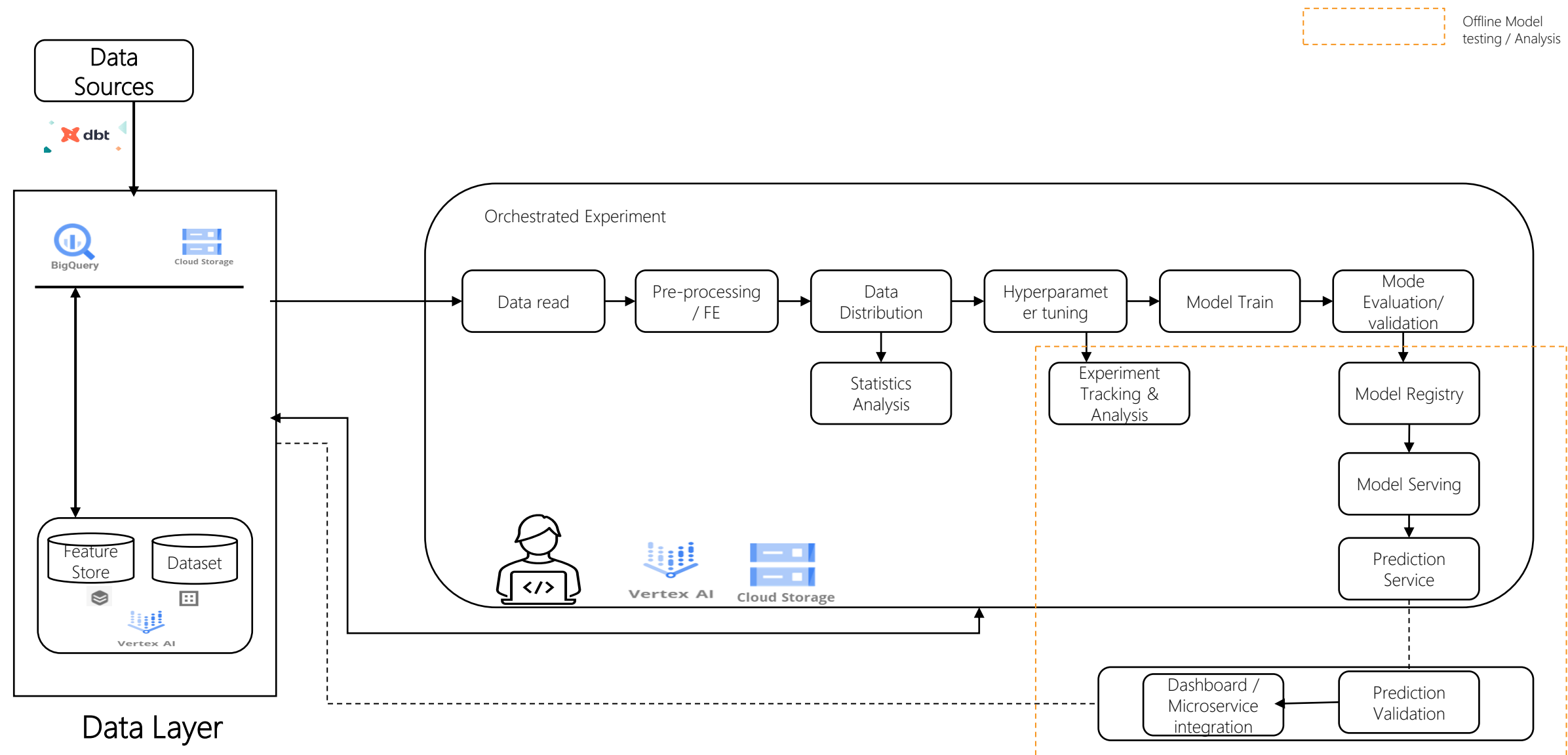# MLOps Strategy & Framework
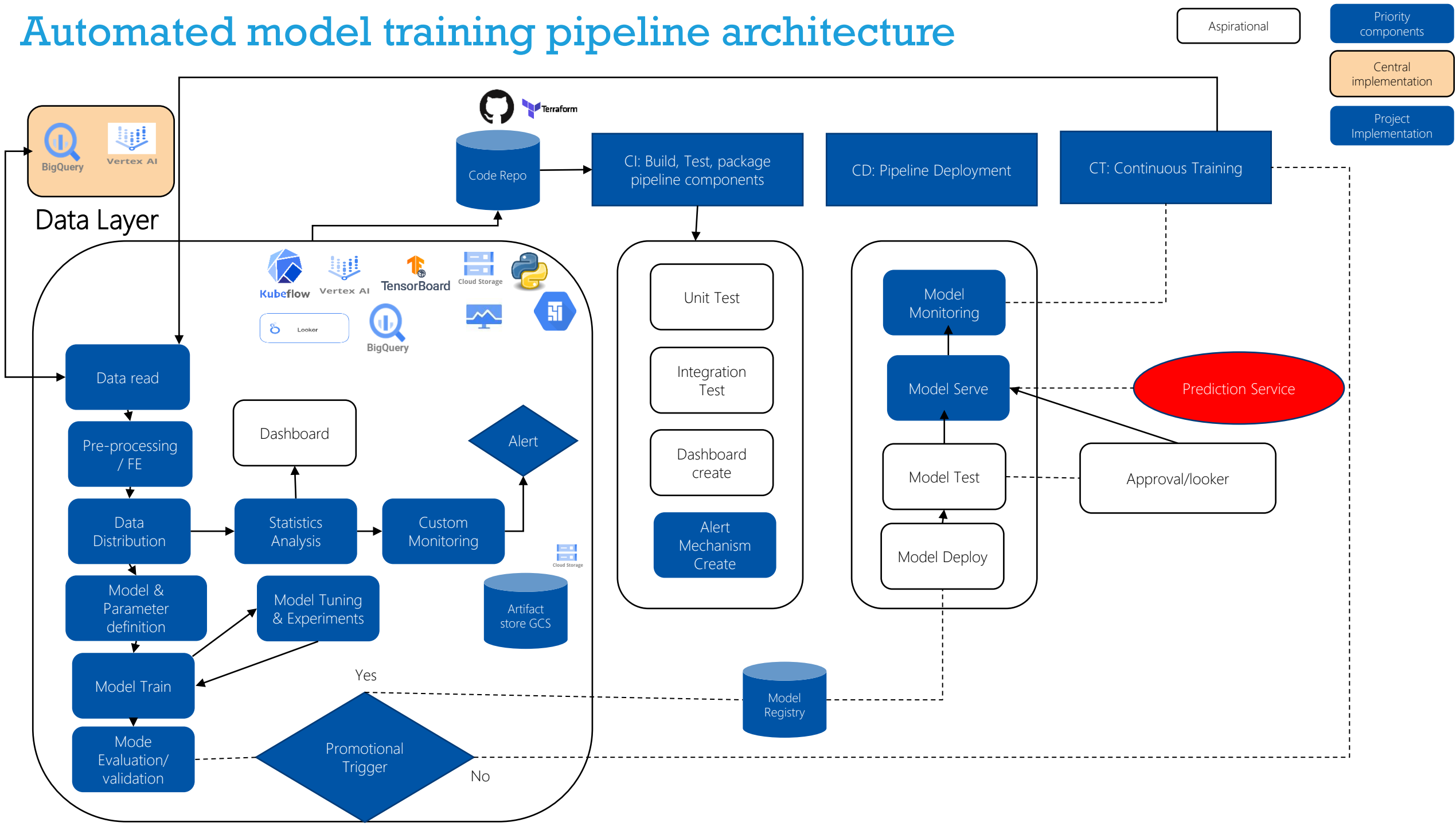
Architecture & Solution Design (Tools)

March 23rd, 2023
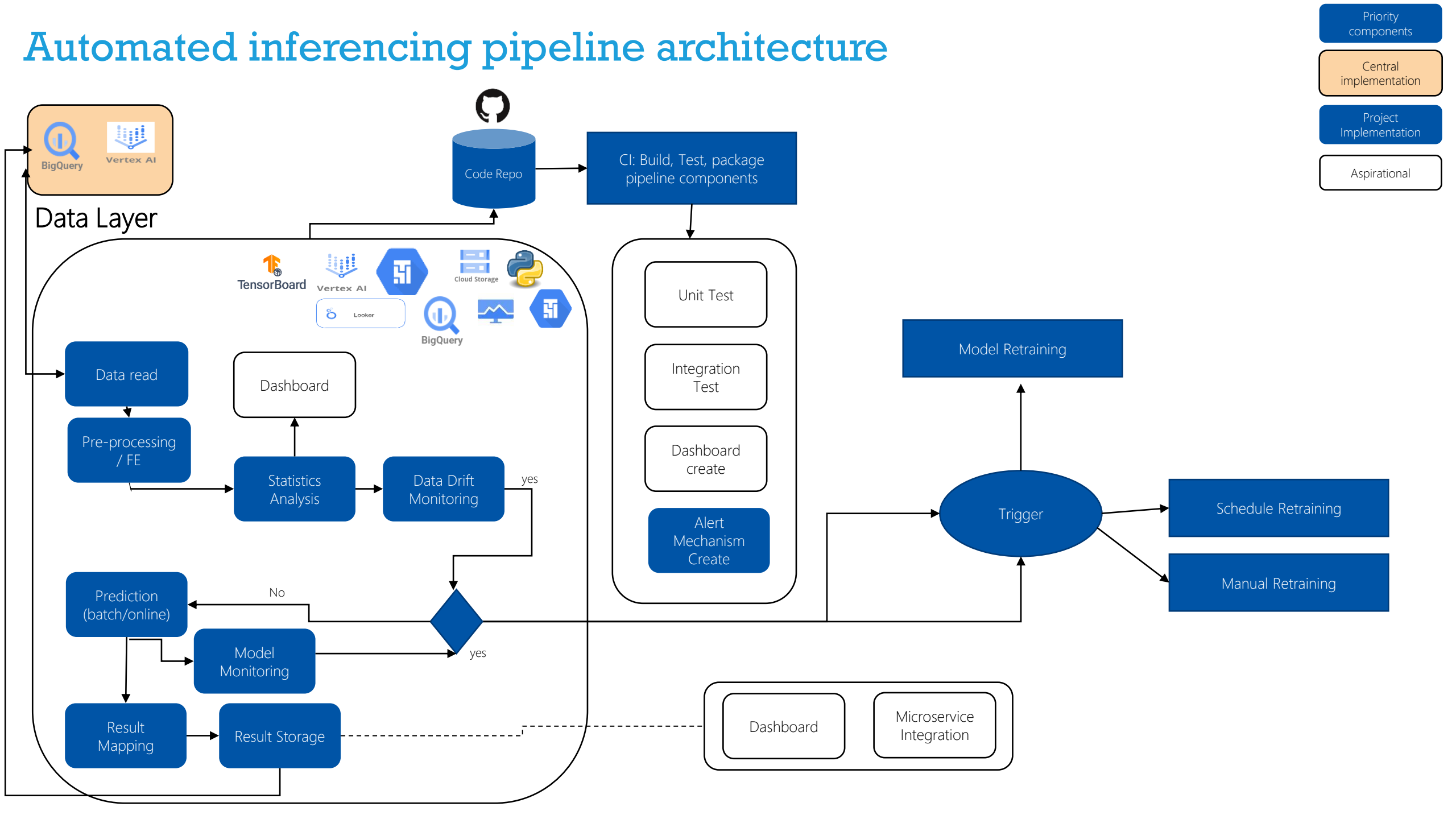
Data Science Experimentation Zone

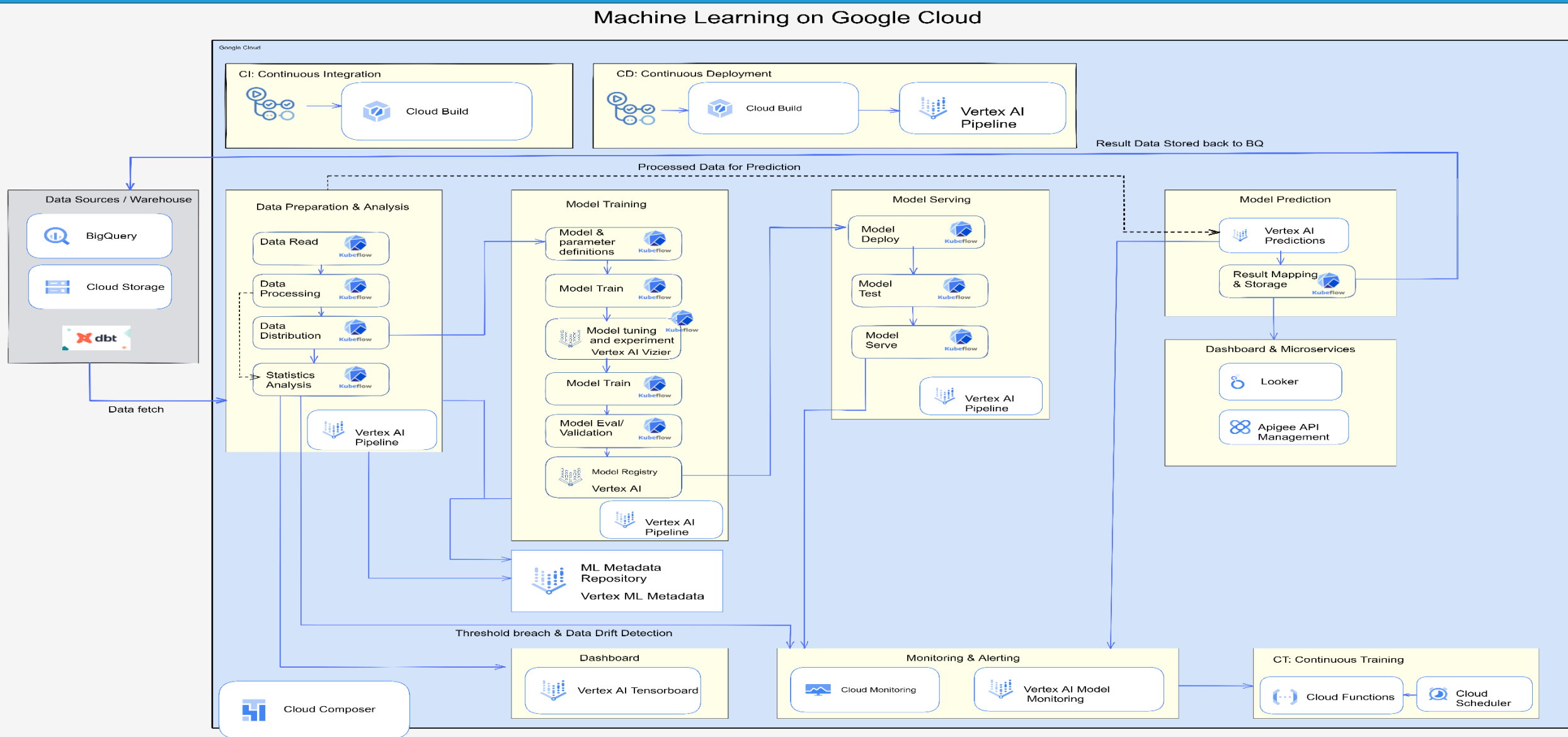# Automated model training pipeline architecture

Automated inferencing pipeline architecture

Priority components

Central implementation

Project Implementation

Aspirational

Code Repo

CI: Build, Test, package pipeline components

Data Layer

TensorBoard   Vertex AI   Cloud Storage

Looker   BigQuery

Data read

Dashboard

Pre-processing / FE

Statistics Analysis

Data Drift Monitoring   yes

Unit Test

Integration Test

Dashboard create

Alert Mechanism Create

Model Retraining

Trigger

Schedule Retraining

Manual Retraining

Prediction (batch/online)   No

Model Monitoring   yes

Result Mapping

Result Storage

Dashboard   Microservice Integration

Solution design & tools

CI/CD approach explained (overall)

Lint & Code Style

Unit Test

Image Build

Github Repository

Github Action

Cloud Build

Cloud Composer

Cloud Storage

Code Sync and execution
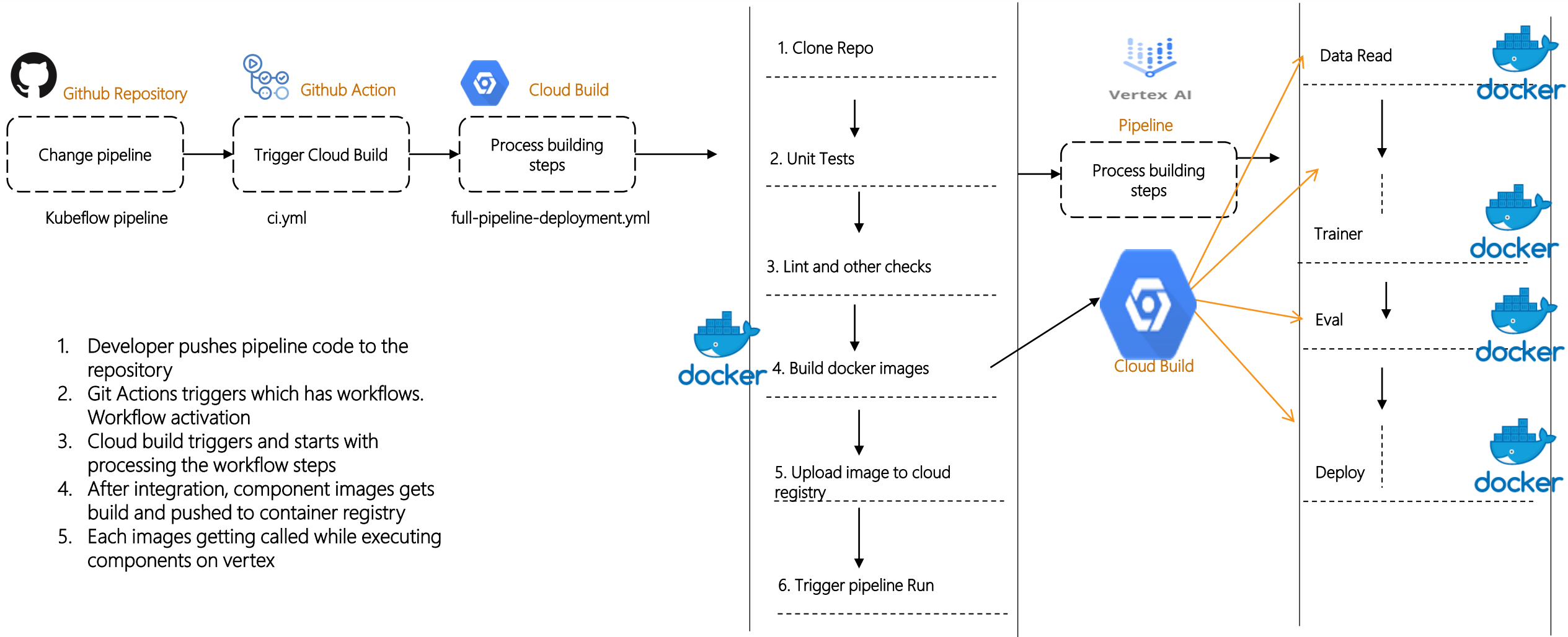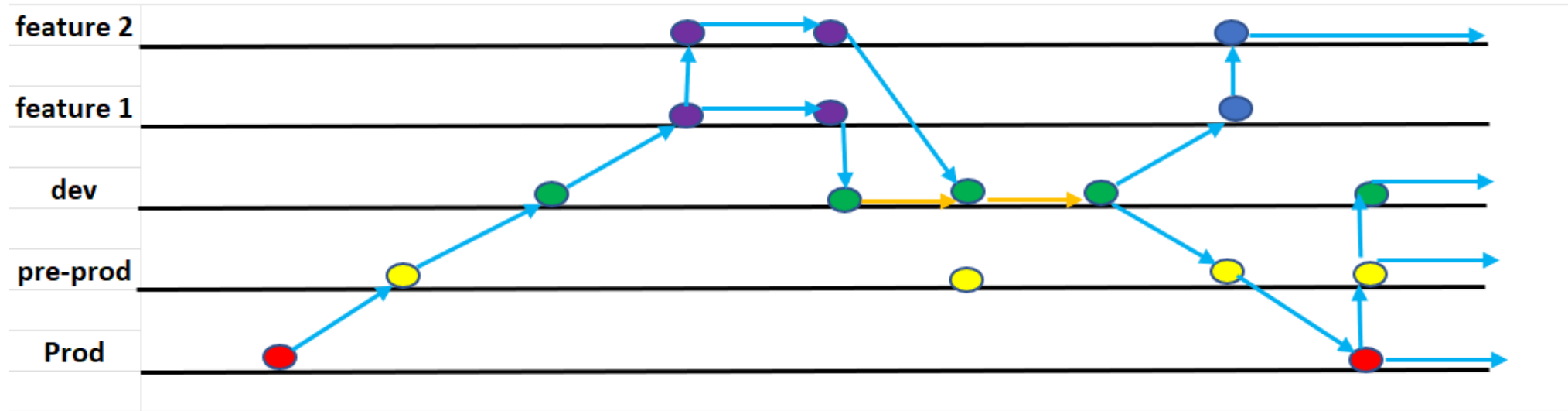
1. Developer pushes code to the repository
2. Git Actions triggers which has workflows. Workflow activation
3. Cloud build triggers and starts with running the workflow
4. Unit test and other test triggers, once passes will build images and push them
5. code sync with composer

CI/CD explained (pipeline)

Github Repository

Change pipeline

Kubeflow pipeline

Github Action

Trigger Cloud Build

ci.yml

Cloud Build

Process building steps

full-pipeline-deployment.yml

1. Clone Repo

2. Unit Tests

3. Lint and other checks

4. Build docker images

5. Upload image to cloud registry

6. Trigger pipeline Run

Vertex AI

Pipeline

Process building steps

Cloud Build

Data Read

Trainer

Eval

Deploy

1. Developer pushes pipeline code to the repository
2. Git Actions triggers which has workflows. Workflow activation
3. Cloud build triggers and starts with processing the workflow steps
4. After integration, component images gets build and pushed to container registry
5. Each images getting called while executing components on vertex

Branching strategy

Legend:
- 🔴 Main master branch, production ready code
- 🟡 Pre-prod, before releasing to main/master
- 🟢 Up-to-date Dev branch, with latest developed features
- 🔵 Feature branch for each developer to work independently on new functionality

- Code migration will occur via gitaction yaml files post pull request by developer
- Each yaml file will call actions within and execute the integration steps

* - We will utilize GMI repo structure for our development phase