


# 마이크로서비스개념과 특성 (MSA특징-2)

SK(주) C&C 한정현

## 5

# MSA 특징 - 분권 데이터 관리

- 단일 데이터베이스를 유지하는 방식은 벤더의 라이선스 Model과 데이터베이스의 기능 확장에 그 뿌리를 두고 있음
- Microservice는 “Polyglot Persistence” 접근방법을 선택하며, 서비스 별로 데이터베이스를 갖도록 설계

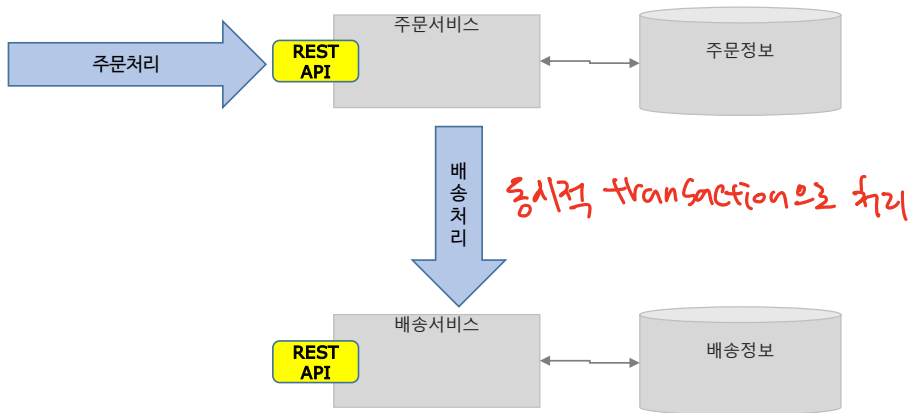
- 
1. 각 저장소 분산 필수
  2. 다른 서비스 저장소 호출 불가능
  3. API만 통해 접근 가능



데이터 일관성 문제 발생

## 5

# MSA 특징 - 분권 데이터 관리

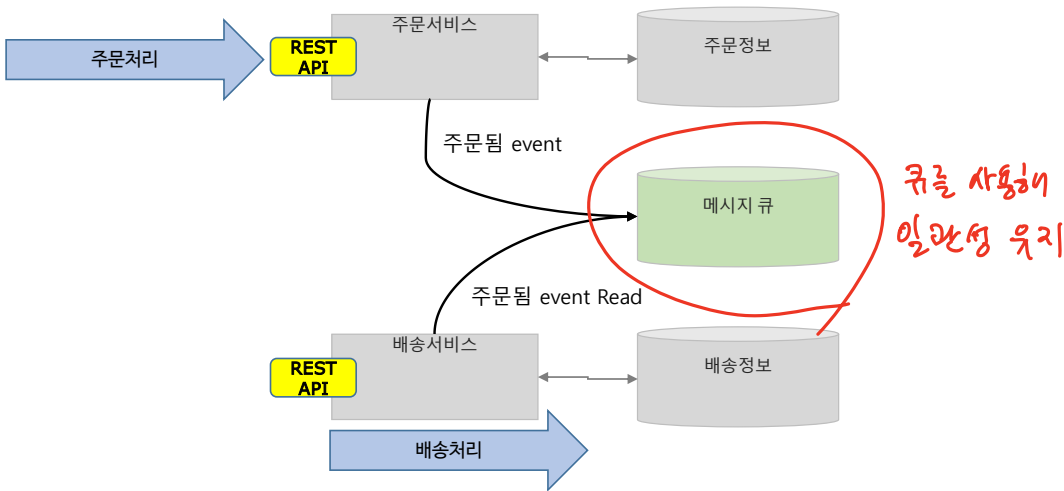


## 5

## MSA 특징 - 분권 데이터 관리

## Eventual Transaction (결과적 일관성)

- 데이터 일관성(consistency)를 유지하기 위해 두 서비스 간의 **트랜잭션이 아닌 협업을 강조**
- 데이터 일관성 목표가 좀 늦게 달성하더라도 원래대로 돌려놓는 시나리오가 Biz 개념에 적합



## 6

# MSA 특징 - 똑똑한 끝지점 단순한 파이프

- 통신을 위해서 ESB 등과 같은 똑똑한 제품을 사용해 옴. ← 라우팅, 변환, 비즈룰, 조율
- Microservice는 커뮤니티는 “**smart endpoints and dumb pipes**” 방식 선호
- Domain Logic은 서비스 속에 **high cohesion** 되어야 하고, 연결은 **loose coupling** 되어야 함.
- 따라서 **REST**와 같은 단순한 도구를 Webservice, BPEL과 같은 복잡한 도구보다 선호
- HTTP 요청/응답을 자원API와 함께 사용하거나 **경량 메시지 버스**(Rabbit MQ, ZeroMQ)를 주로 사용
- 고운 입자(fine-grained) 인터페이스를 **거친 입자(coarse-grained)**로 변경하여 사용  
*작은 흐름*                      *거친 흐름*

## 7

## MSA 특징 - 실패를 위한 설계

- 서비스는 언제든지 실패할 수 있으며, 실패해서 더 이상 진행할 수 없을 때, 자연스럽게 대응할 수 있도록 설계해야 함
- 제품화 단계에 다양한 실패에 대비하여 자동으로 테스트할 수 있는 환경을 마련해야 함.
- 기술 요소와 Biz 내용에 대한 실시간 모니터링 체계를 갖추어야 함.
- 이러한 설계는 애플리케이션이 긴급 상황에 빠르고 유연하게 대응할 수 있도록 함.

ex) 서킷 브레이크 패턴

## 8

# 아키텍처, 프로세스, 조직문화

- 점진 반복적인 개발프로세스
- 유연하고 자동화된 개발환경 *infra*
- 어플리케이션 아키텍처
- 자율적인 조직 문화

