



Title: Architecture for Model Deployment

— Big Data Project

Professor: Madan krish

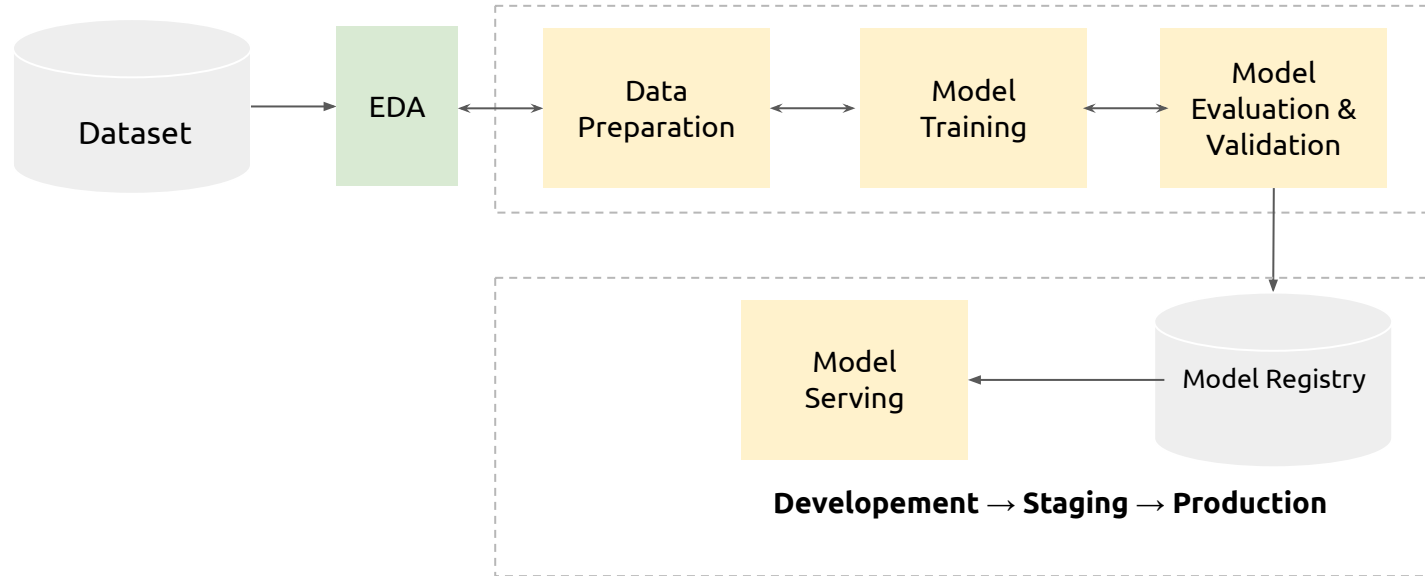
By: Azza Kamoun

Master ADEO 2

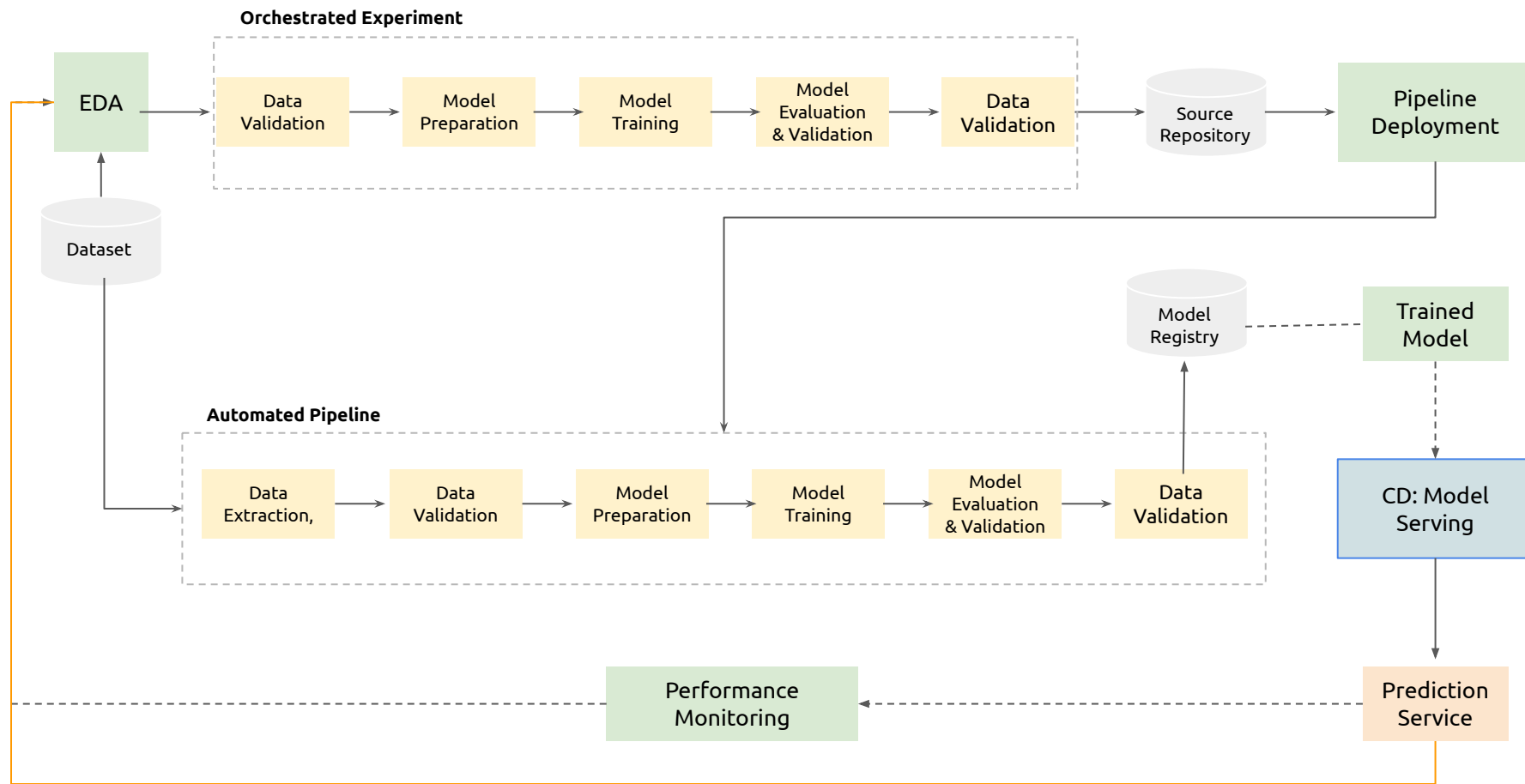
Table of Contents

1. Architecture for Model Deployment - Traditional Approach
2. Architecture for Model Deployment - Container-based Approach
3. Architecture for Model Deployment - Serverless Approach
4. Explanation of the Architectures

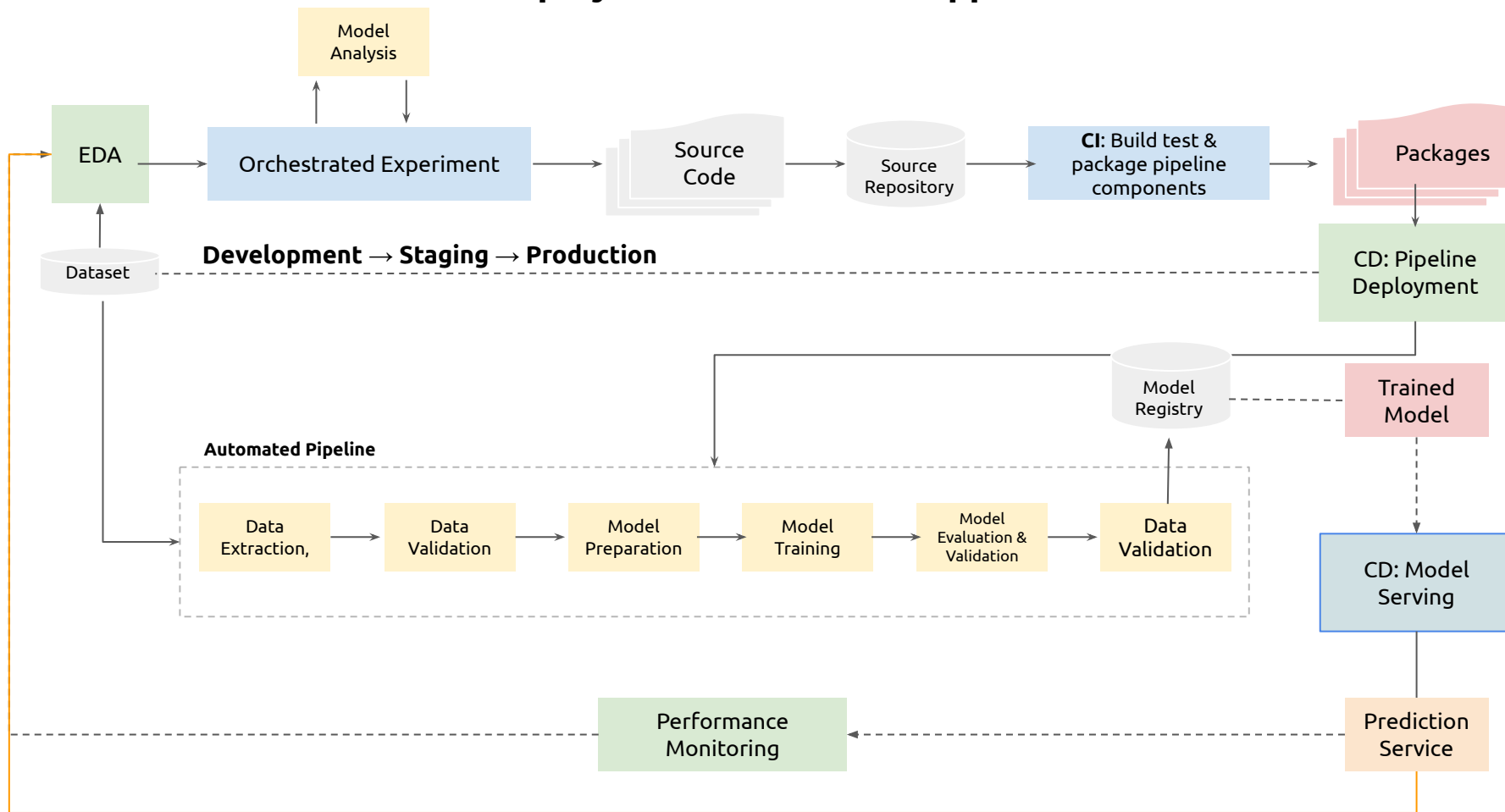
1 Architecture for Model Deployment - Traditional Approach



2 Architecture for Model Deployment - Container-based Approach



3 Architecture for Model Deployment - Serverless Approach



4 Explanation Architecture for Model Deployment

1 Architecture for Model Deployment - Traditional Approach

- The first architecture, known as the "traditional approach," entails manually deploying and managing the model infrastructure. This includes managing the servers and resources required for model training and serving, as well as developing and deploying custom automation scripts. While this approach can be tailored to meet specific requirements, it is also time-consuming and error-prone.

2 Architecture for Model Deployment - Container-based Approach

- A "container-based approach" is used in the second architecture, which involves packaging the model and its dependencies into containers for easier deployment and management. This strategy enables more efficient and consistent deployment, as well as the ability to scale resources as needed. However, some manual configuration and management of the containerized infrastructure is still required.

3 Architecture for Model Deployment - Serverless Approach

- The third architecture, known as a "serverless approach," involves using cloud-based services such as AWS Lambda or Azure Functions for model training and serving. This approach eliminates the need for infrastructure management and configuration because the cloud provider handles this automatically. Furthermore, serverless computing can save money by charging only for actual usage rather than maintaining a dedicated infrastructure. However, in terms of customization and control over the underlying infrastructure, this approach may have some limitations.

References:

Monoliths vs Microservices vs Serverless [link](#)

MLOps: Continuous delivery and automation pipelines in machine learning [link](#)

[MLOps: Continuous delivery and automation
pipelines in machine learning | Cloud
Architecture Center | Google Cloud](#)

Thank you for your Attention

Feel free to ask any questions or give your feedback