AI Essentials

Machine Learning Operations (MLOps)

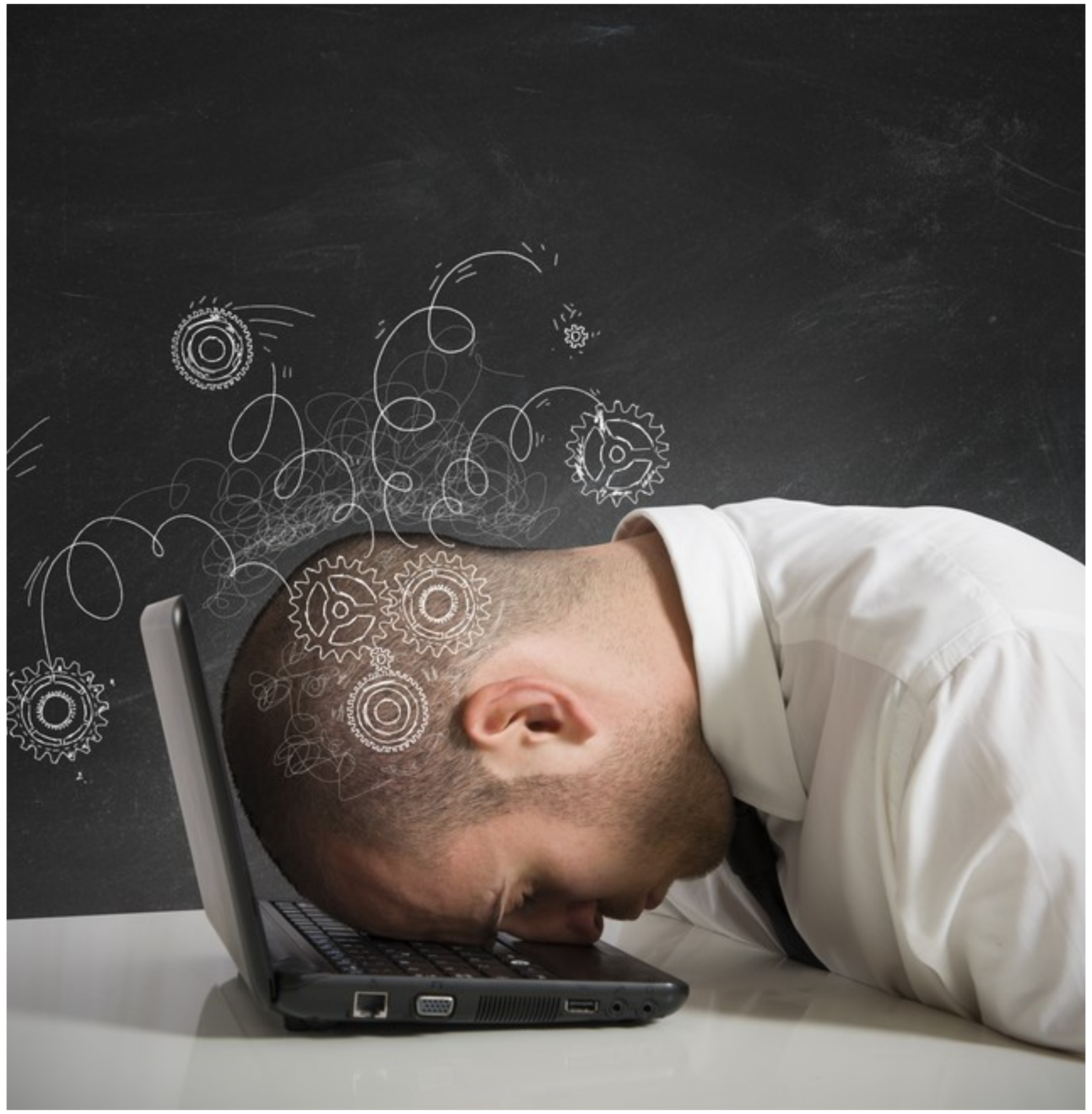
ir. Hennion Domien

Introduction

- The Machine Learning community had focused extensively on the building of ML models,
- but not on building production-ready ML products
- and providing the necessary coordination of the resulting often complex ML system components and infrastructure

Introduction

- In many industrial applications, data scientists still manage ML workflows manually
- Resulting in many issues during the operations of the respective ML solutions
- How can manual ML processes be automated and operationalized so that more ML POC’s can be brought into production



DevOps

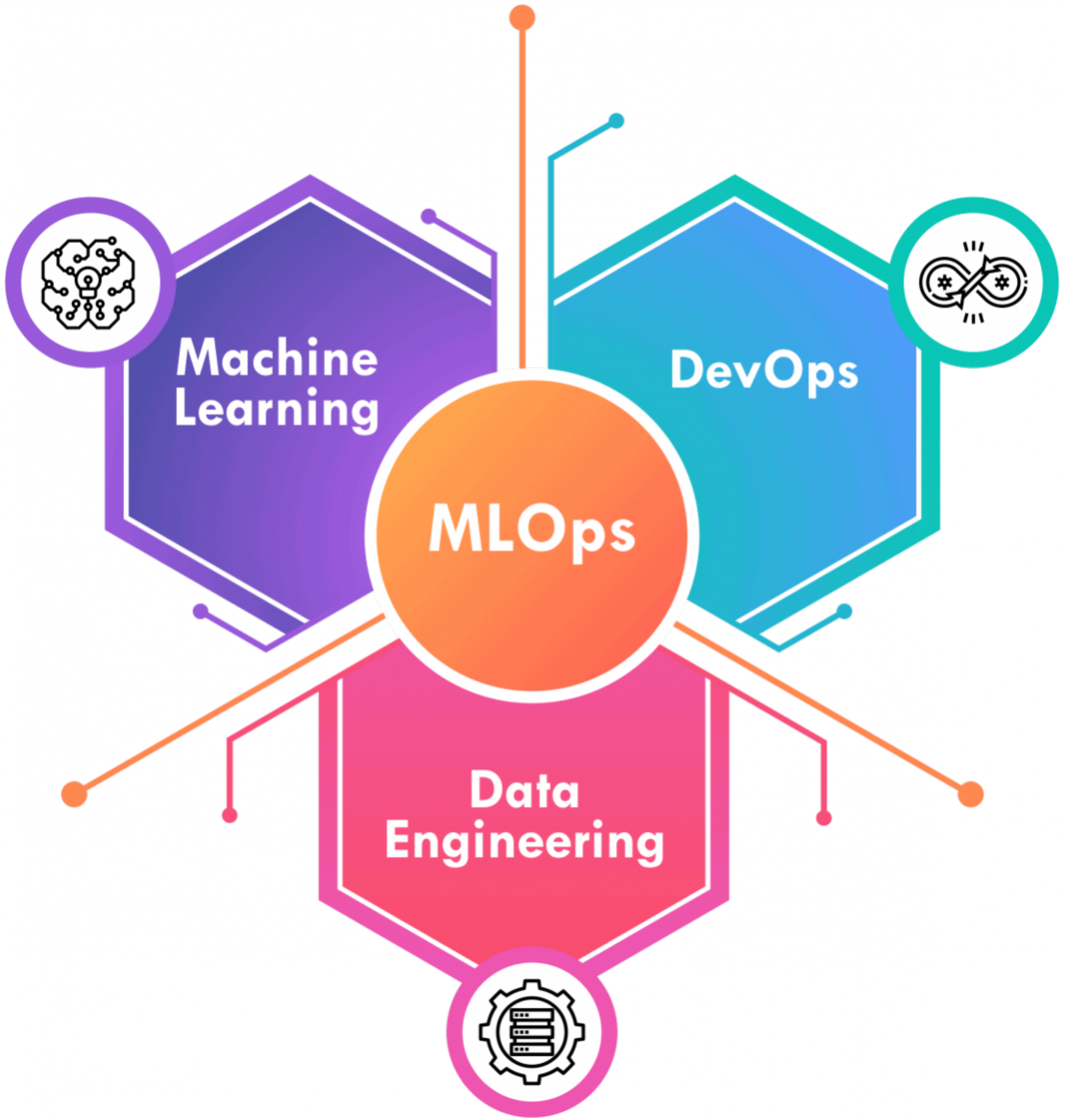
Development Operations

- DevOps represents a paradigm addressing social and technical issues in organization engaged in software development.
- Ensure automation with continuous integration, continuous delivery and continuous deployment (CI/CD)

DevOps

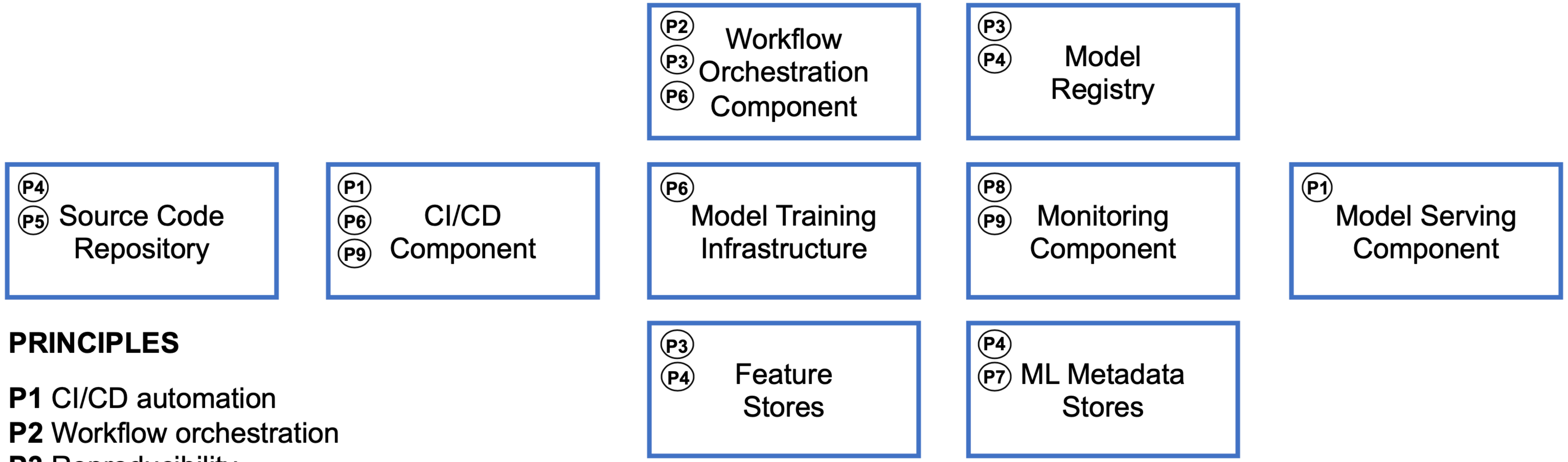
Tools

- Differentiated into 6 groups
  - Collaboration and knowledge sharing (Slack, Trello, GitLab wiki)
  - Source code management (GitHub, GitLab)
  - Build process (Maven)
  - Continuous integration (Jenkins, GitLab CI)
  - deployment automation (Kubernetes, Docker)
  - Monitoring and logging (Prometheus, Logstash)


Machine Learning
DevOps
MLOps
Data Engineering

MLOps

9 principles



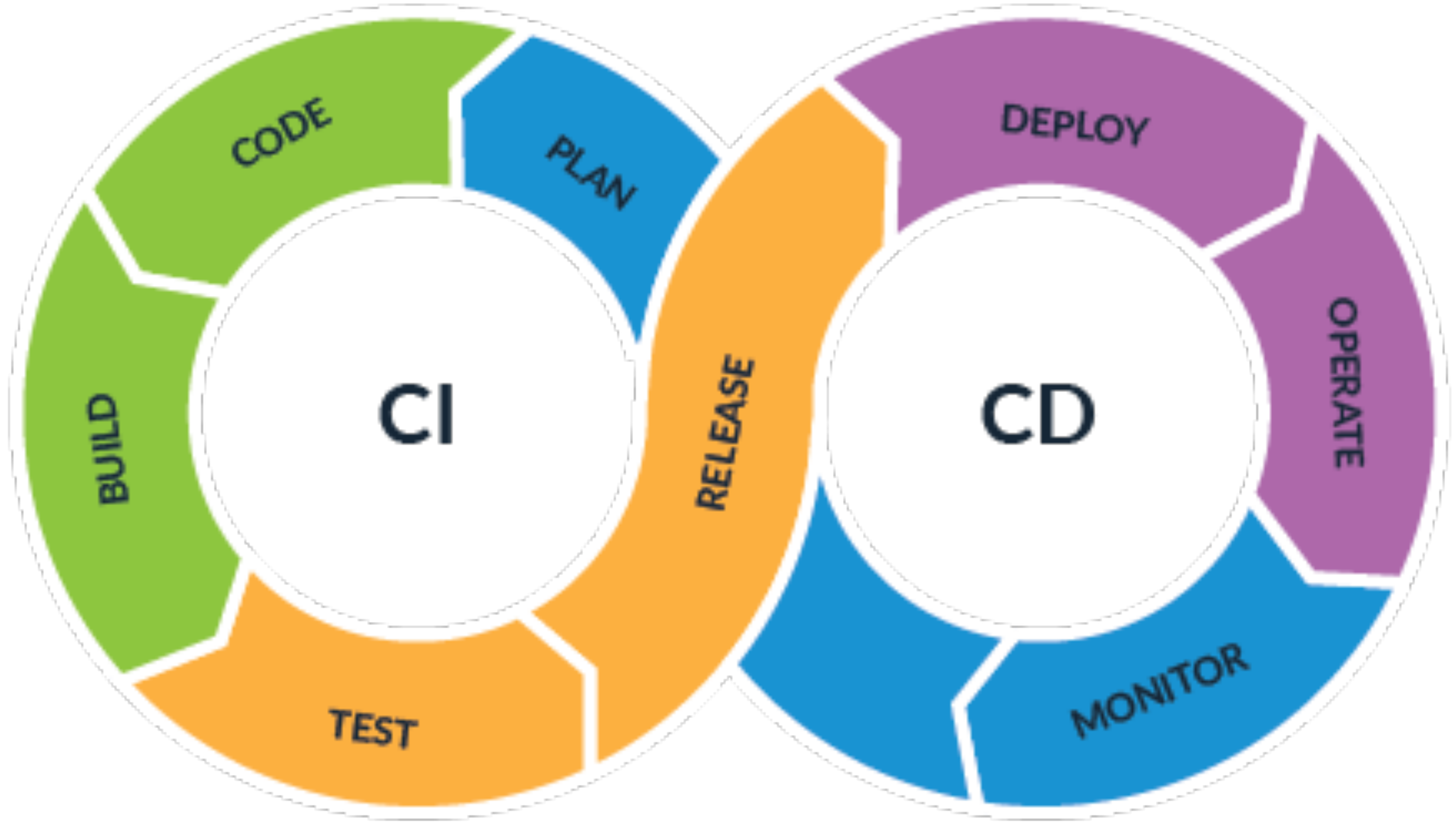
PRINCIPLES

P1 CI/CD automation
P2 Workflow orchestration
P3 Reproducibility
P4 Versioning of data, code, model
P5 Collaboration
P6 Continuous ML training & evaluation
P7 ML metadata tracking
P8 Continuous monitoring
P9 Feedback loops

COMPONENT

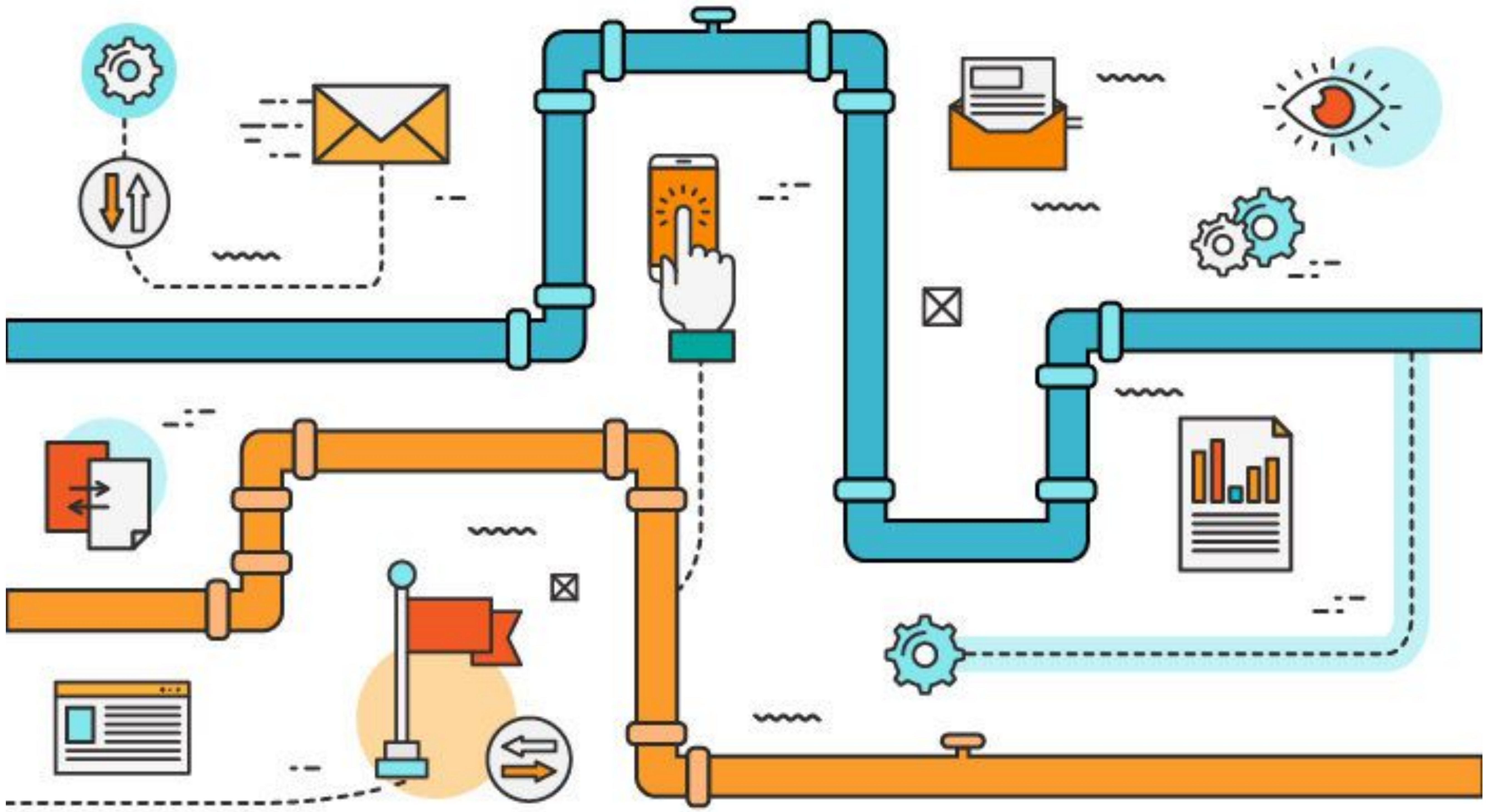
P1 CI/CD automation

- Continuous integration
- Continuous delivery
- Continuous delivery
- Build, test, delivery and deploy steps
- Fast feedback on success or failure of steps



P2 Workflow orchestration

- Coordinates tasks of an ML workflow pipeline according to DAGs



P3 Reproducibility

- Ability reproduce an ML experiment
- Obtain same results

P4 Versioning

- Versioning of data, model and code
- Reproductively and traceability

P5 Collaboration

- Work collaboratively on data, model and code
- Reduce domain silos between different roles

P6 Continuous ML training & evaluation

- Periodic retraining of the ML model based on new feature data
- An evaluation run to assess the change in model quality

P7 ML metadata tracking/logging

- Metadata is tracked and logged for each orchestrated ML workflow task
- Training data and time, duration, …
- Used parameters and the resulting performance metrics
- Ensure full traceability of experiment runs

P8 Continuous monitoring

- Periodic assessment of data, model, code, infrastructure resources and model serving performance
- To detect potential errors or changes

P9 Feedback loops

- Integrate insights for the quality assessment step into the dev or engineering precess
- Feedback from the monitoring component to the schedular

Techinal Components

- Incorporated the principles into MLOps
- Which components do we need?
- How do we implement them in the ML systems design

Technical Components

C1 CI/CD Component (P1, P6, P9)

- Ensures continuous integration, continuous delivery and continuous deployment
- Build, test, delivery and deploy steps
- Feedback to developers regarding success or failure of steps

- Jenkins, GitHub actions, TeamCity

Technical Components

C2 Source Code Repository (P4, P5)

- Code storing and versioning
- Mulitple developers commit and merge their code

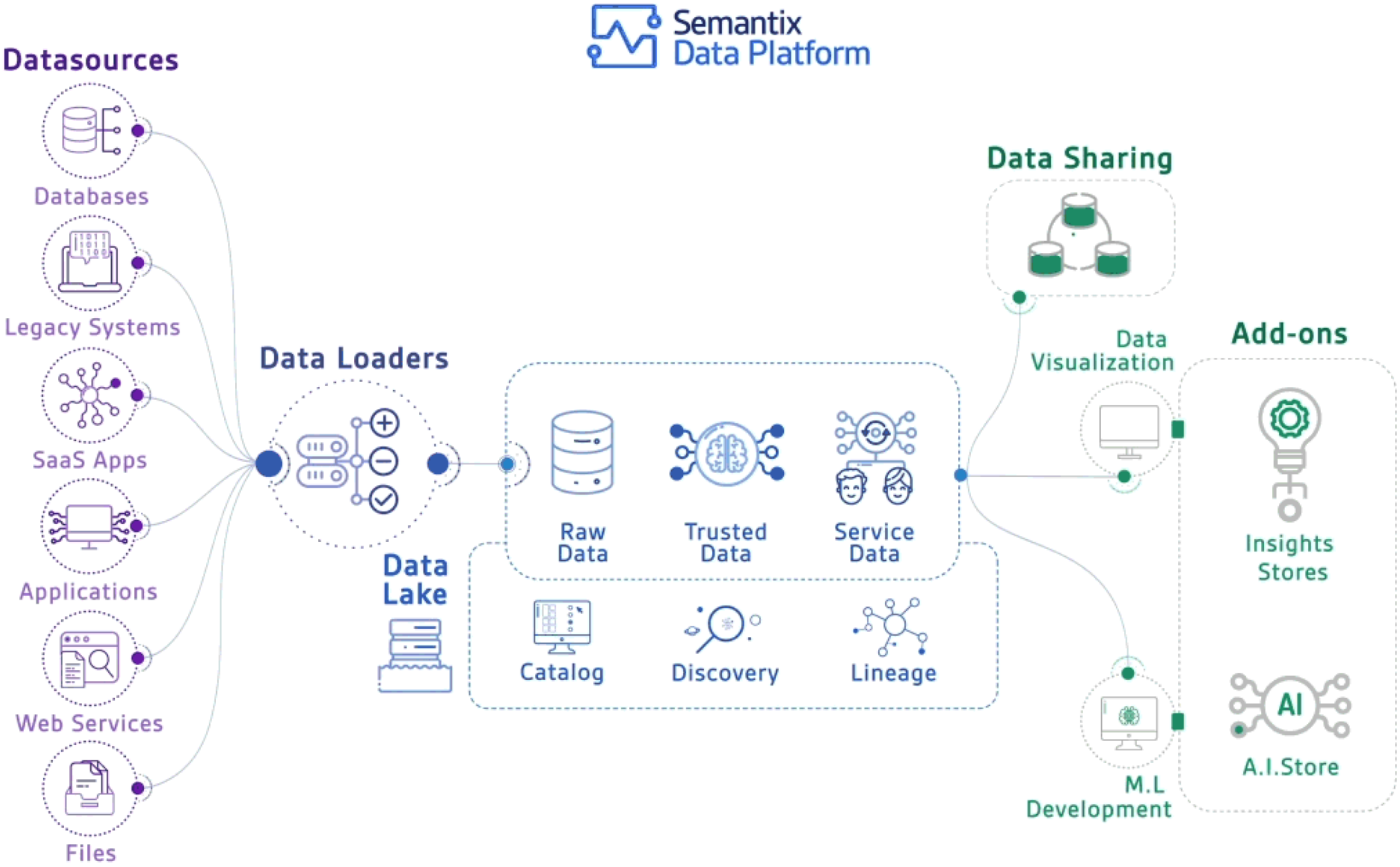
- Bitbucket, GitLab, GitHub, Gitea

Technical Components

C3 Workflow Orchestration Component (P2, P3, P6)

- Task orchestration of an ML workflow

- Apache Airflow, Kubeflow pipelines, Luigi, AWS SageMake Pipelines, Azure Pipelines, Dagster


Semantix
Data Platform
Datasources
Databases
Legacy Systems
SaaS Apps
Applications
Web Services
Files
Data Loaders
Data Lake
Raw Data
Trusted Data
Service Data
Catalog
Discovery
Lineage
Data Sharing
Data Visualization
M.L Development
Add-ons
Insights Stores
AI
A.I.Store

Technical Components

C4 Feature Store System (P3, P4)

- Central storage of commonly used features
- Offline feature store, normal latency
- Online feature store, low latency for predictions in production

- Google Feast, Amazon AWS Feature Store, Section.ai

Technical Components

C5 Model Training Infrastructure (P6)

- Providing the foundational computation resources: CPUs, RAM and GPUs
- Distributed or non-distributed
- Scalable and distributed infrastructure

- Kubernetes, Red Hat OpenShift

Technical Components

C6 Model Registry

- Stores the trained machine learning models together with their metadata

- Advanced storage: MLflow, AWS SageMaker, Model Registry, Microsoft Azure ML Model Registry, Neptune.ai

- Simple storage: Microsoft Azure Storage, Google Cloud Storage, Amazon AWS S3

Technical Components

C7 ML Metadata Stores

- Tracking of various kinds of metadata
- Can be configured in the model registry
- Training job information (training date, time, duration, …)
- Used parameters, resulting performance metrics, used data and code

- Orchestrators with built-in metadata stores: Kubeflow Pipelines, AWS SageMaker Pipelines, Azure ML, IBM Watson Studio

Technical Components

C8 Model Serving Component (P1)

- Online inference for real-time predictions
- Batch inference for predications using large volumes of input data
- Scalable and distributed model serving infrastructure is recommended

- Kubernetes or Docker to containerize the ML model
- Flask for REST API

Technical Components

C9 Monitoring Component (P8, P9)

- Continuous monitoring of the model serving performance
- Monitoring ML infrastructure, CI/CD and orchestration

- Prometheus with Grafan, ELK stock (Elasticseach, Logstash and Kibana), TensorBoard
- Built-in monitoring capabilities: Kubeflow, MLflow, AWS SageMaker

Roles

- MLOps is an interdisciplinary group process
- Interplay of different roles is crucial to design, manage, automate and operate an ML system

Roles

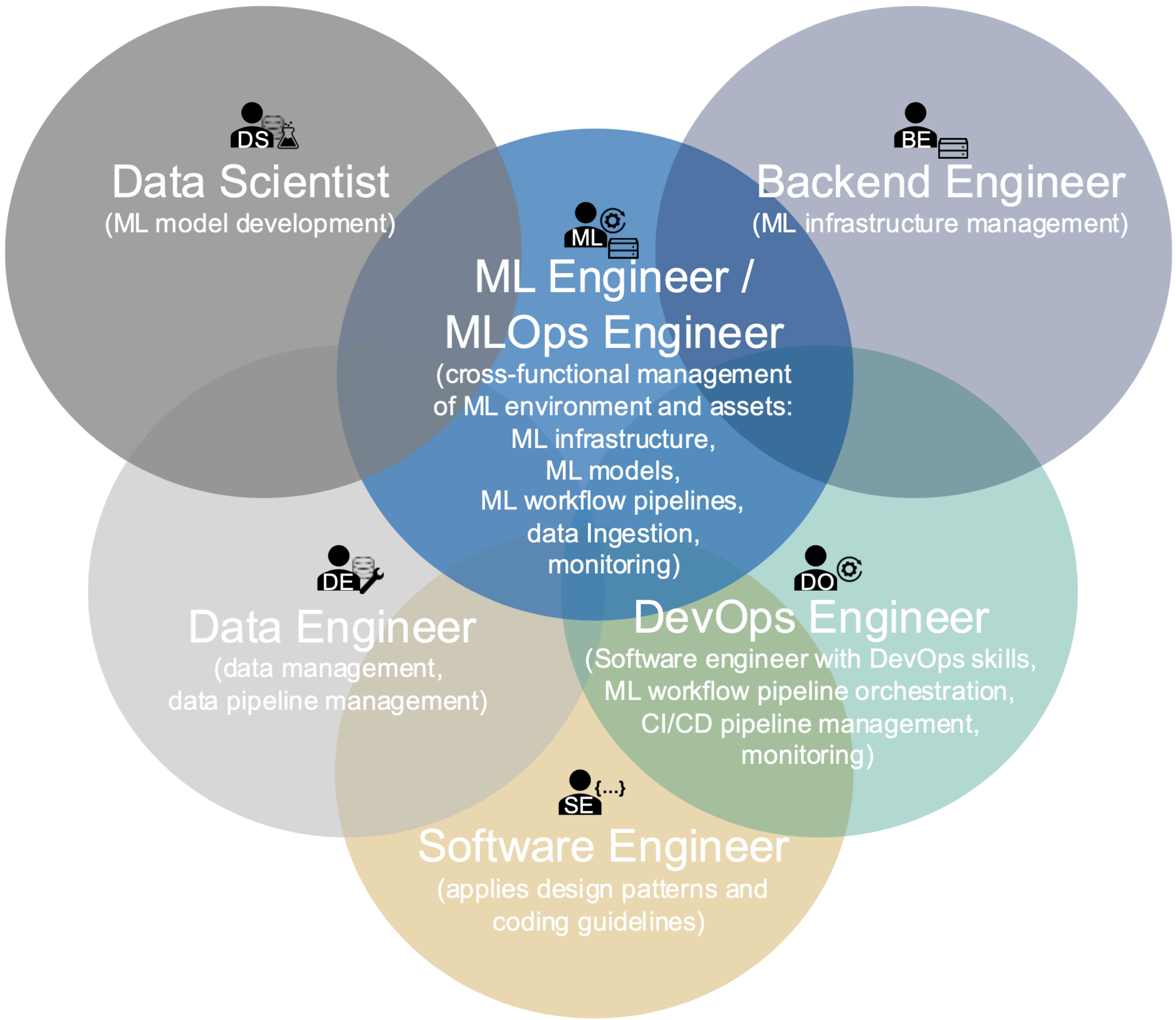
- R1 Business Stakeholder
    - Define the business goal to be achieved with ML
- R2 Solution Architect
    - Deines technologies to be used
- R3 Data Scientist
    - Translates business problem into ML problem
    - Model engineering: selection of algorithm and hyperparameters

Roles

- R4 Data Engineer
  - Builds and manages data and feature engineering pipelines
  - Ensures data ingestion to the database
- R5 Software Engineer
  - Applies software design patterns, coding guidelines and best practices
  - Turn a raw ML problem into a well-engineered product

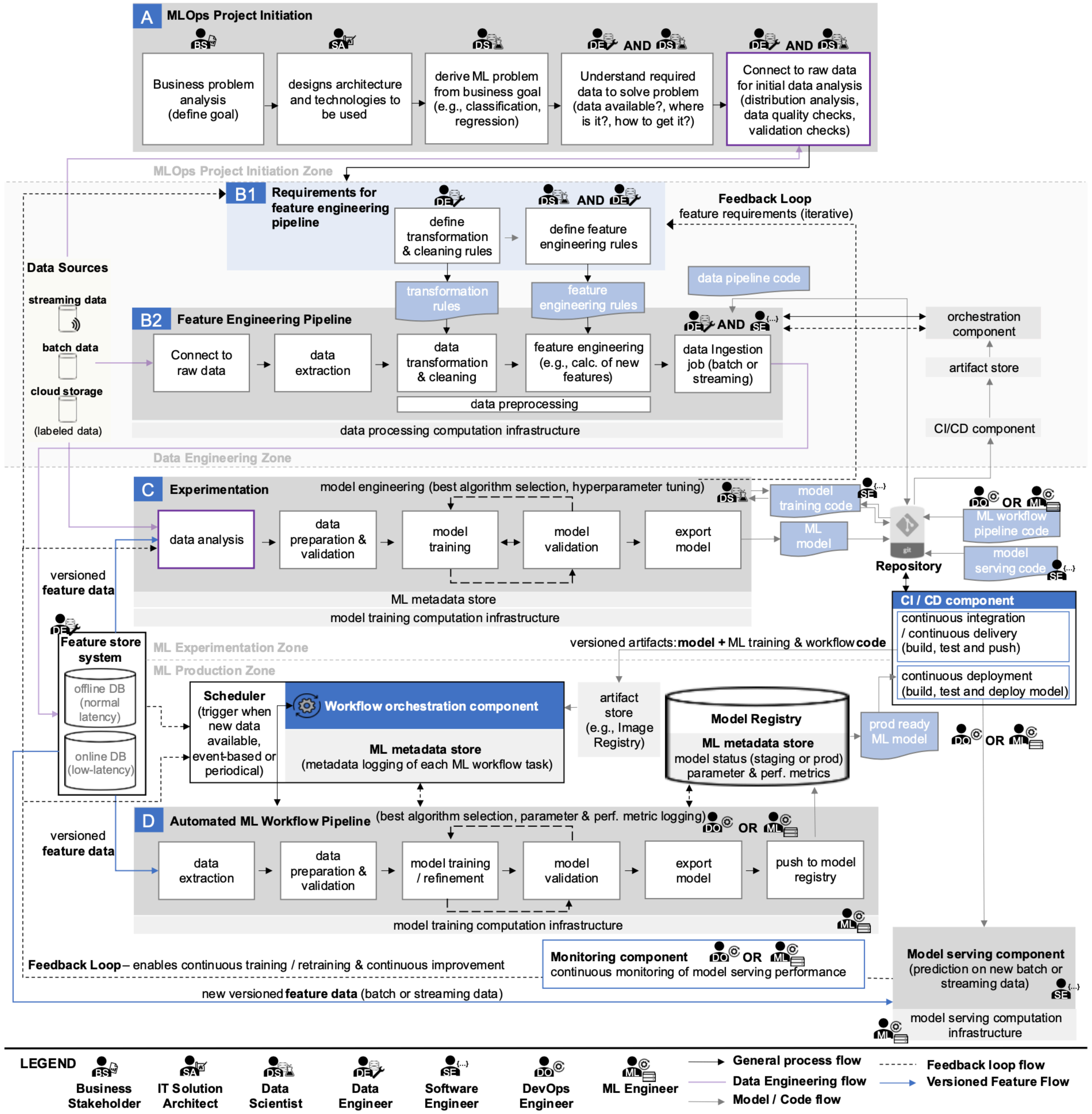
Roles

- R6 DevOps Engineer
  - Bridges the gap between development and operations
  - CI/CD automation, ML workflow orchestration, model deployment and monitoring
- R7 ML Engineer/MLOps Engineer
  - Incorporates knowledge from data scientists, data engineers, software engineers, DevOps engineers and backend engineers


DS
Data Scientist
(ML model development)
BE
Backend Engineer
(ML infrastructure management)
ML
ML Engineer /
MLOps Engineer
(cross-functional management
of ML environment and assets:
ML infrastructure,
ML models,
ML workflow pipelines,
data Ingestion,
monitoring)
DE
Data Engineer
(data management,
data pipeline management)
DO
DevOps Engineer
(Software engineer with DevOps skills,
ML workflow pipeline orchestration,
CI/CD pipeline management,
monitoring)
SE
{...}
Software Engineer
(applies design patterns and
coding guidelines)

Architecture and Workflow

- And end-to-end process from MLOps project initiation to the model serving.
- (A) The MLOps project initiation steps
- (B) The feature engineering pipeline, including the data ingestions to the feature store
- (C) The experimentation
- (D) The automated ML workflow pipeline up to the model serving


A MLOps Project Initiation
BS
Business problem analysis (define goal)
SA
designs architecture and technologies to be used
DS
derive ML problem from business goal (e.g., classification, regression)
DE AND DS
Understand required data to solve problem (data available?, where is it?, how to get it?)
DE AND DS
Connect to raw data for initial data analysis (distribution analysis, data quality checks, validation checks)
MLOps Project Initiation Zone
B1 Requirements for feature engineering pipeline
DE
define transformation & cleaning rules
DS AND DE
define feature engineering rules
Feedback Loop
feature requirements (iterative)
transformation rules
feature engineering rules
data pipeline code
Data Sources
streaming data
batch data
cloud storage
(labeled data)
B2 Feature Engineering Pipeline
Connect to raw data
data extraction
data transformation & cleaning
feature engineering (e.g., calc. of new features)
DE AND SE
data Ingestion job (batch or streaming)
data preprocessing
data processing computation infrastructure
orchestration component
artifact store
CI/CD component
Data Engineering Zone
C Experimentation
model engineering (best algorithm selection, hyperparameter tuning)
DS
data analysis
data preparation & validation
model training
model validation
export model
ML metadata store
model training computation infrastructure
model training code
SE
ML model
DO OR ML
ML workflow pipeline code
model serving code
SE
Repository
versioned feature data
DE
Feature store system
offline DB (normal latency)
online DB (low-latency)
CI / CD component
continuous integration / continuous delivery (build, test and push)
continuous deployment (build, test and deploy model)
ML Experimentation Zone
ML Production Zone
versioned artifacts: model + ML training & workflow code
Scheduler (trigger when new data available, event-based or periodical)
Workflow orchestration component
ML metadata store (metadata logging of each ML workflow task)
artifact store (e.g., Image Registry)
Model Registry
ML metadata store
model status (staging or prod)
parameter & perf. metrics
prod ready ML model
DO OR ML
D Automated ML Workflow Pipeline
(best algorithm selection, parameter & perf. metric logging)
DO OR ML
versioned feature data
data extraction
data preparation & validation
model training / refinement
model validation
export model
push to model registry
model training computation infrastructure
ML
Feedback Loop – enables continuous training / retraining & continuous improvement
Monitoring component
DO OR ML
continuous monitoring of model serving performance
Model serving component (prediction on new batch or streaming data)
SE
model serving computation infrastructure
ML
new versioned feature data (batch or streaming data)
LEGEND
BS Business Stakeholder
SA IT Solution Architect
DS Data Scientist
DE Data Engineer
SE Software Engineer
DO DevOps Engineer
ML ML Engineer
General process flow
Data Engineering flow
Model / Code flow
Feedback loop flow
Versioned Feature Flow