



# Integrasi Sistem

---

SIROJUL MUNIR | [DEV.XBATA.COM](https://dev.xbata.com)

STT TERPADU NURULFIKRI



# Micro Service

---

SIROJUL MUNIR | @ROJULMAN

# Timeline – To Micro Service

1990s and earlier

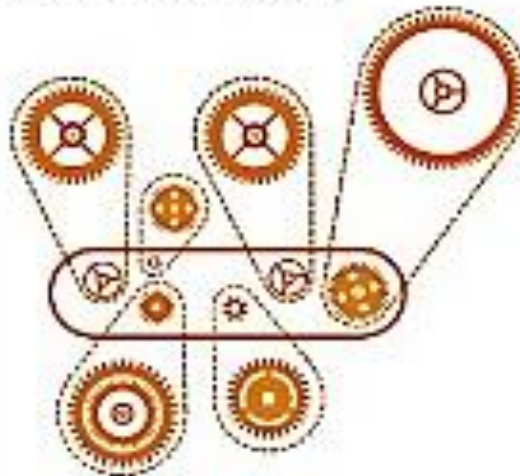
**Coupling**

**Pre-SOA (monolithic)**  
Tight coupling



2000s

**Traditional SOA**  
Looser coupling

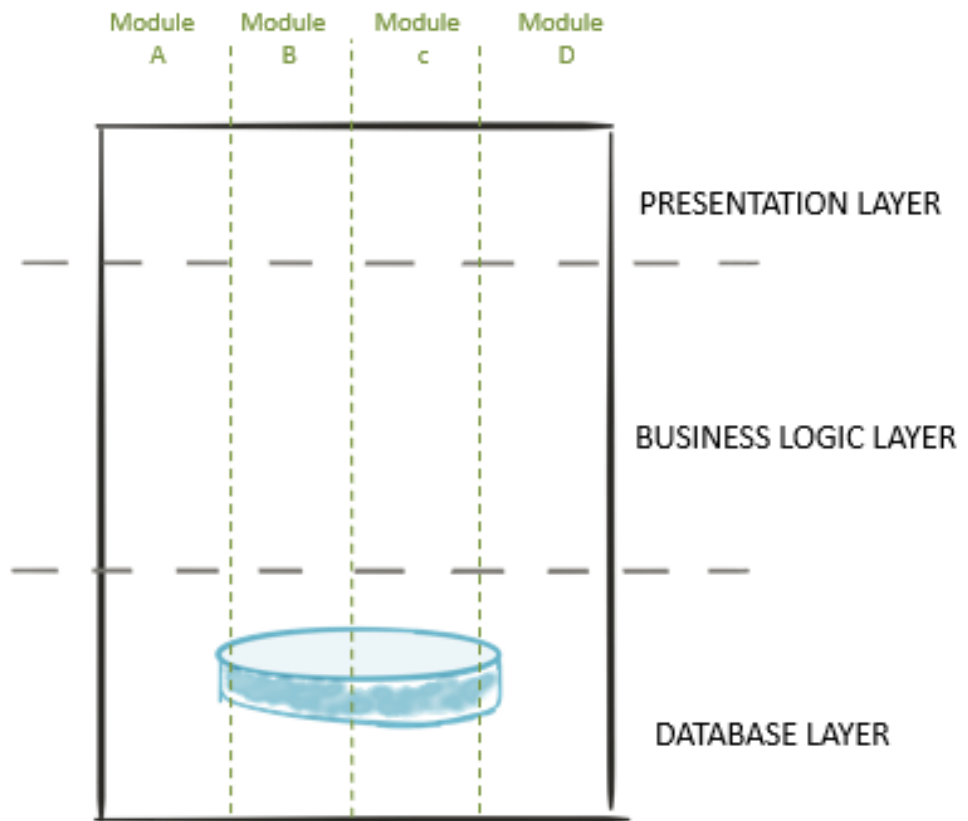


2010s

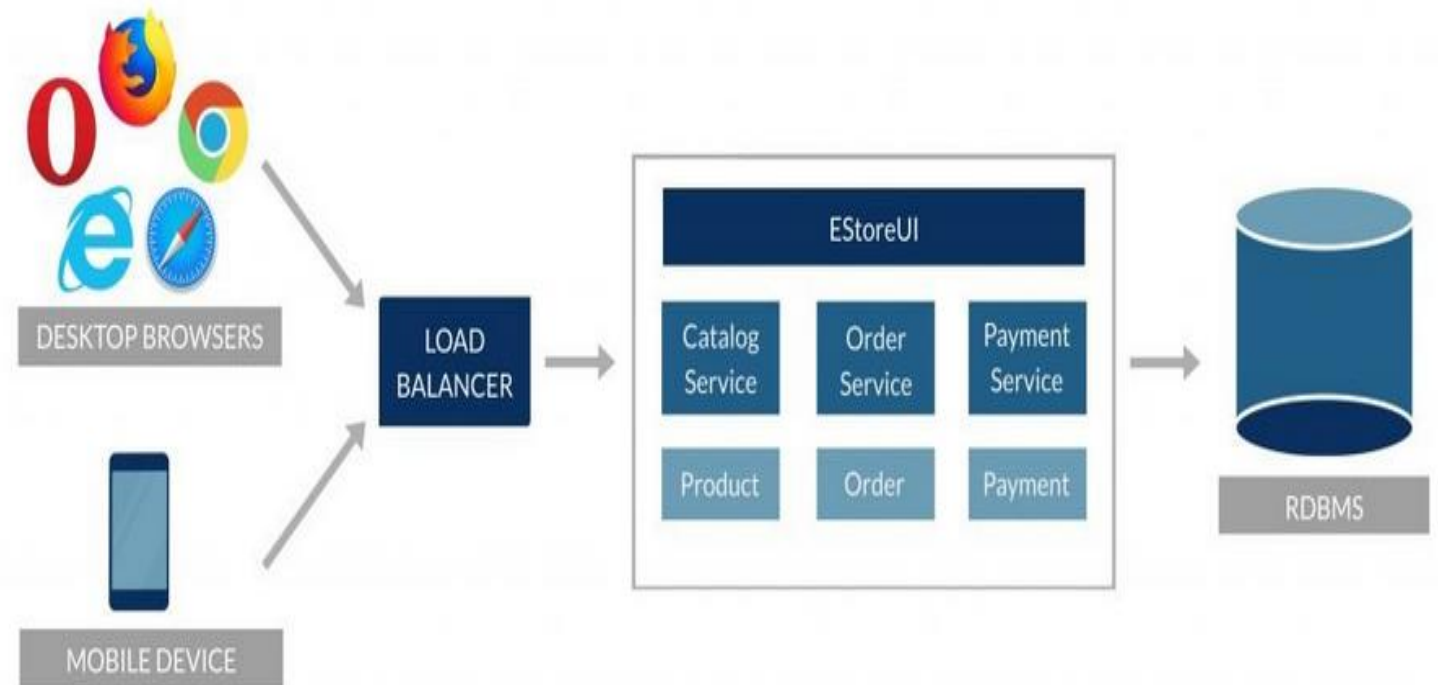
**Microservices**  
Decoupled



# Monolithic



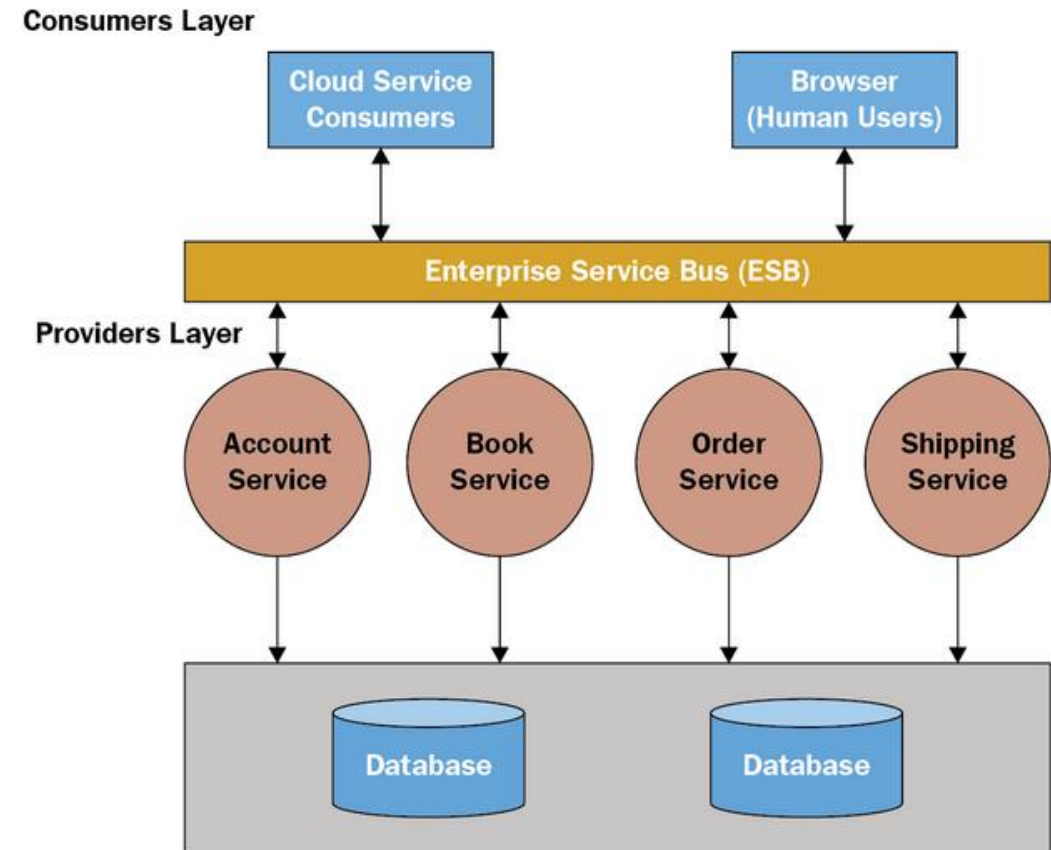
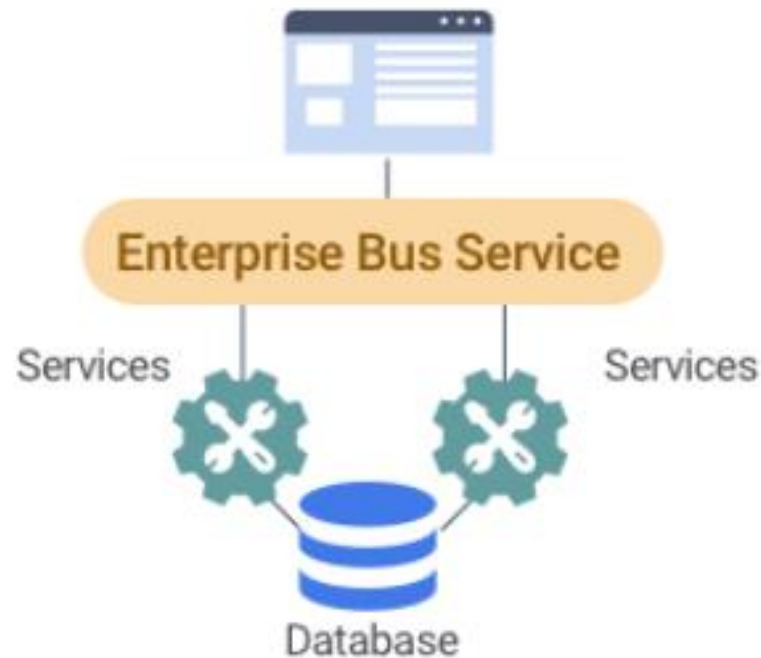
Sebutkan karakteristik aplikasi monolithic ?



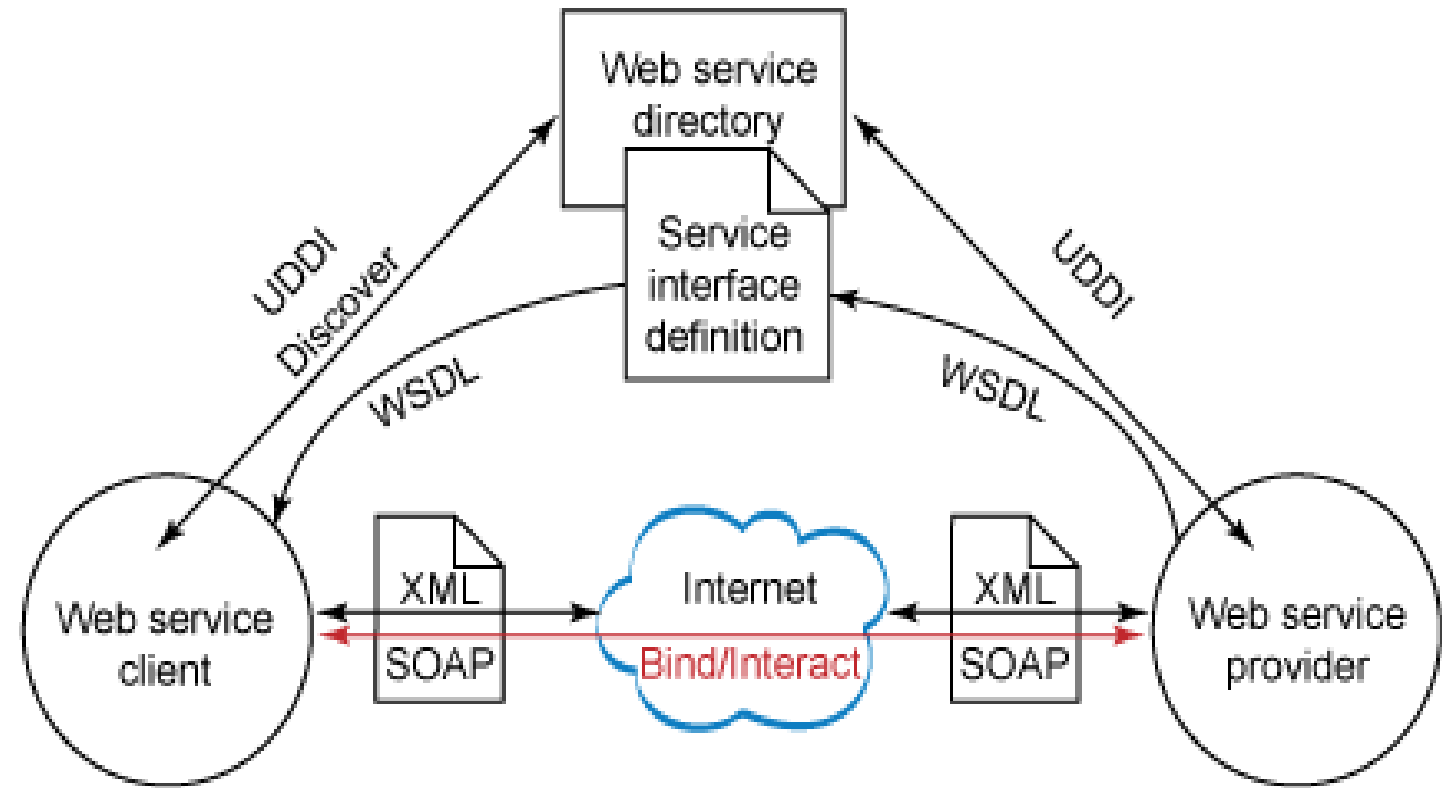
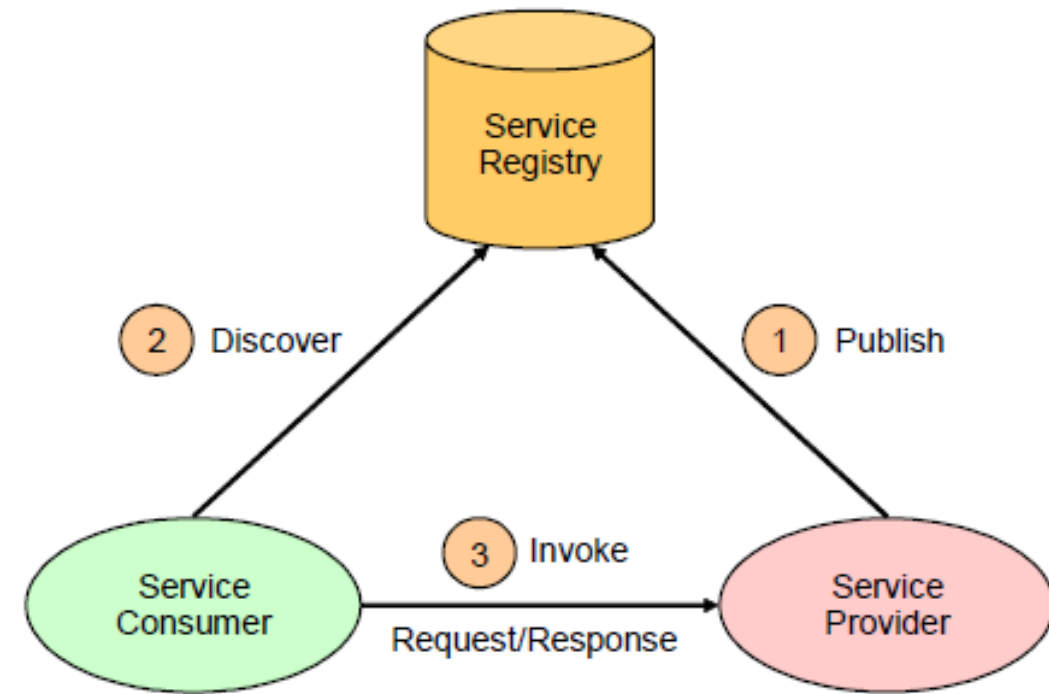
Apa keuntungan aplikasi monolithic ?

# SOA

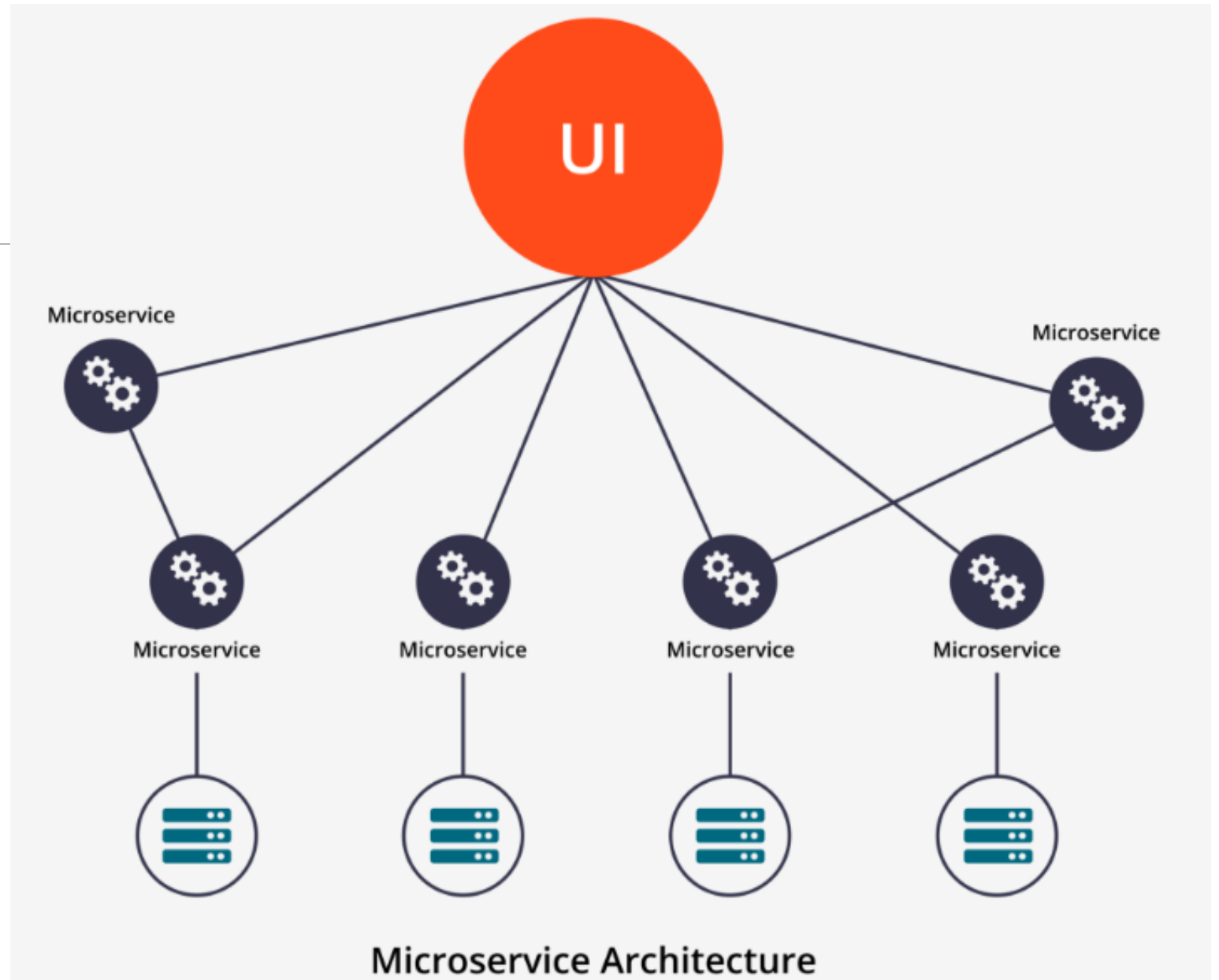
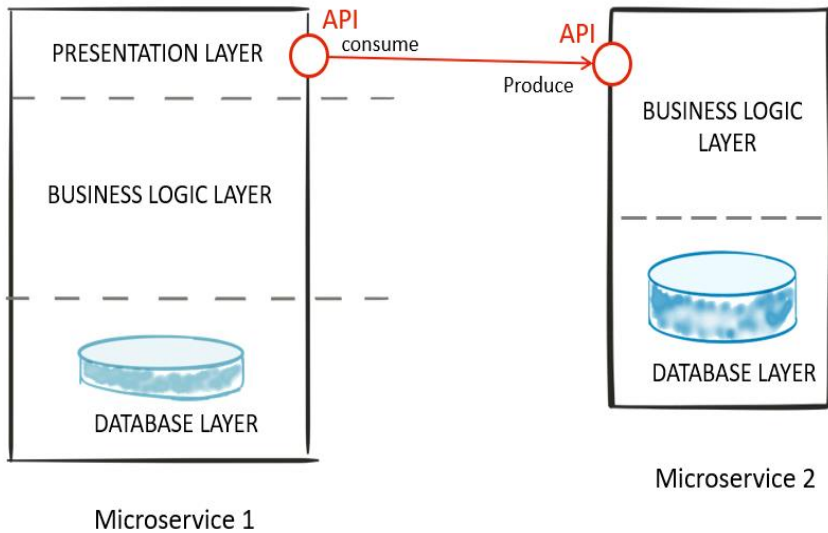
## Service Oriented Architecture



# SOA - Component

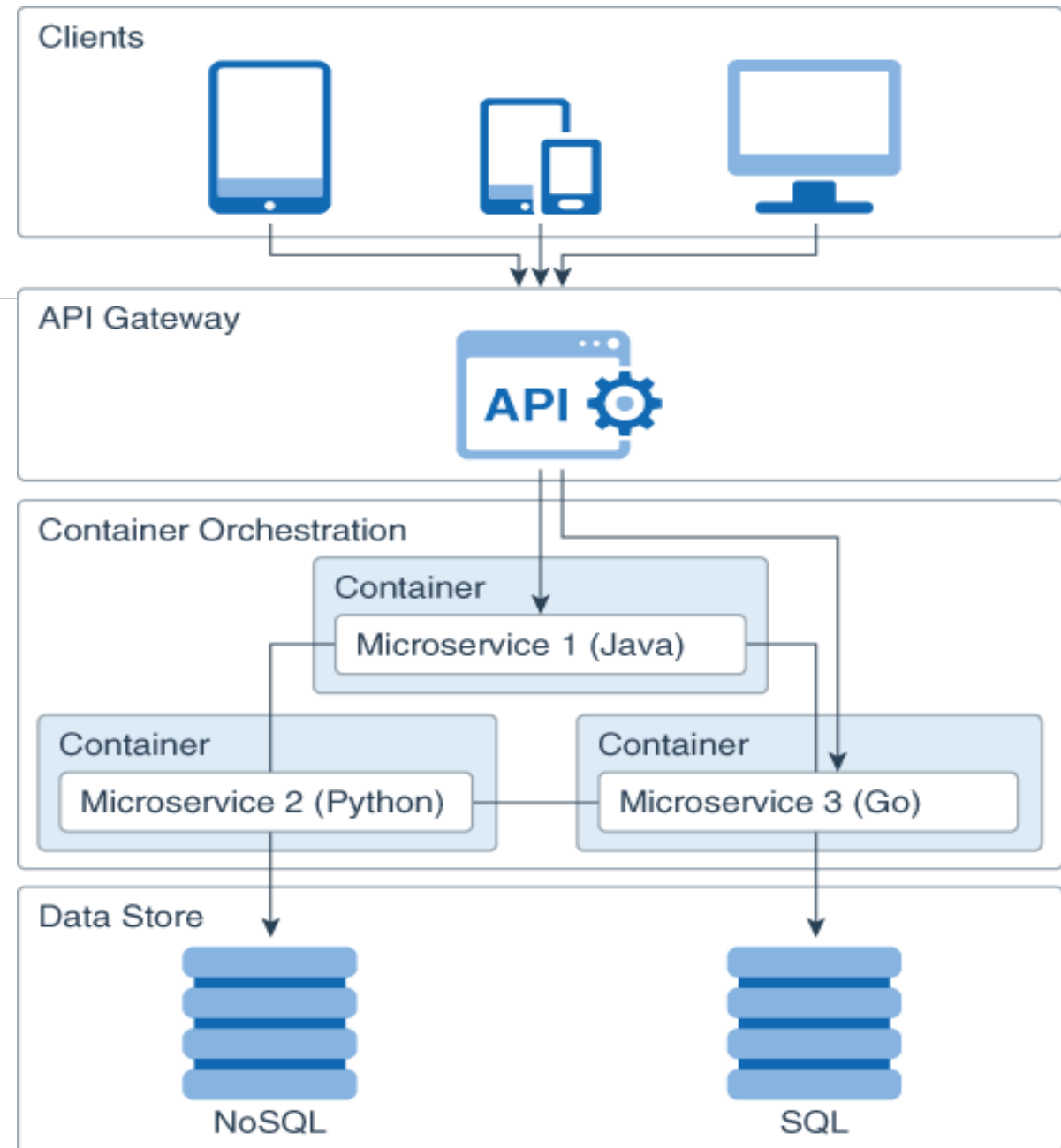
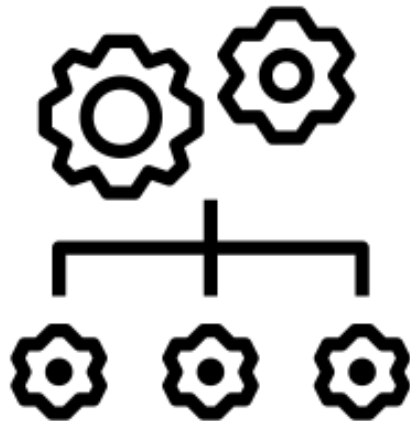


# Microservice



# Microservice

Membagi aplikasi menjadi banyak layanan (services) yang lebih kecil (micro) dan saling terhubung, dengan prinsip kemudahan maintenance, independent software, secure & reliable, adaptive new technology



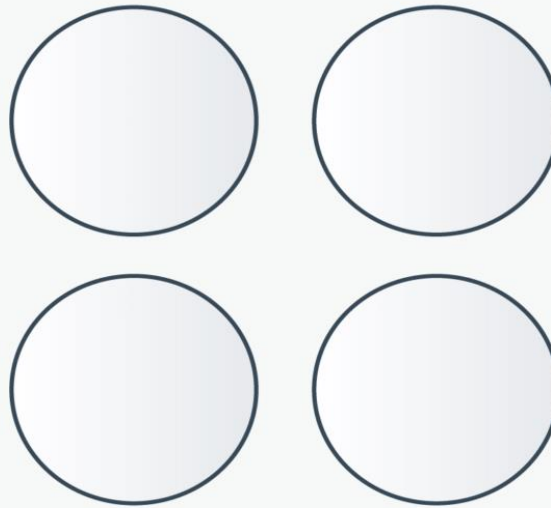


# Monolithic vs. SOA vs. Microservices



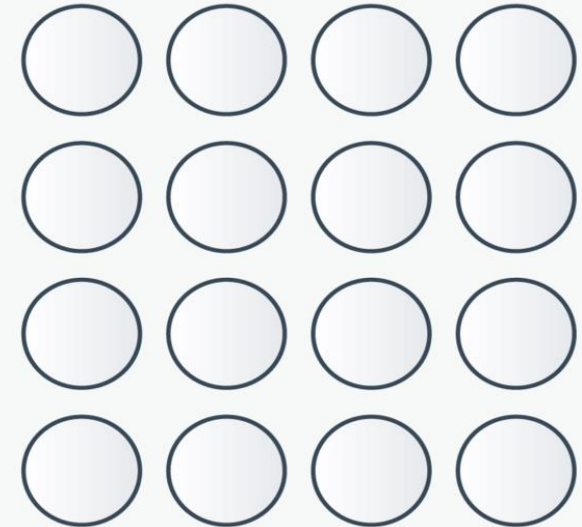
**Monolithic**

**Single Unit**



**SOA**

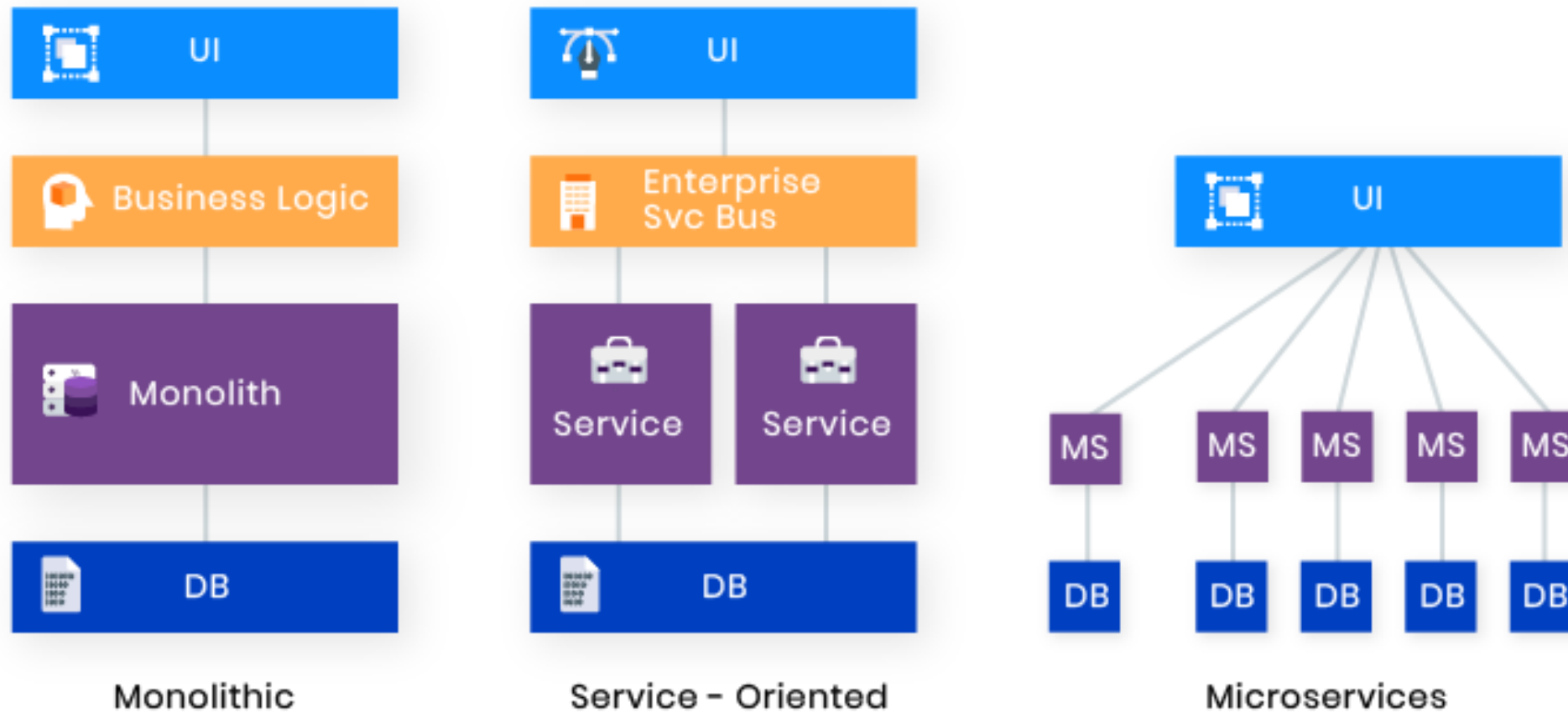
**Coarse-grained**



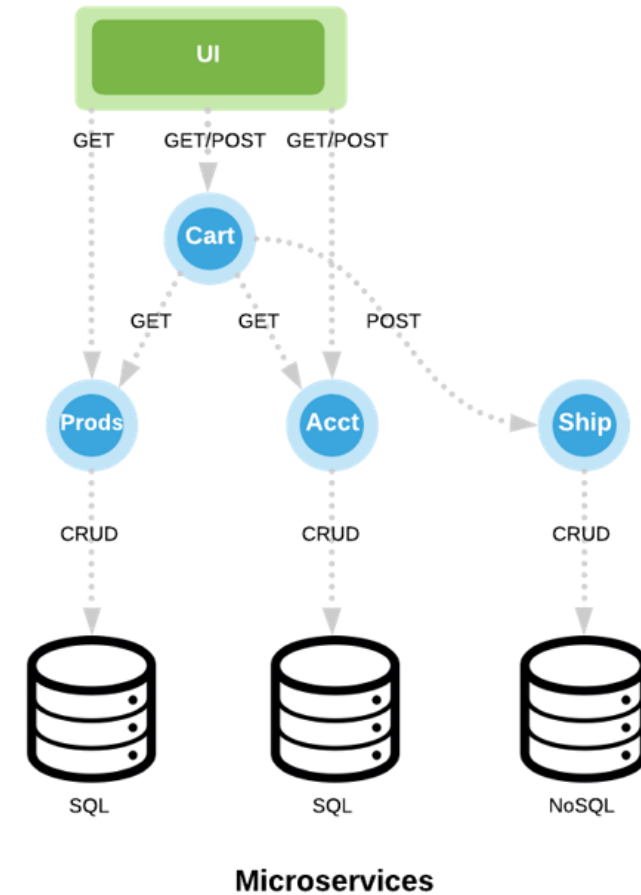
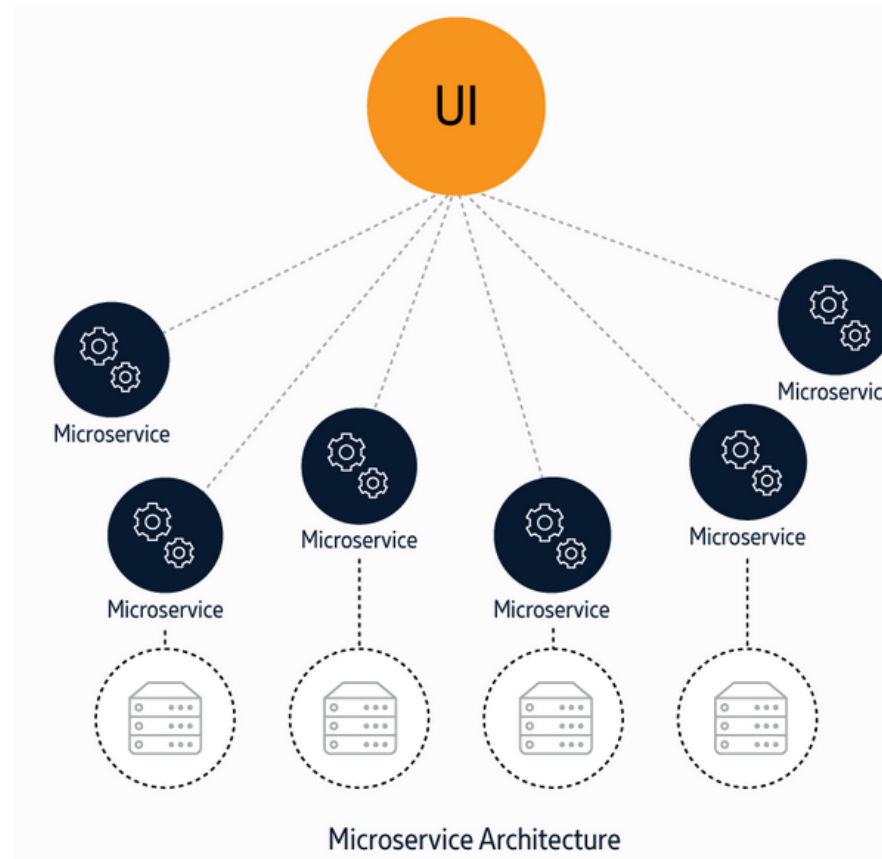
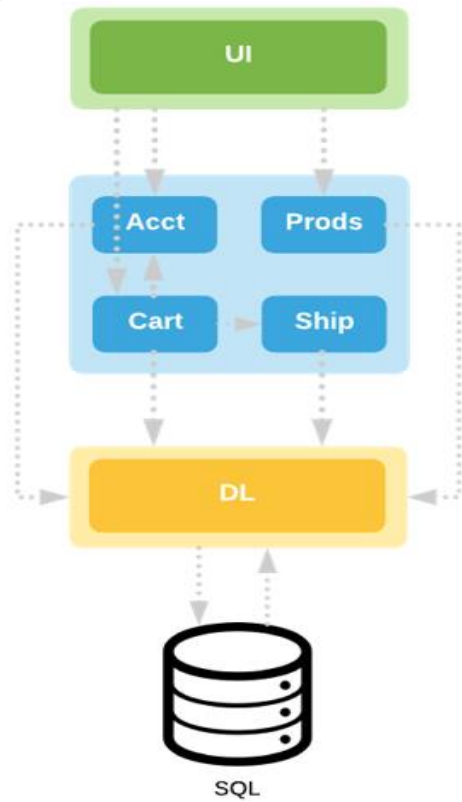
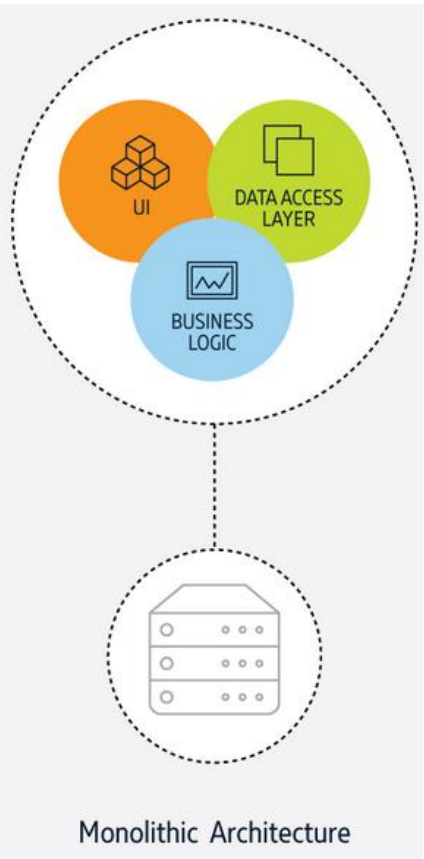
**Microservices**

**Fine-grained**

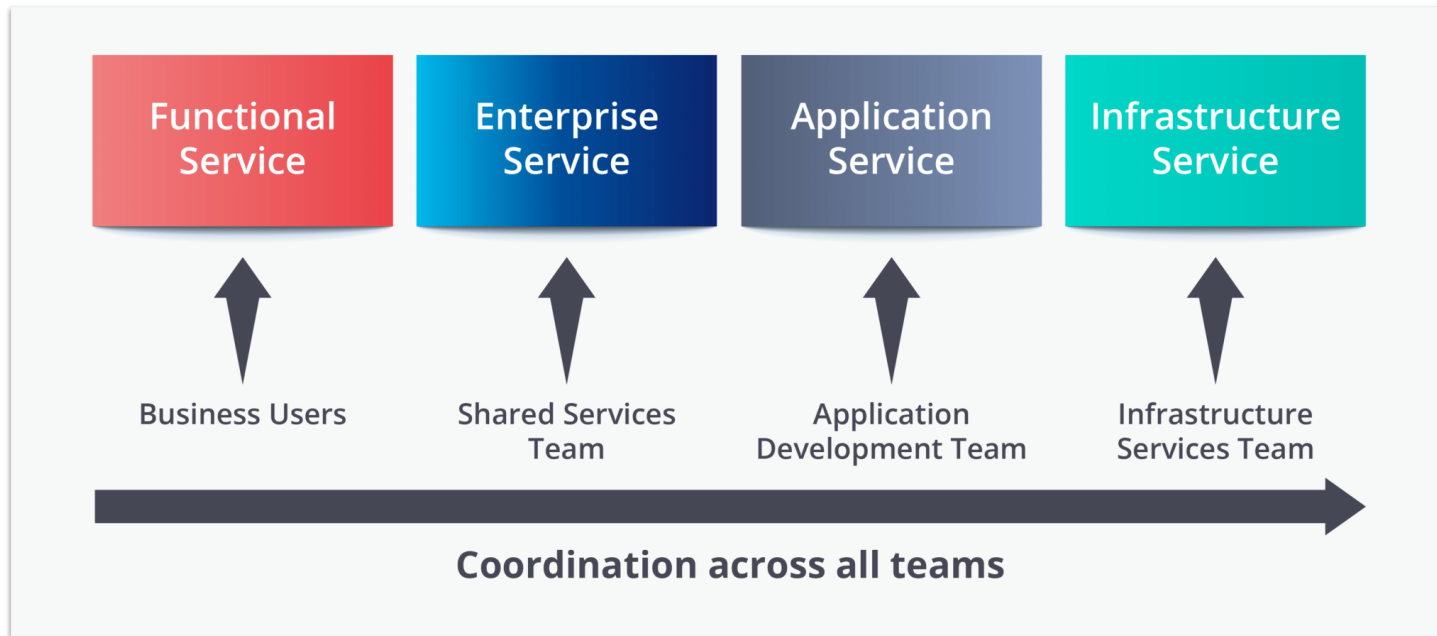
# Monolithic vs SOA vs Microservice



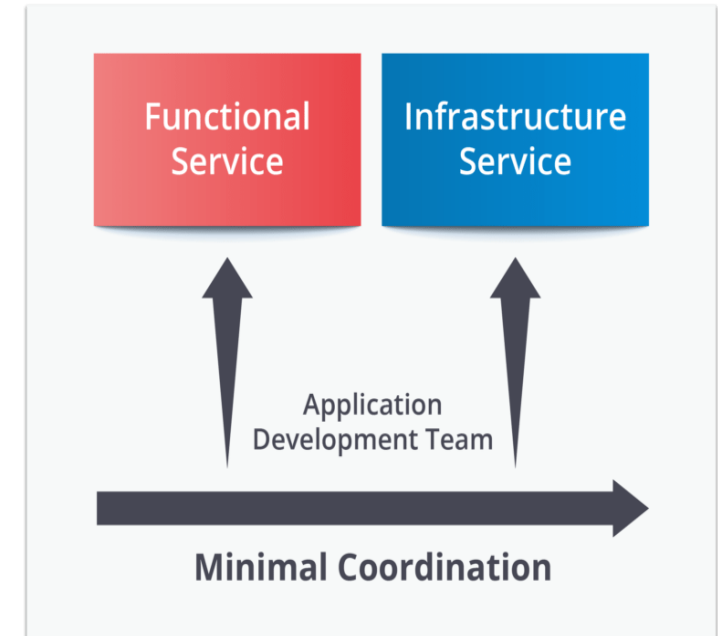
# Monolithic vs Microservice



# SOA – Microservice :: Development Team



SOA



Microservice

# Pros - Microservice

---

- 1) Aplikasi skalabale, secure dan reliable;
- 2) Setiap service berdiri sendiri;
- 3) Maintenance-nya lebih mudah;
- 4) Tidak ada hambatan dalam menggunakan teknologi baru;
- 5) Setiap tim developer dapat mengembangkan setiap services-nya tanpa mengganggu services yang lain.

# Min - Microservice

---

- 1) Komunikasi antar service adalah komplek; harus hati2 dalam handle request antar modul-modul aplikasi
- 2) Banyak service yang disediakan perlu tambahan resource;
- 3) Setingan global akan sangat sulit;
- 4) Fase deployment yang sangat menantang; (alias sulit)
- 5) Microservice baik untuk perusahaan besar, namun untuk perusahaan kecil akan sangat *complicated* & cenderung akan memperlambat penerapan bisnis proses yang sederhana

# Lingkungan Teknologi Microservice

---

- ❖ Operating System : Linux
- ❖ Docker and Kubernetes
- ❖ REST
- ❖ REDIS
- ❖ Prometheus
- ❖ Consul

<https://www.educative.io/blog/top-technologies-microservices-architecture>

# Popular Java – Python - PHP Microservice Framework

---

## Java

- ❖ Spring & Spring Boot
- ❖ Dropwizard
- ❖ Eclipse MicroProfile
- ❖ Micronaut

## Python

- ❖ Flask
- ❖ Django
- ❖ Pyramid
- ❖ Falcon
- ❖ Eve
- ❖ Cherrypy
- ❖ Bottle
- ❖ Tornado

## PHP

- ❖ Lumen
- ❖ Guzzle
- ❖ Slim
- ❖ LeafPHP
- ❖ Swift
- ❖ Ubiquity
- ❖ Fat Free Framework



<https://www.jrebel.com/blog/popular-java-microservices-frameworks>

<https://nordicapis.com/8-open-source-frameworks-for-building-apis-in-python/>  
<https://rapidapi.com/blog/php-api-frameworks/>



# Tools support microservice

---

- ❖ Documentation :

- ❖ <https://redoc.ly>
- ❖ <https://swagger.io/tools/swagger-ui/> ,
- ❖ <http://dapperdox.io/>

- ❖ Testing :

- ❖ <https://www.postman.com/>
- ❖ <https://jmeter.apache.org/> ,
- ❖ <https://www.katalon.com/>

- ❖ Security : OAUTH 2.0 , JWT , API Keys