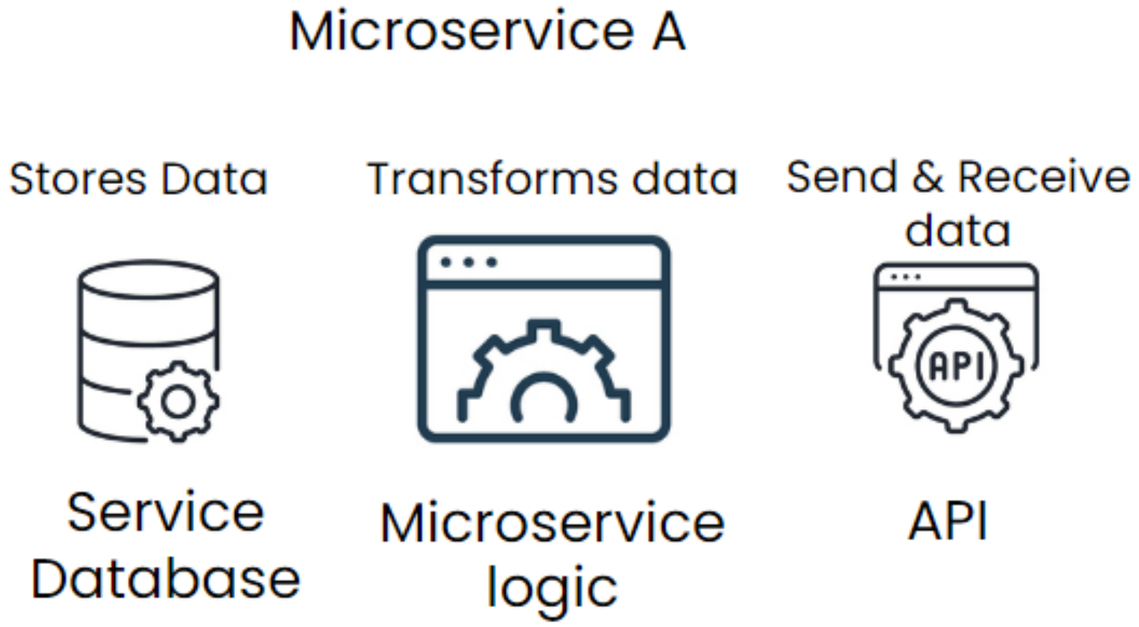
Data Engineering Implications of the DevOps Architecture

INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

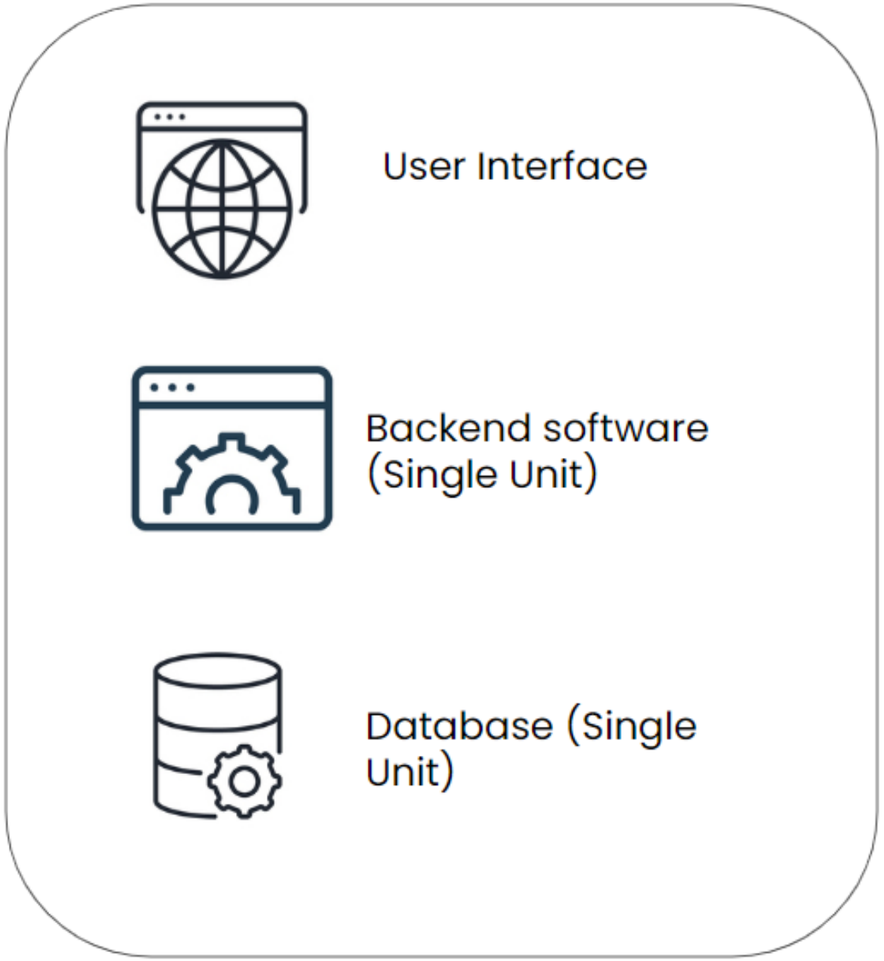
datacamp

Microservices architecture



- Microservices are small-scale software programs
- Microservices deployed separately
- Each microservice take care of a different functionality
- Each microservice has its own data and logic
- They store the data in private databases

Monolithic architecture



- Opposite of the microservices architecture
- A big single unit
- Much simpler compared to microservices
- Maintenance and change is very hard and risky

Monolithic Architecture

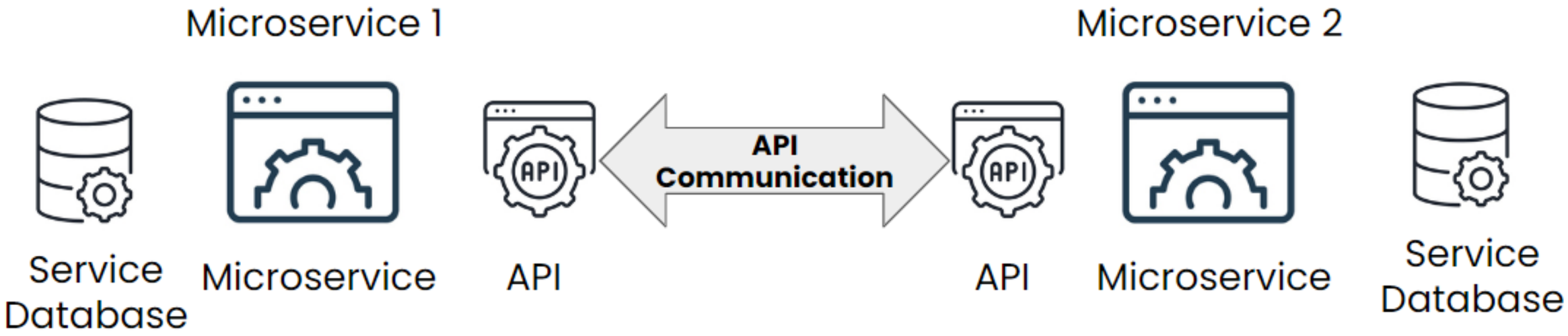
- Limited number of databases
- All application uses the same databases
- Could be a viable option for small scale applications

Microservices Architecture

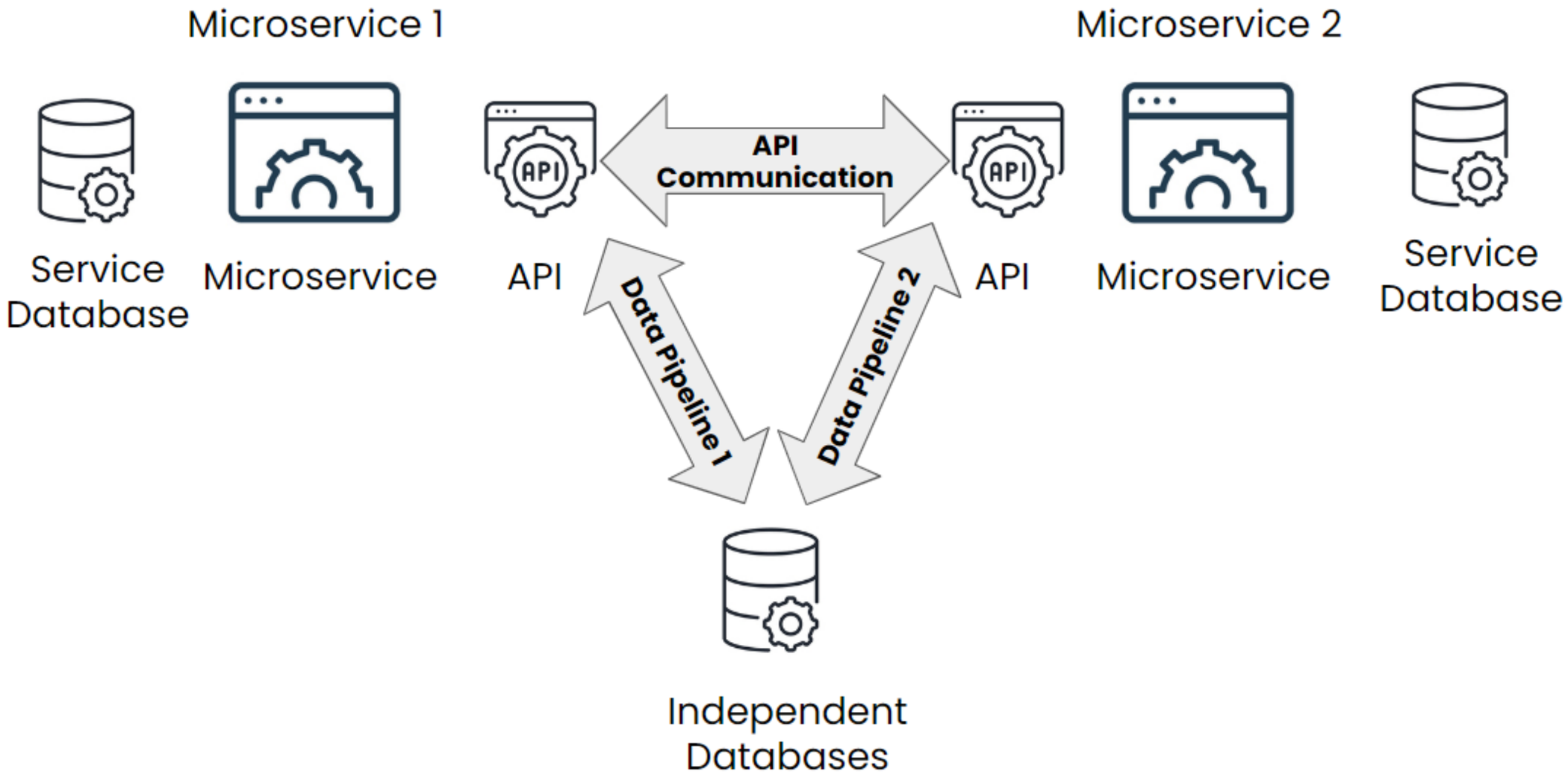
- A database for each microservice
- Microservices must do API calls to reach another services database
- Microservices are effective in large organizations and complex products

Microservices private databases

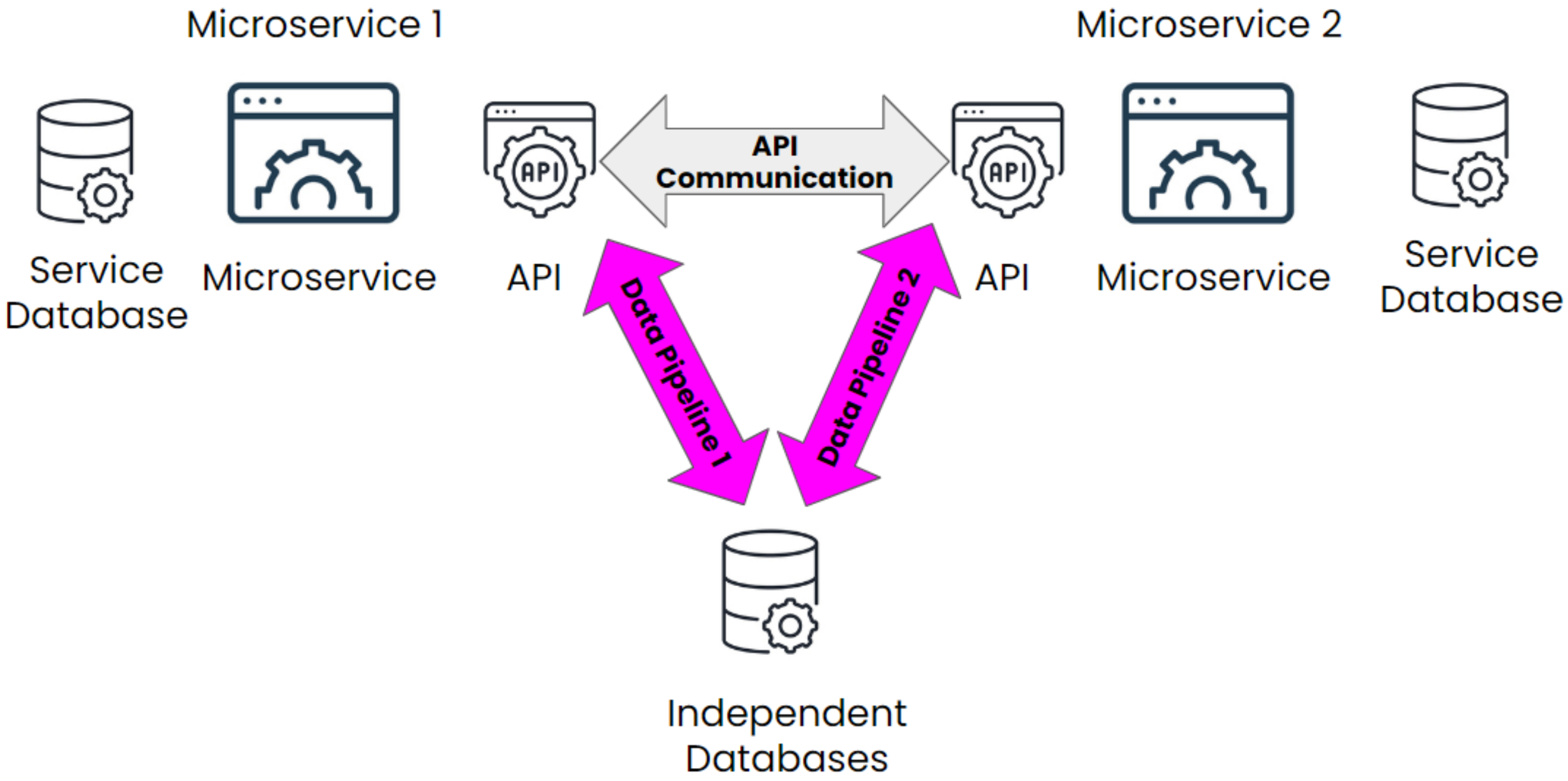
- Some product functionalities require collaboration from multiple microservices
- Microservices can not access each others databases freely, so they do API calls
- Microservices always communicate with each other.



Data Engineering applications in microservices



Data Engineering applications in microservices



Let's practice!

INTRODUCTION TO DEVOPS

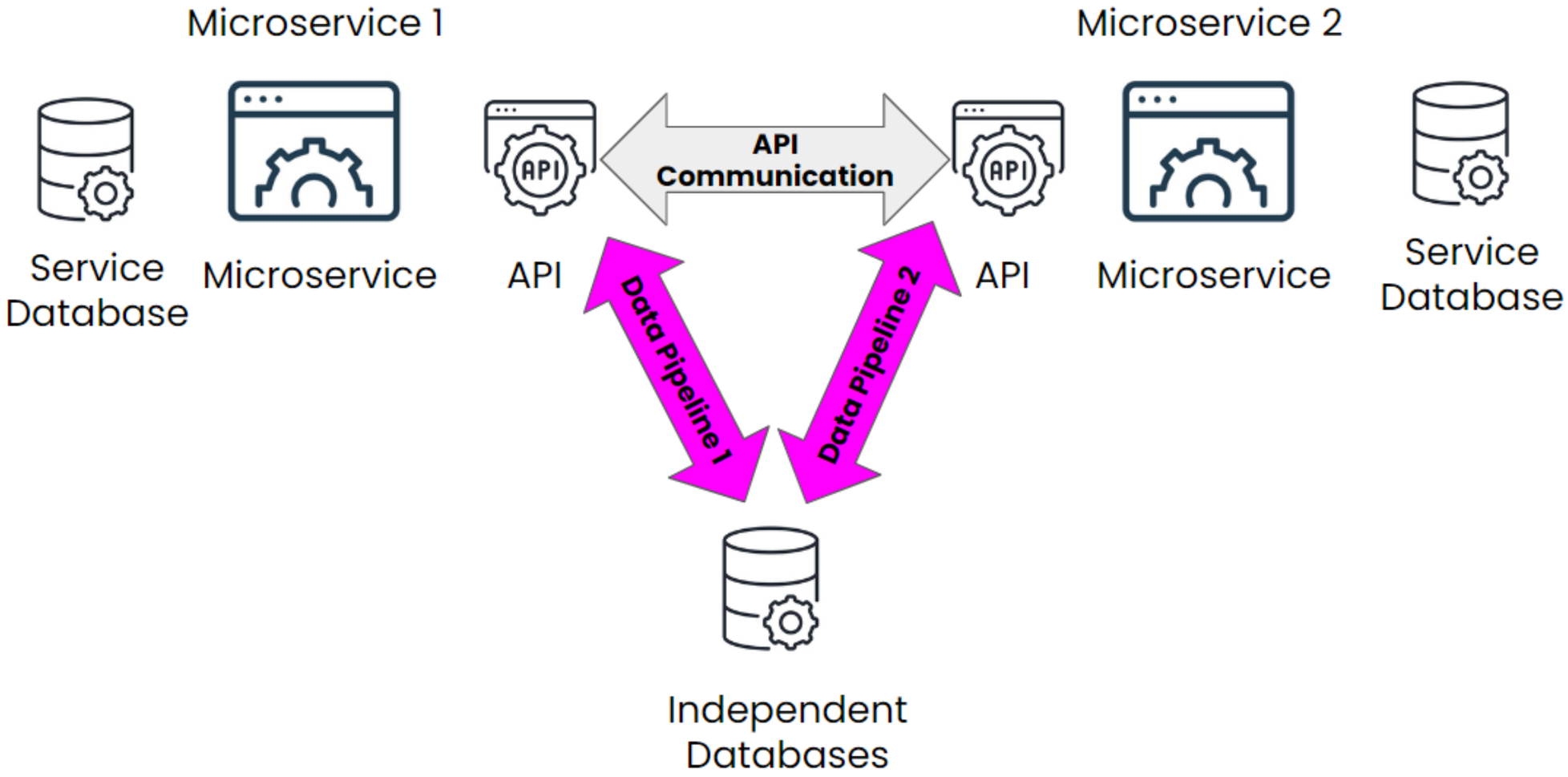
Data Pipelines

INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

datacamp

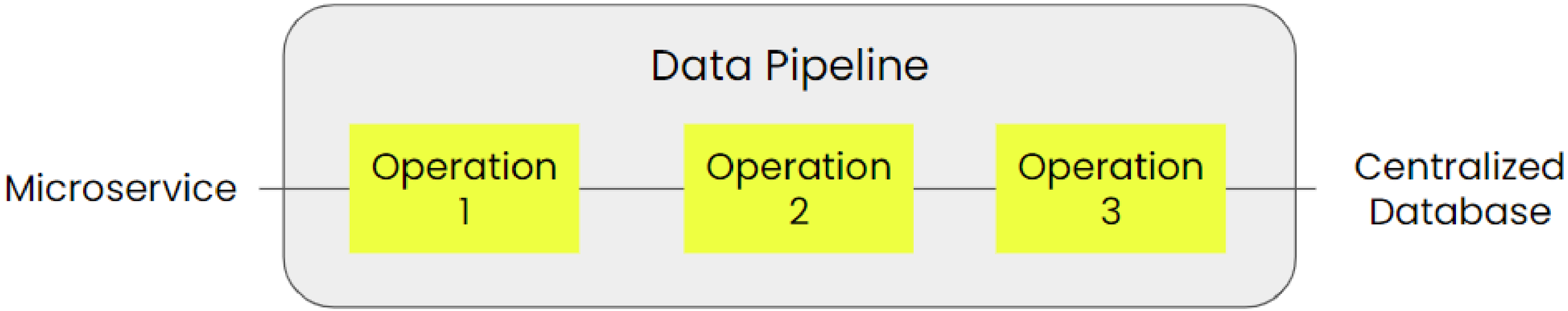
Data pipelines



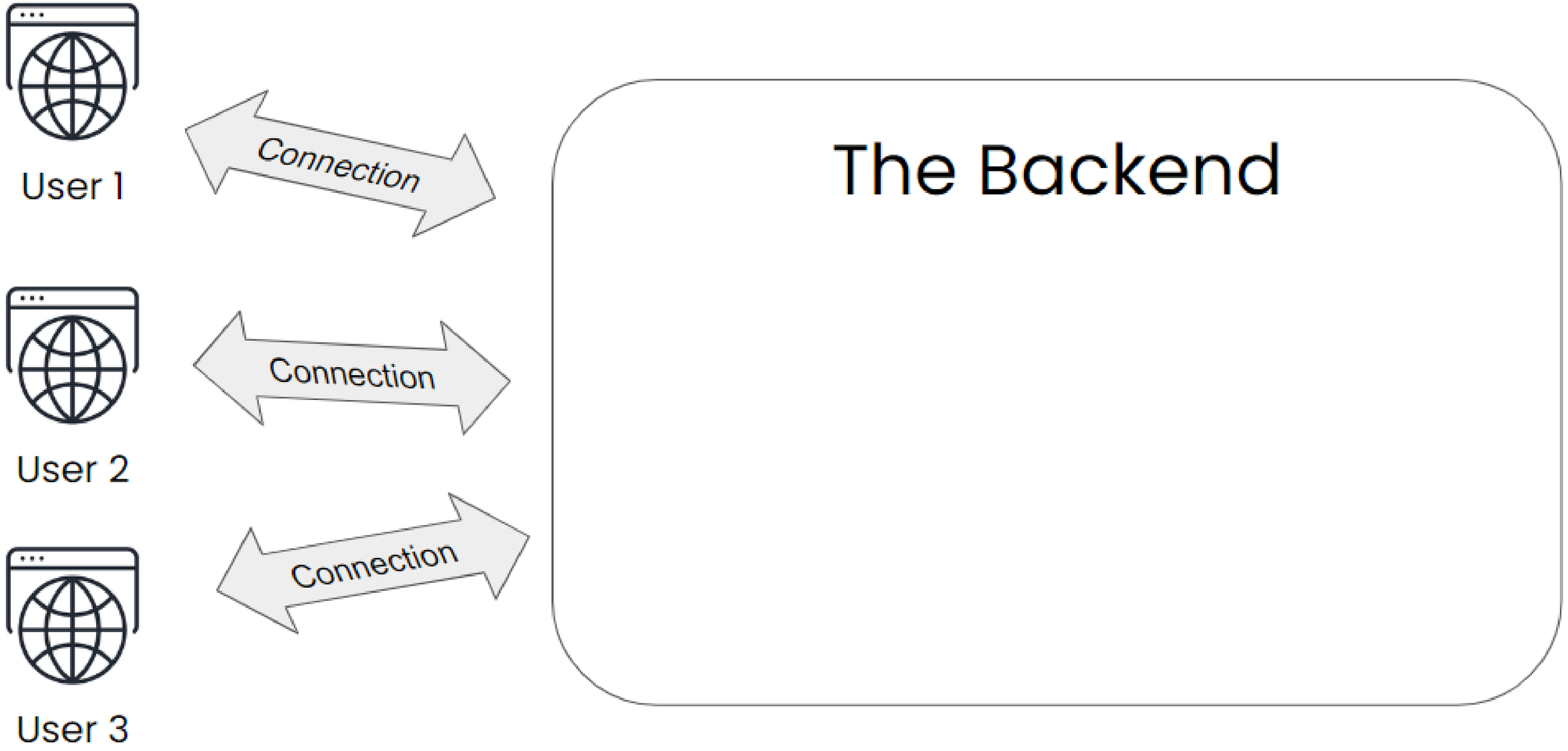
ETL

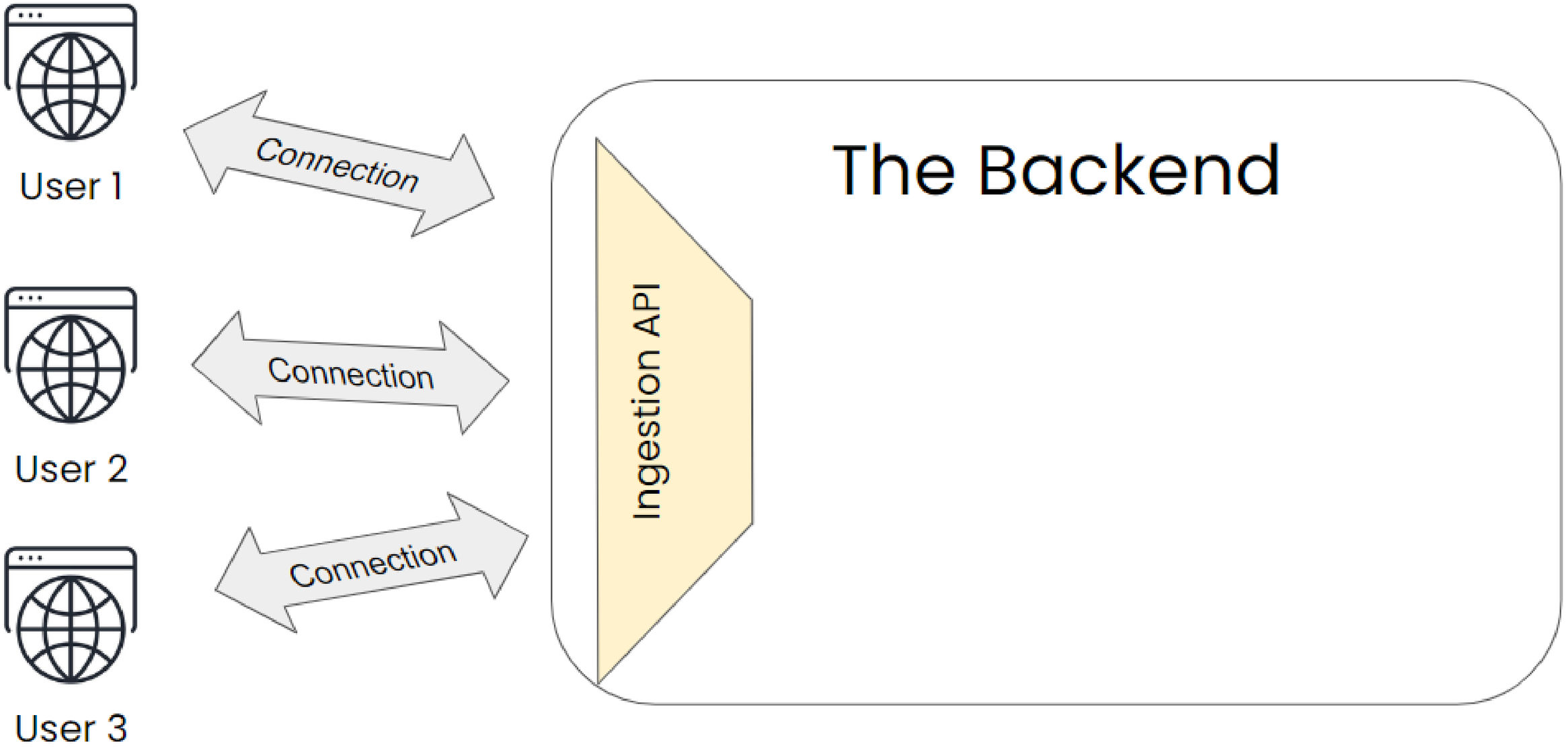
[1] https://www.istockphoto.com/nl/vector/data-extract-transform-load-gm1361894912-434102842?phrase=ETL

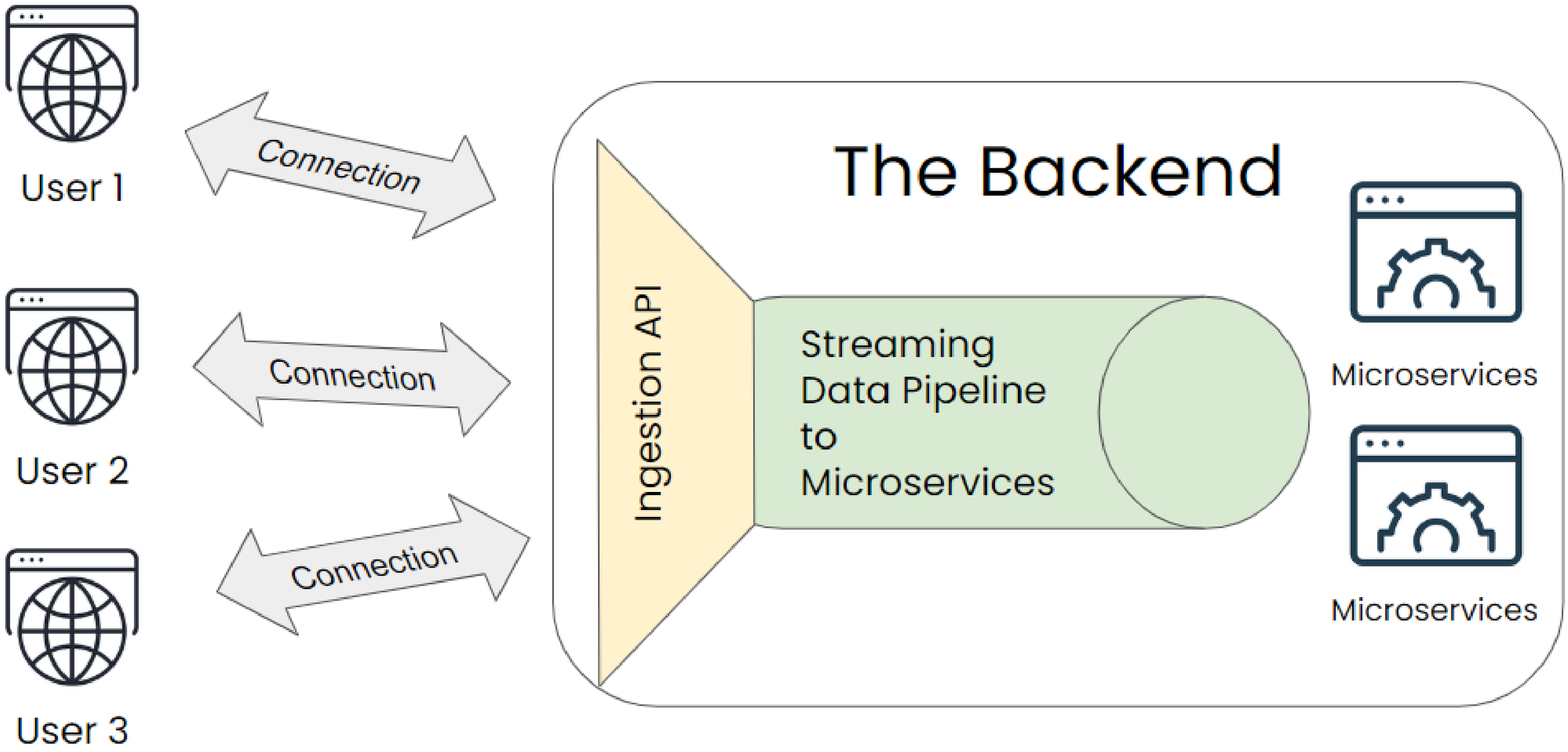
Batch processing



- Move offline data in batches
- Batch Processing: A large amount of data


User 1
Connection
User 2
Connection
User 3
Connection
The Backend


User 1
Connection
User 2
Connection
User 3
Connection
The Backend
Ingestion API


User 1
User 2
User 3
Connection
Connection
Connection
The Backend
Ingestion API
Streaming
Data Pipeline
to
Microservices
Microservices
Microservices

Recap

- Data Engineering is an integral part of Infrastructure Engineering
- Data pipelines are robust tools important for handling data
- Batch processing works on regular schedules, stream processing works continuously
- Batch processing is useful for handling the accumulated data
- Stream processing handles the real-time

Let's practice!

INTRODUCTION TO DEVOPS

Analytics & Reporting with DevOps

INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

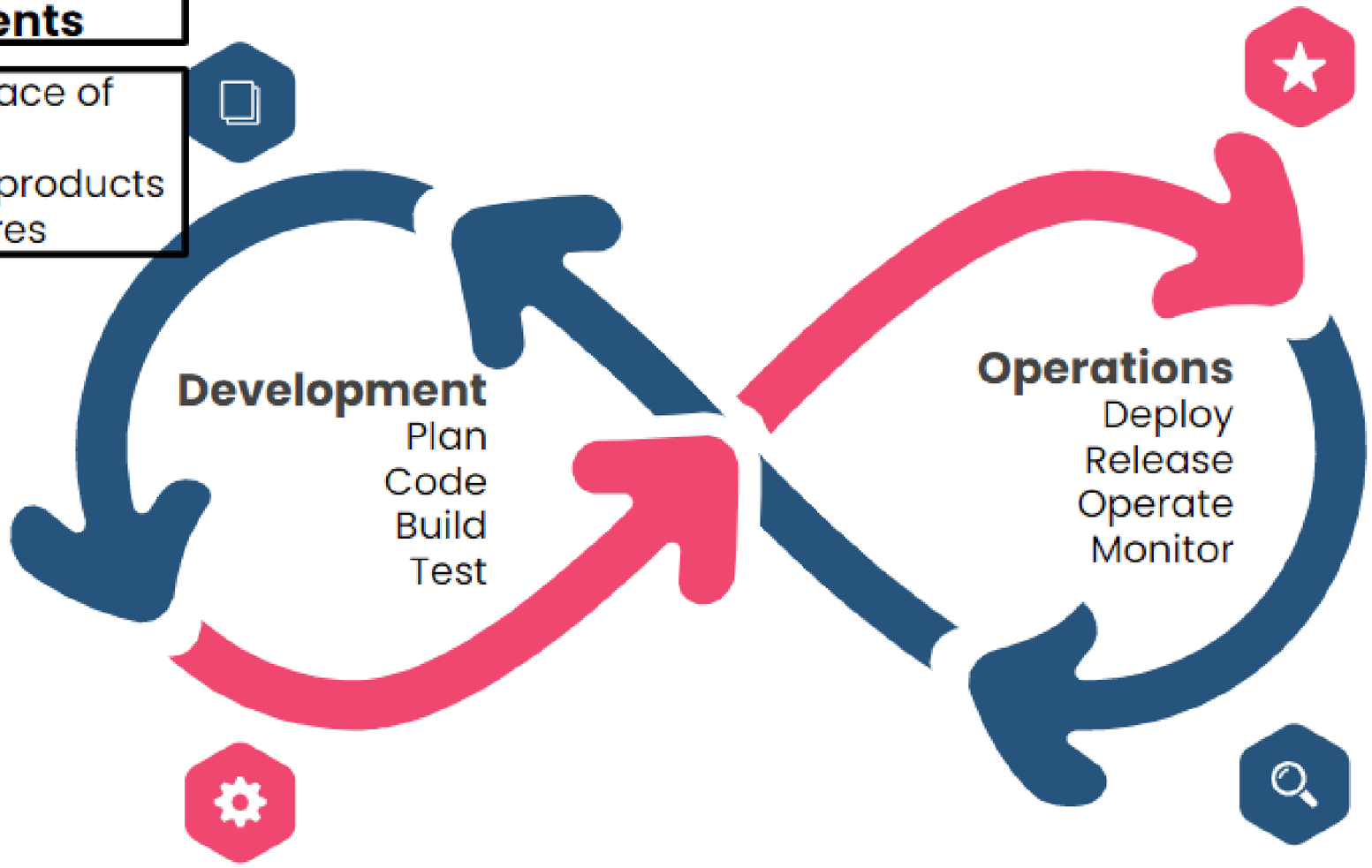
datacamp

DevOps benefits for reporting

- Handling Multiple Data Sources in harmony
- Producing many logs during the change management process

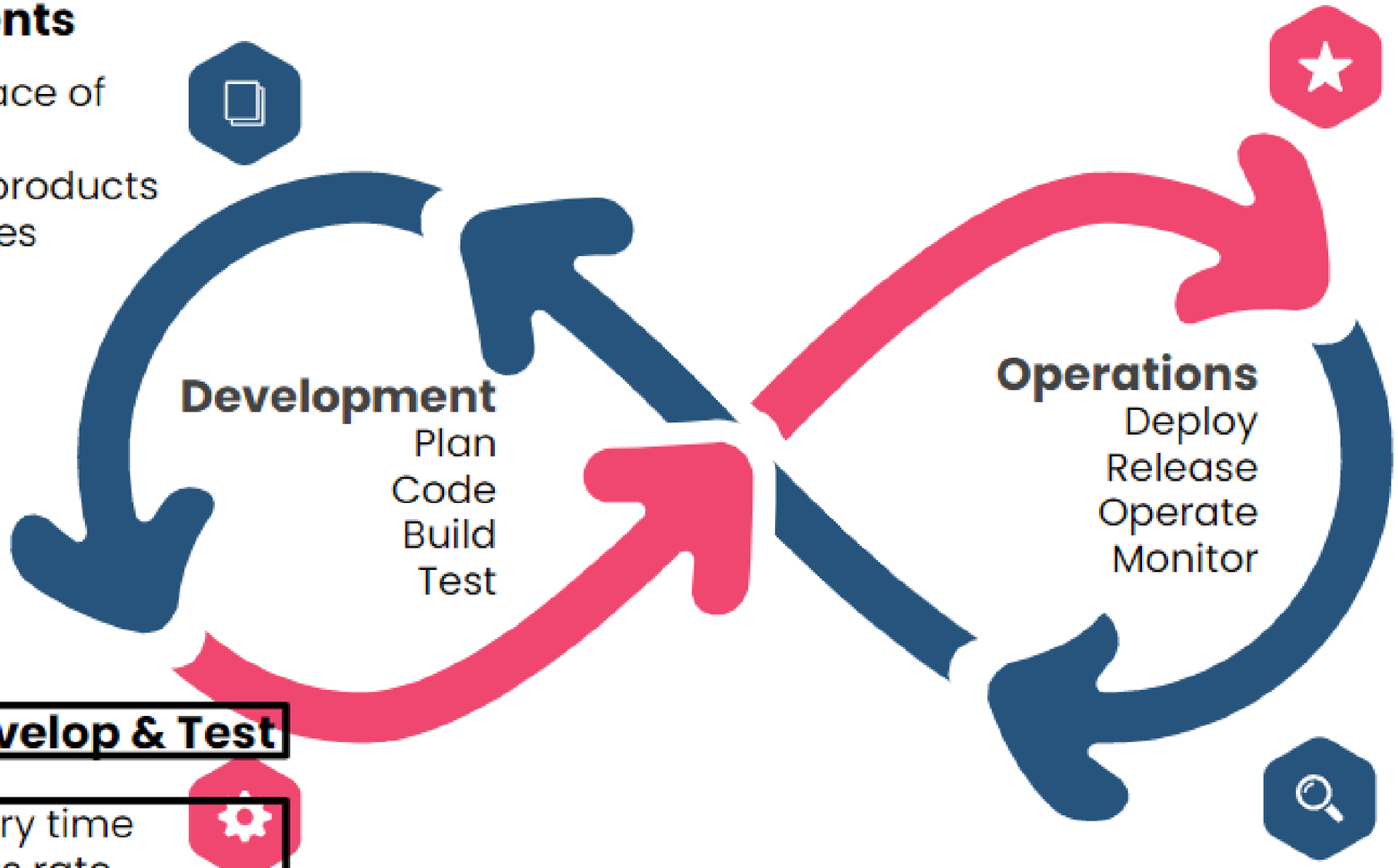
Requirements

-Track the pace of innovation
-Number of products & new features



Requirements

-Track the pace of innovation
-Number of products & new features

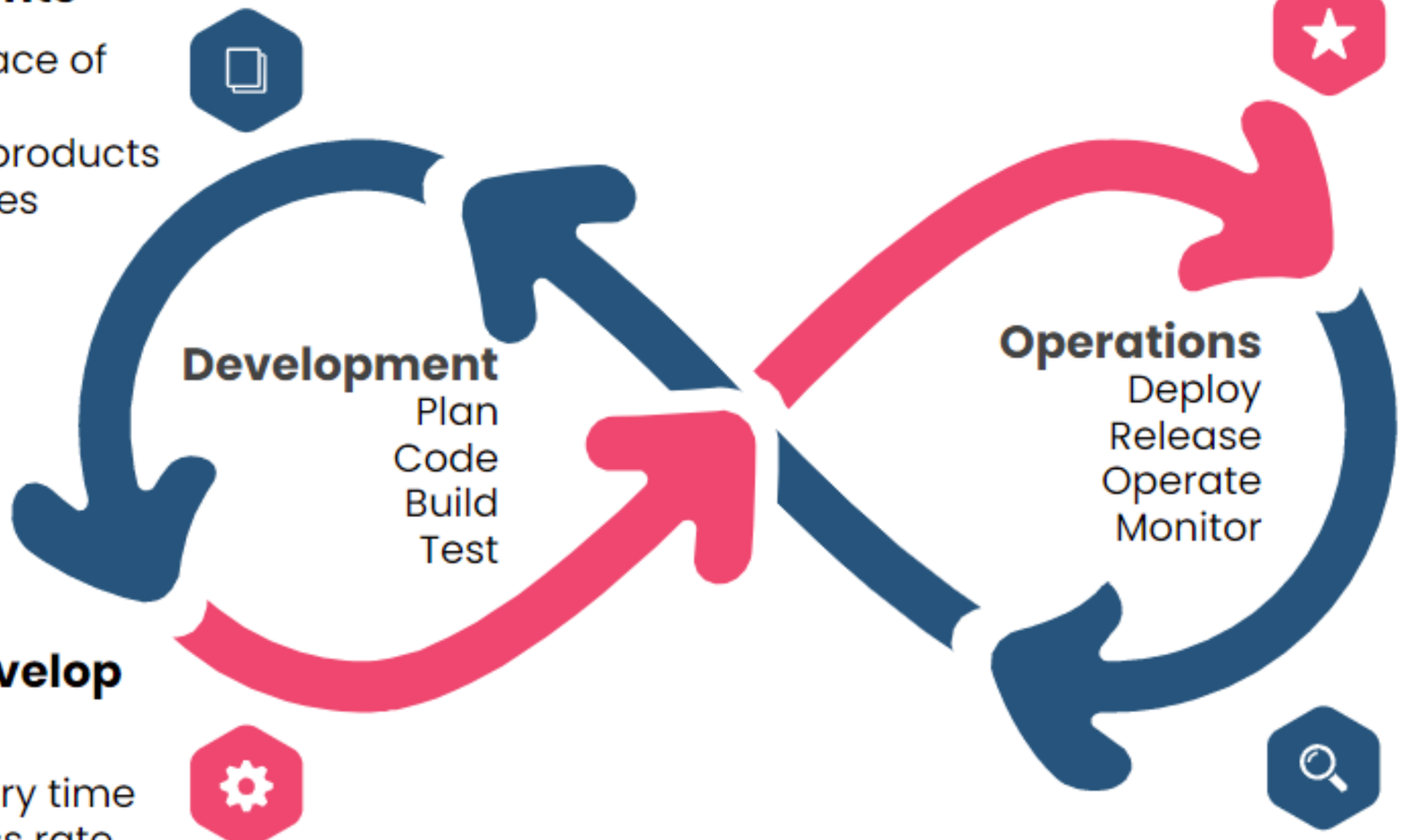


Design, Develop & Test

-Track delivery time
-Pipeline pass rate
-Track development time

Requirements

-Track the pace of innovation
-Number of products & new features



Deploy, Review

-Measure the success of experiments
-Quality of the Products

Design, Develop & Test

-Track delivery time
-Pipeline pass rate
-Track development time

Requirements

-Track the pace of innovation
-Number of products & new features

Development
Plan
Code
Build
Test

Operations
Deploy
Release
Operate
Monitor

Deploy, Review

-Measure the success of experiments
-Quality of the Products

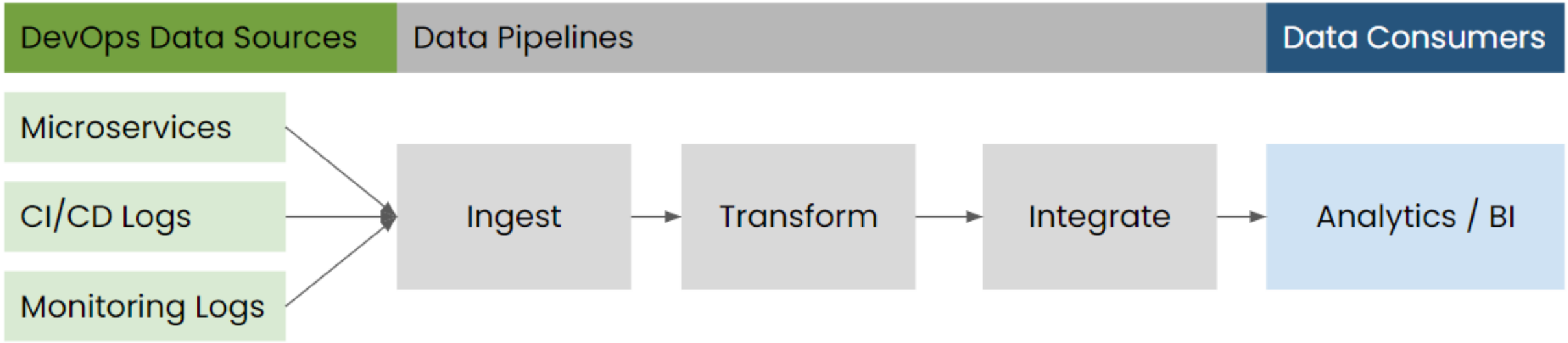
Design, Develop & Test

-Track delivery time
-Pipeline pass rate
-Track development time

Launch, Operate & Monitor

-Measure Post-Release Issues

Reporting architecture For DevOps



- DevOps produces a lot of data
- Data sources are spread out
- Data needs to be collected and moved to a database
- Data Scientists can use the data for insights, reports, and dashboards

Let's practice!

INTRODUCTION TO DEVOPS

Tools: Ecosystem

INTRODUCTION TO DEVOPS

Cem Sakarya
DevOps Risk Advisor

Requirements

- Business and engineering interaction
- Used throughout the software change management process
- Project Management Tools
- Communication Tools

[1] https://atlassian.design/foundations/logos [2] https://slack.com/media-kit [3] https://discord.com/branding

Version Control Software

- Main change management technology is git
- Git is a protocol used for version control
- Git ensures multiple developers can work on the same software
- Git has many implementations, most common ones are GitHub and GitLab

[1] https://git-scm.com/downloads/logos [2] https://github.com/logos [3] https://about.gitlab.com/press/press-kit/

Build Tools

- Software needs to be built to become executable
- Main build tools are: Maven and Gradle

[1] https://maven.apache.org/ [2] https://gradle.com/brand/

CI/CD Tools

- CI/CD pipelines are the main principles of DevOps
- CI/CD pipelines ensure automated building, testing, and deployment of software
- Main CI/CD tools are Jenkins and CircleCI

circleci

[1] https://www.jenkins.io/press/ [2] https://circleci.com/legal/trademark-guidelines/

Deployment

- Microservices are developed and deployed independently from each other
- Containers imitate separate machines
- Microservices are deployed on separate containers
- Containers: Docker and Podman
- Container orchestration: Kubernetes

[1] https://www.docker.com/company/newsroom/media-resources/ [2] https://podman.io/ [3] https://kubernetes.io/

Monitoring Tools

- Products need to be closely monitored and observed for quality and reliability issues
- Monitor the DevOps health and change management metrics
- Example tools used for monitoring are SignalFX and AppDynamics

[1] https://www.splunk.com/en_us/products/observability.html?301=/en_us/devops.html

Data management tools: Kafka

- Kafka is a message publishing system
- Kafka is heavily used in microservices architecture
- Microservices keep a journal of the work they do on Kafka

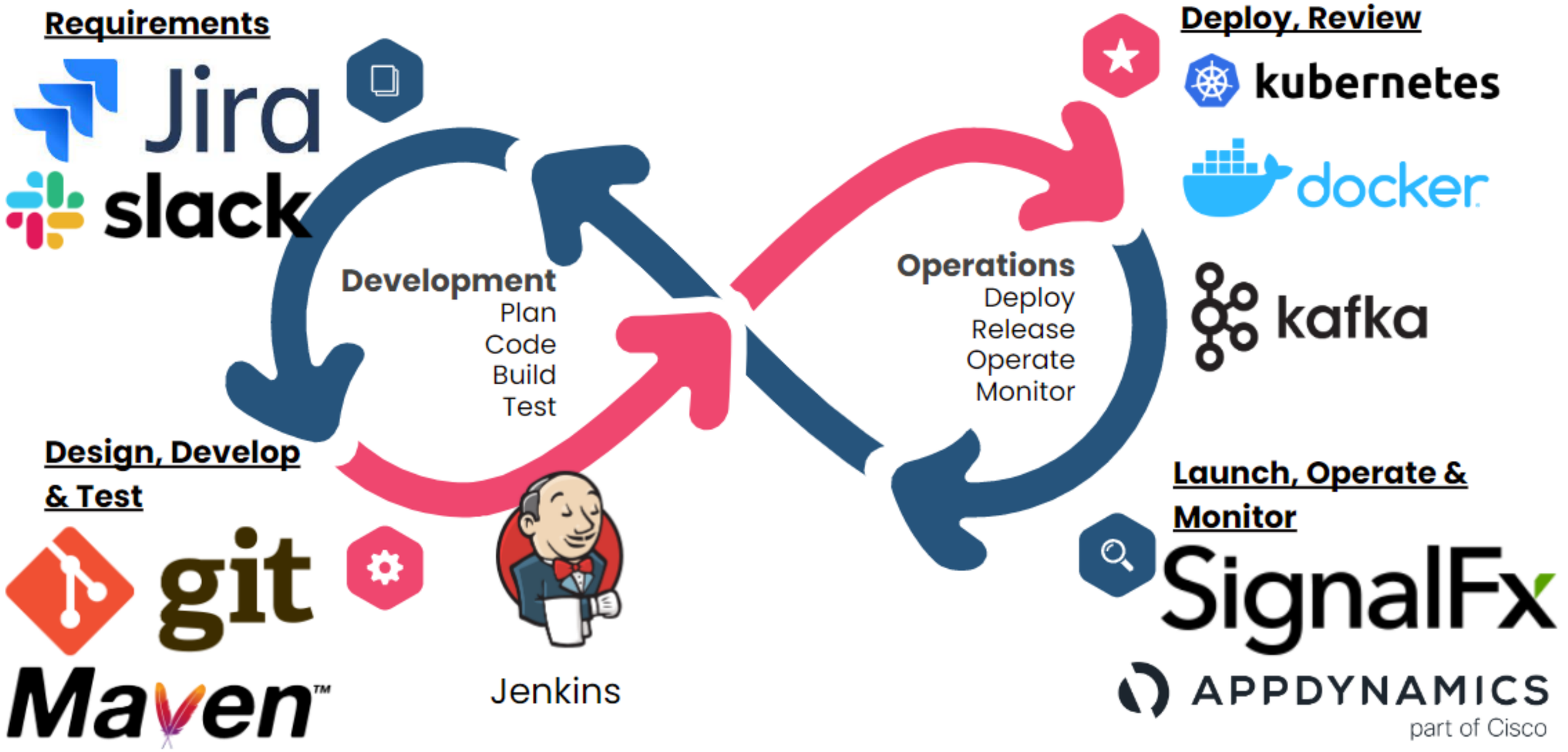
[1] https://kafka.apache.org/trademark

Data pipeline management tools

- Most tools used for both batch and streaming processing
- Main tools used for data pipeline management are: Apache Airflow, Hevo Data, and Prefect

[1] https://cwiki.apache.org/confluence/display/AIRFLOW/Airflow%20logos [2] https://design.hevodata.com/ [3] https://www.prefect.io/newsroom/logos/

Recap


Requirements
Jira
slack
Development
Plan
Code
Build
Test
Operations
Deploy
Release
Operate
Monitor
Deploy, Review
kubernetes
docker
kafka
Design, Develop & Test
git
Maven™
Jenkins
Launch, Operate & Monitor
SignalFx
APPDYNAMICS
part of Cisco

Let's practice!

INTRODUCTION TO DEVOPS