

# 마이크로서비스개념과 특성

## (마이크로서비스 vs 모노리스)

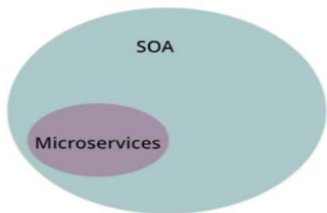
SK(주) C&C 한정현

# SOA , MSA

개체지향 문법 설계  
Component 기반 개발

- 하나의 어플리케이션을 여러 개의 서비스 집합으로 구성
- MSA는 SOA의 연장선, SOA를 제대로 수행하는 것

Service Oriented Architecture



구분	Msa	Soa
사상	서비스지향	서비스지향
오너쉽	조직(팀)단위 자율성 부여	조직(팀)간 협업
크기	Soa대비 작음	Msa대비 큼
공유정보	서비스간독립	서비스공유
공유방식	Api	서비스공유를 위한 미들웨어
서비스통신방식	Restful api등 ↳ 구현이 쉽다	SOAP,WSDL,UDDI,ESB
저장소	서비스 저장소 분리하지 않음	서비스 별 저장소 분

참조 : 박성훈(2018), 자바기반의 마이크로서비스 이해와 아키텍처 구축하기, pp.22., 제이펍출판

↳ 서비스 분리의 어려움

# 1

## Martin Fowler를 통해 본 Microservice

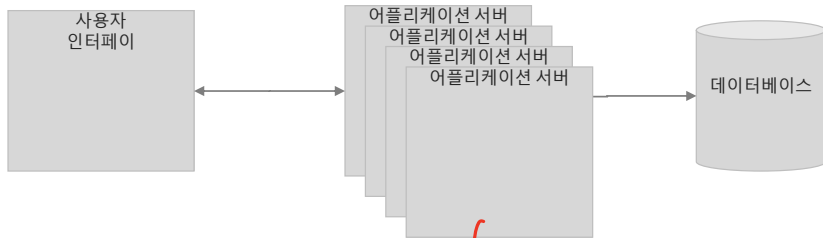
---

## 2

# 일체식 (monolithic) Application 설계

- 사용자 인터페이스와 데이터베이스, 그리고 서버 쪽 Application 세 개로 구성
- 서버 쪽 Application이 일체 (monolith), 즉 논리적인 단일 실행체, 아무리 작은 변화에도 새로운 버전을 전체 빌드해야 함. <sup>버전까지</sup> 단일 프로세스에서 실행
- 필요한 특정 기능만 확장 (scale-out) 할 수 없고, 전체 애플리케이션을 동시에 확장해야 함

Load balancer 사용

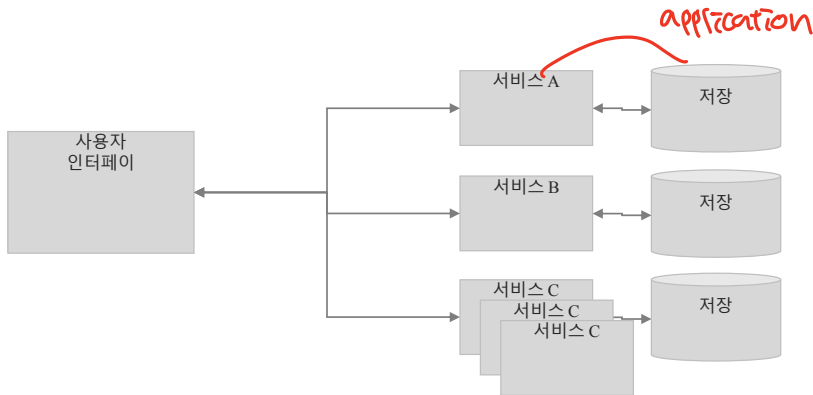


→ 논리적인 단일 실행체

## 3

# 마이크로서비스 설계

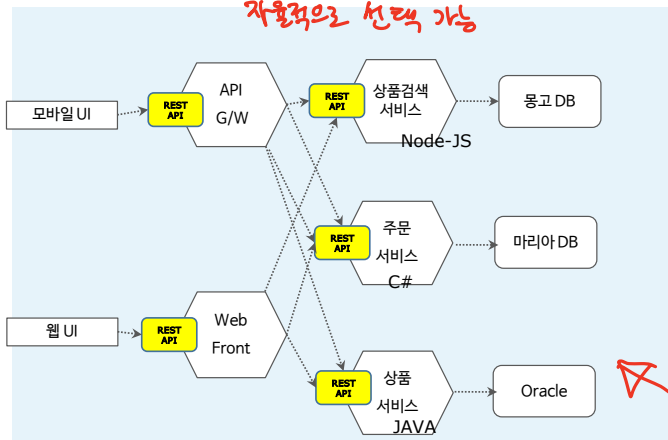
- 마이크로서비스는 확고한 모듈 경계를 가지며,
- 특정 기능 별로, 독립적으로 배포, 확장(scale-out)할 수 있고,
- 서로 다른 언어로 개발되는 것도 허용됨  
→ 각 서비스는 서로 다른 팀이 관리를 할 수 있음.



## 4

# Microservices 정의

Polyglot: Interface 만족시 서비스 간의 기술은  
자유롭게 선택 가능



- 여러 개의 작은 서비스 집합으로 개발하는 접근방법
- 각 서비스는 개별 프로세스에서 실행
- HTTP 자원 API와 같은 가벼운 수단을 사용하여 통신
- 서비스는 Biz 기능 단위로 구성
- 자동화된 배포 방식을 이용/ 독립 배포.
- 서비스들에 대한 중앙 집중적인 관리는 최소화,
- 각 서비스는 서로 다른 언어와 데이터 저장기술을 사용할 수 있음.