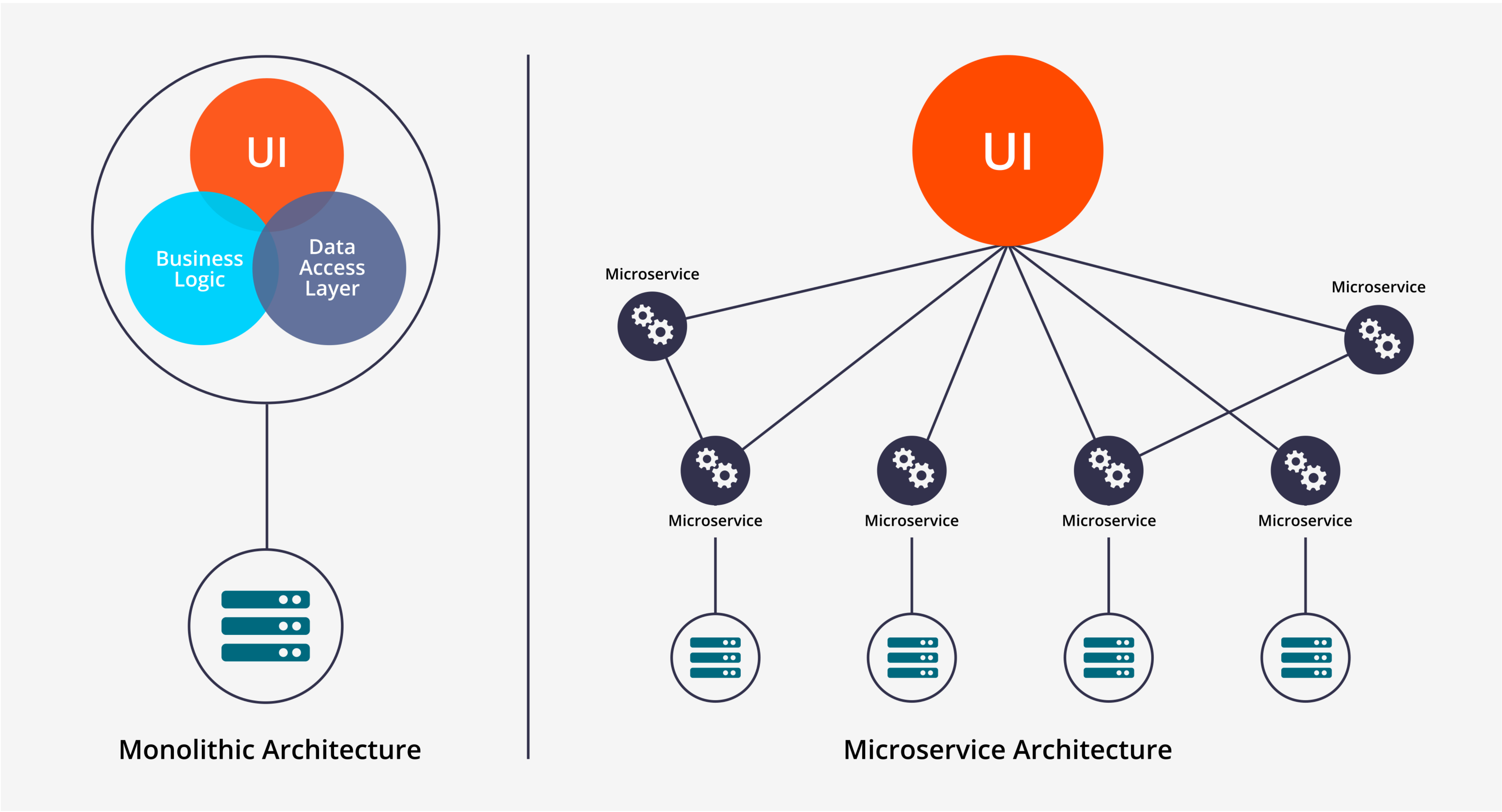
Spring boot And Micro Services

Kayartaya Vinod
https://vinod.co
vinod@vinod.co

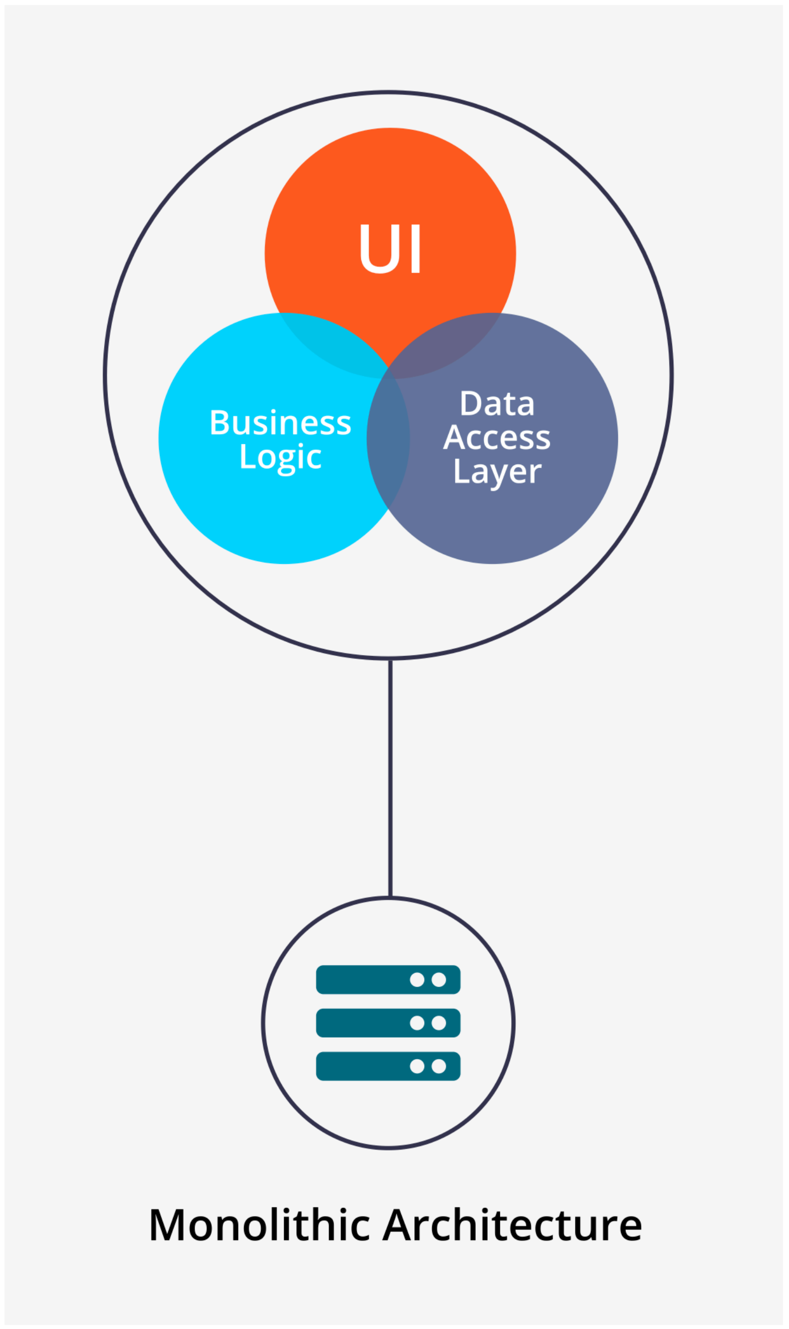
Monolith vs Micro services



source: https://medium.com/startlovingyourself/microservices-vs-monolithic-architecture-c8df91f16bb4

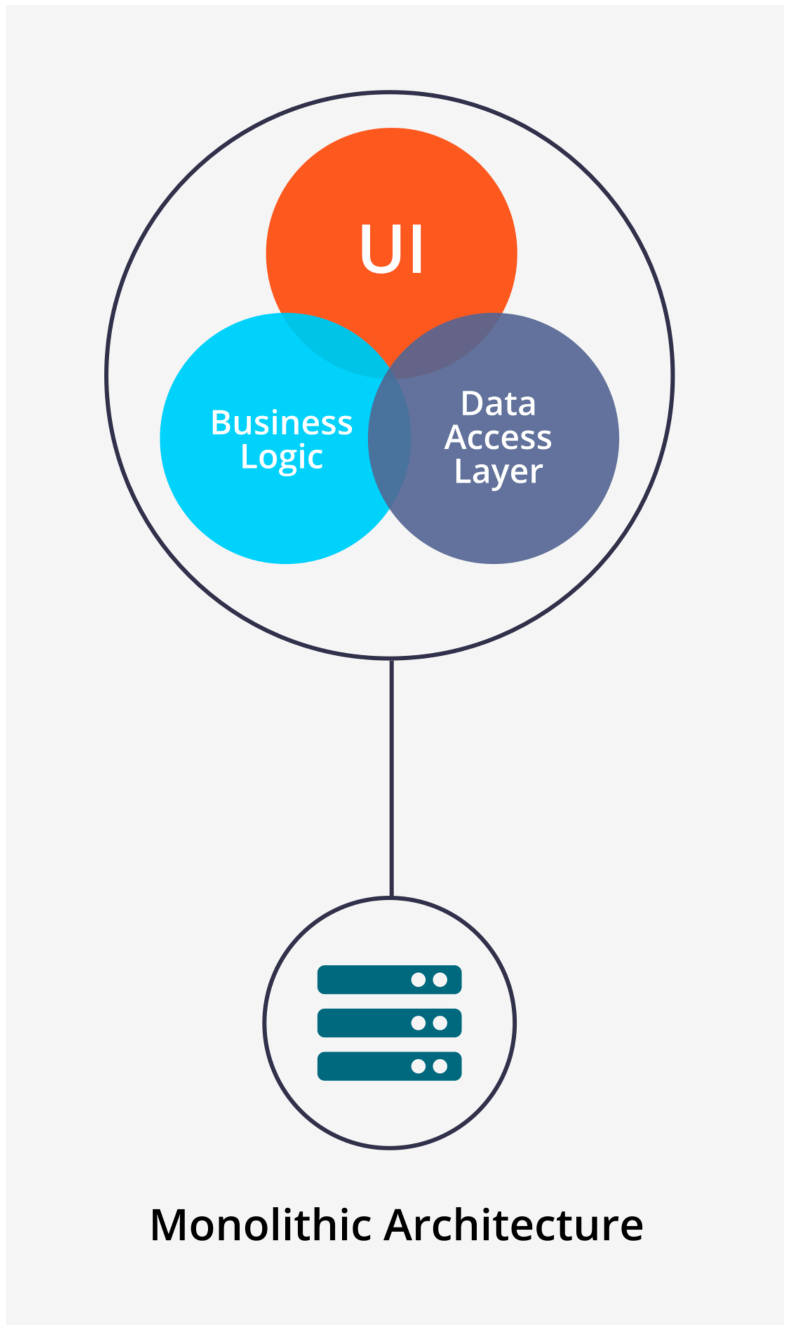
Monolith architecture

- Before Microservices, a common approach to design an application was to use a monolithic architecture.
- In this mode of development, the application is deployed as a single deployment artefact.



Monolith architecture

- Challenging to release new services and difficult to in maintaining large code base
- Modules can't scale independently
- Difficult to enable/disable services without affecting existing features
- Small changes in the code lead to many regression testing/complex deployment scenarios.
- Will have challenges in continuous deployments

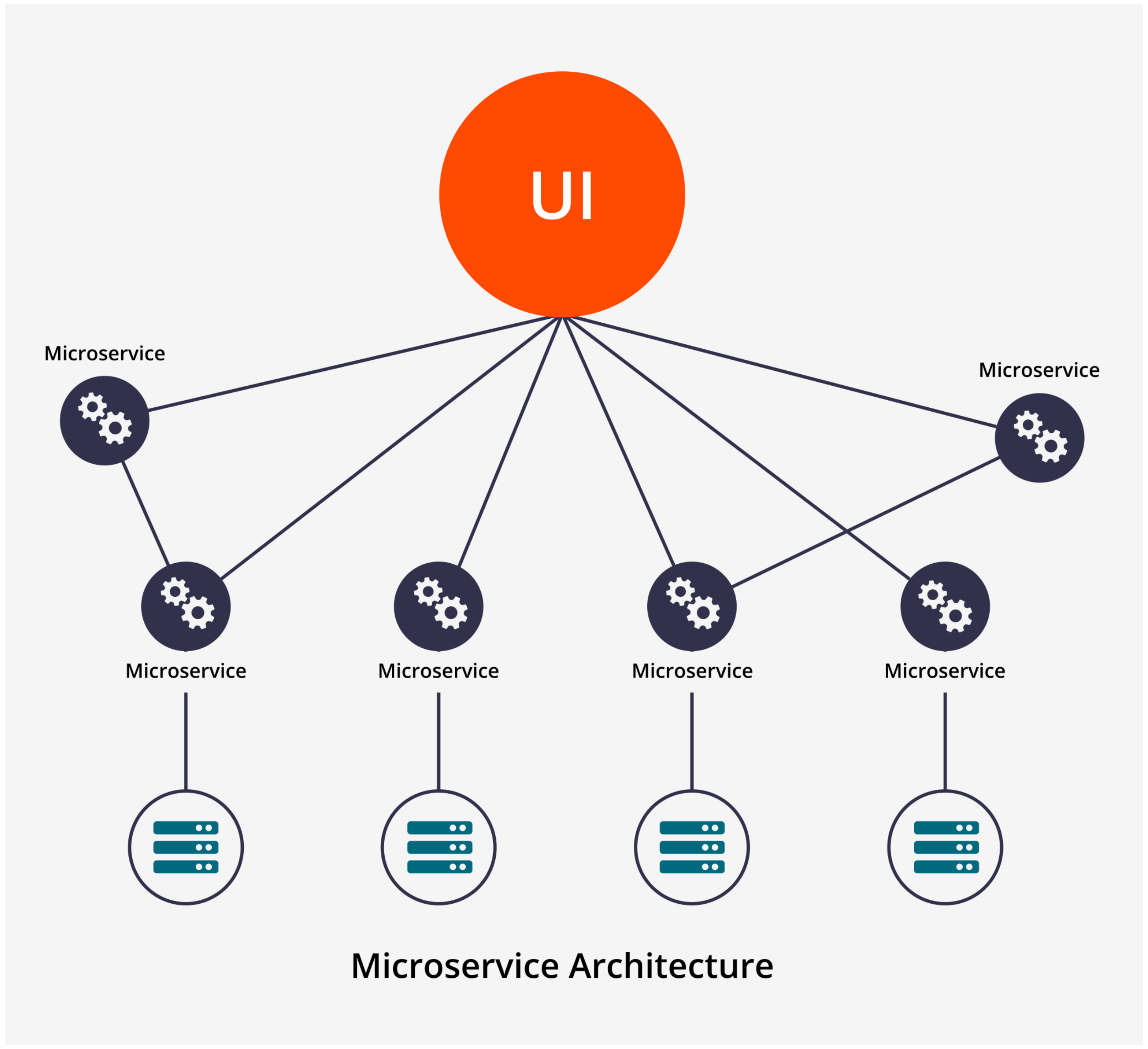


Microservice architecture

- It is an architectural style to develop a small/coarse grained services encapsulated by clear business boundaries.

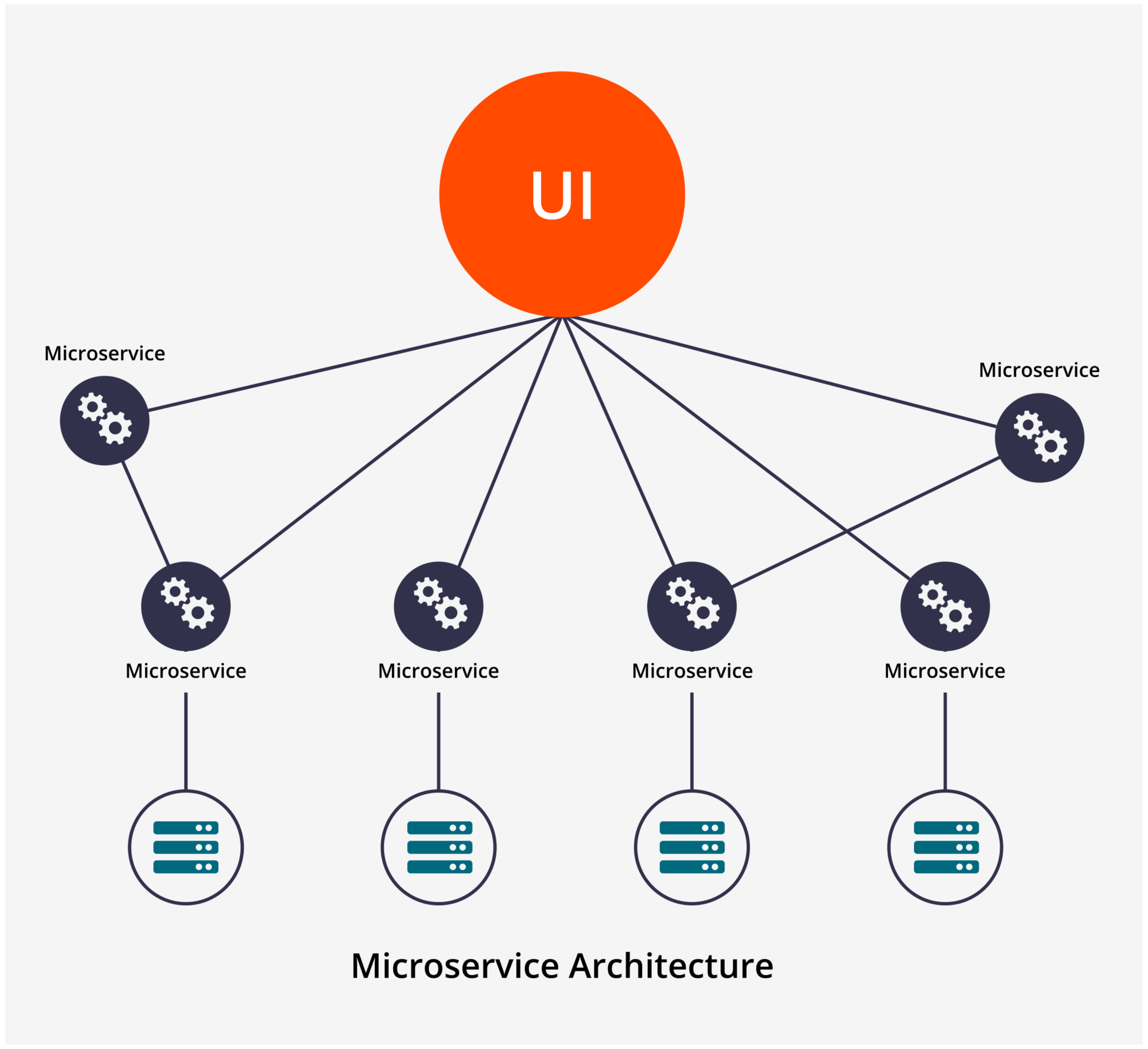
- Thus each service is autonomous in nature and can scale independently.

- Each service can communicate each other via REST APIs and can be deployed remotely or locally.



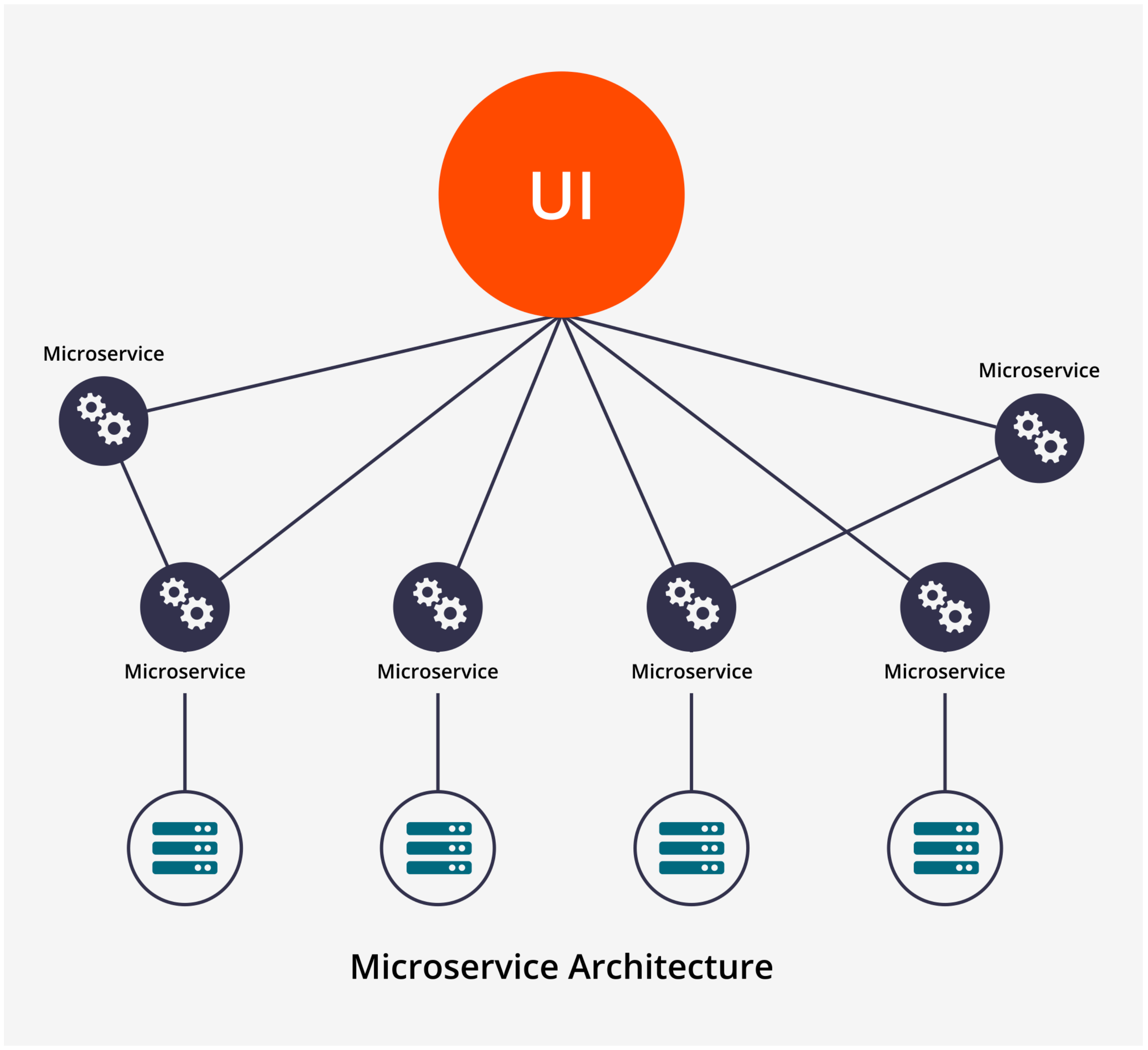
Pros

- Decoupled services can have isolated life cycle
- Faster time to market; less regressions, smaller team size
- Modular in nature, hence flexible development and deployment plans
- Can scale independently



Cons

- Monitoring many services
- System may become complex easily – For example: Microservices are distributed in nature. Hence there will be added complexity such as network latency, network unreliability, also understanding the DevOps culture
- Technology learning curve – new set of tools to learn



Spring boot

Spring Boot is an opinionated, convention-over-configuration focused addition to the Spring platform – highly useful to get started with minimum effort and create stand-alone, production-grade applications.

Features

- Zero or minimal configuration
- Includes a web container (Tomcat/Jetty/Undertow)
- Automatically creates Spring container to manage beans
- Automatically loads all components
- Can do many things by just adding JARs and annotations

Spring boot

DEMO

What next?

- Dockerization, Docker-swarm, K8s
- Spring cloud
  - Spring Cloud Config
  - Spring Cloud Netflix
  - Spring Cloud Security
  - Spring Cloud Sleuth
  - Spring Cloud Stream
  - … and many more

Check out my courses

- Spring Framework MasterClass:
  - https://vinod.co/view/spring-framework-masterclass
- Develop RESTful Java Web Services using Spring boot:
  - https://vinod.co/view/developing-restful-java-web-services-using-spring-boot

Spring boot

Thank You