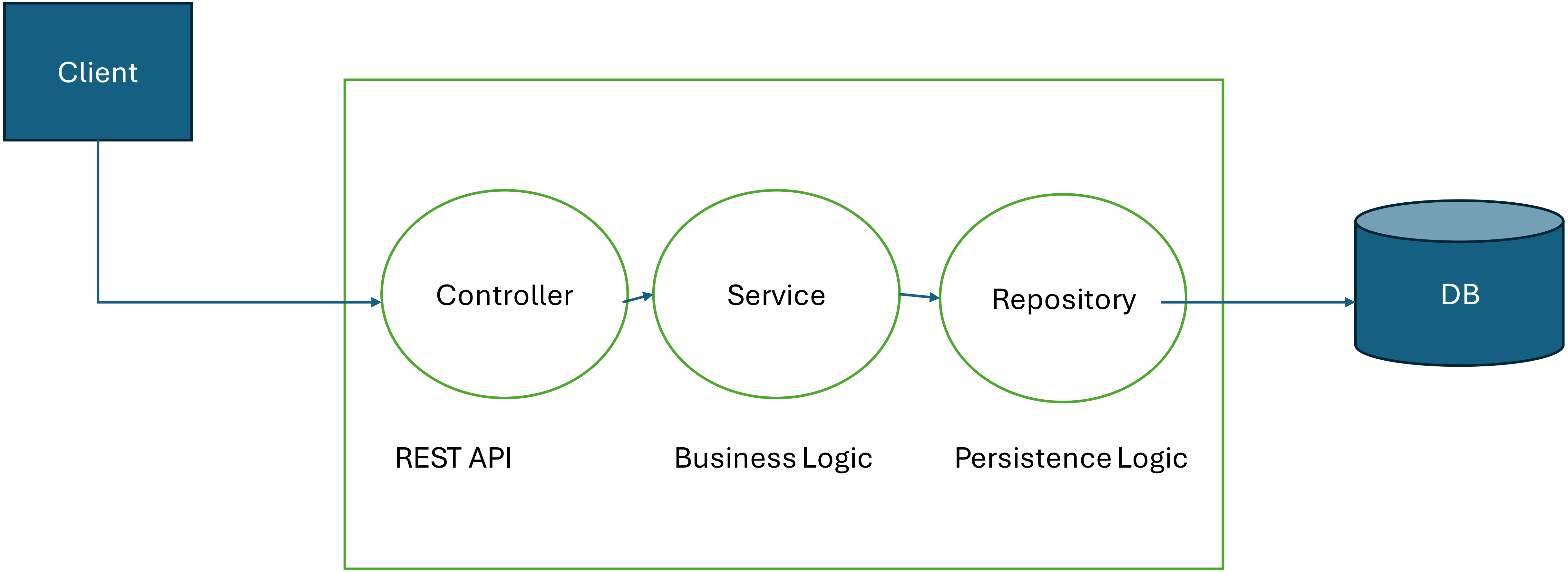
Roughwork



Client

Controller

Service

Repository

DB

REST API

Business Logic

Persistence Logic

TBL_CUST

CUST_ID	NAME	EMAIL	PASSWORD	PROFILE	ADDR_ID (FK)
10	MAJRUL	...	...	..	

TBL_ADDR

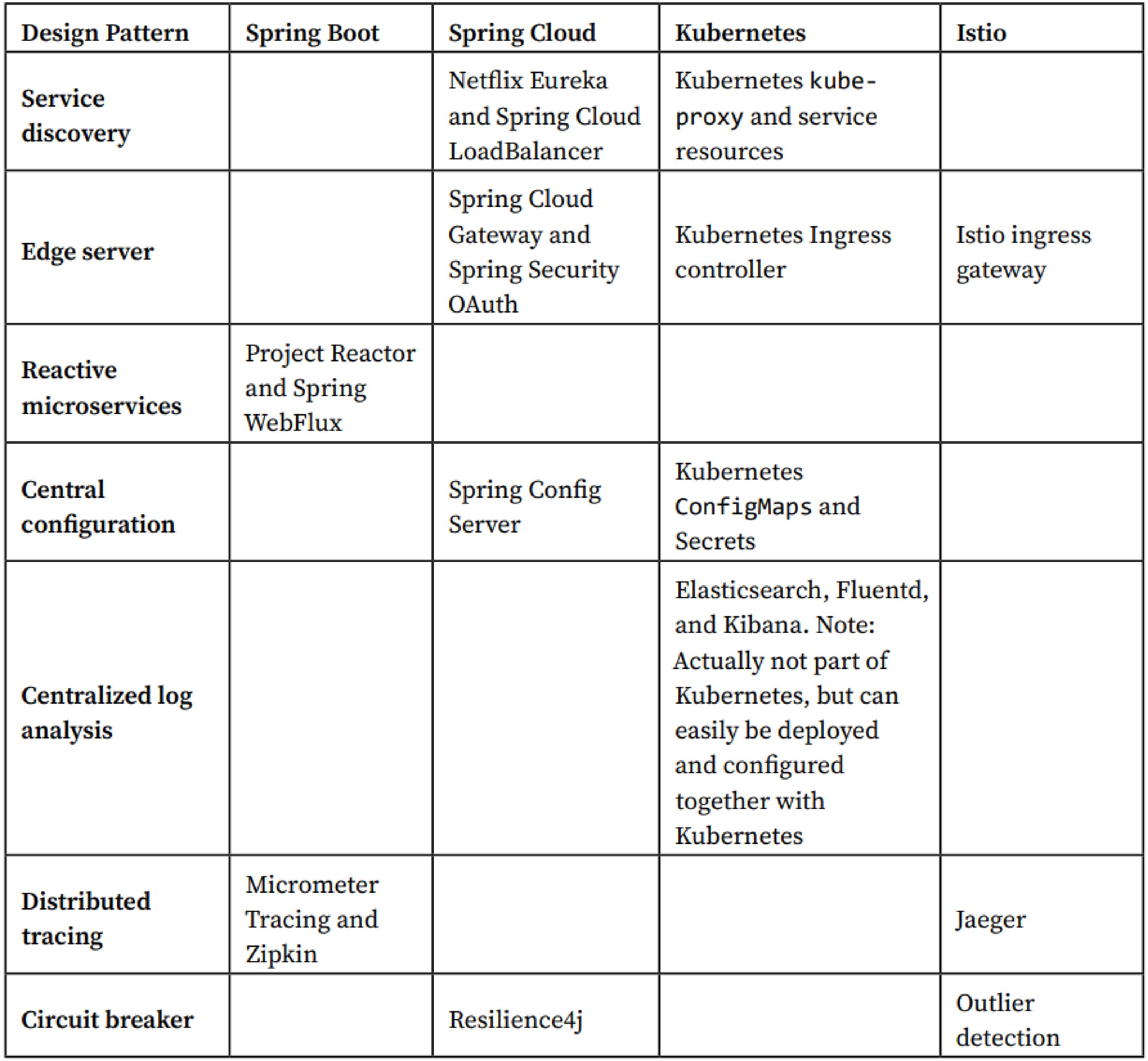
ADDR_ID	CITY	PINCODE			
100	MUMBAI	...			

Microservices

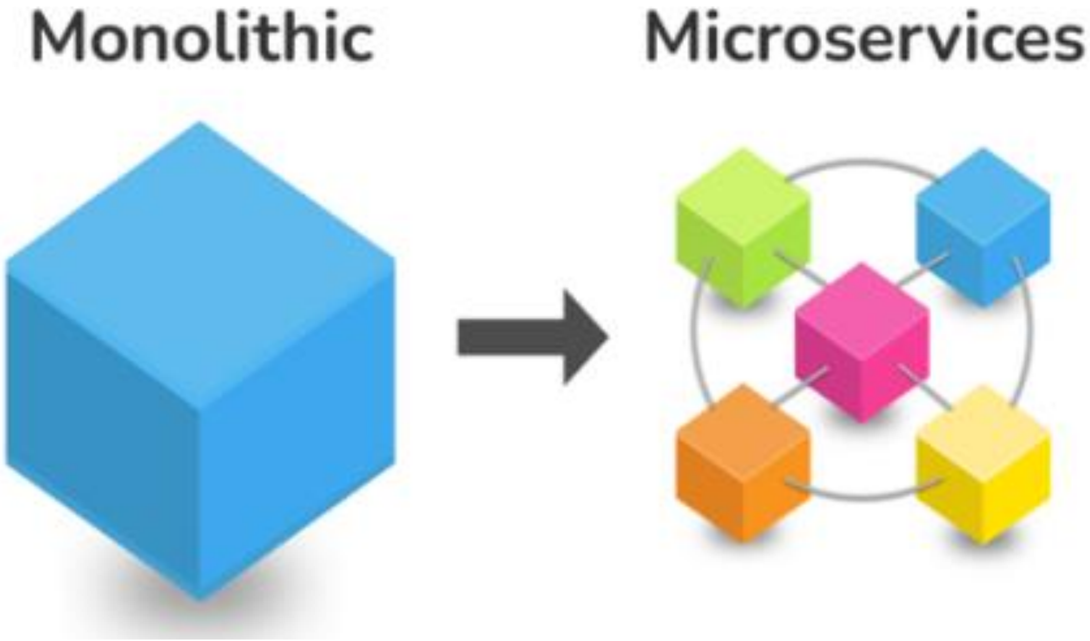
- Pre-requisite: REST API Development

- Modern software development technique in which an application is broken down into parts and each part is developed independently

- Each microservice will have it’s own DB and it will provide data to others through well defined APIs

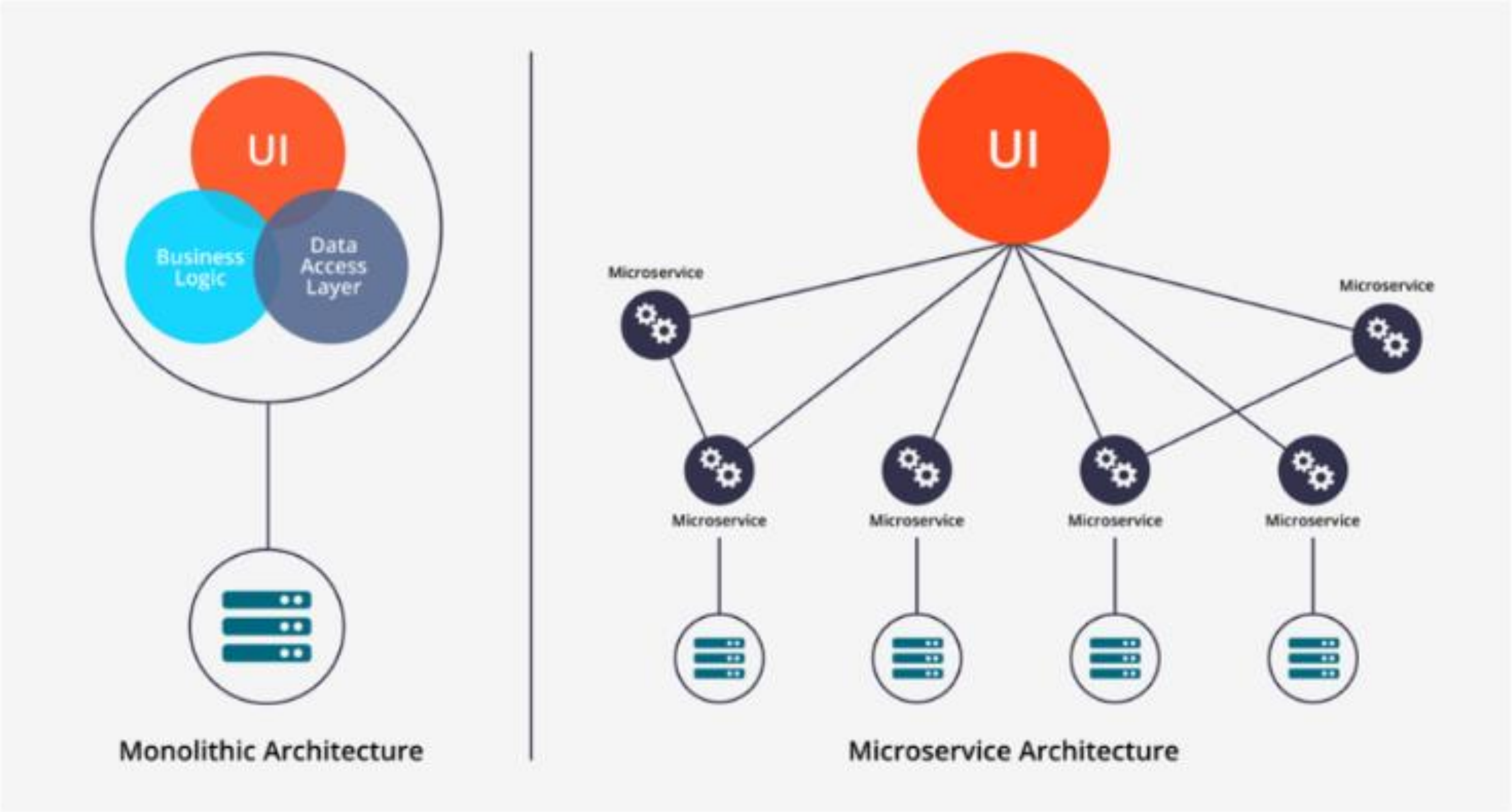


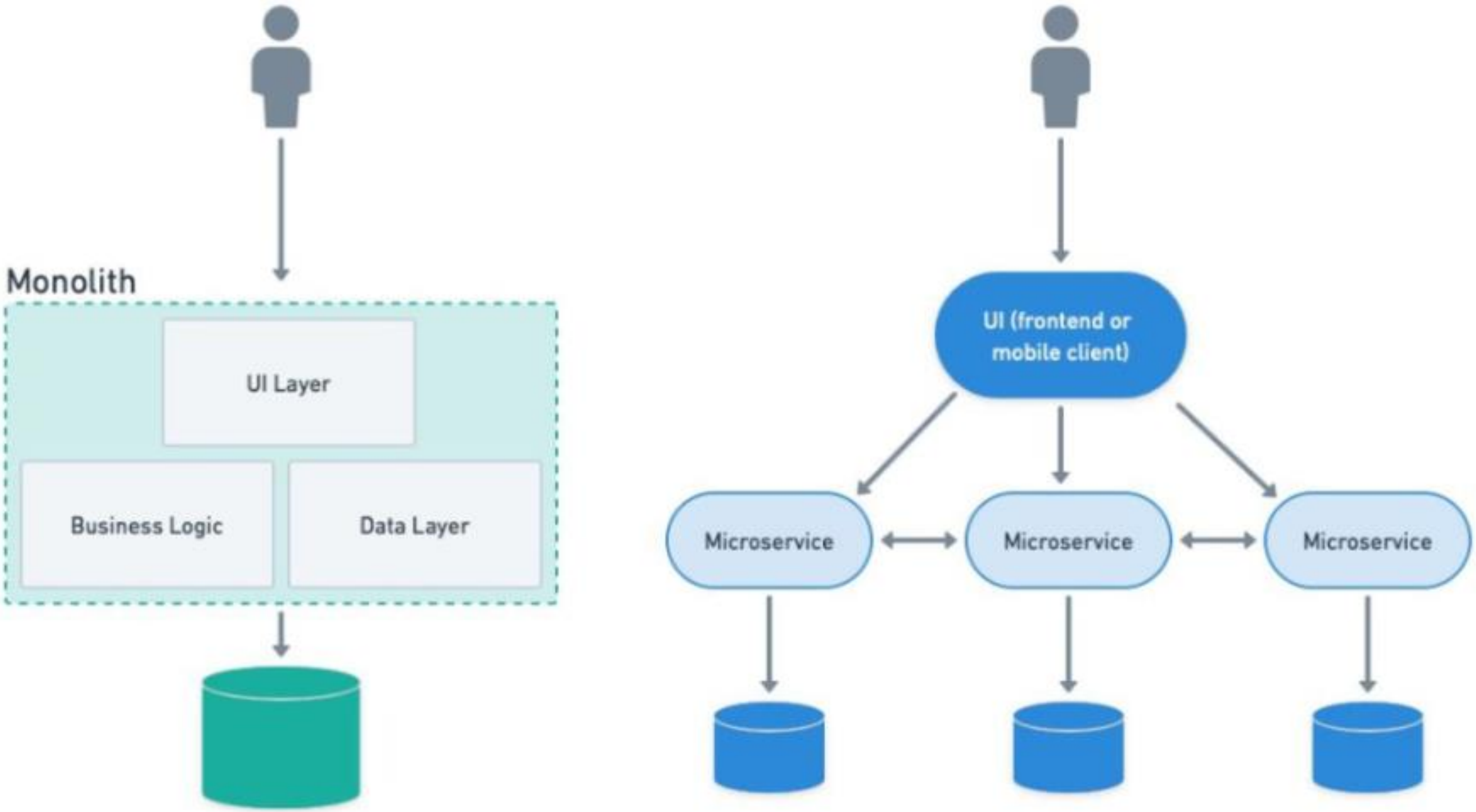
Design Pattern	Spring Boot	Spring Cloud	Kubernetes	Istio
Service discovery		Netflix Eureka and Spring Cloud LoadBalancer	Kubernetes `kube-proxy` and service resources	
Edge server		Spring Cloud Gateway and Spring Security OAuth	Kubernetes Ingress controller	Istio ingress gateway
Reactive microservices	Project Reactor and Spring WebFlux			
Central configuration		Spring Config Server	Kubernetes `ConfigMaps` and Secrets	
Centralized log analysis			Elasticsearch, Fluentd, and Kibana. Note: Actually not part of Kubernetes, but can easily be deployed and configured together with Kubernetes	
Distributed tracing	Micrometer Tracing and Zipkin			Jaeger
Circuit breaker		Resilience4j		Outlier detection



Monolithic

Microservices


UI
Business Logic
Data Access Layer
Monolithic Architecture
UI
Microservice
Microservice
Microservice
Microservice
Microservice
Microservice
Microservice Architecture



Monolith

UI Layer

Business Logic

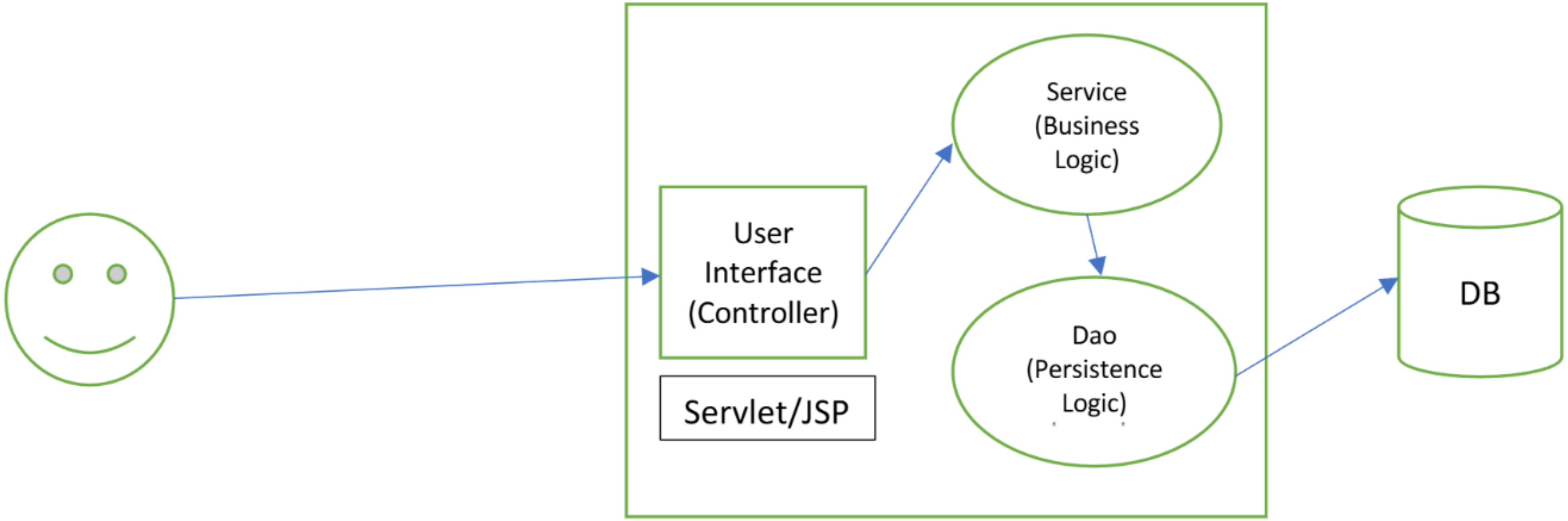
Data Layer

UI (frontend or mobile client)

Microservice

Microservice

Microservice



User Interface (Controller)

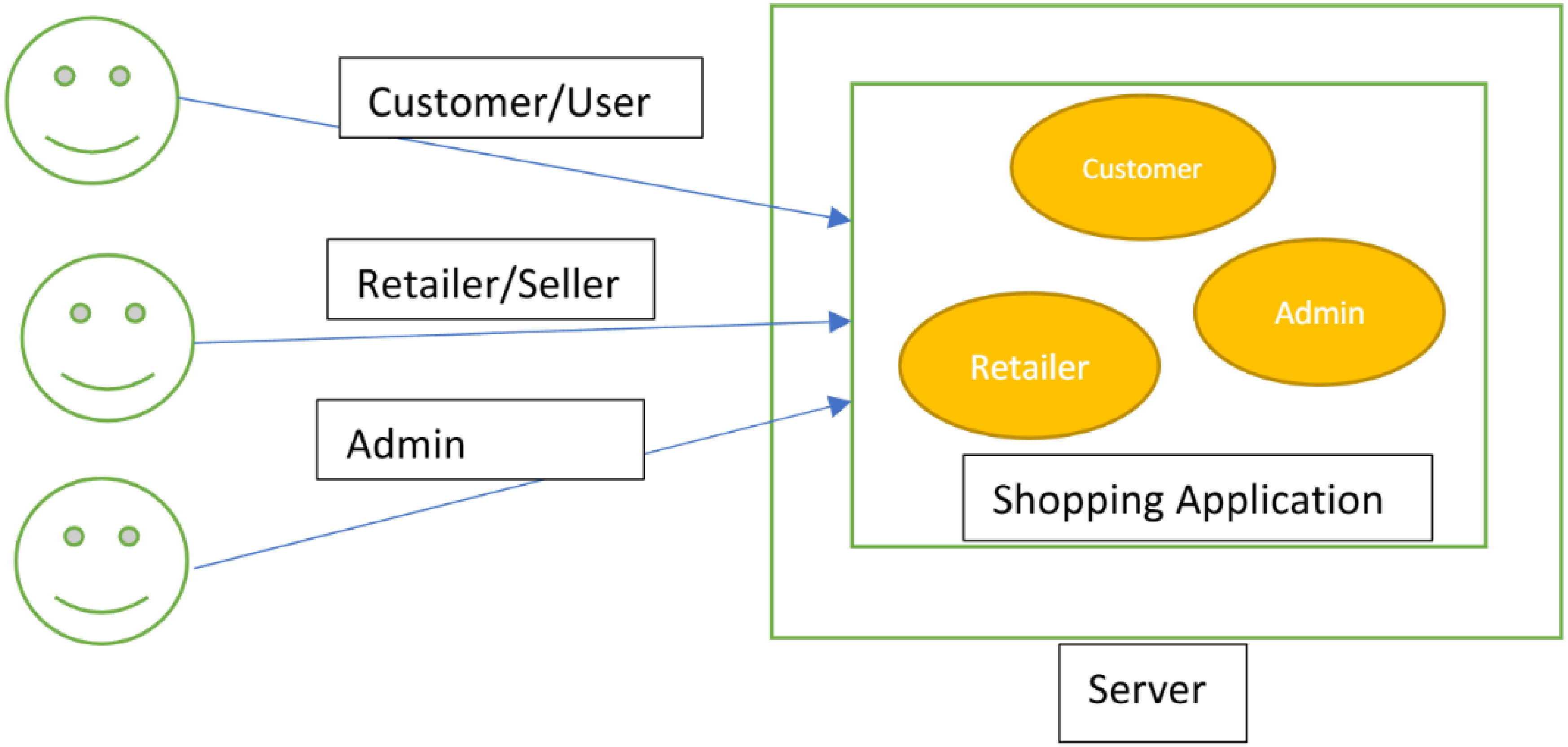
Servlet/JSP

Service (Business Logic)

Dao (Persistence Logic)

DB

Tomcat Server
Deploy: WAR file


Customer/User

Retailer/Seller

Admin

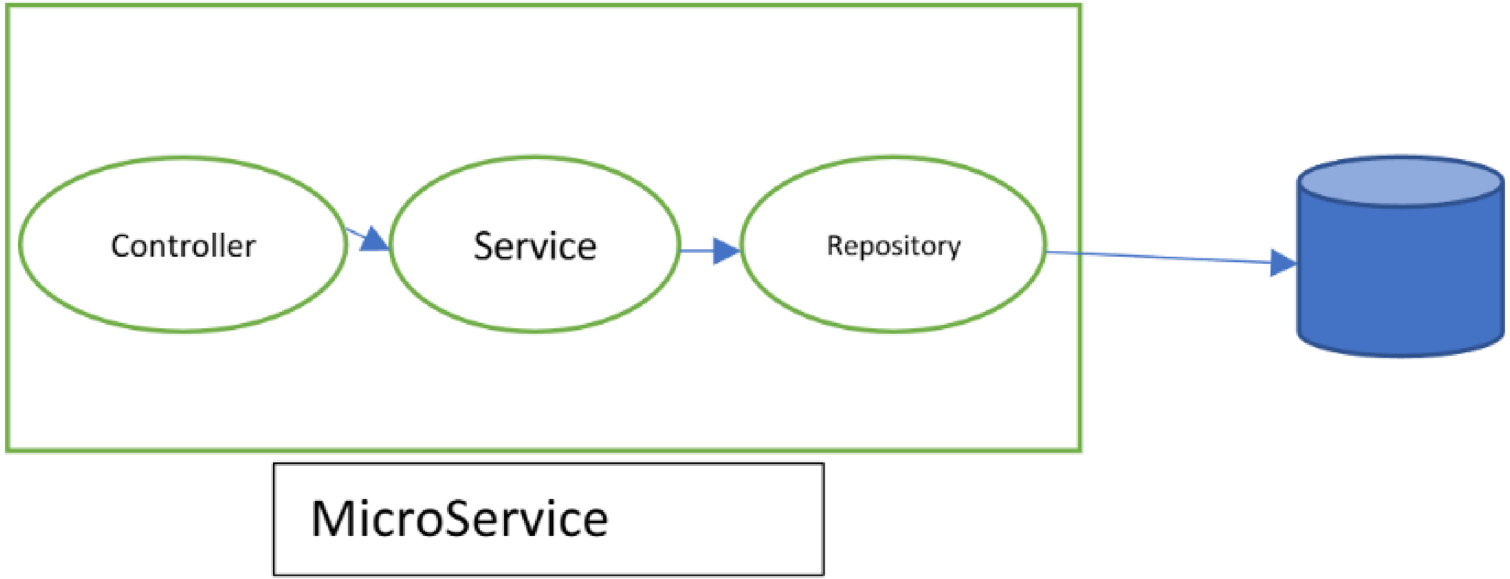
Customer

Admin

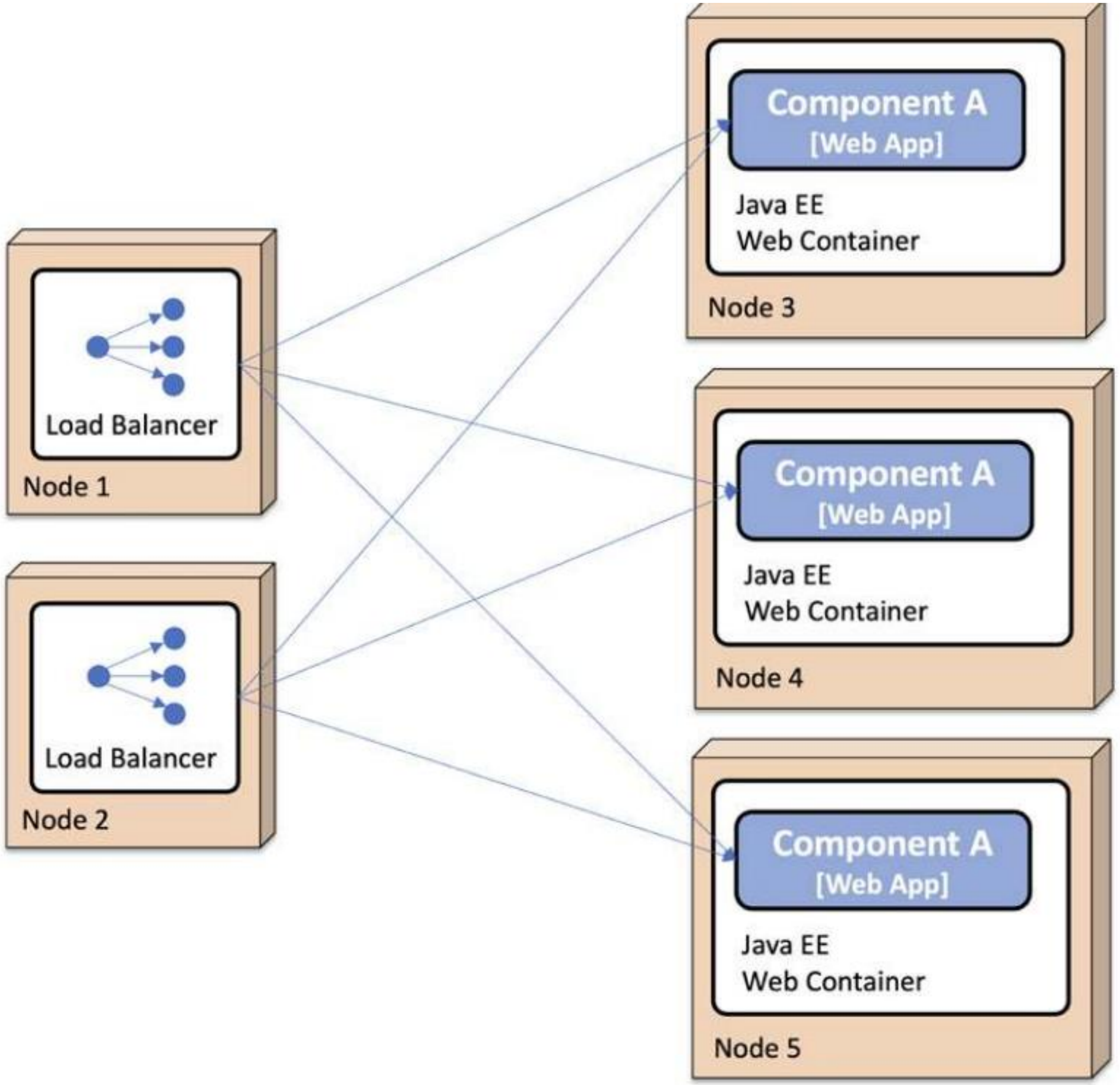
Retailer

Shopping Application

Server

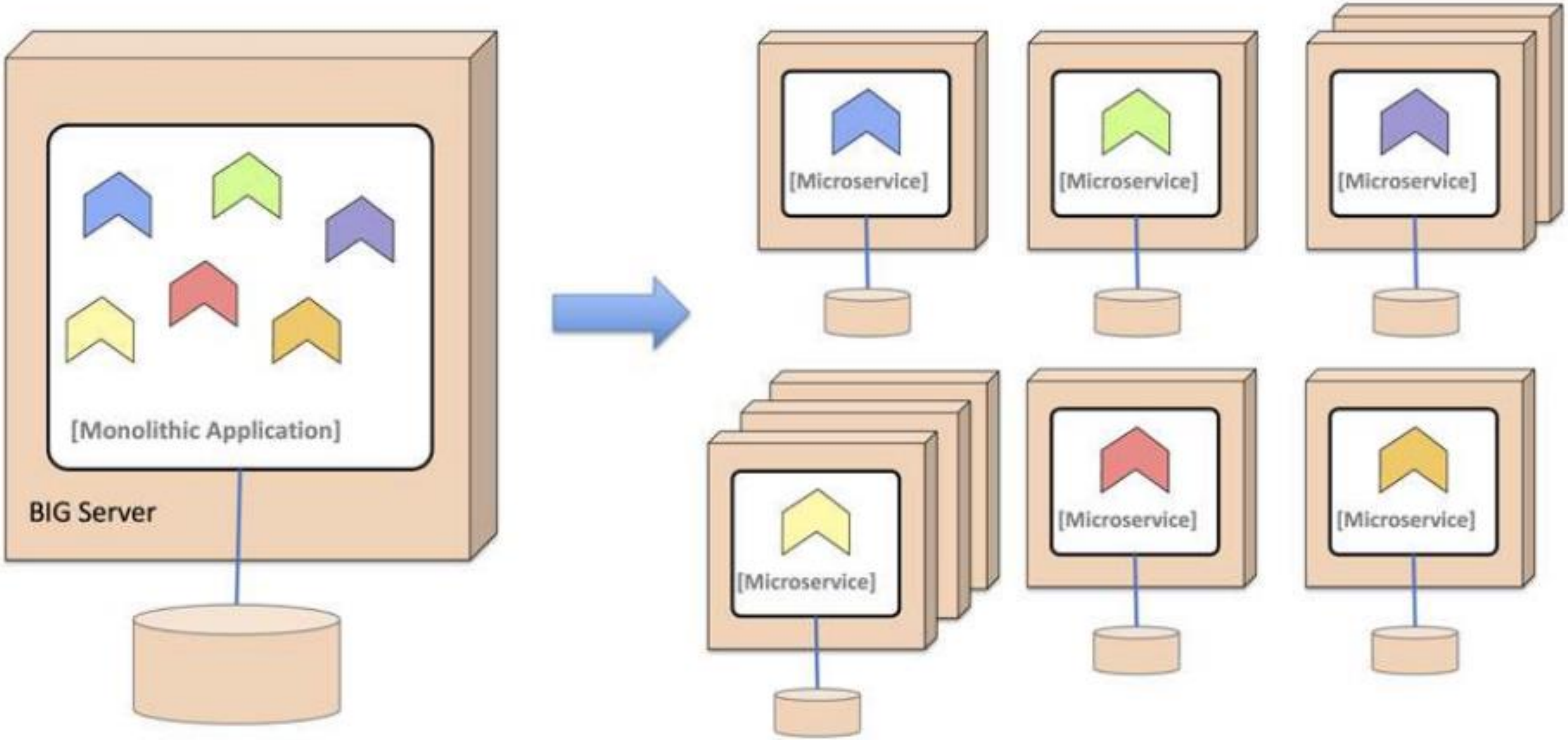

Controller
Service
Repository
MicroService

Before Microservices



Some of the challenges

- Adding new instances of a component required manually load balancer configurations
- Chain of failures
- Central configuration
- Monitoring the usage of resources
- Collecting logs from the distributed instances
- Vertical scaling no issues, but horizontal scaling a challenge

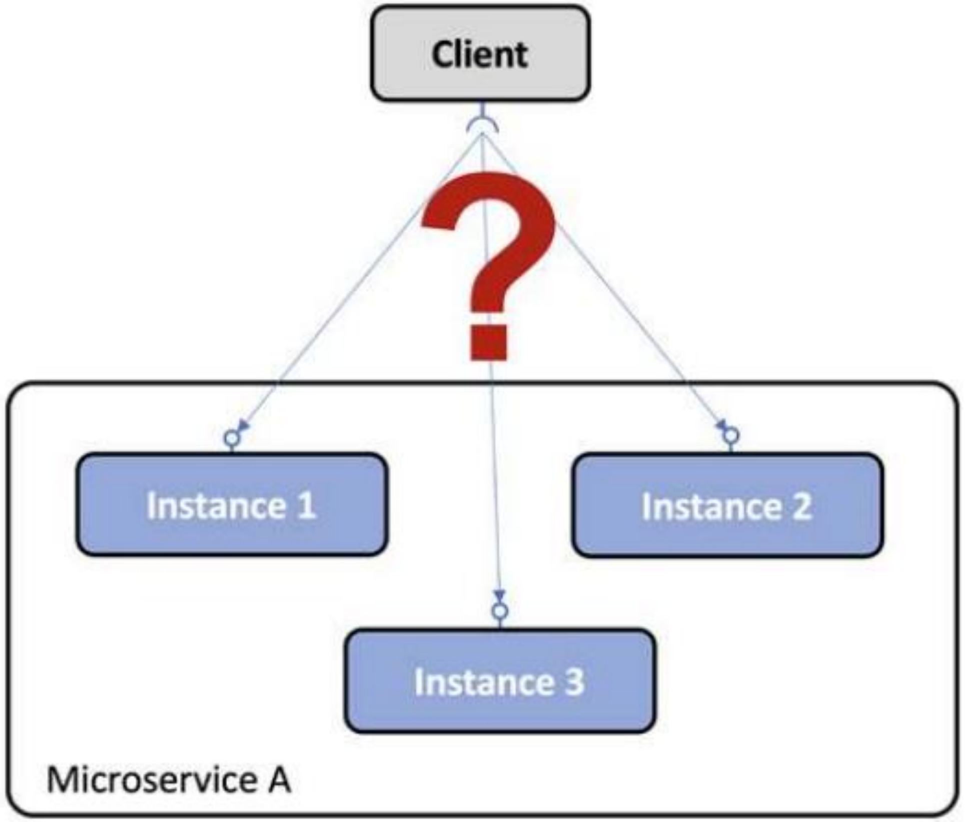

[Monolithic Application]
BIG Server
[Microservice]
[Microservice]
[Microservice]
[Microservice]
[Microservice]
[Microservice]

Design patterns for microservices

- Service discovery
- Edge server
- Central configuration
- Central log management
- Distributed tracing
- Circuit breaker
- Control loop
- Centralized monitoring

Service discovery

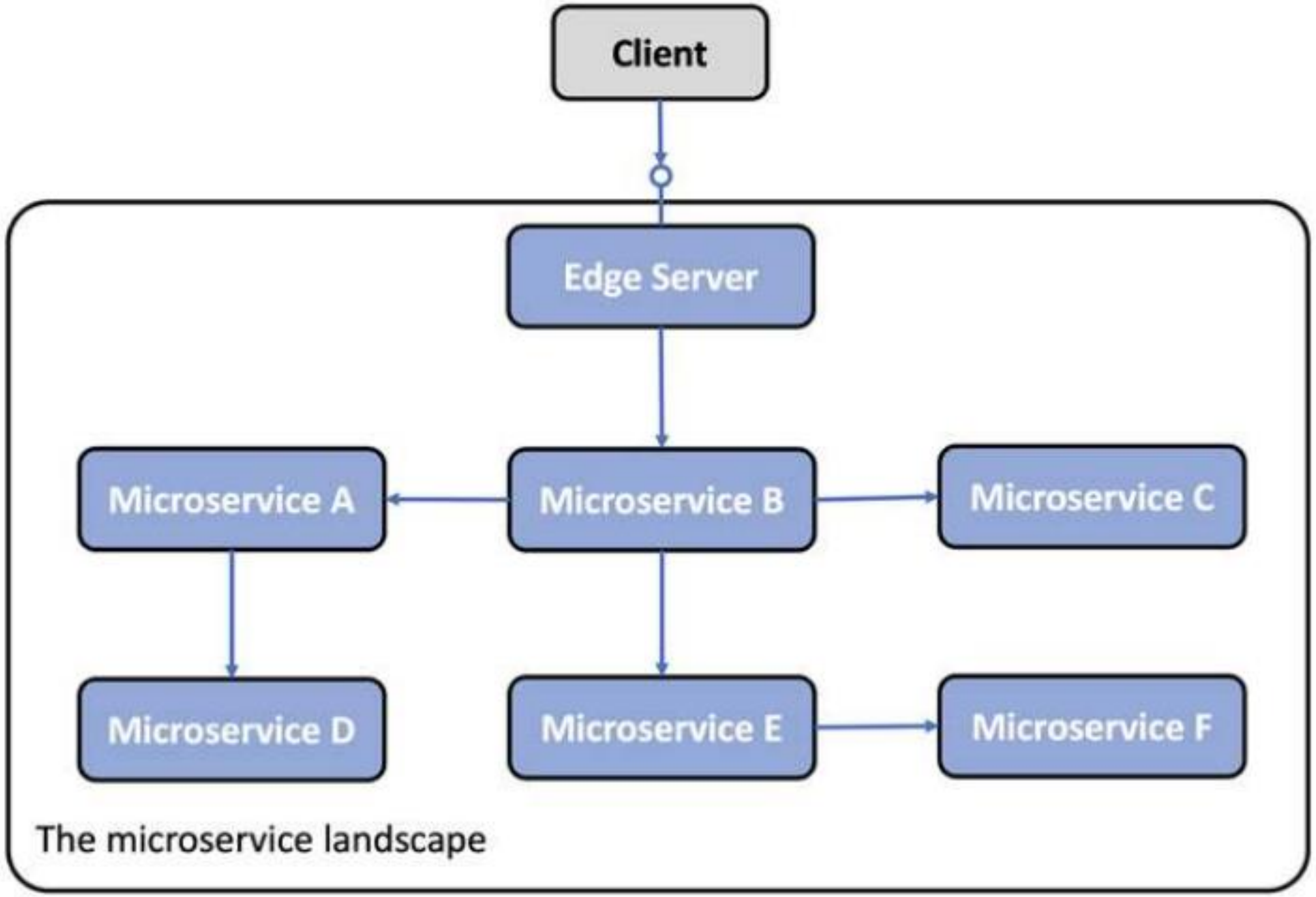
- Client wants to talk to a microservice instance



We can use a service registery & discovery service

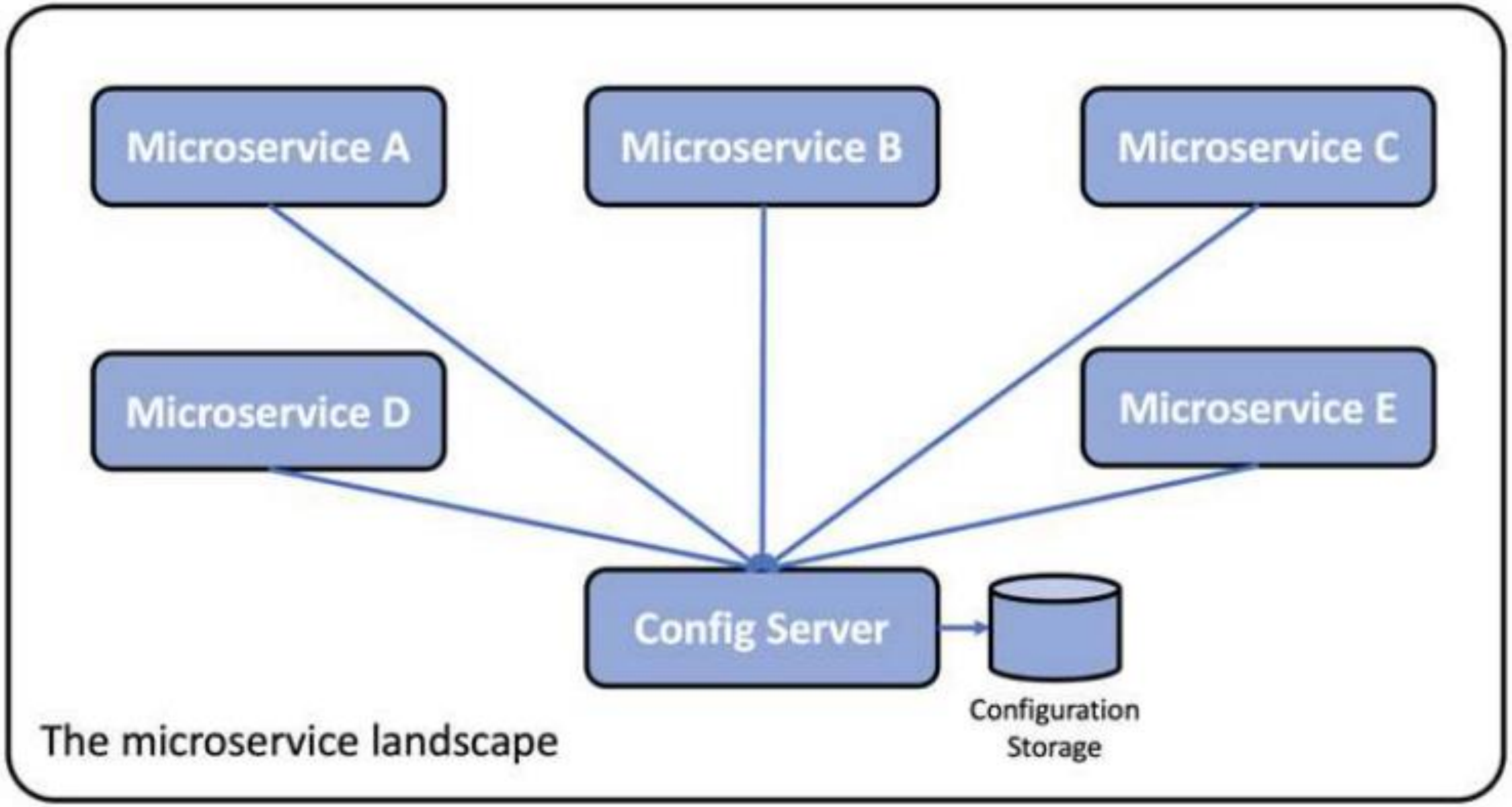
- Each instance of the microservice will register with this register
- Clients will use the registry to locate the available instance

Edge server

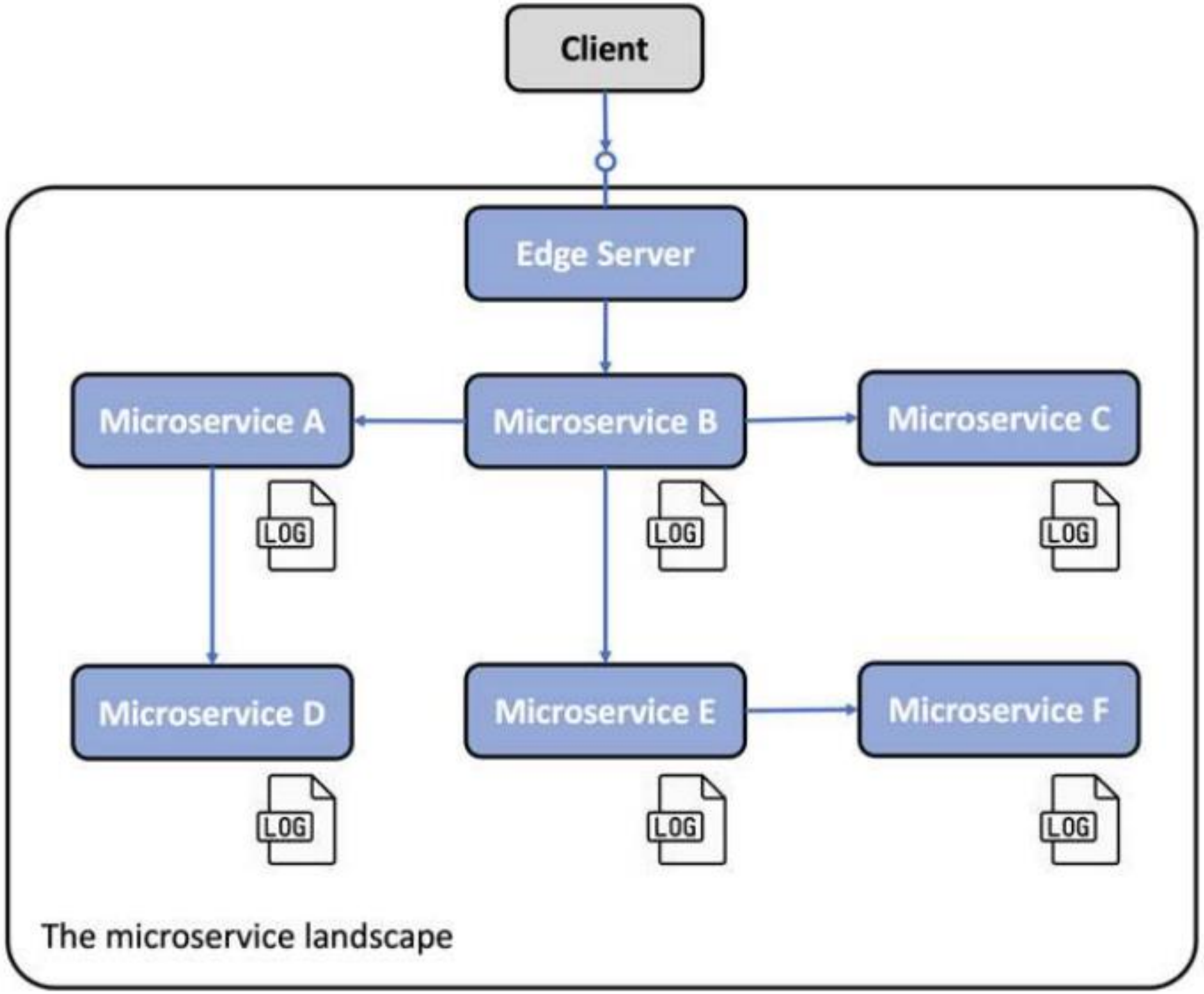


An edge server typically acts like a reverse proxy which can be integrated with a discovery service to control access as well as balance load dynamically

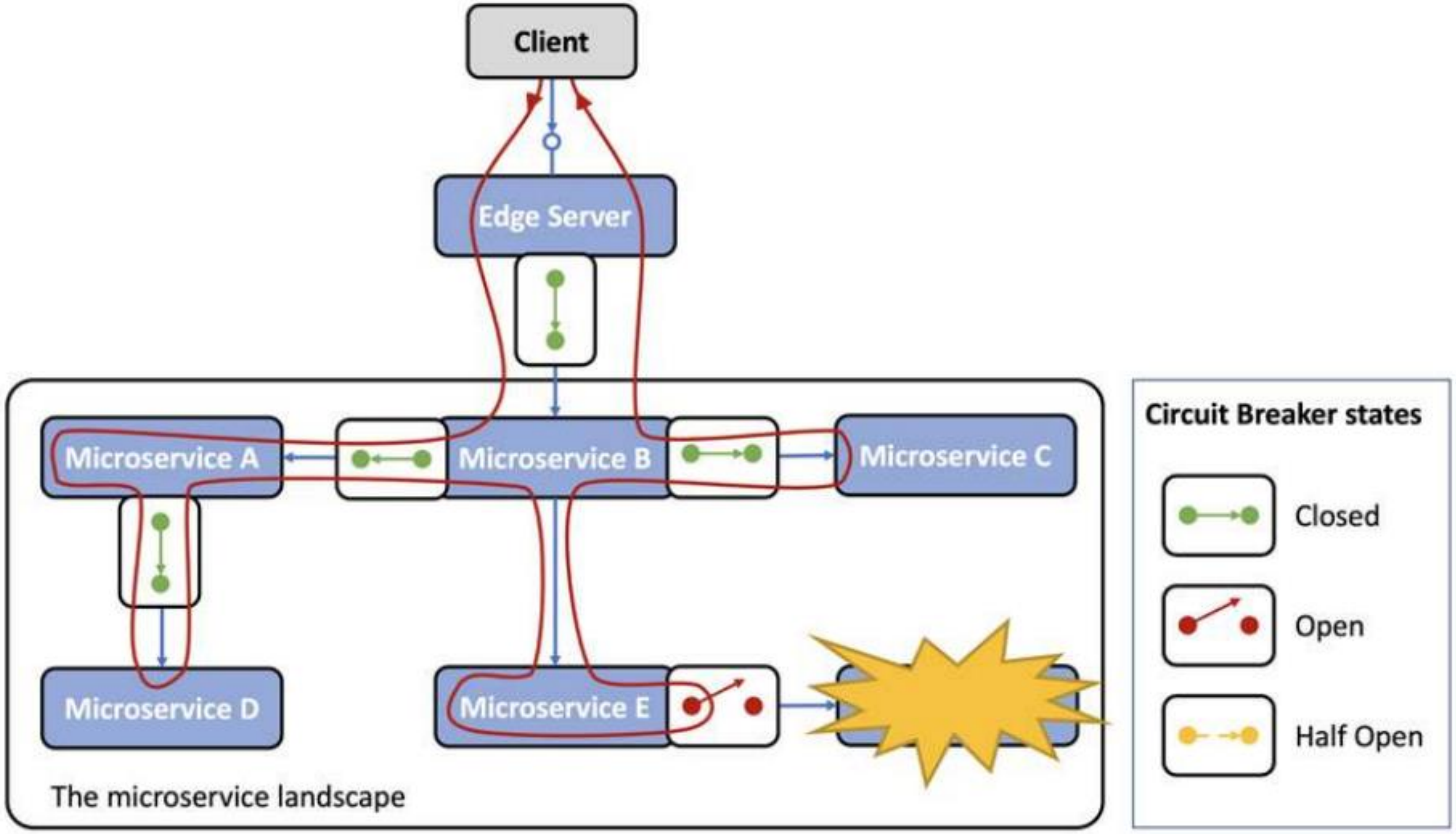
Configuration server

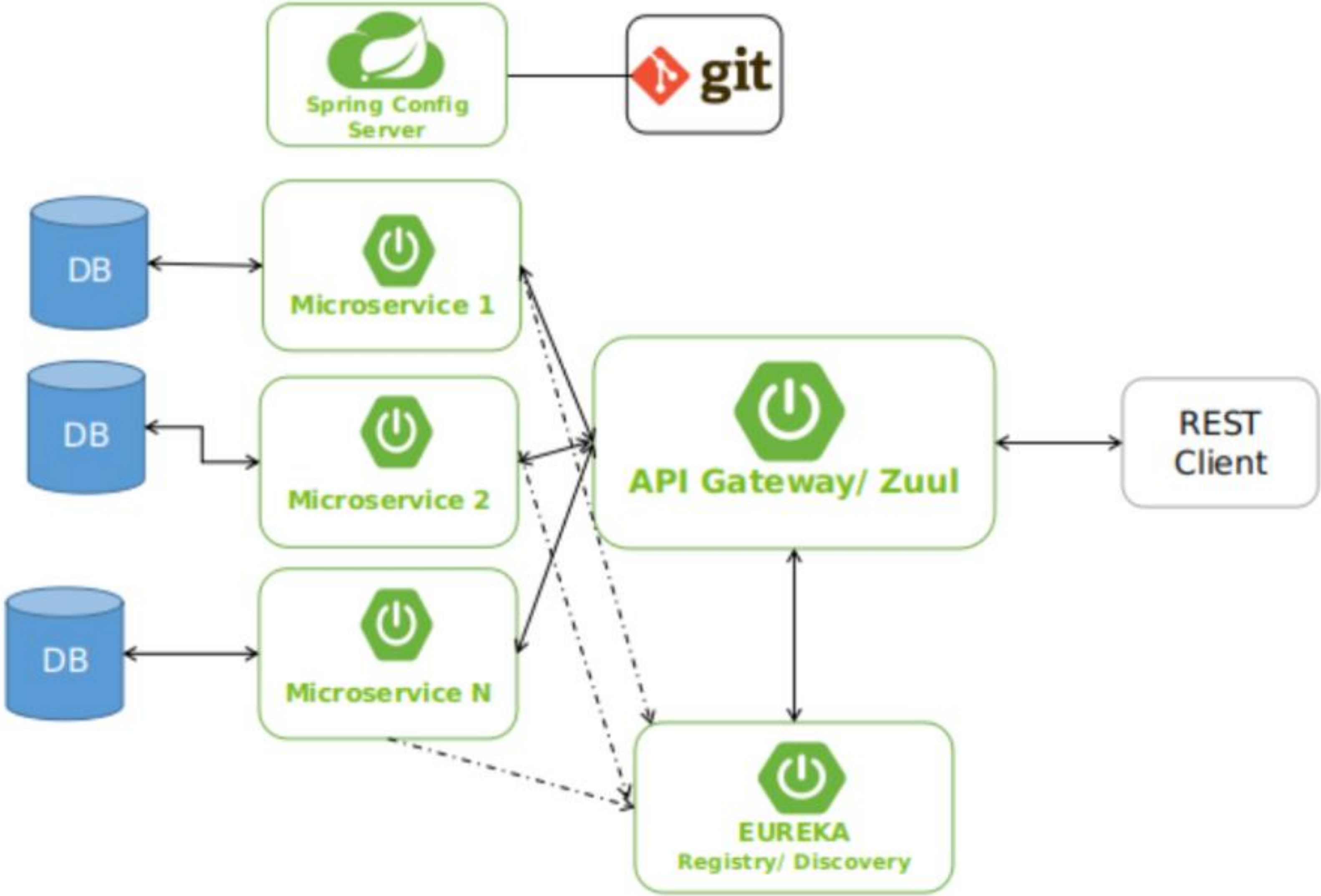


Centralized logging

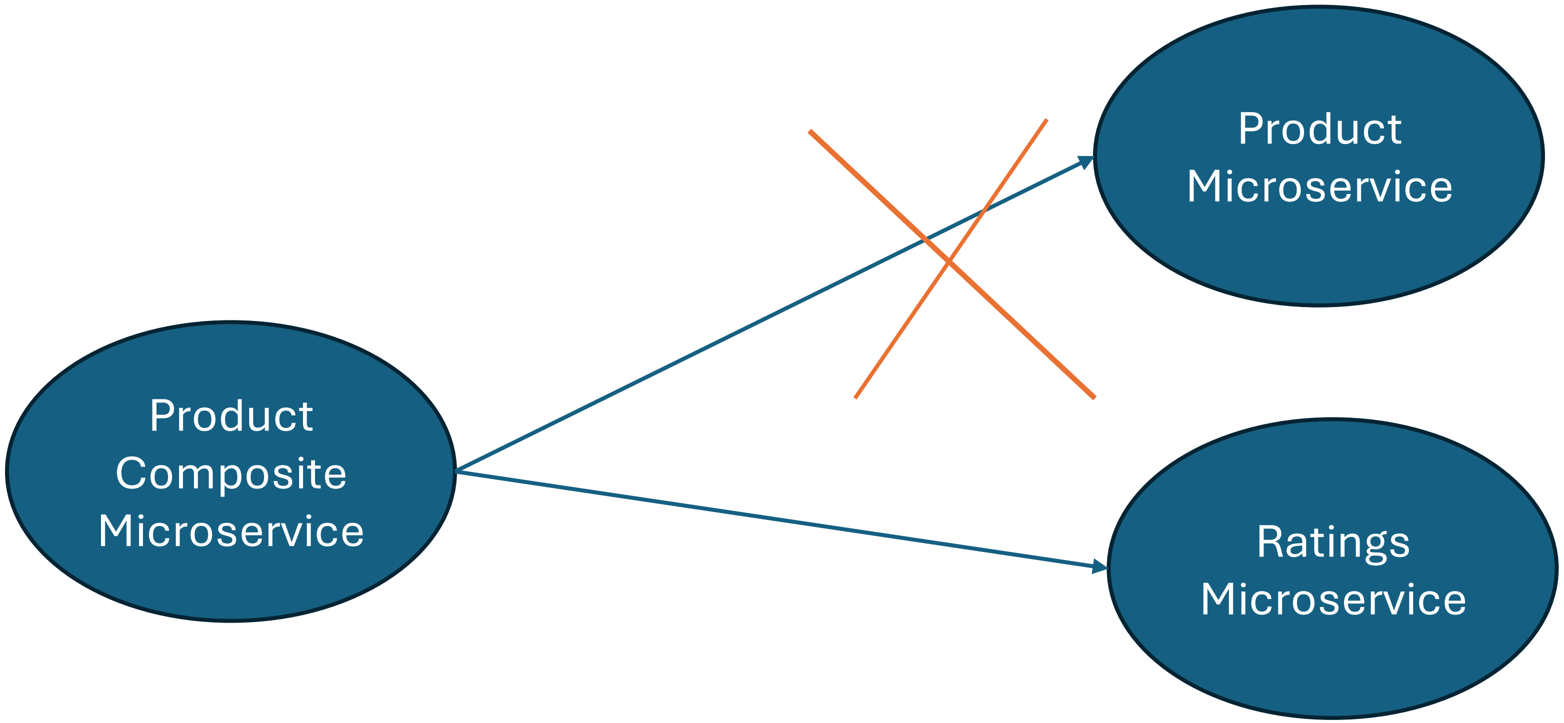


Circuit breaker

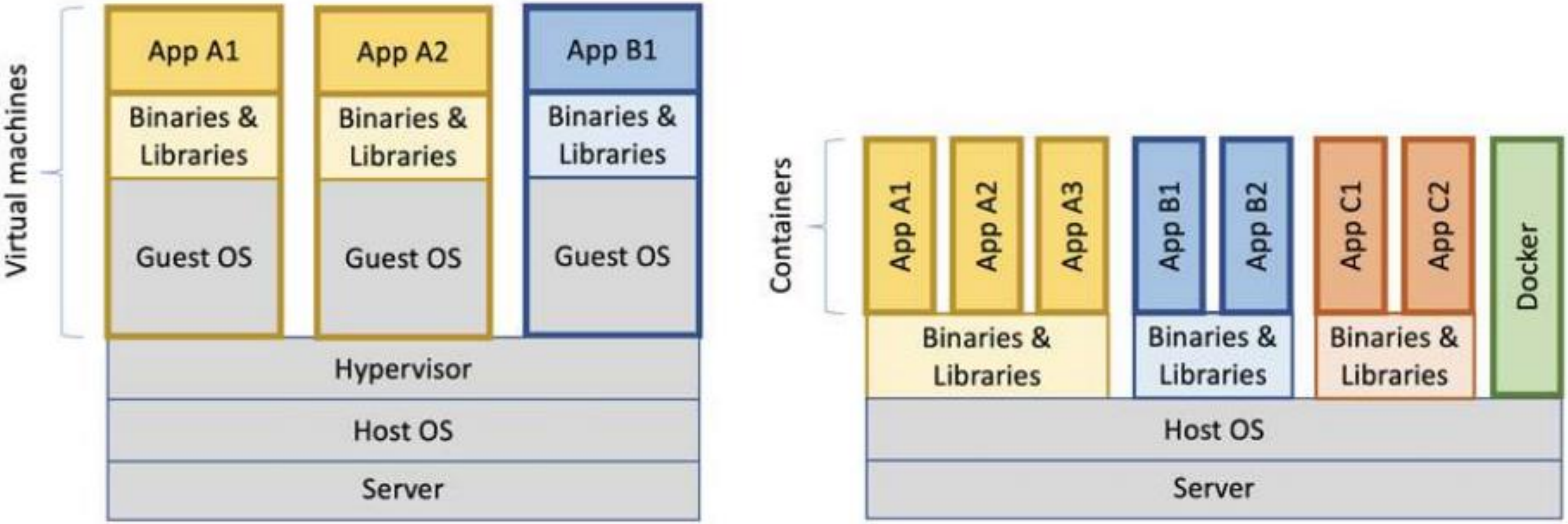



Spring Config Server
git
DB
Microservice 1
DB
Microservice 2
DB
Microservice N
API Gateway/ Zuul
REST Client
EUREKA
Registry/ Discovery

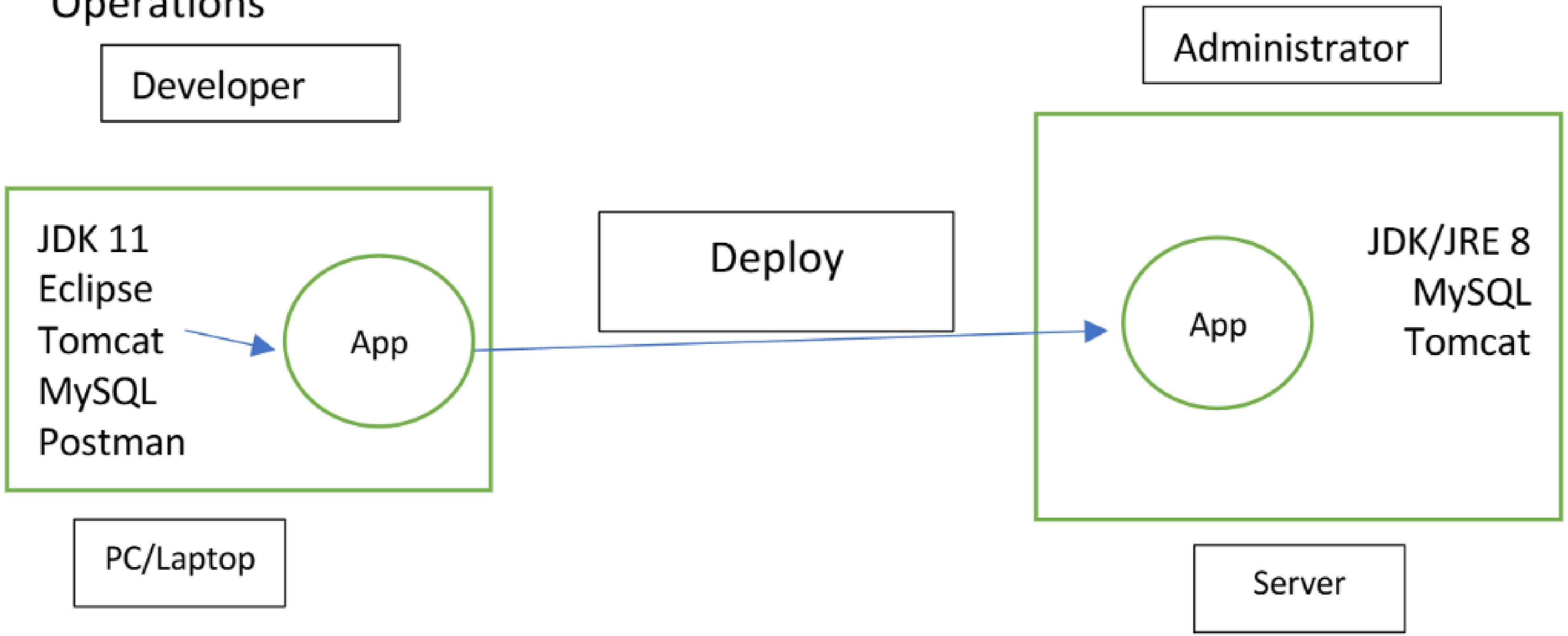
Microservices example



Docker


Virtual machines
App A1
App A2
App B1
Binaries & Libraries
Binaries & Libraries
Binaries & Libraries
Guest OS
Guest OS
Guest OS
Hypervisor
Host OS
Server
Containers
App A1
App A2
App A3
App B1
App B2
App C1
App C2
Docker
Binaries & Libraries
Binaries & Libraries
Binaries & Libraries
Host OS
Server

Developer Operations



Developer

JDK 11
Eclipse
Tomcat
MySQL
Postman

App

PC/Laptop

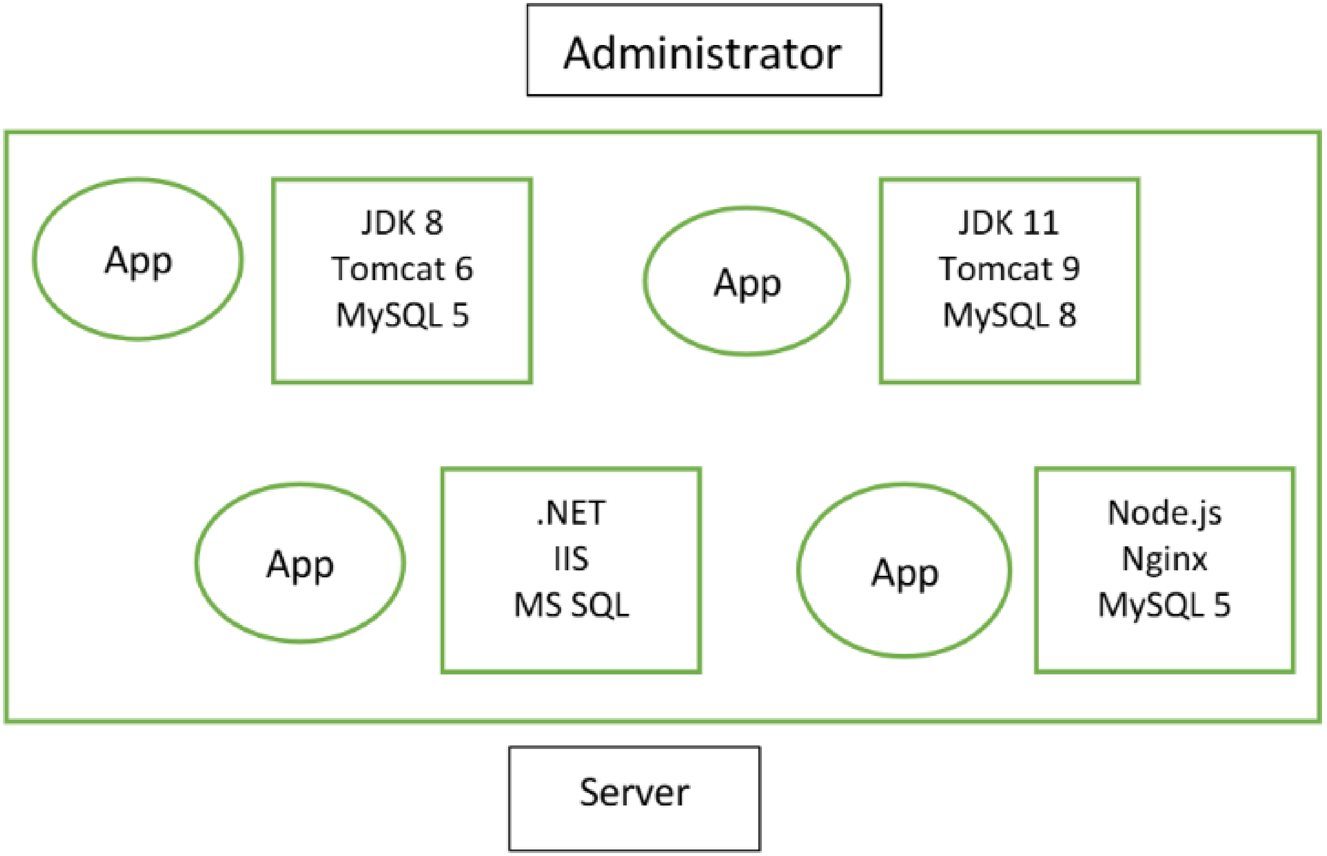
Deploy

Administrator

App

JDK/JRE 8
MySQL
Tomcat

Server



Administrator

App

JDK 8
Tomcat 6
MySQL 5

App

JDK 11
Tomcat 9
MySQL 8

App

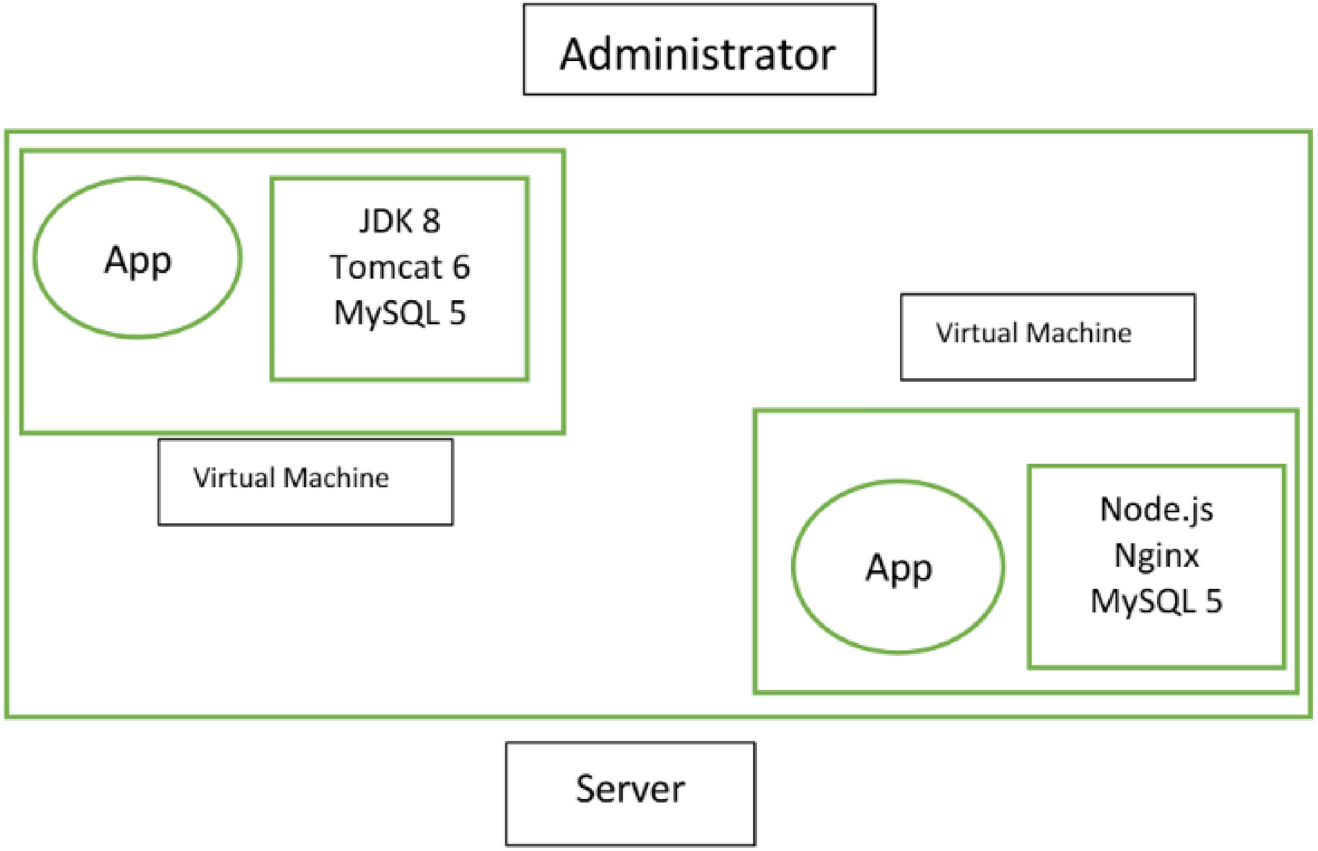
.NET
IIS
MS SQL

App

Node.js
Nginx
MySQL 5

Server

- -


Administrator
App
JDK 8
Tomcat 6
MySQL 5
Virtual Machine
Virtual Machine
App
Node.js
Nginx
MySQL 5
Server