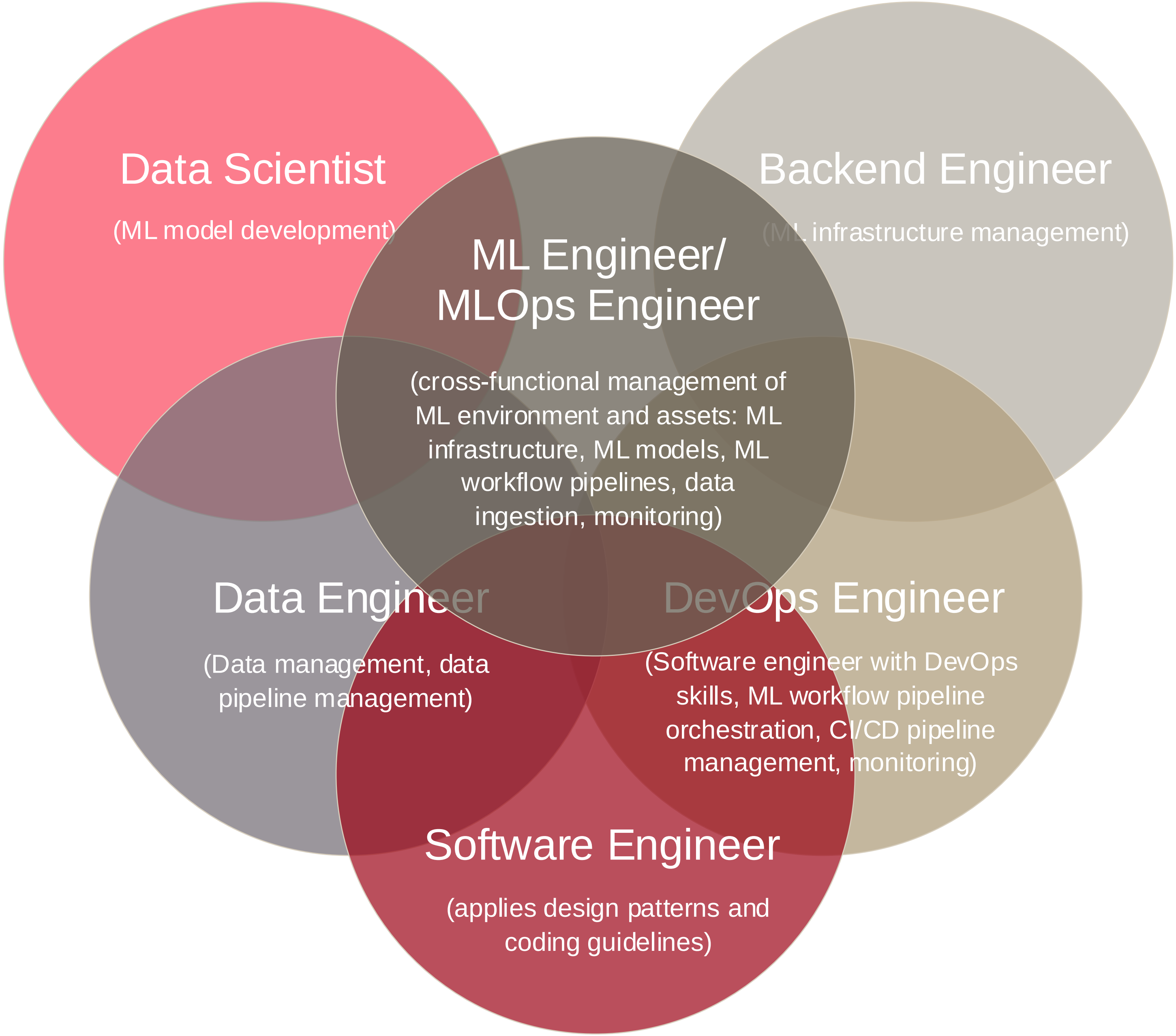
MLOPS ADDS ONE ADDITIONAL COMPONENT



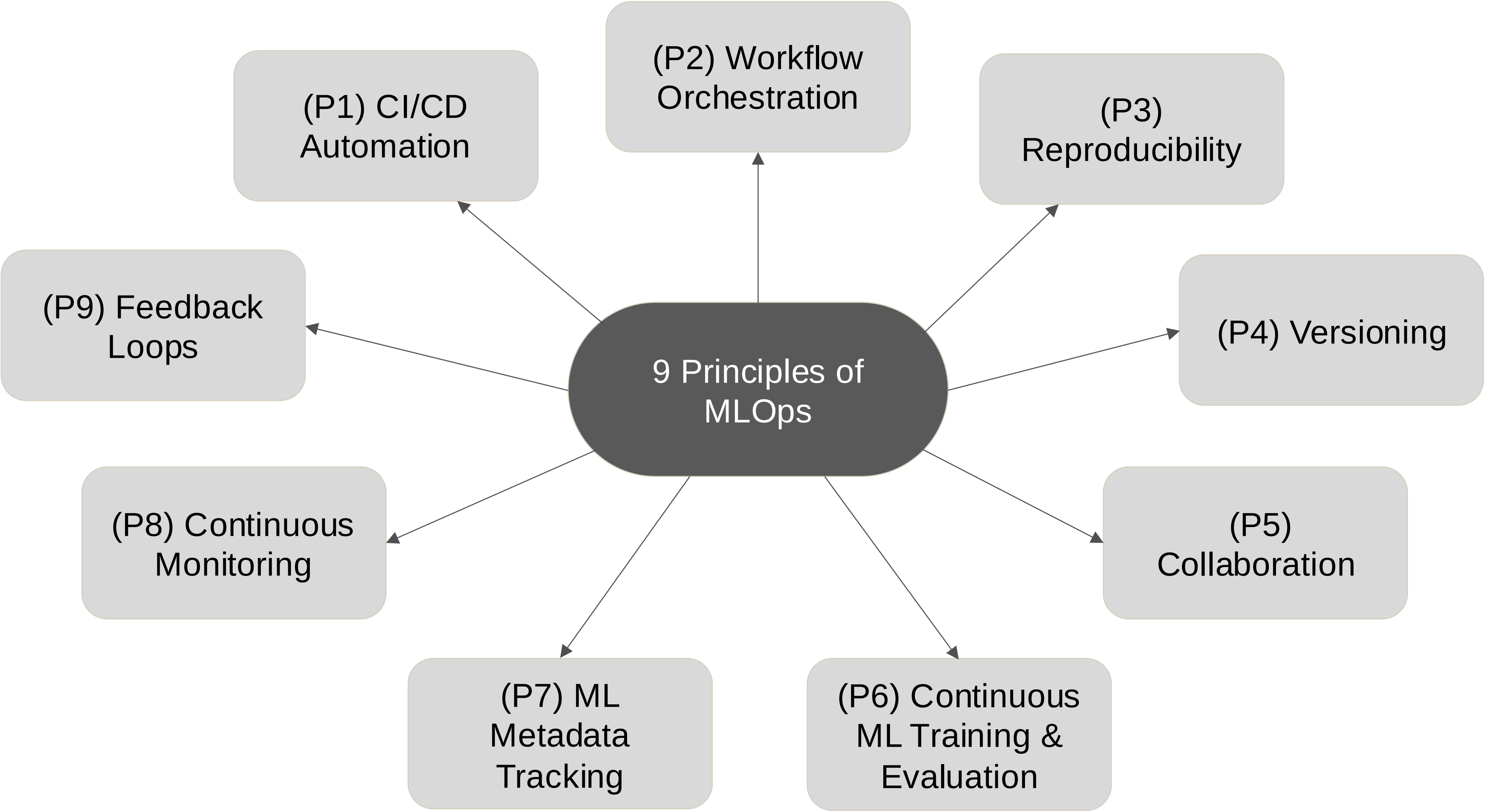
MLOPS HELP TO PROFESSIONALIZE/AUTOMATE PIPELINES

- **Problem:**
  - More and more companies rely on ML models
  - Difficult to scale ML projects only in Python notebooks/files
- **MLOps:**
  - Combines ML, DevOps, and data engineering
- **Goal:**
  - Deploy and maintain ML models in production reliably and efficiently
  - Automate the ML lifecycle to efficiently manage multiple ML projects
- **Benefits:**
  - Improves model reliability and reproducibility
  - Enables continuous integration and delivery of ML models
  - Facilitates monitoring and management of models in production

MLOPS IS A MULTIDISCIPLINARY TASK

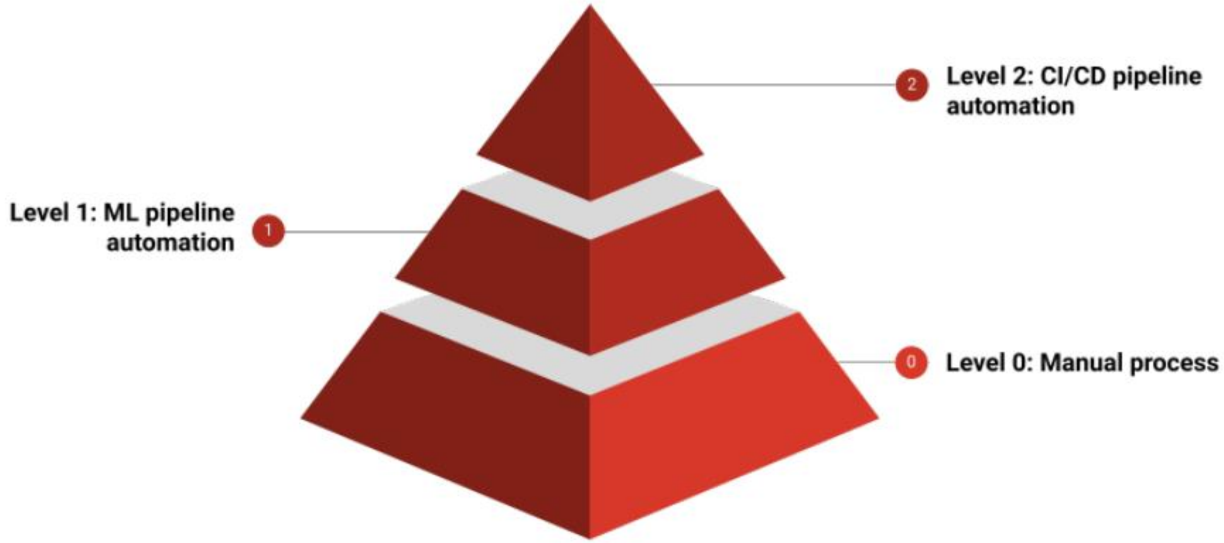


WHAT PRINCIPLES HELP US DOING MLOPS SUCCESSFULLY?

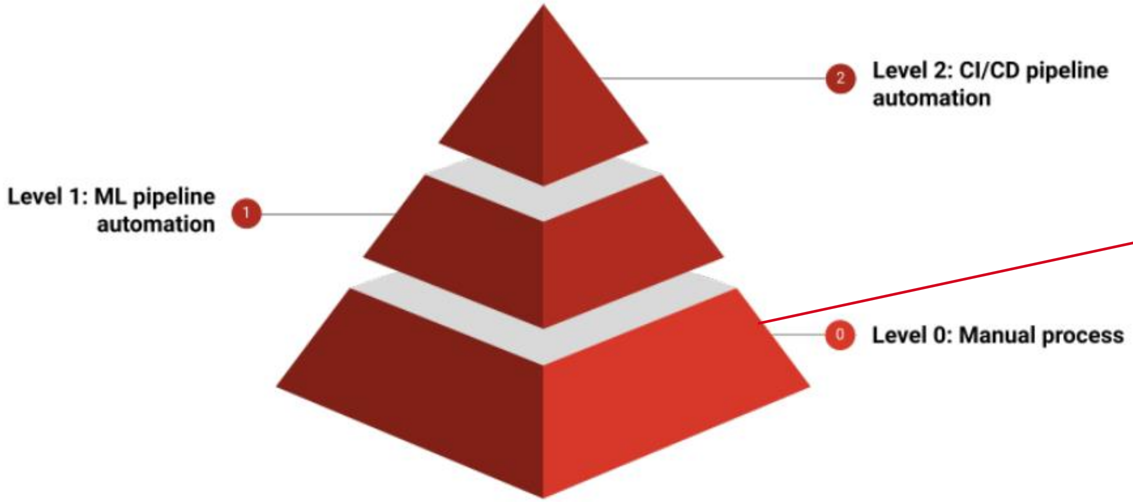


MLOPS MATURITY MODELS

Models to define the maturity of MLOps projects



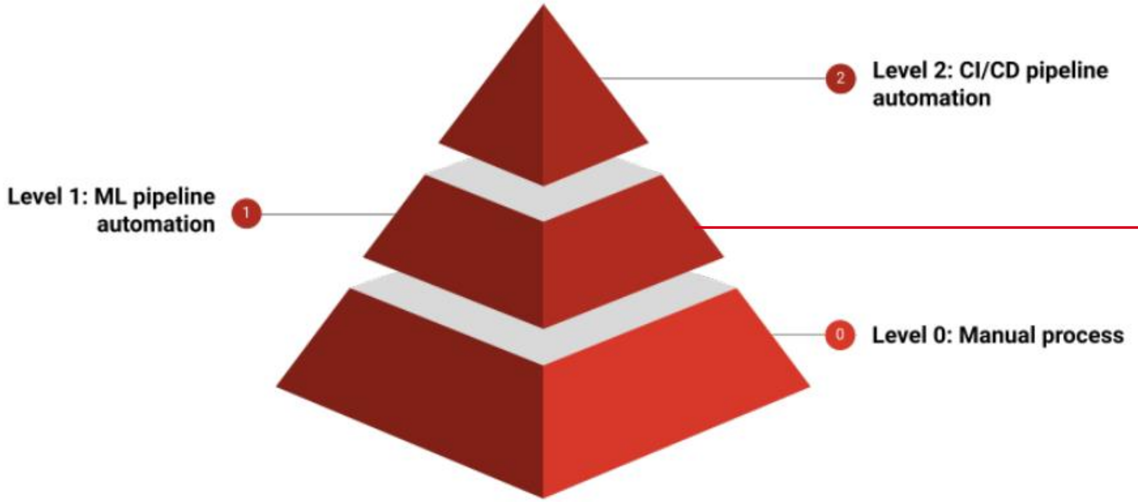
LEVEL 0 – NO AUTOMATION



Characteristics

- **Manual Python notebook runs**
- Disconnection between ML and operations
- Infrequent release iterations
- No CI/CD
- No integration into front-end
- Lack of active performance monitoring

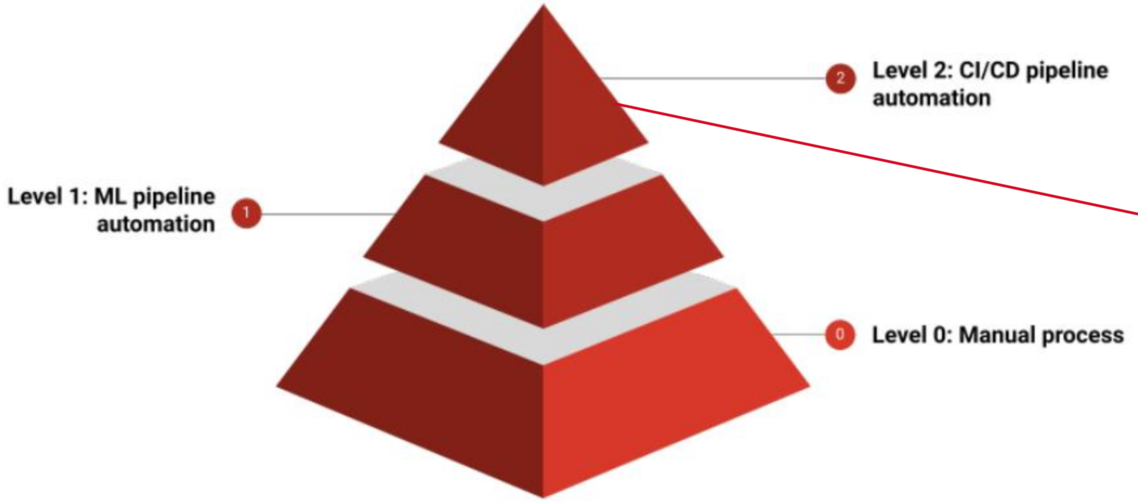
LEVEL 1 – ML PIPELINE AUTOMATION



Characteristics

- Rapid iterations
- Continuous retraining of model automated
- Python files with Classes and methods interacting with each other
- **Continuous delivery of models**
- Pipeline deployment

LEVEL 2 – CI/CD PIPELINE AUTOMATION



Characteristics

- Development and experimentation
- **End-to-end ML pipeline integration**
- Automated triggering of re-training
- Model delivered to front-end via API
- Monitoring mechanism in place
- Model registered

ML PIPELINE: OVERVIEW

- Helps automate preparing data and training an ML model with the data
- Consists of several stages/building blocks
- Each stage feds its output as input into the next stage
- Allows raw data to flow through the different building blocks into the input for ML model training

COMPLEXITY OF MLOPS PIPELINE

ETL

101011010
010101010
000011110

Collect and prepare training data

Build

Choose or build an ML algorithm

Train & Tune

Set up and manage environments for training

Train, debug, and tune models

Manage training runs

Deploy & Manage

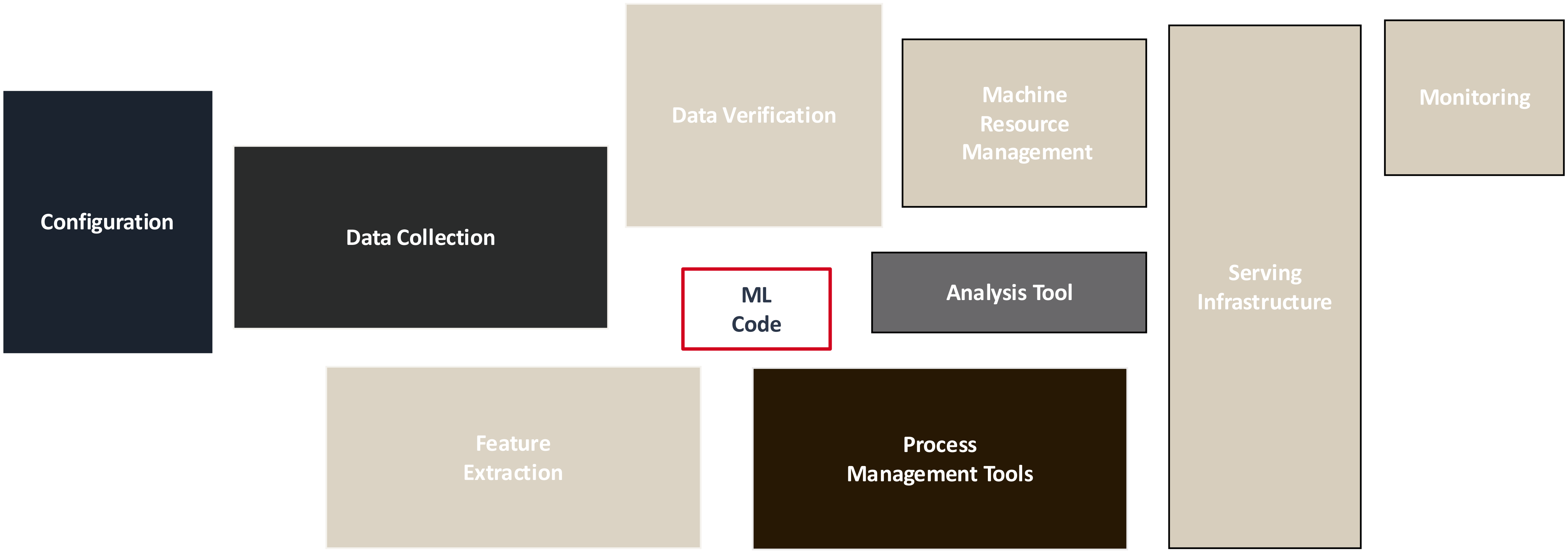
Deploy model in production

Monitor models

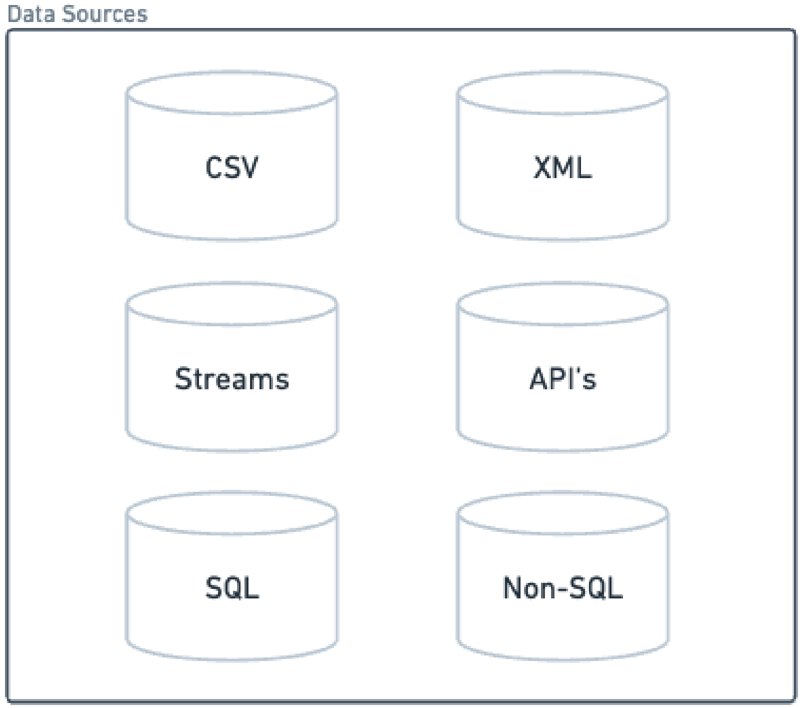
Validate predictions

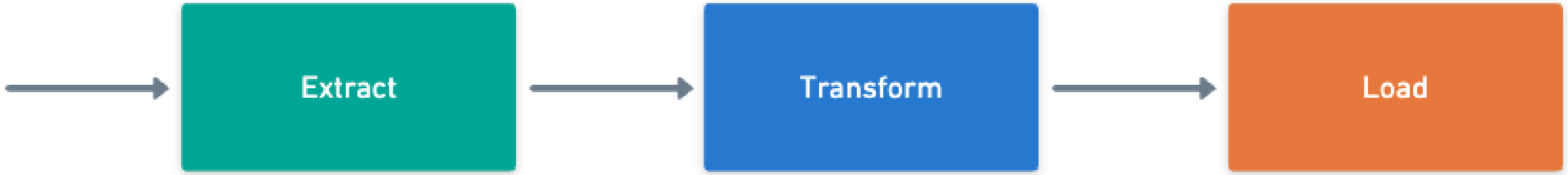
Scale and manage the production environment

ML CODE IS A SMALL FRACTION OF REAL-WORLD MLOPS



ETL PIPELINE NOT ONLY USED FOR MLOPS





Extract

- Stream or load data from different sources
- Authentication to remote data source

Transform

- Fix bad datapoints
- Standardize different data sources
- Join data sources

Load

- Store transformed data
- Used for BI, ML, monitoring, etc.

DATA PRE-PROCESSING USUALLY TAKES THE MOST TIME

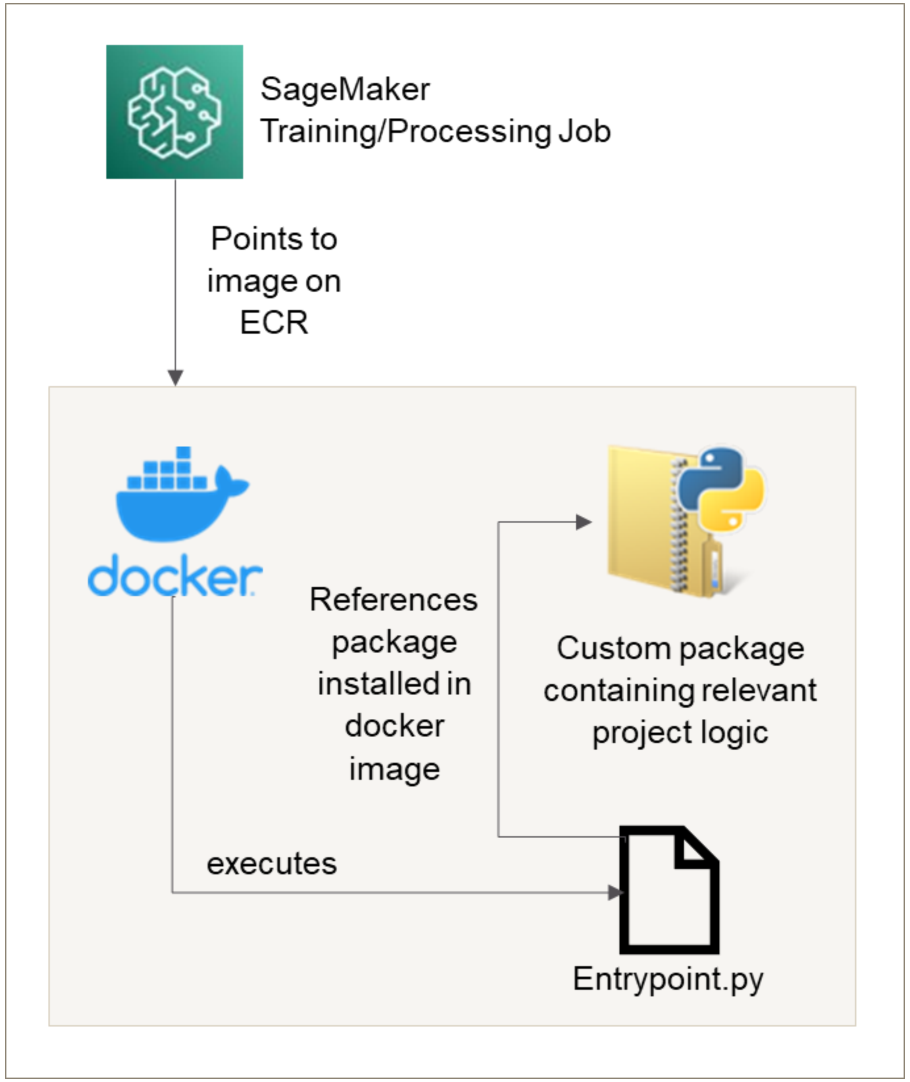
- **Data Cleaning**
  - Duplicates
  - Missing Data
    - Ignore the datapoint
    - Fill the missing values (mean, median, etc.)
  - Noisy data
    - Binning method
  - Mismatched data types
  - Outlier detection
    - Measuring errors
    - Type conversion errors
    - Integer Overflow

- **Data Transformation**
  - Normalization
    - Linear/Log scaling to (-1, 1)
  - Feature encoding
    - Classes → 1-hot-encoding
  - Feature engineering
  - Discretization (Int → Classes)
  - Handling imbalanced data
    - Oversampling
    - Undersampling

- **Data Reduction**
  - Dimensionality reduction
    - Principle Component Analysis (PCA)
  - Aggregation
  - Sampling
    - Random Sampling
    - Cluster Sampling
  - Feature selection
    - Correlation Analysis

TURNING ML CODE INTO MLOPS PROJECT

- Project has **completed ETL & Experimentation stages**
- Convert codebase into **python package** following object-oriented programming principles
- **Modular code** structure: package "jobs" in executable modules
- Create **entry-points** (Python scripts calling the executable modules) for Docker image

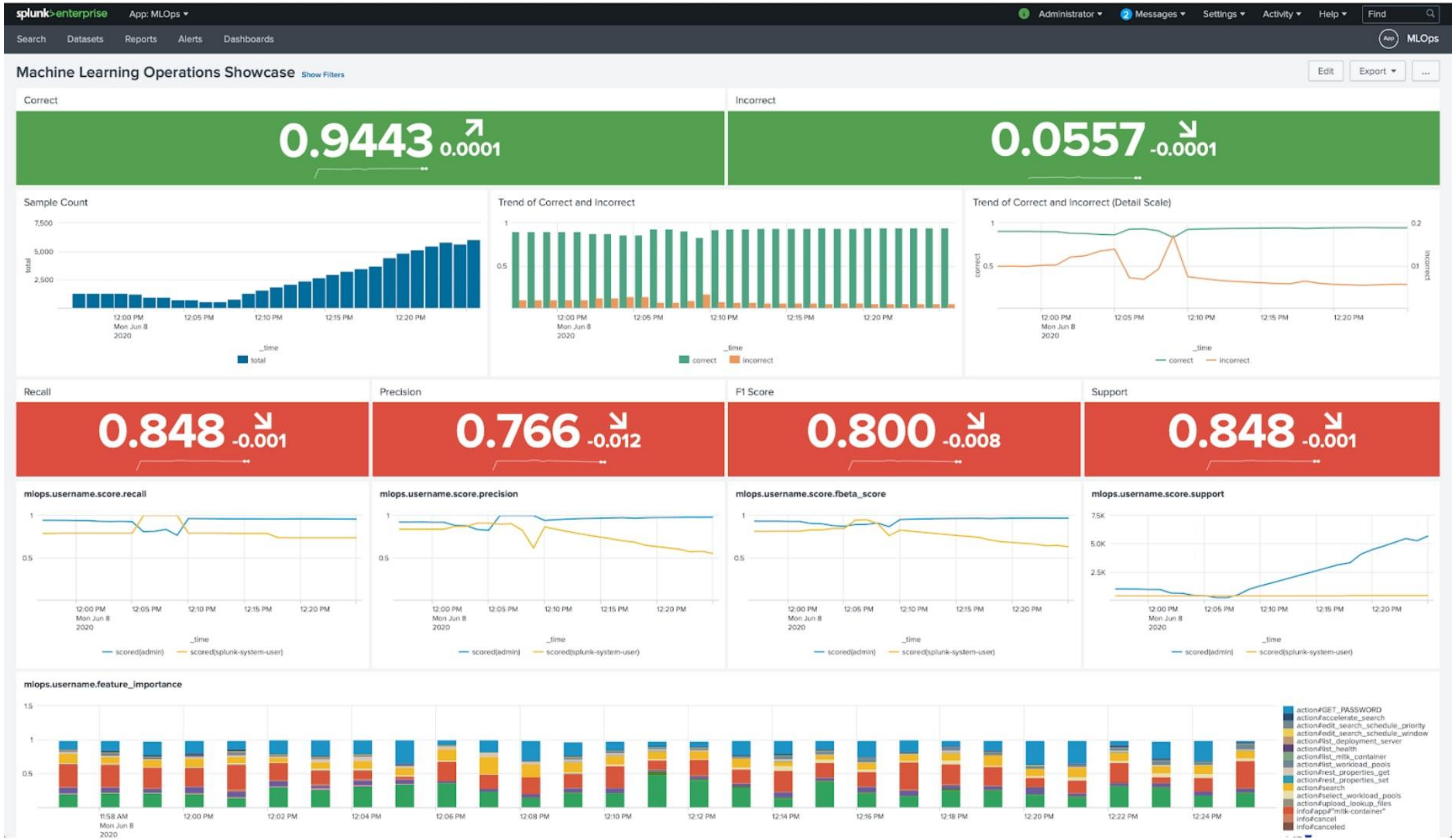


ML PIPELINE: MODEL EVALUATION

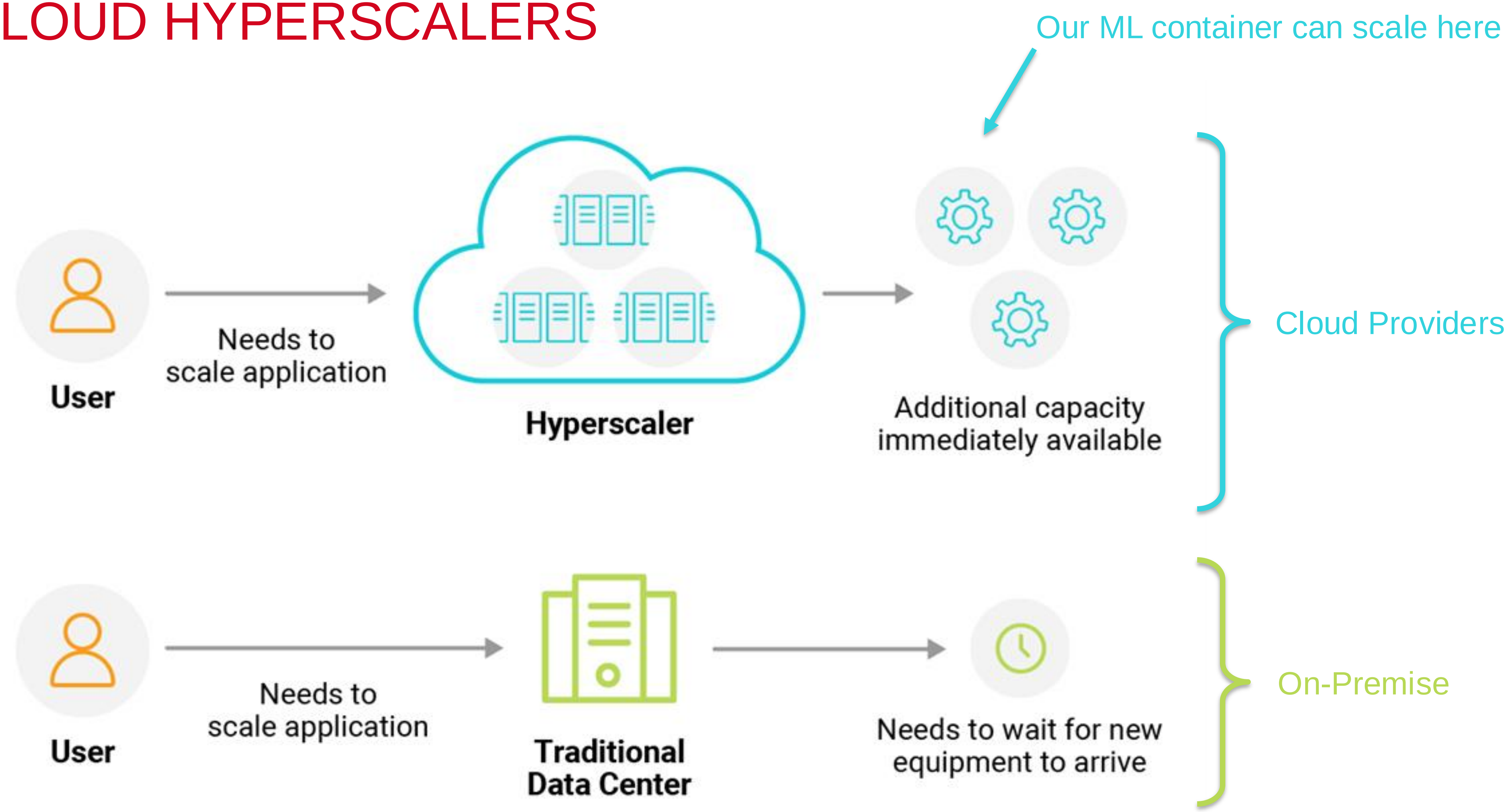
- Assess the performance of the ML model
- Essential for understanding a model’s strength and weaknesses to decide whether to deploy to production
- Compare to previous models and benchmark models (e.g. more traditional algorithms)
- Discuss with business stakeholders how precise a model should be
- Monitor the performance of a model over time
  - Model performance can degrade over time due to changes in data distribution

→ **We need quantitative measures to achieve this**

MONITORING DASHBOARD



CLOUD HYPERSCALERS

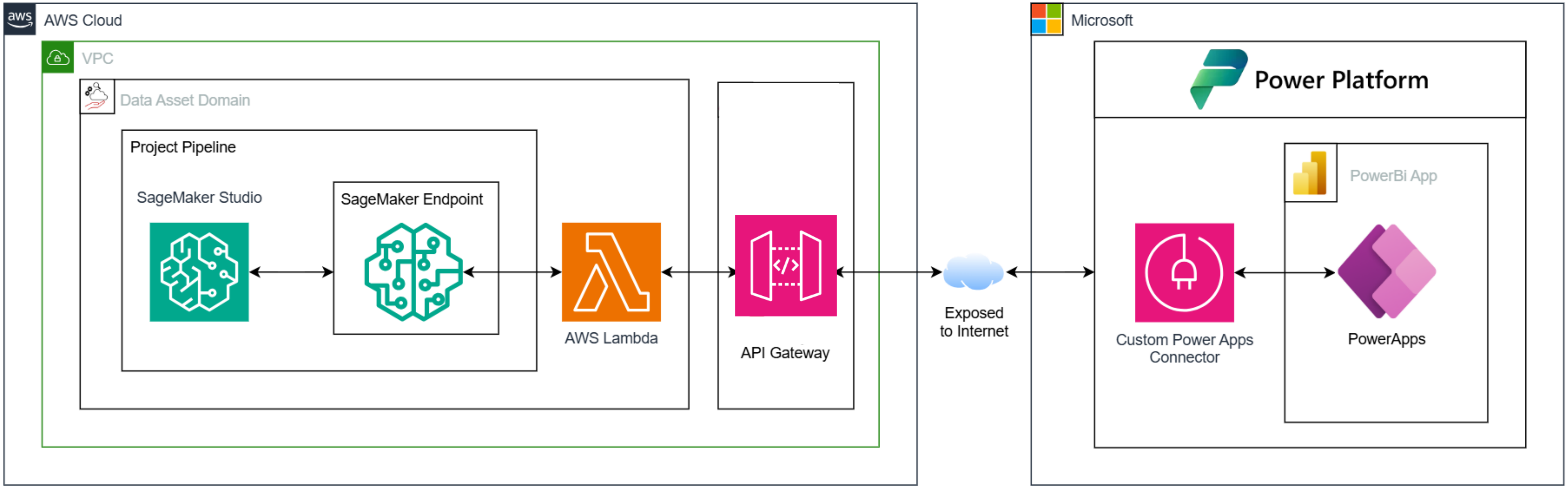


CLOUD HYPERSCALERS

Virtual Servers	Instances	VM Instances	VMs
Platform-as-a-Service	Elastic Beanstalk	App Engine	Cloud Services
Serverless Computing	Lambda	Cloud Functions	Azure Functions
Docker Management	ECS	Container Engine	Container Service
Kubernetes Management	EKS	Kubernetes Engine	Kubernetes Service
Object Storage	S3	Cloud Storage	Block Blob
Archive Storage	Glacier	Coldline	Archive Storage
File Storage	EFS	ZFS / Avere	Azure Files
Global Content Delivery	CloudFront	Cloud CDN	Delivery Network
Managed Data Warehouse	Redshift	Big Query	SQL Warehouse

Source: Moses Mtende

AMAZON WEB SERVICES (AWS)



DISCUSSION & QUESTIONS

Thank you for your attention!