

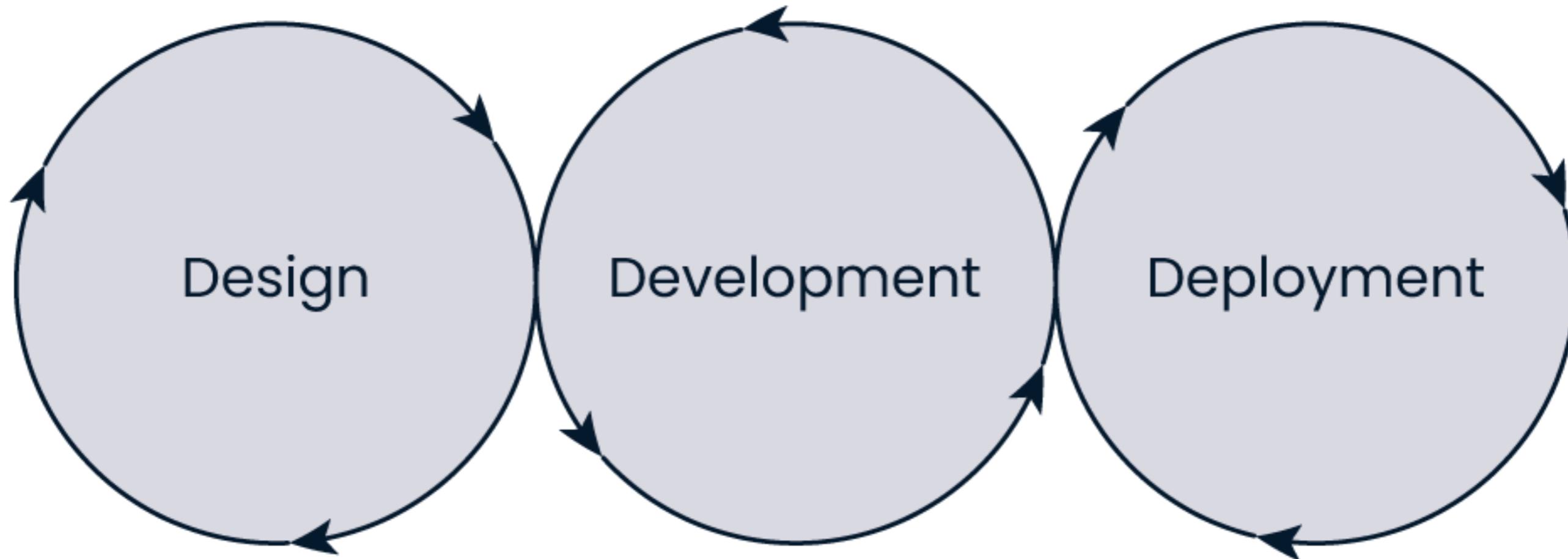
Preparing model for deployment

MLOPS CONCEPTS



Folkert Stijnman
ML Engineer

Runtime environment

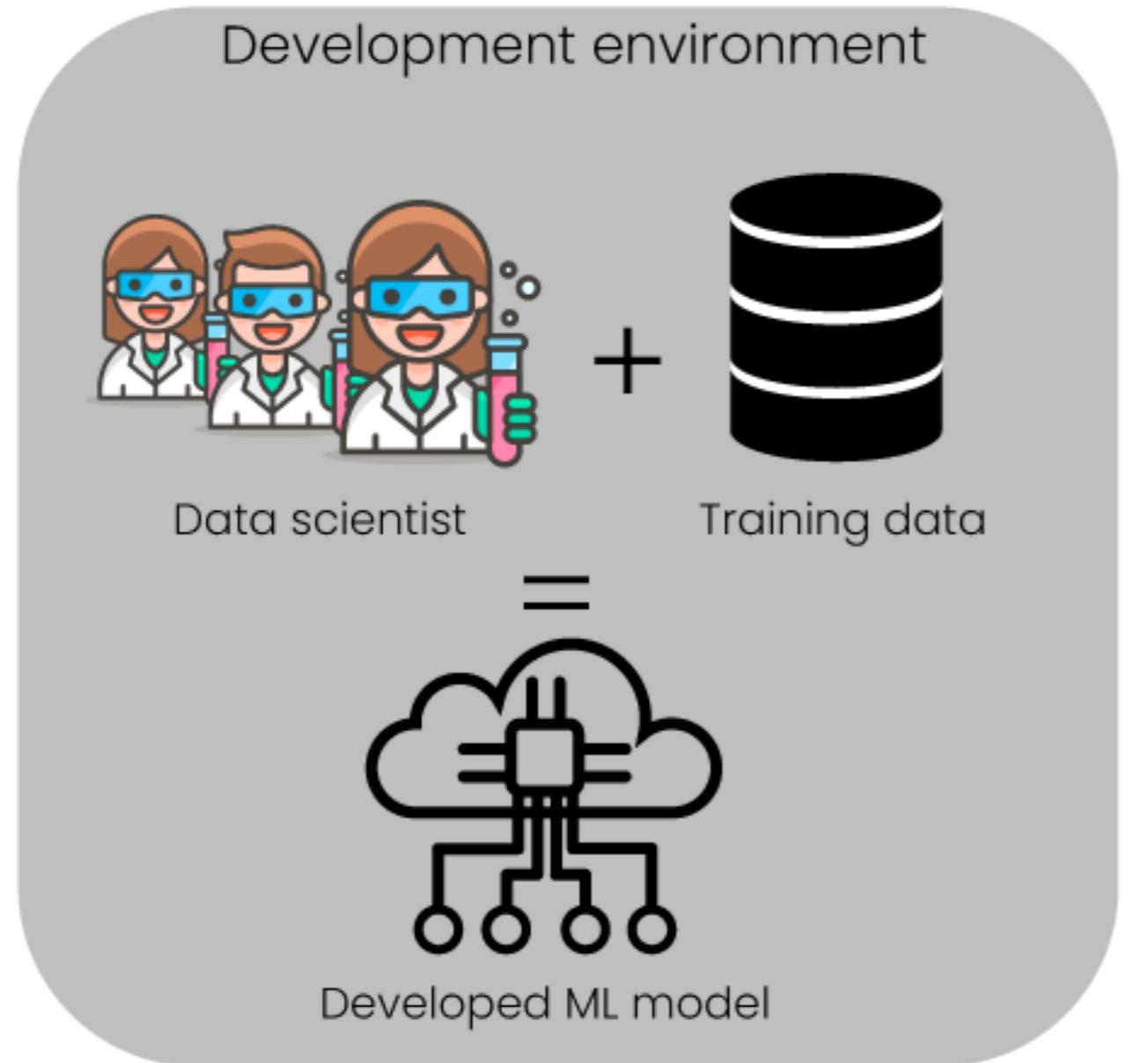


- Added value
- Business requirements
- Key metrics
- Data processing

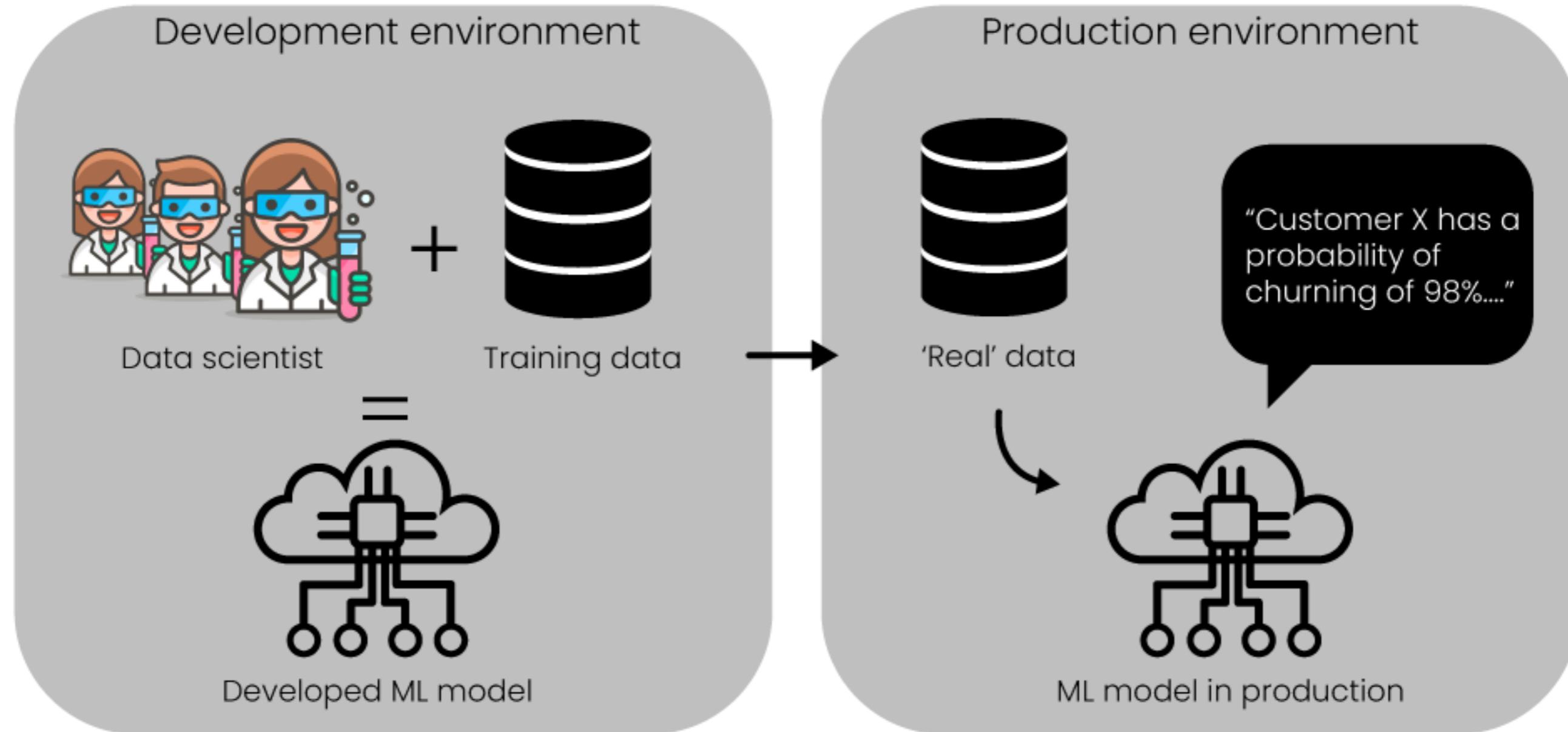
- Feature engineering
- Experiment tracking
- Model training & evaluation

- Runtime environments
- Microservices architecture
- CI/CD pipeline
- Monitoring & retraining

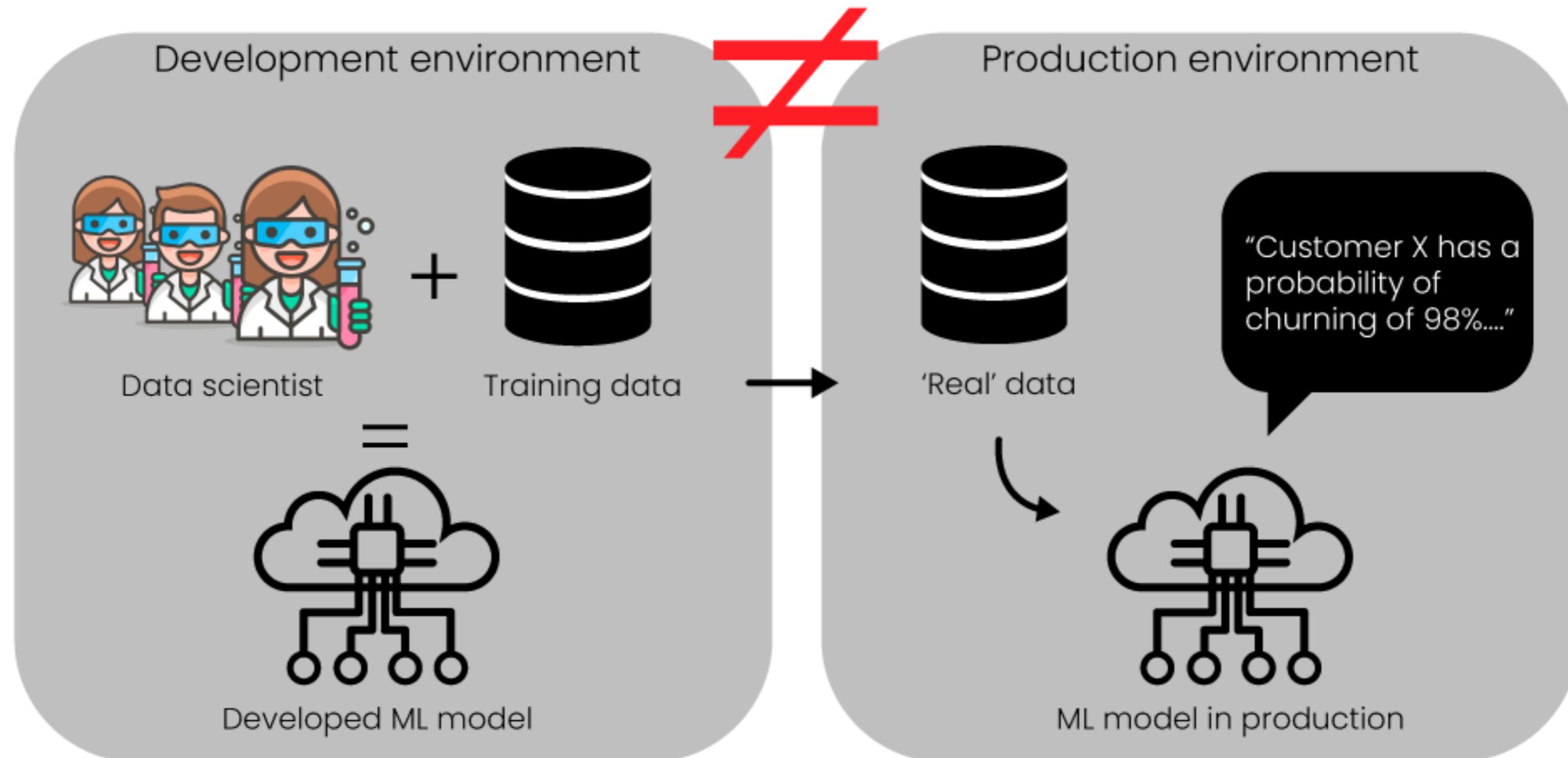
Development to deployment



Development to deployment



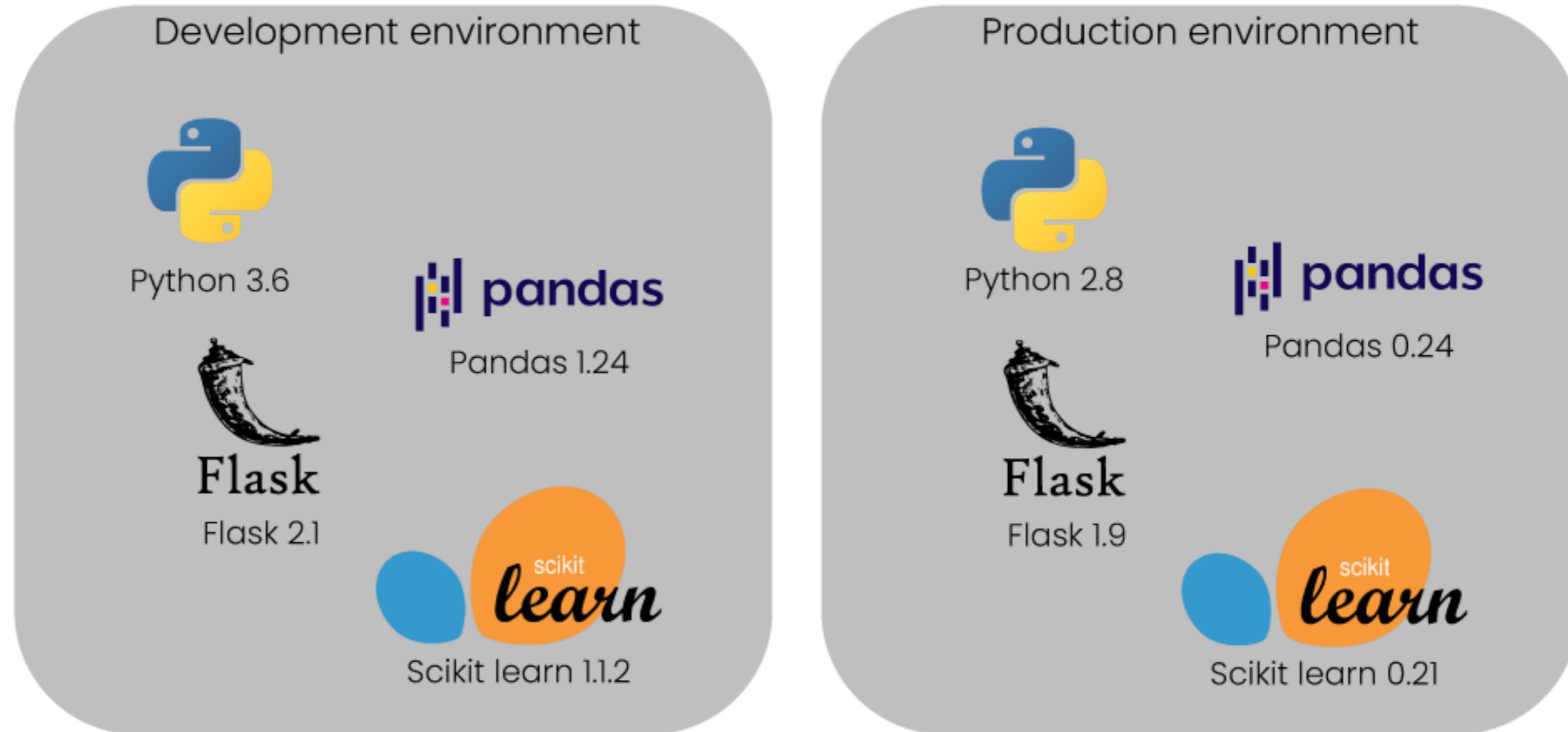
Development to deployment



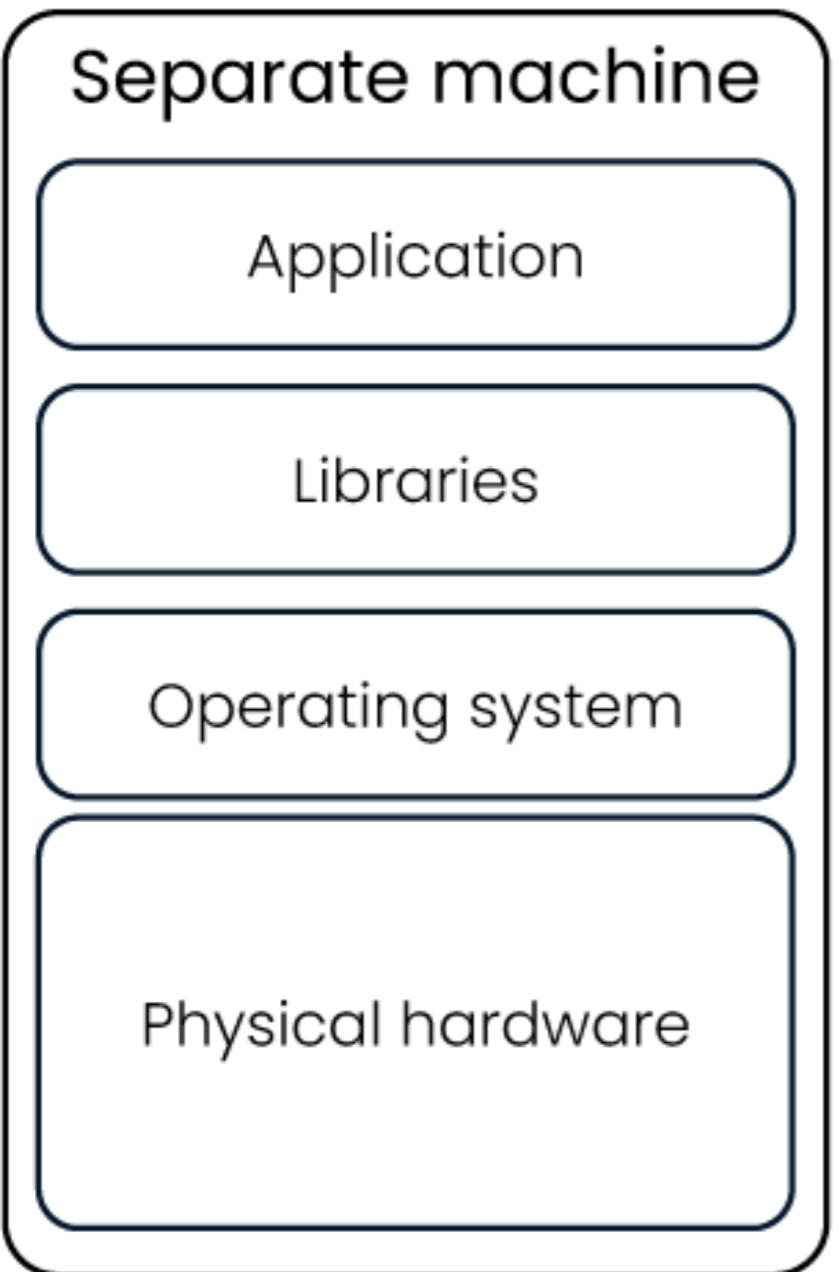
Runtime environments



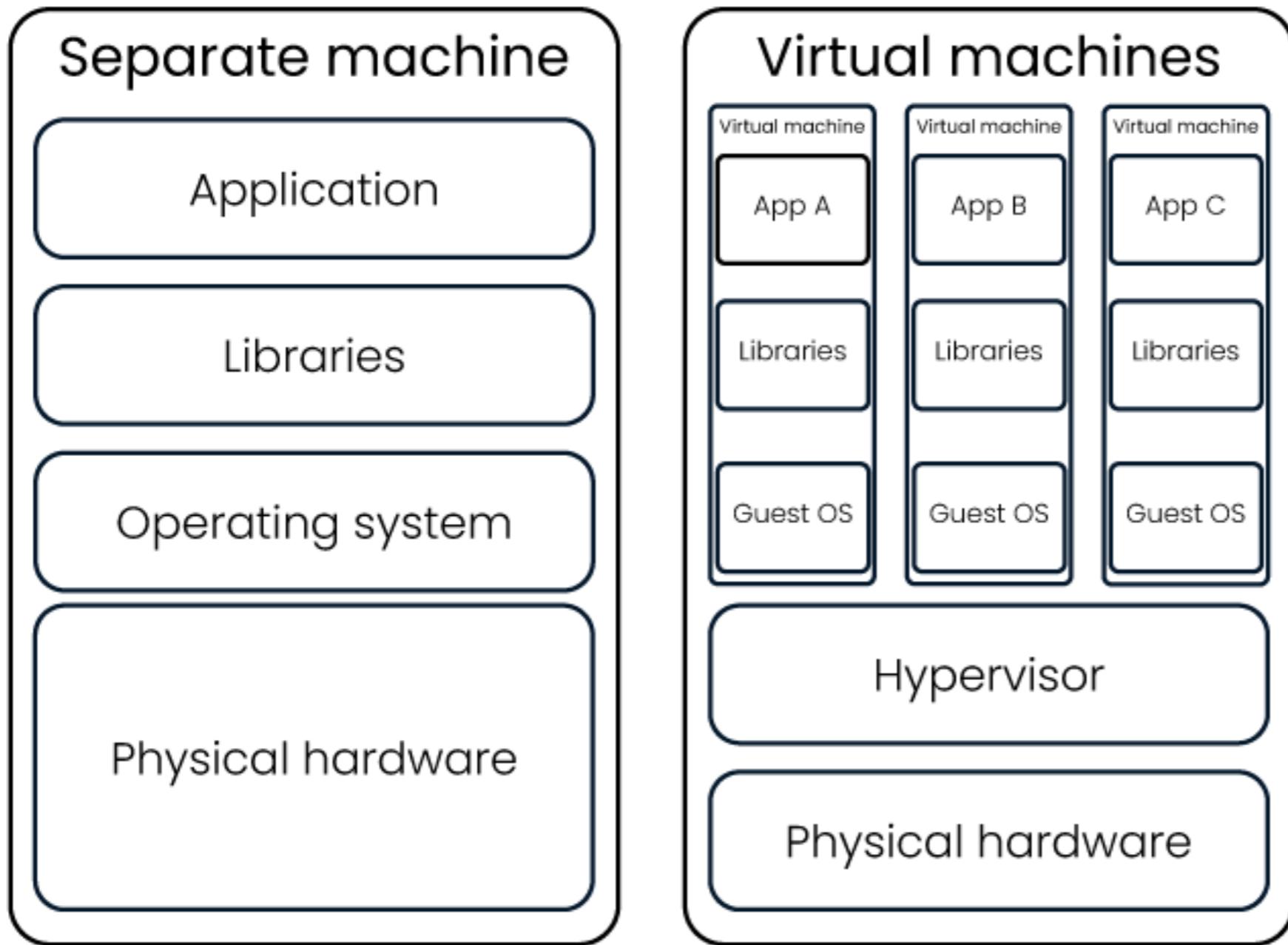
Runtime environments



Mitigate different environments



Mitigate different environments



Mitigate different environments

Separate machine

Application

Libraries

Operating system

Physical hardware

Virtual machines

Virtual machine

App A

Libraries

Guest OS

Virtual machine

App B

Libraries

Guest OS

Virtual machine

App C

Libraries

Guest OS

Hypervisor

Physical hardware

Containers

Container

App A

Libraries

Container engine

Hypervisor

Physical hardware

Benefits containers

- Easier to maintain
- Portable
- Fast to start up



Let's practice!

MLOPS CONCEPTS

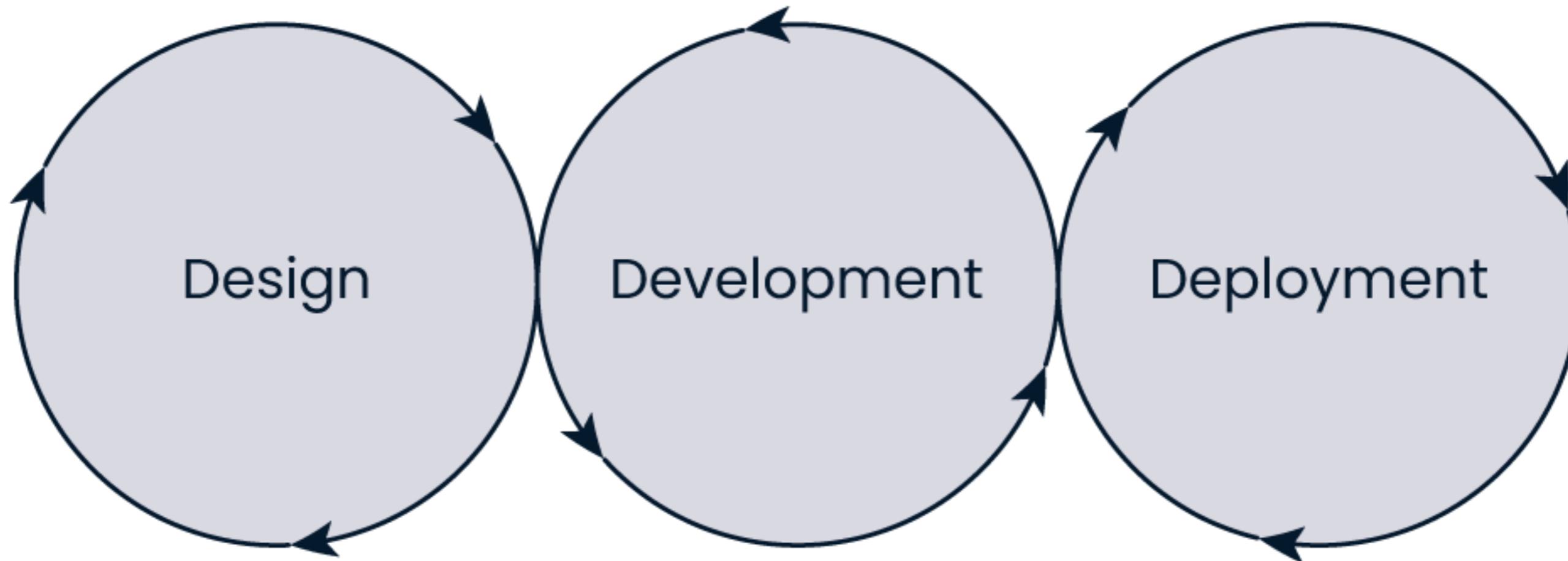
Machine learning deployment architecture

MLOPS CONCEPTS



Folkert Stijnman
ML Engineer

Microservices architecture



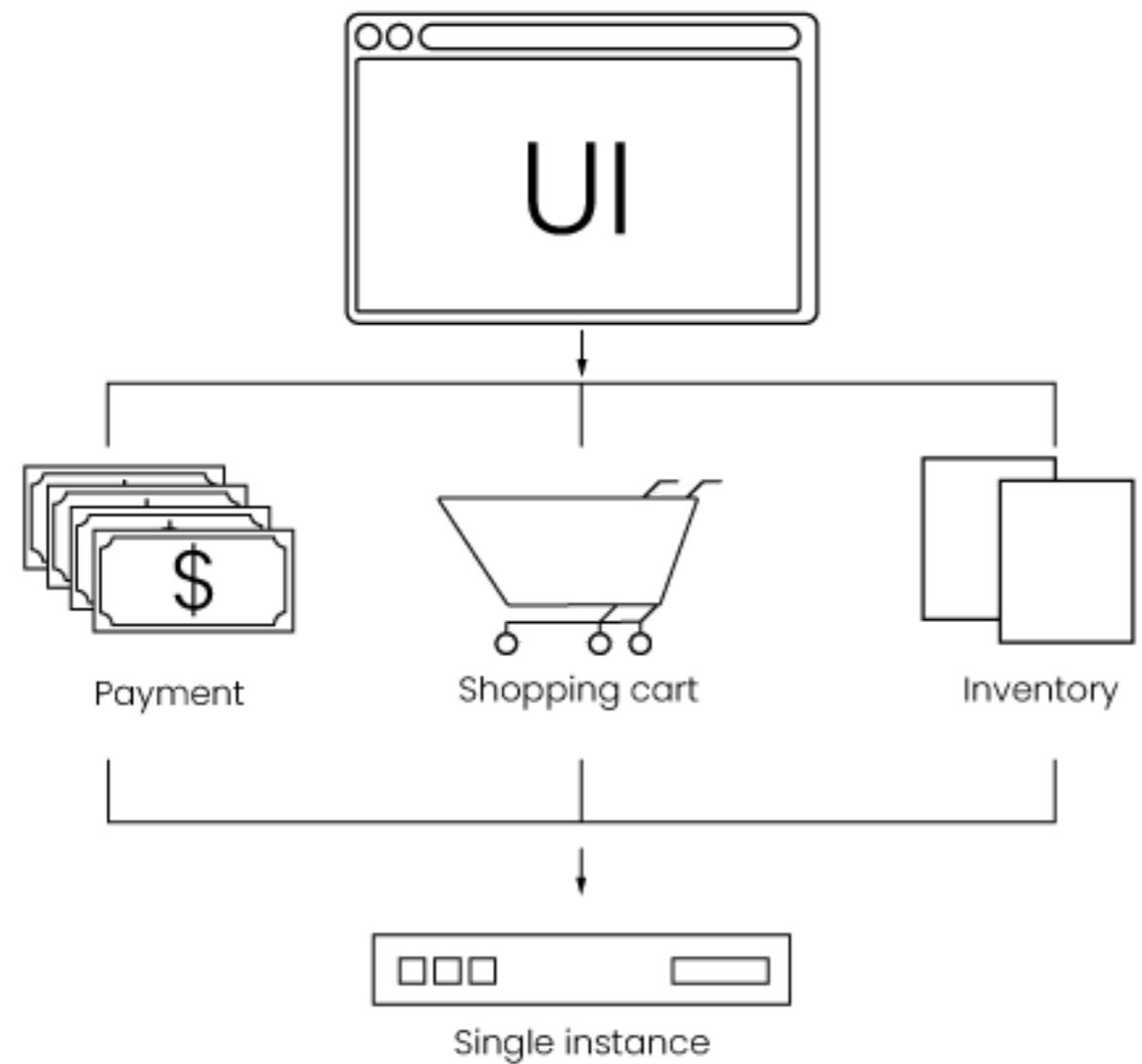
- Added value
- Business requirements
- Key metrics
- Data processing

- Feature engineering
- Experiment tracking
- Model training & evaluation

- Runtime environments
- Microservices architecture
- CI/CD pipeline
- Monitoring & retraining

Monolith vs. microservice architecture

Monolithic architecture

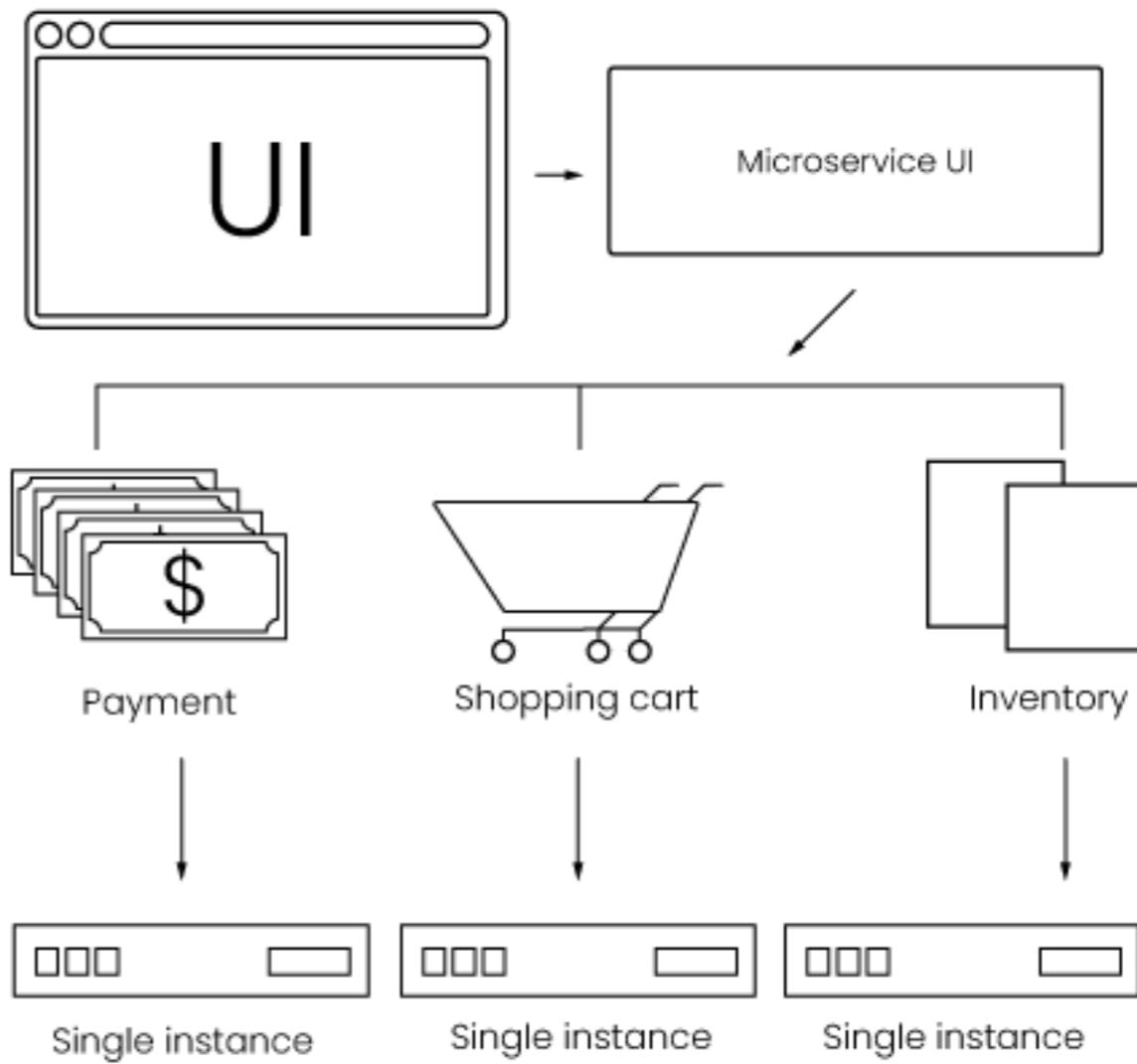


- One uniform application containing all services

Monolith vs. microservice architecture

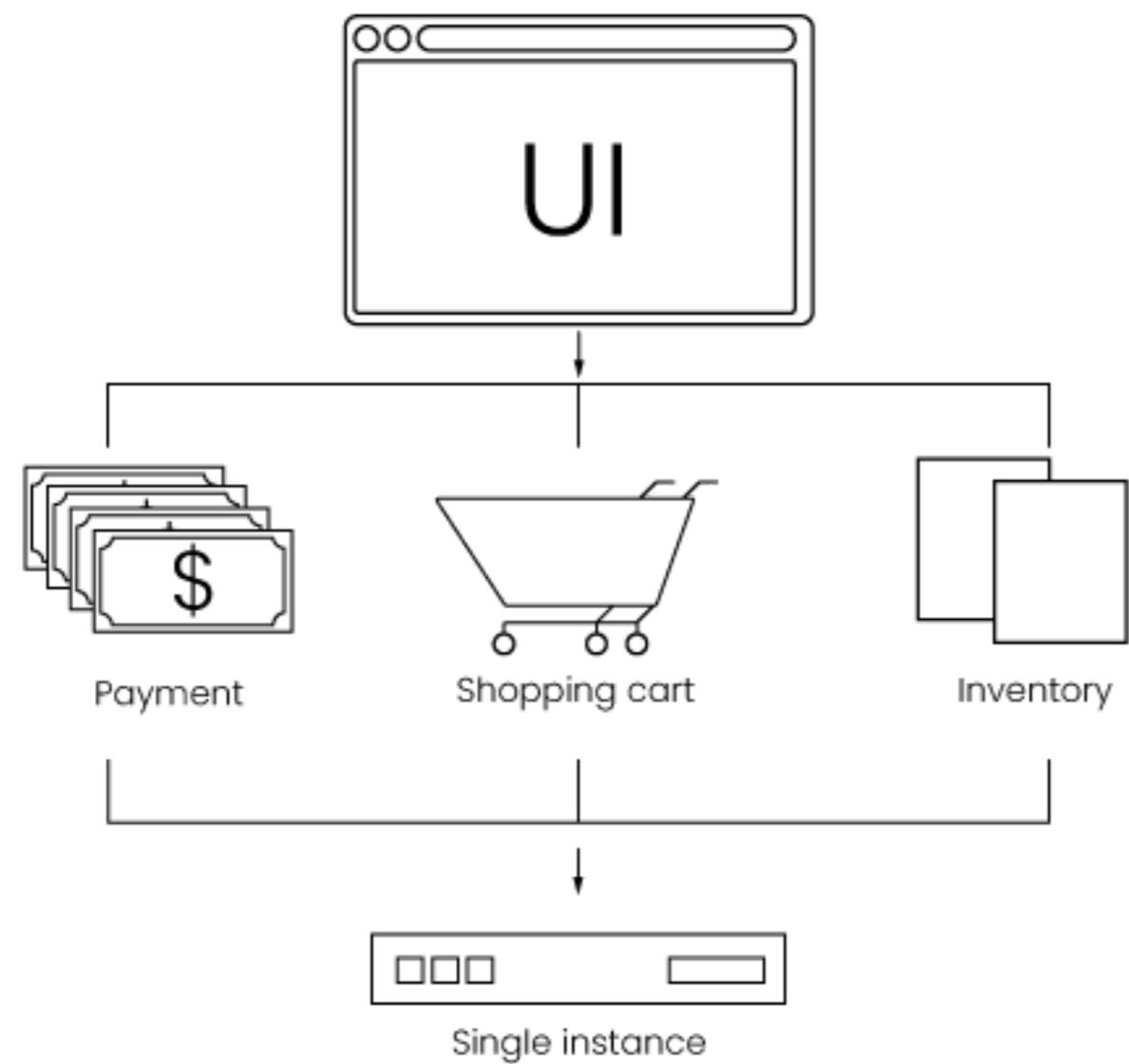
- Collection of smaller, independent services

Microservices architecture

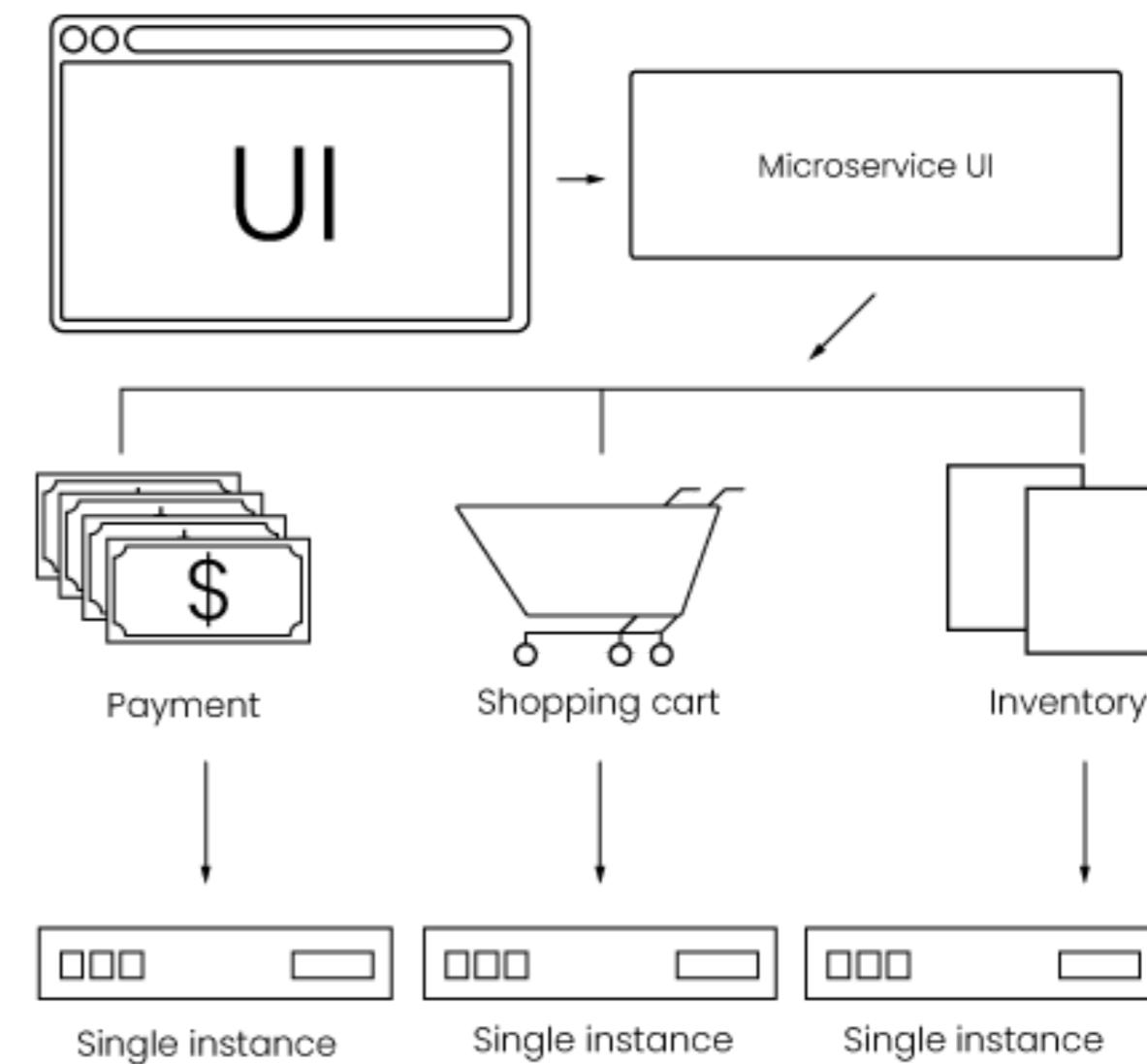


Monolith vs. microservice architecture

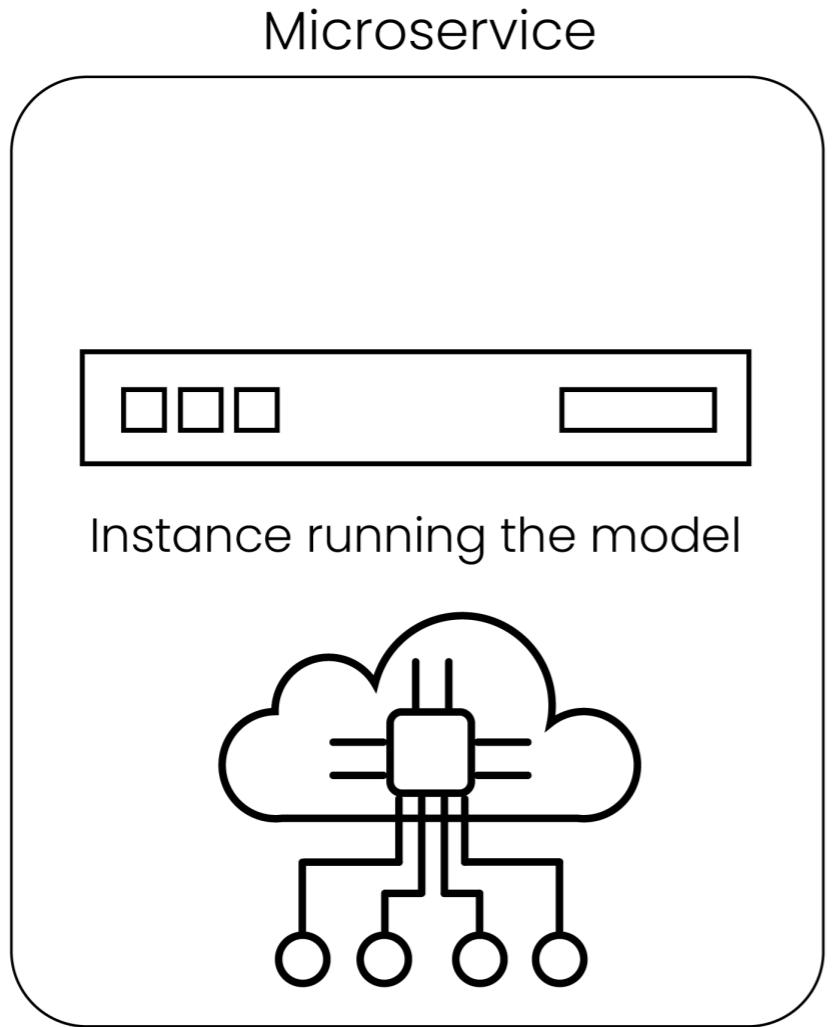
Monolithic architecture



Microservices architecture

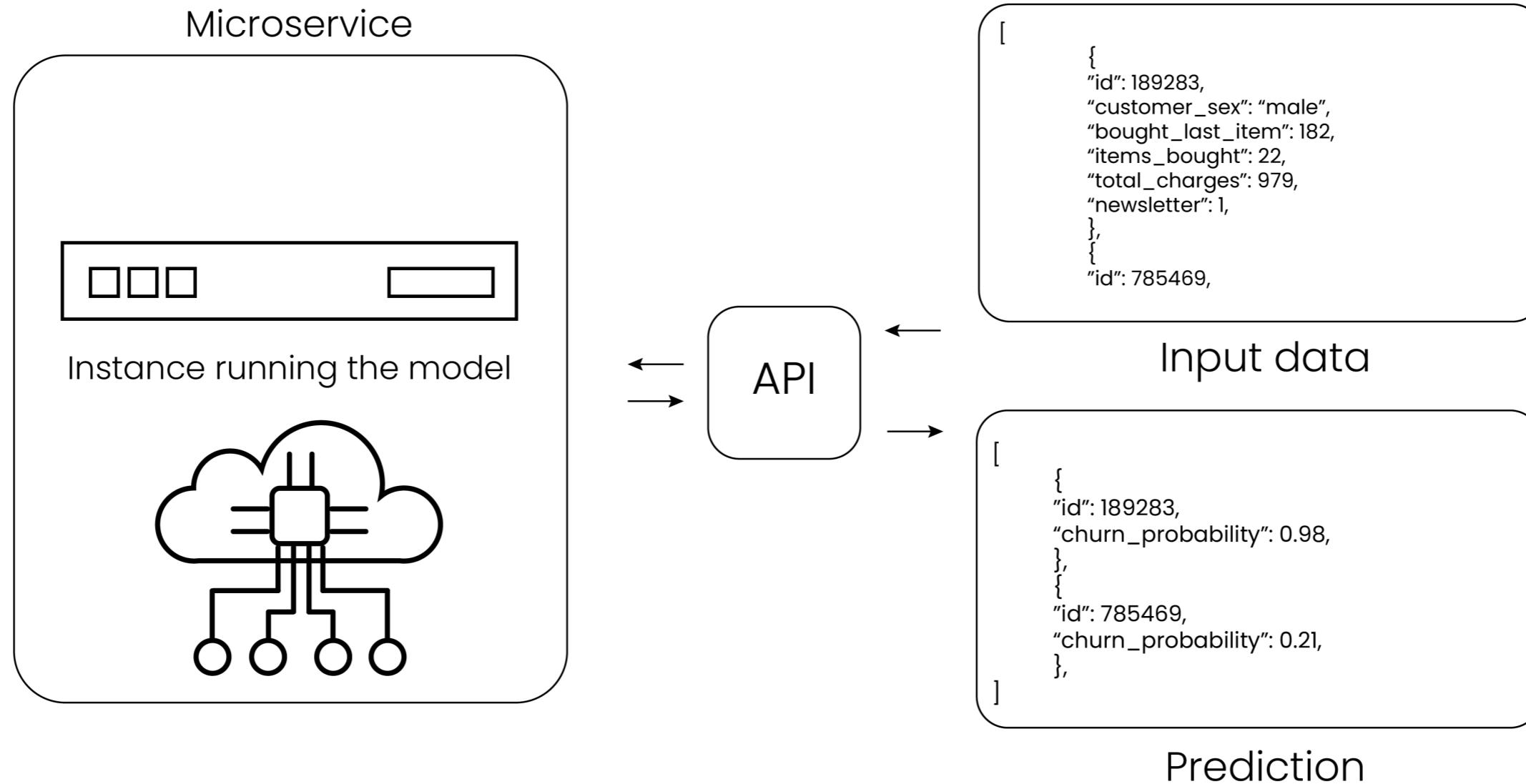


Inferencing

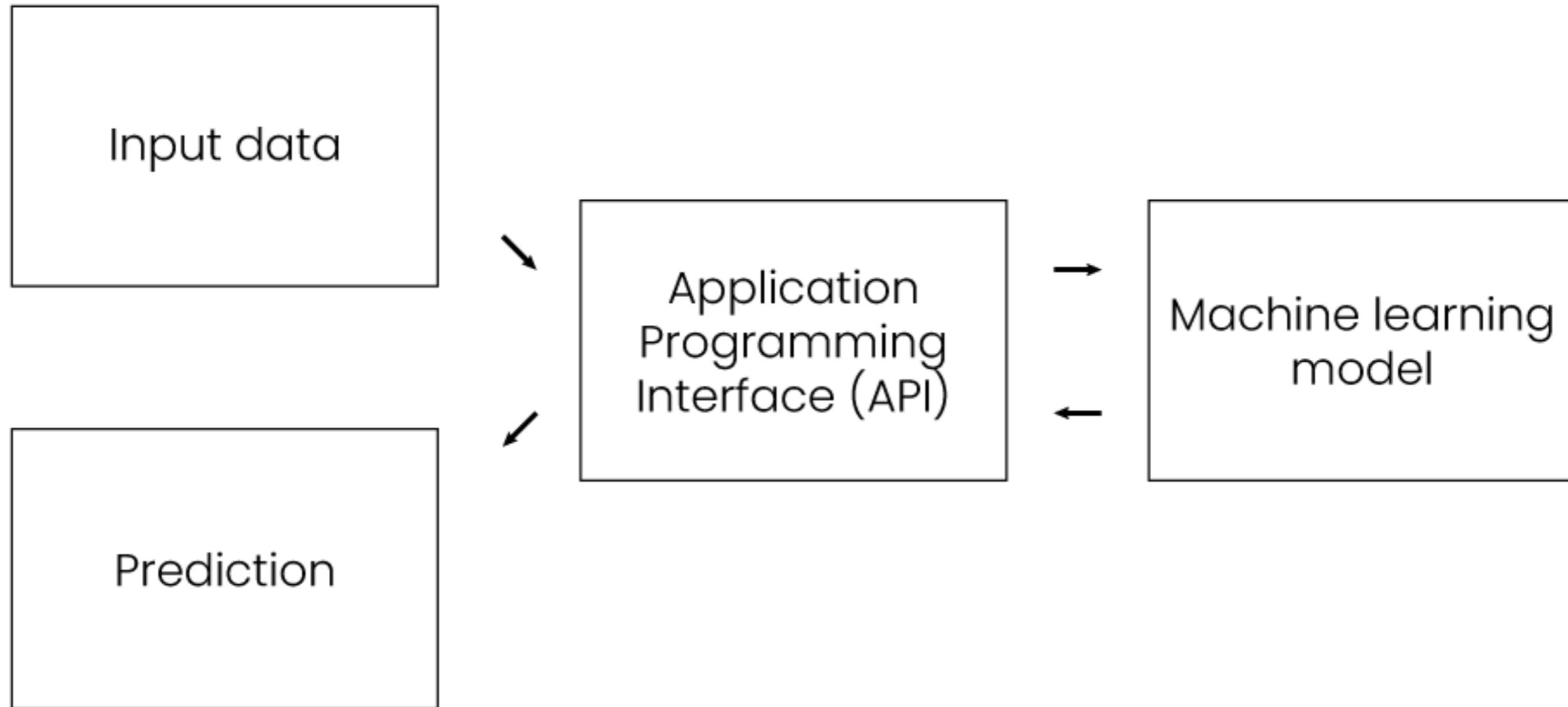


Inferencing

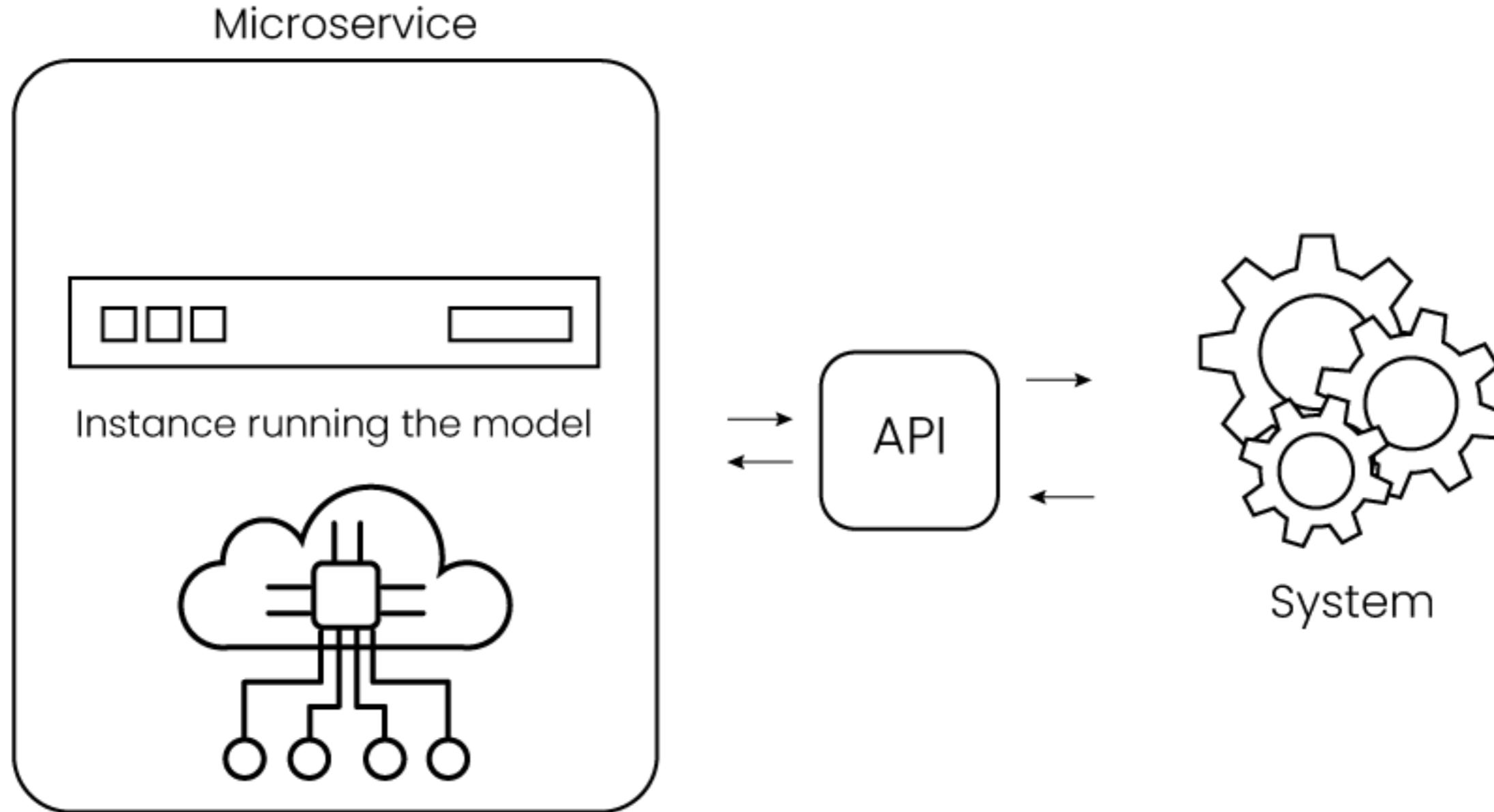
Inferencing is the process in which we send new input to the machine learning model and receive output from the model.



Application Programming Interface (API)



Integration



Let's practice!

MLOPS CONCEPTS

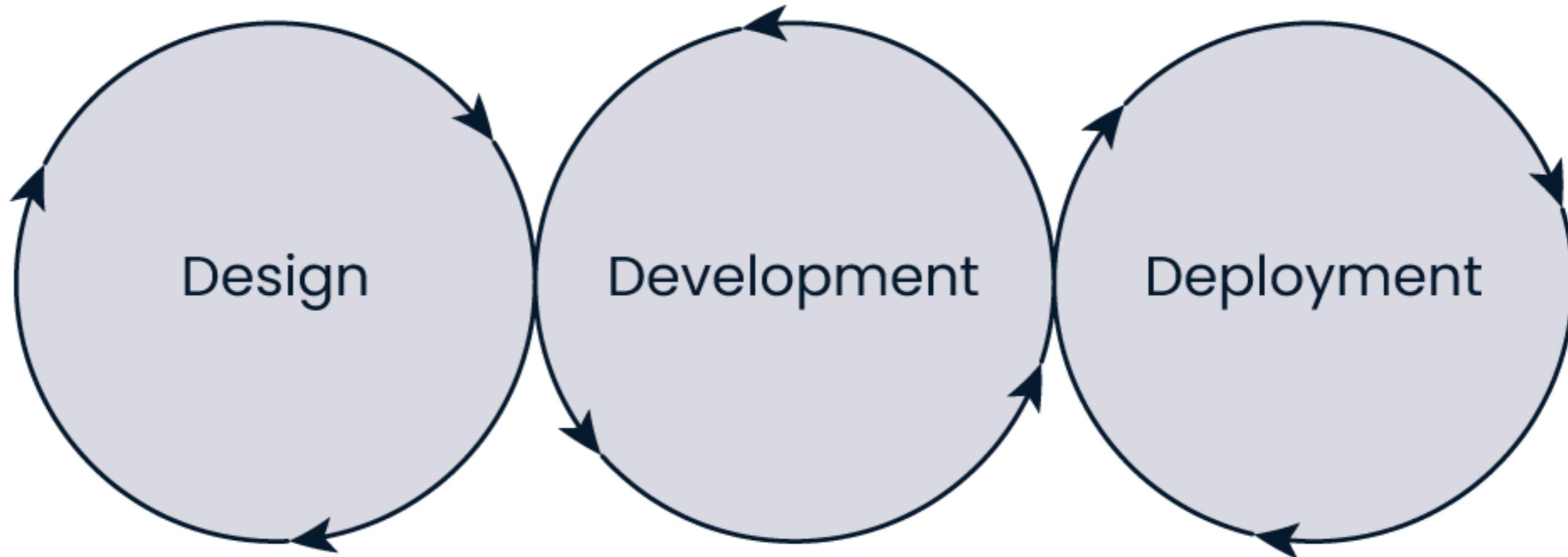
CI/CD and deployment strategy

MLOPS CONCEPTS



Folkert Stijnman
ML Engineer

CI/CD pipeline

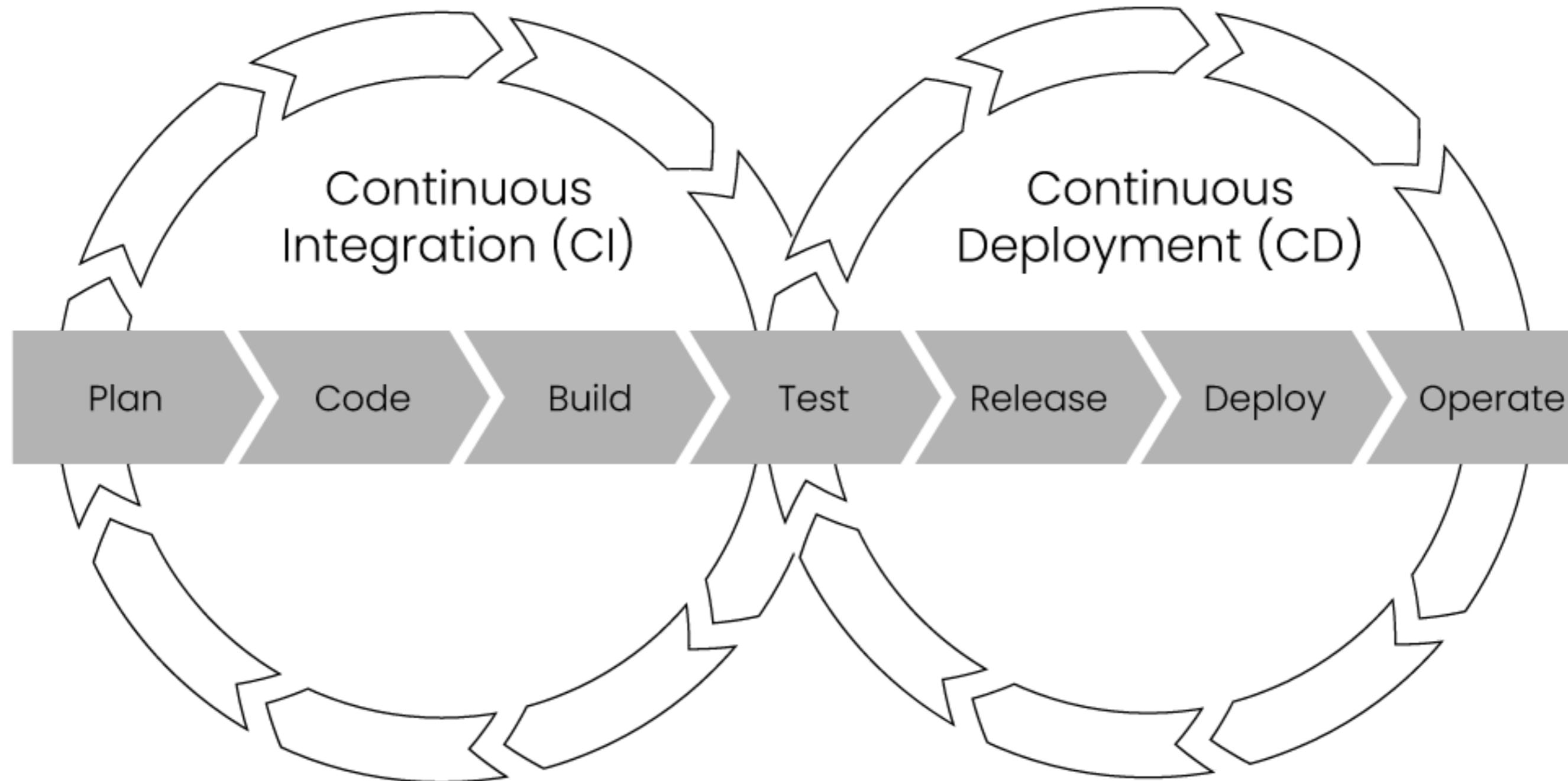


- Added value
- Business requirements
- Key metrics
- Data processing

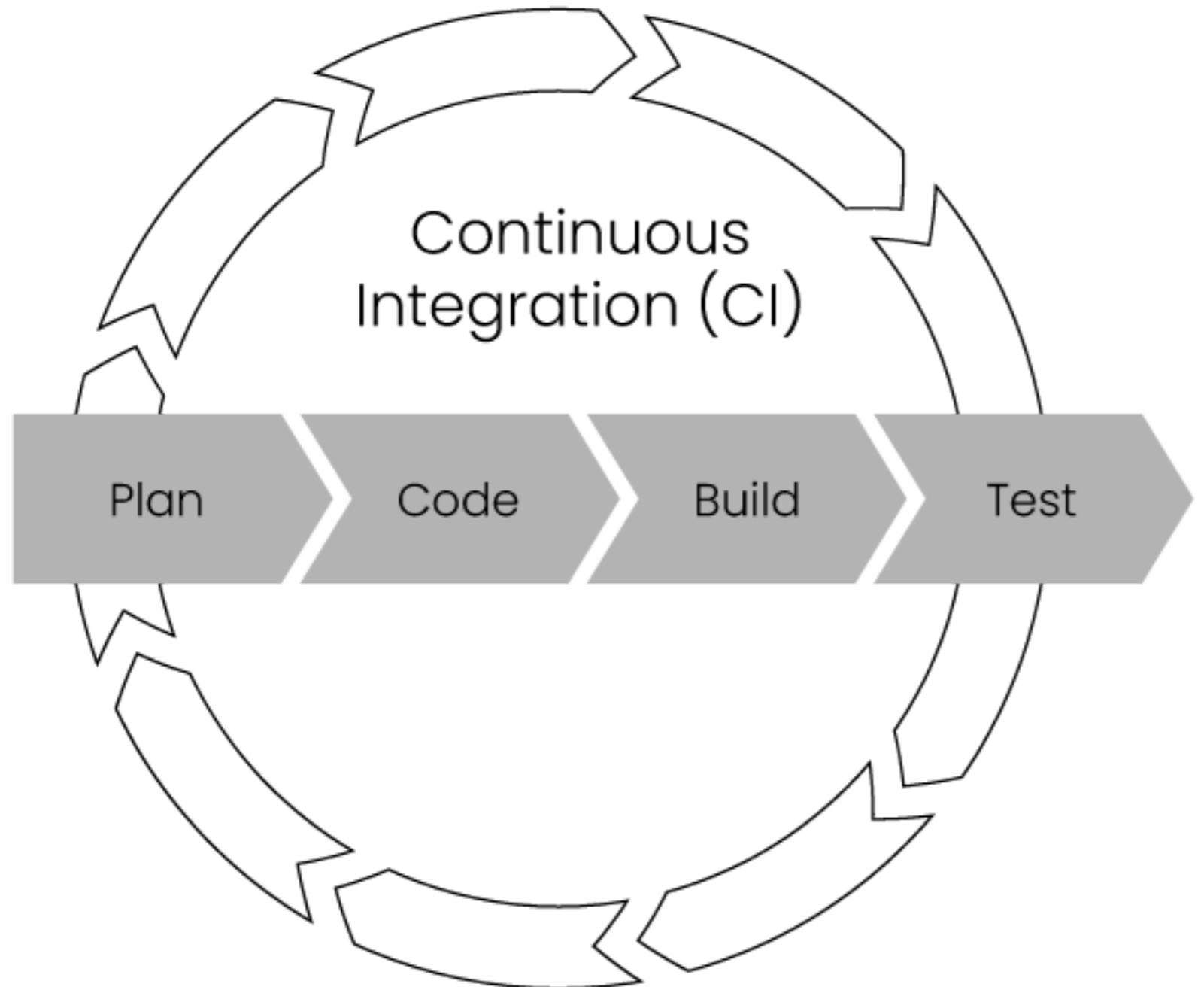
- Feature engineering
- Experiment tracking
- Model training & evaluation

- Runtime environments
- Microservices architecture
- CI/CD pipeline
- Monitoring & retraining

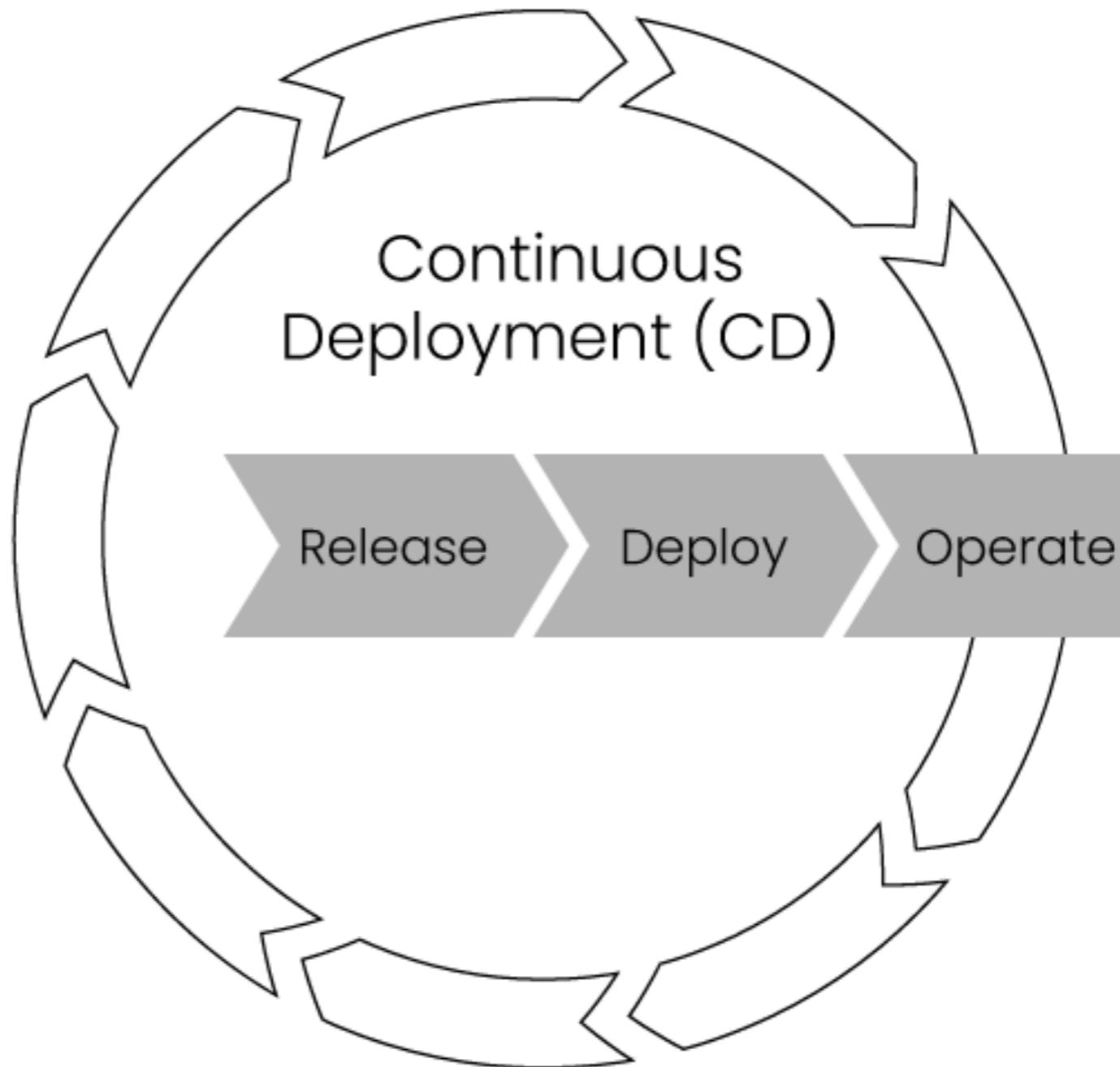
CI/CD



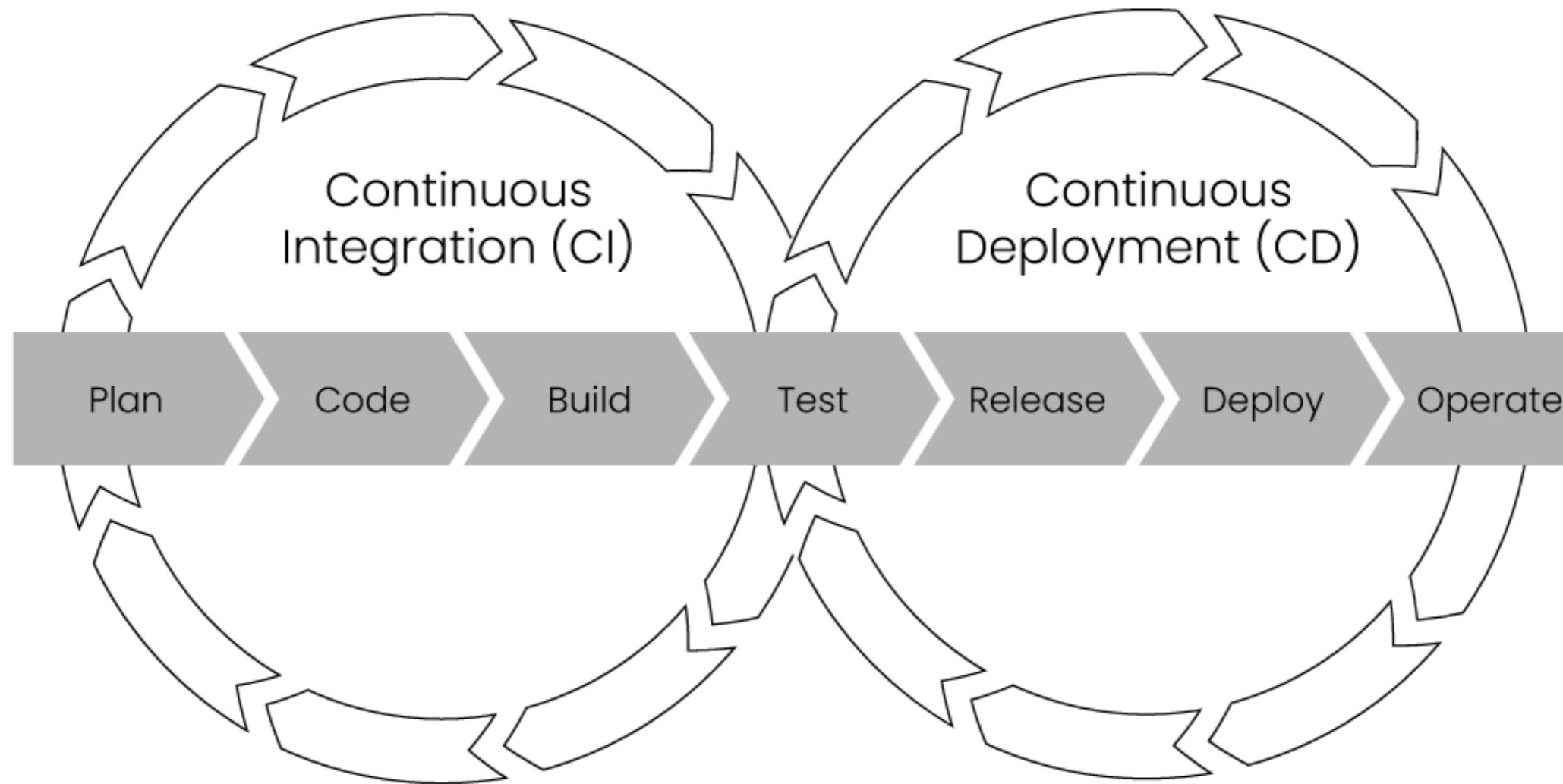
Continuous Integration



Continuous Deployment



CI/CD pipeline

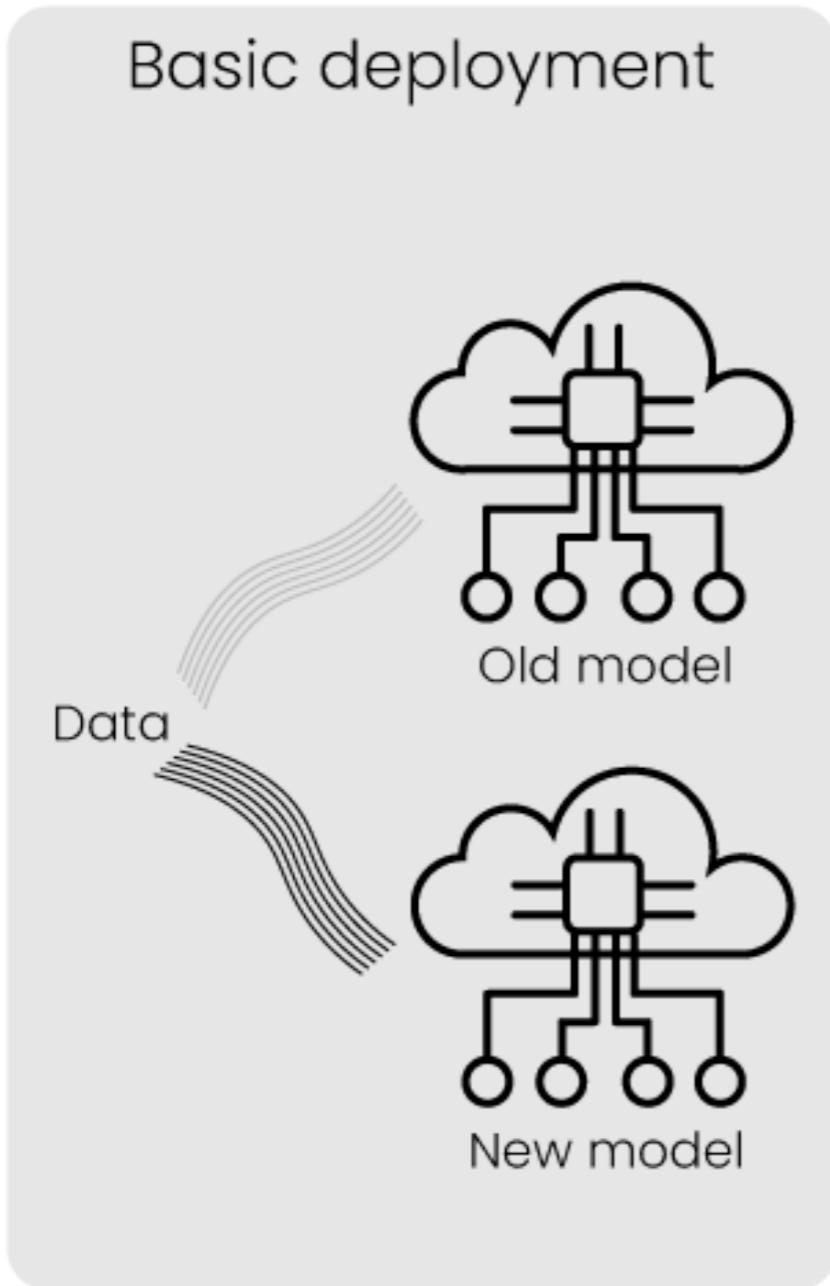


- Continuous integration: practices while code is being written
- Continuous deployment: practices after code is completed

Deployment strategies

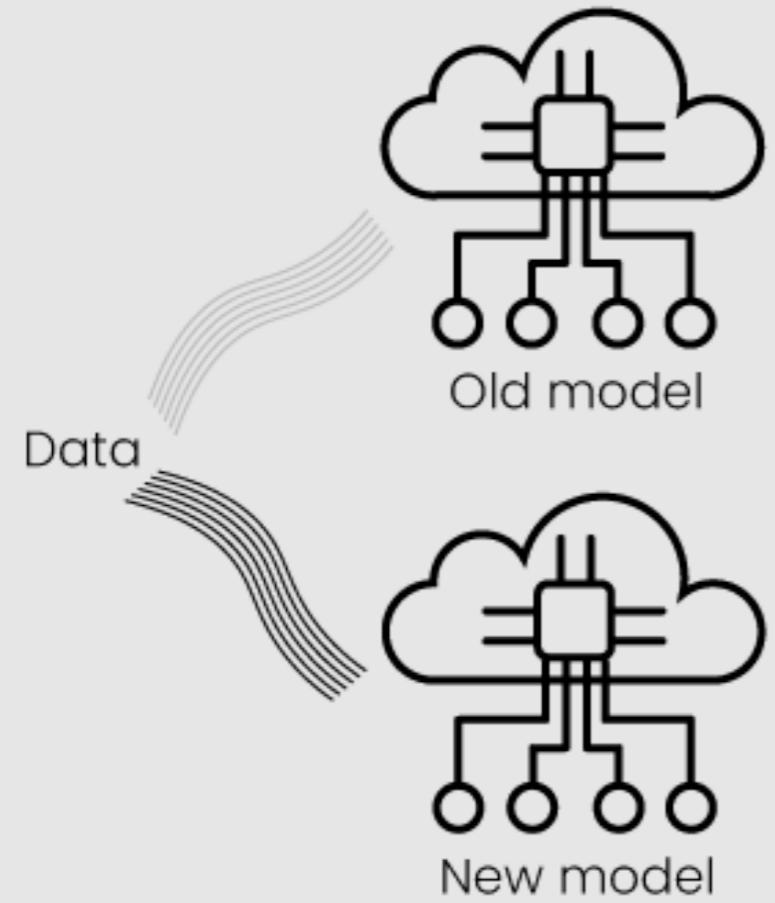
- Basic deployment
- Shadow deployment
- Canary deployment

Deployment strategies

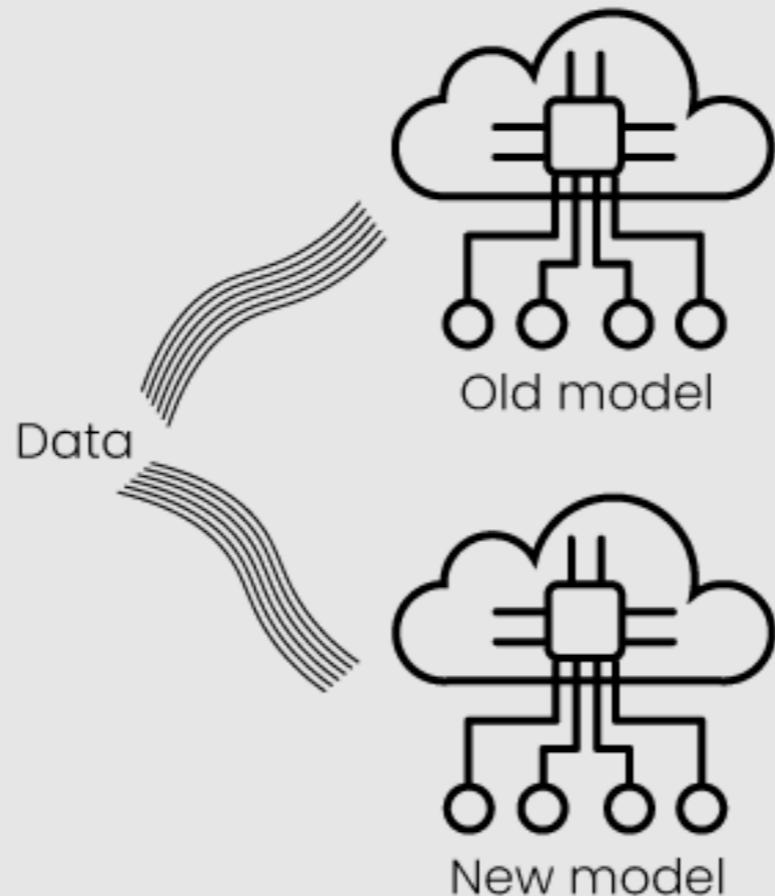


Deployment strategies

Basic deployment

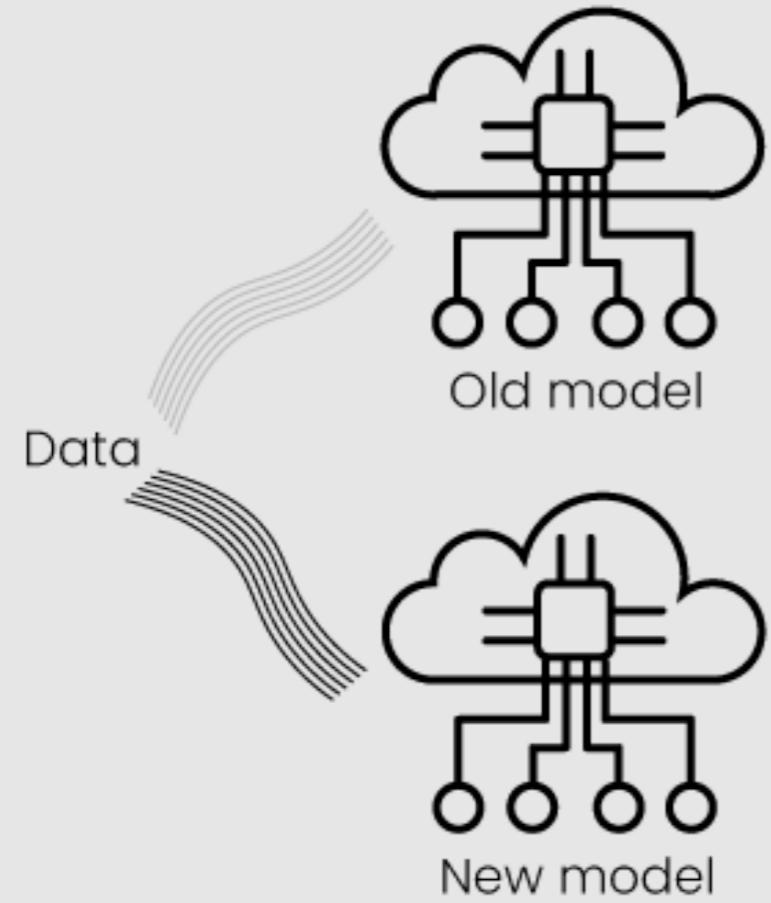


Shadow deployment

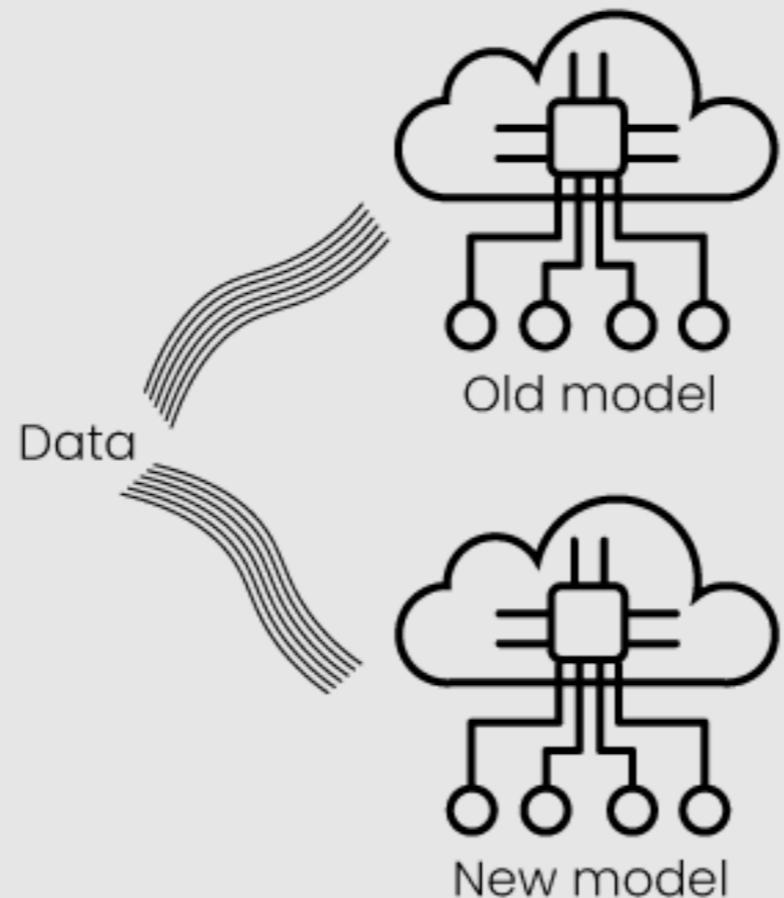


Deployment strategies

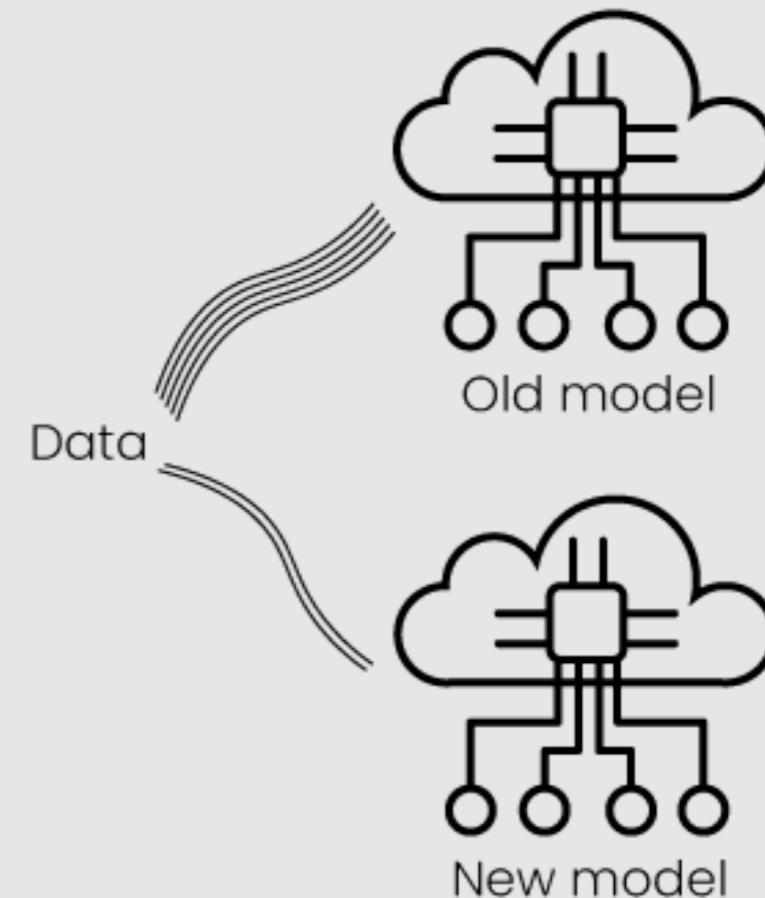
Basic deployment



Shadow deployment



Canary deployment



Deployment strategies

Strategy	Pros	Cons
Basic deployment	Straightforward, easy to implement, low resources	High risk if the model does not work as expected.
Shadow deployment	Easy to implement, no risk if model does not work as expected	Double resources.
Canary deployment	Slightly harder to implement, medium amount of resources	Small risk if model does not work as expected.

Let's practice!

MLOPS CONCEPTS

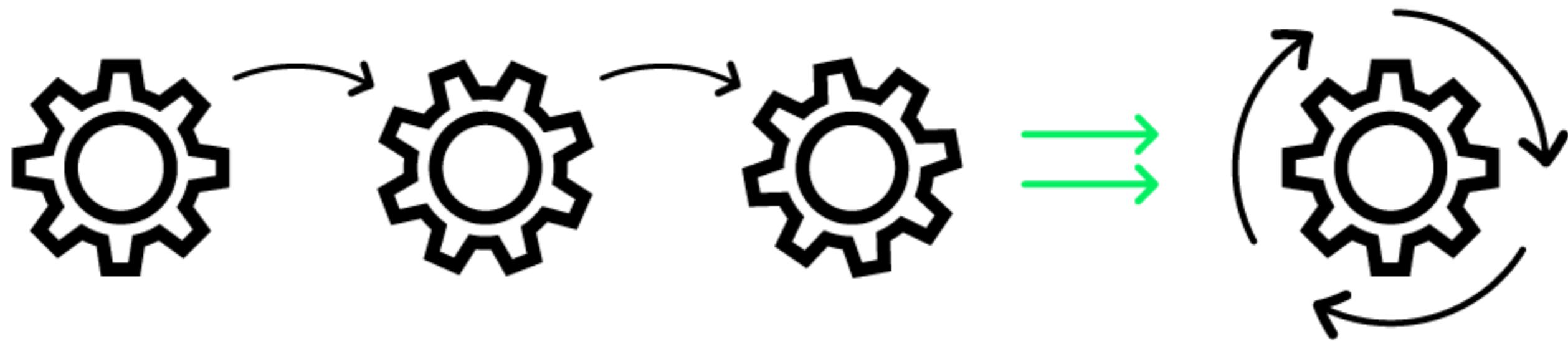
Automation and scaling

MLOPS CONCEPTS

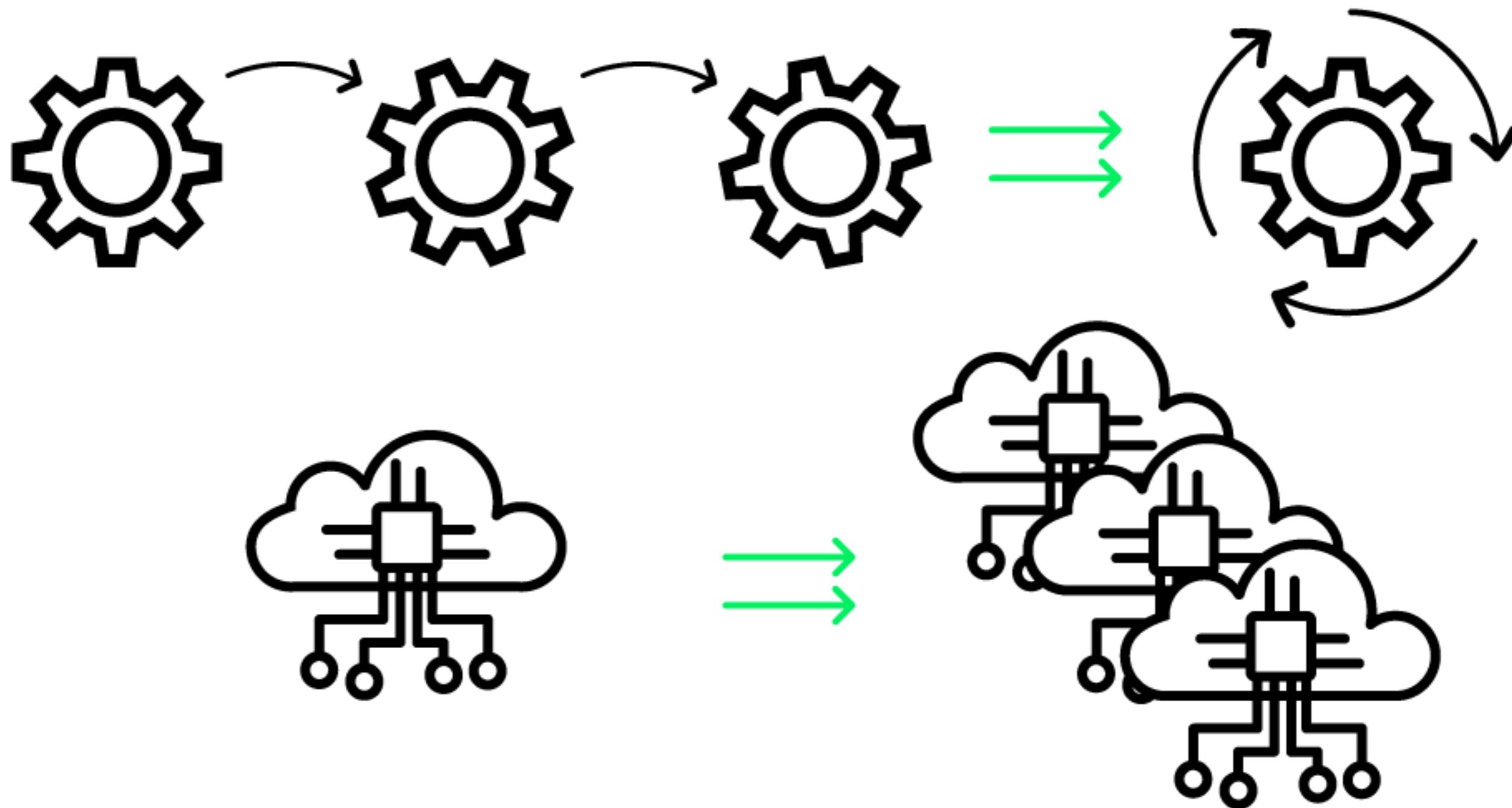


Folkert Stijnman
ML Engineer

Automation and scaling



Automation and scaling



Design phase

Project requirements



Project design

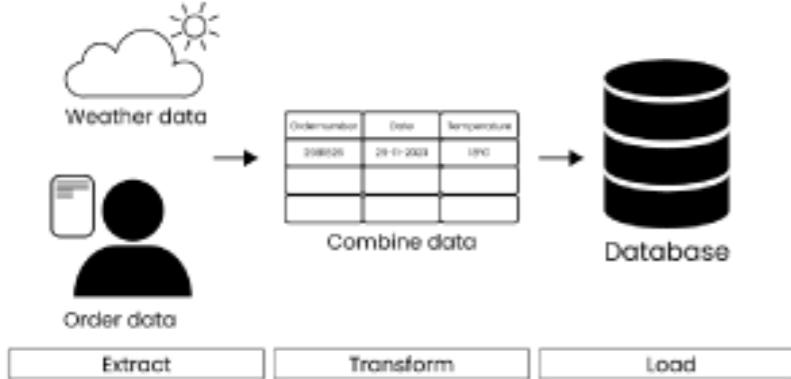
- Project design remains a manual process
- Use templates to automate and scale

Design phase

Project requirements



ETL pipeline



Data quality checks



Project design

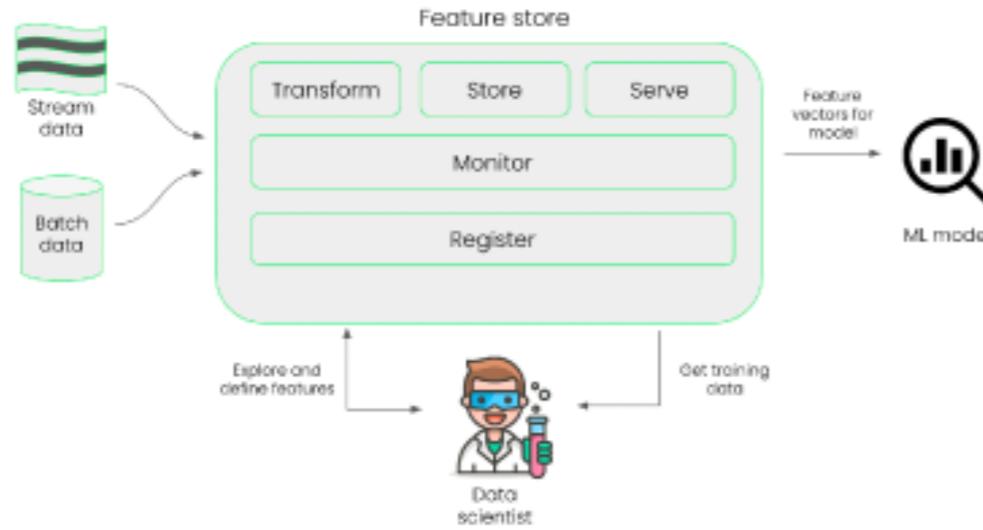
- Project design remains a manual process
- Use templates to automate and scale

Data acquisition

- Can be automated
- Enables high data quality

Development phase

Feature store

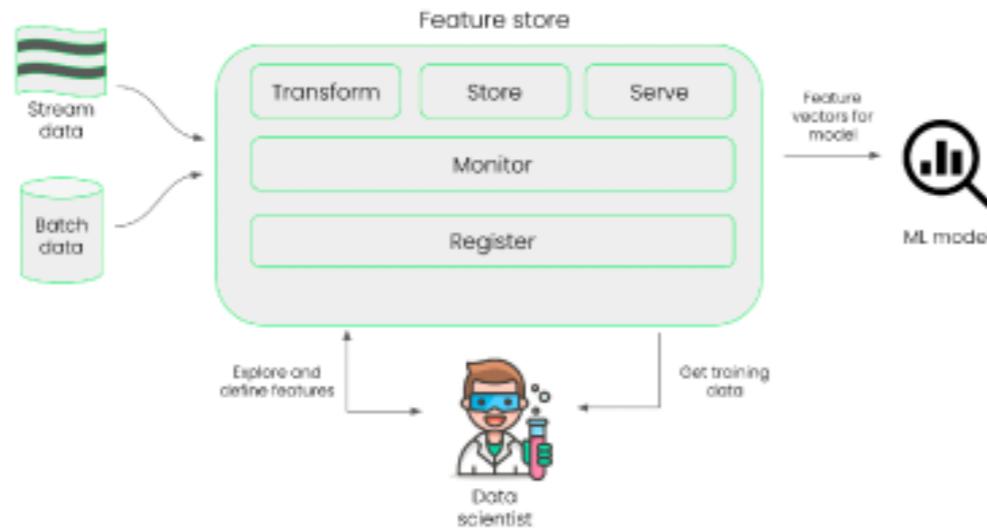


Feature store

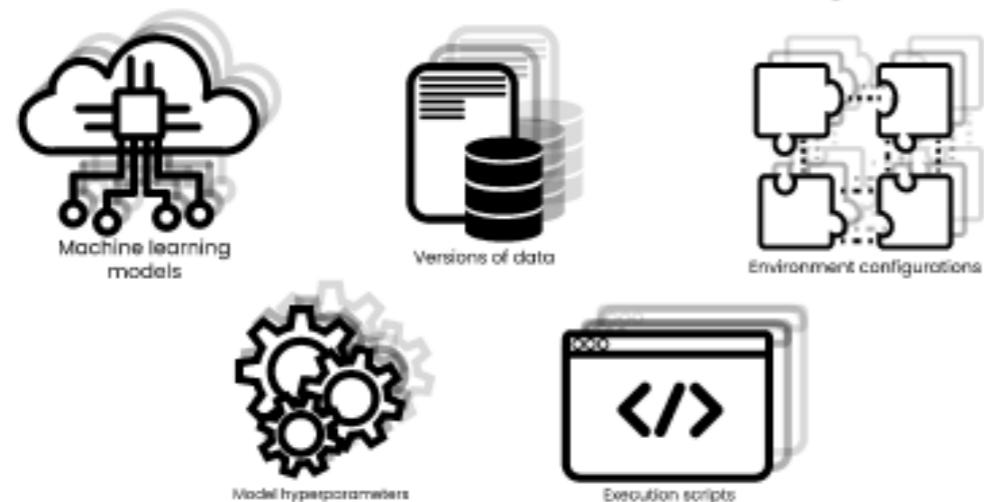
- Saves time building the same features
- Helps to scale

Development phase

Feature store



Experiment tracking



Feature store

- Saves time building the same features
- Helps to scale

Experiment tracking

- Automates tracking
- Ensures reproducibility

Deployment phase

Containerization



Containerization

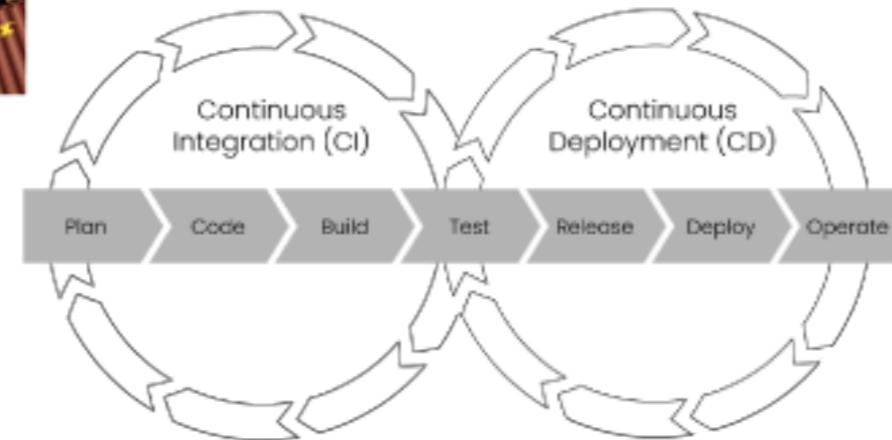
- Easy to start up copies of same application
- Improves scalability

Deployment phase

Containerization



CI/CD pipeline



Containerization

- Easy to start up copies of same application
- Improves scalability

CI/CD pipeline

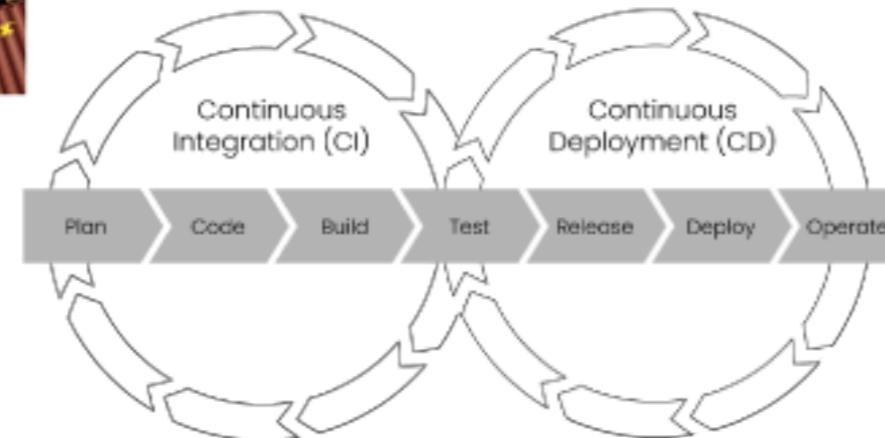
- Automates development and deployment
- Increases velocity of processes

Deployment phase

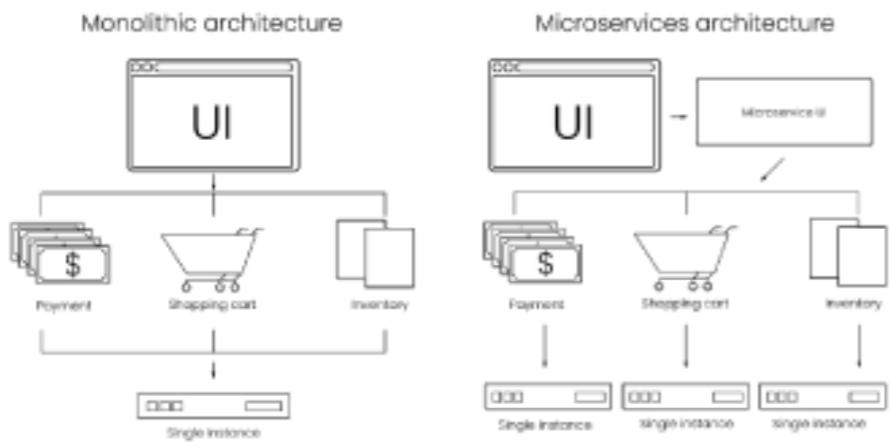
Containerization



CI/CD pipeline



Microservice architecture



Containerization

- Easy to start up copies of same application
- Improves scalability

CI/CD pipeline

- Automates development and deployment
- Increases velocity of processes

Microservices architecture

- Improves scalability
- Independent development and deployment

Let's practice!

MLOPS CONCEPTS