

MSA와 Monolithic

장정훈

[담당하시게 될 업무를 소개합니다.]

- 쏘카를 이용하는 고객들을 위한 서비스 개발
- [가입 → 예약 → 주행/제어 → 반납 → 결제]에 이르는
전체 서비스 라이프 사이클 관리 및 외부 서비스 연동
- 최신 기술의 이해와 이를 활용한 확장성 있는 기능 개발 및
구현
- Legacy 서비스를 마이크로 서비스 구조로 개선
- 모빌리티 도메인의 컨텍스트를 이해하고 코드로 구현

*'쏘카가 일하는 방식'을 소개합니다.

[<https://blog.socar.kr/10391?category=724974>]

MSA란?

- MicroService Architecture의 약어
- 하나의 큰 어플리케이션을 여러개의 작은 어플리케이션으로 쪼개어 변경과 조합이 가능하도록 만든 아키텍처
- 마이크로서비스는 완전히 독립적으로 배포가 가능하고, 다른 기술 스택(개발 언어, 데이터베이스 등)이 사용 가능한 단일 사업 영역에 초점을 둔다.

Monolithic Architecture

MONOLITHIC ARCHITECTURE



Monolithic Architecture

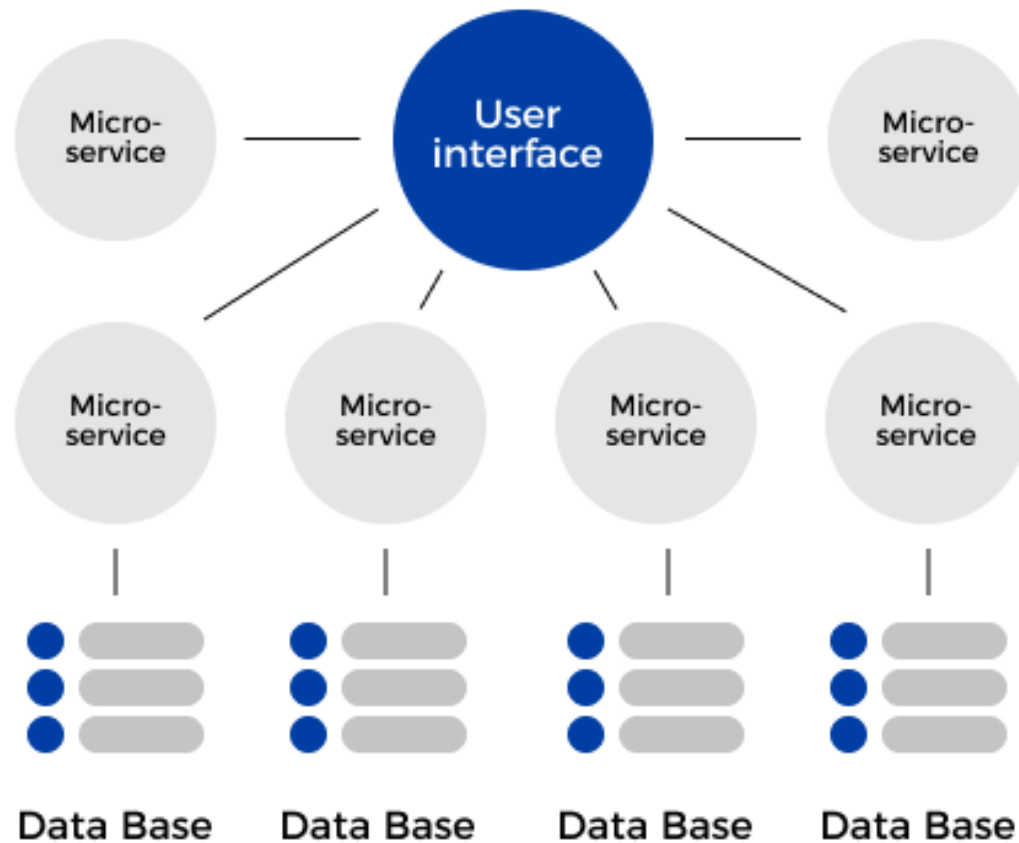
- Monolithic Architecture란, 소프트웨어의 모든 구성요소가 한 프로젝트에 통합 되어있는 형태.
- 아직까지는 많은 소프트웨어가 Monolithic 형태로 구현되어 있다.
- 소규모 프로젝트에는 Monolithic Architecture가 훨씬 합리적.
 - 간단한 Architecture이고, 유지보수가 용이하기 때문.

Monolithic Architecture

- 일정 규모 이상의 서비스에서는 한계가 뚜렷해진다.
- 부분 장애가 전체 서비스의 장애로 확대될 수 있다.
- 서비스를 부분적으로 스케일아웃(scale-out)하기가 힘들다.
- 서비스의 변경이 어렵고, 수정 시 장애의 영향도 파악하기 힘들다.
- 한 언어에 종속적일 수 있다.

MSA

MICROSERVICE ARCHITECTURE



MSA 등장 배경

- 앞의 Monolithic Architecture의 문제점들을 보완하기 위해 MSA가 등장하게 되었다.
- 기존의 특정한 물리적인 서버에 서비스를 올리던 on-premise 서버 기반의 MA에서 이제는 클라우드 환경을 이용하여 서버를 구성하는 MSA로 많은 서비스들이 전환되고 있다.

MSA 등장 배경

- MSA 이전에도 CBD, SOA 등 Monolith를 논리/물리적으로 구조화 하기 위한 노력들이 있었다.
- MSA는 큰 의미에서 SOA의 부분집합으로 여겨지고 있다.
 - SOA : 대규모 컴퓨터 시스템을 구축할 때의 개념으로, 업무상 일 처리에 해당하는 소프트웨어 기능을 서비스로 판단하고 그 서비스를 네트워크상에 연동하여 시스템 전체를 구축해 가는 방법론.

MSA의 조건

- 제대로 설계 된 마이크로 서비스는 하나의 비즈니스 범위에 맞춰 만들어지므로 하나의 기능만 수행한다.
- 어플리케이션은 항상 기술 중립적 프로토콜을 사용해 통신하므로 서비스 구현 기술과는 무관하다.
 - 즉, MSA기반 어플리케이션은 다양한 언어와 기술로 구축할 수 있다.
- REST 등 가벼운 통신 아키텍처, 또는 Kafka 등을 이용한 message stream을 주로 사용한다.

MSA 장점

- 서비스가 커지면서 생겼던 MA의 문제점을 어느정도 보완.
- 배포 관점
 - 서비스 별 개별 배포 가능(배포 시 전체 서비스 중단이 없음)
 - 요구 사항을 신속하게 반영하여 빠르게 배포할 수 있음.
- 확장 관점
 - 특정 서비스에 대한 확장성이 용이.
 - 클라우드 사용에 적합한 구조
- 장애 관점
 - 장애가 전체 서비스로 확장될 가능성이 적음.
 - 부분적 장애에 대한 격리가 수월.

MSA의 단점

- 전체 서비스가 커짐에 따라 그 복잡도가 기하급수적으로 늘어날 수 있다.
- 성능: 서비스 간 호출 시 API를 사용하기 때문에, 통신 비용이나, 대기 시간이 그만큼 늘어나게 된다.
- 테스트/트랜잭션: 서비스가 분리되어 있기 때문에 테스트와 트랜잭션의 복잡도가 증가하고, 많은 자원을 필요로 함.
- 데이터 관리: 데이터가 여러 서비스에 걸쳐 분산되기 때문에 한 번에 조회하기 어렵고, 데이터 정합성 또한 관리하기 어렵다.

