

Monolithic & MicroService

맛보기..

Mash up 11th 이소연

Monolithic Architecture

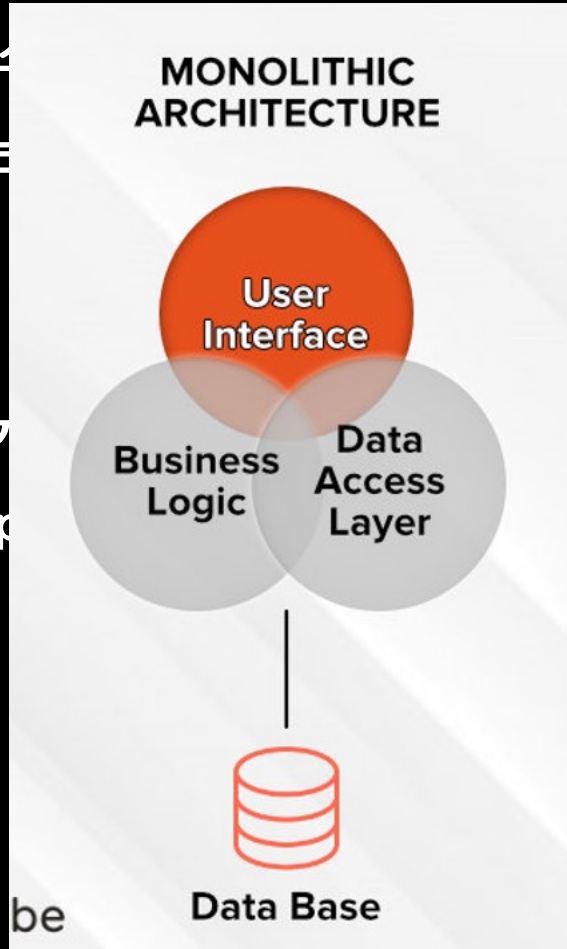
: 데이터 입력, 출력, 프로세싱, 에러 처리, UI와 같은 구분된 기능들이 분리되어 있지 않고 혼합된 구조의 시스템

: 서비스 아키텍처를 구성할 때 모든 서비스
하나의 패키지에 담아 빌드하고 배포하는

DB, UI 등은 논리적으로 모듈화 될 수 있지만

"우리가
Rep

방법 "
관리



be

Data Base

Monolithic Architecture

장점

- 빠르게 개발 가능
- 단일 프로젝트이기 때문에 시스템 설계가 수월
- 운영 복잡도가 낮음
- 버전 관리 수월
- 기술의 단일화
- 단일 코드베이스로 프로세스 내부 통신이 빠름

단점

사업의 규모가 커지고 필요한 기능 늘어나면..

- 성능 문제
- 확장성 문제
- 유지보수 힘들어짐
- 특정 기능과 코드에 관련된 개발자의 책임 소재 불분명
- 사용하던 기술과 언어를 벗어날 수 없음
- 업그레이드 과정에서 다운타임 발생

마이크로서비스는 이러한 대규모 모듈식 모놀리식 상황에서 매우 효과적.

MicroService Architecture

: 작고 자율적으로 협업하는 서비스

: 독립적인 기능을 실행 파일 또는 프로세스 단위로 설계한 서비스

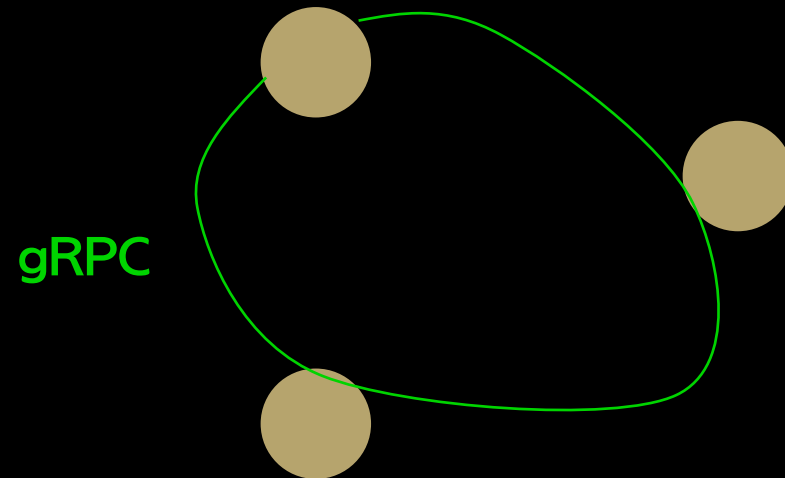
느슨한 결합

맞춤형 확장성

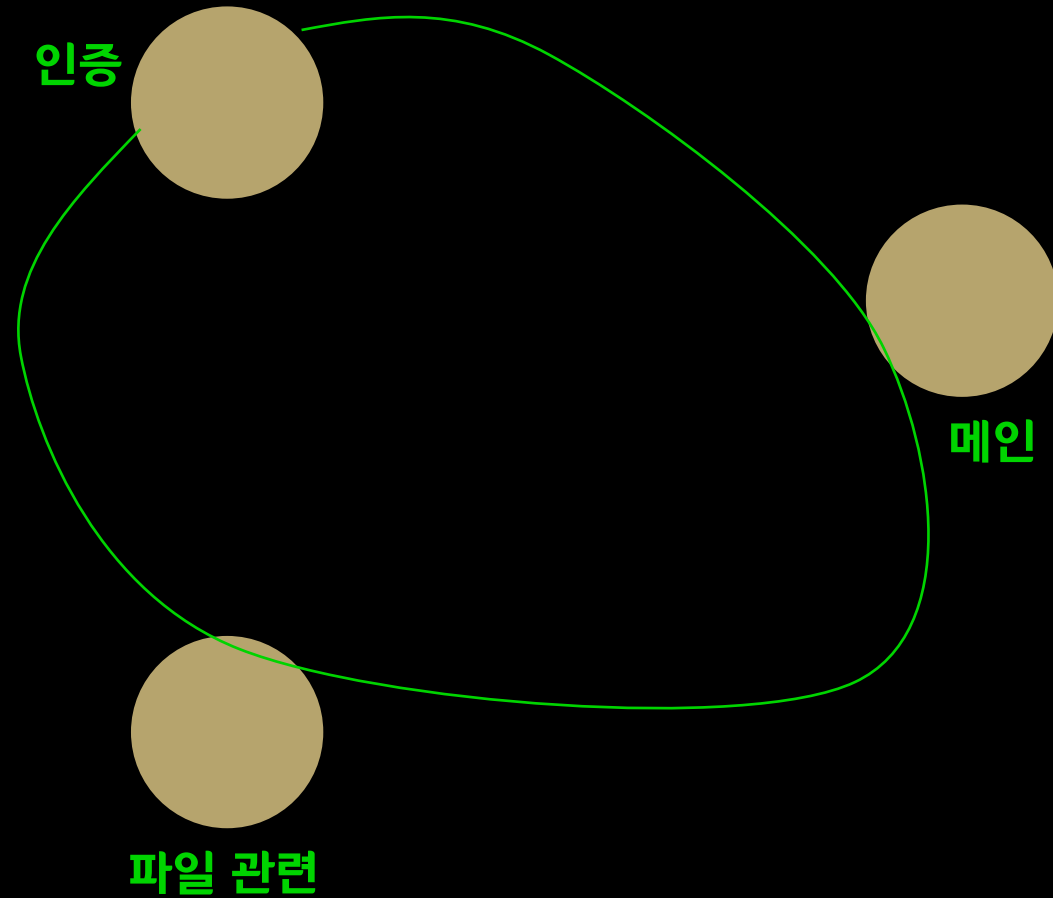
서비스 중심성

느슨한 결합

강한 응집력



MicroService Architecture



MicroService Architecture

gRPC??

다른 서비스 간 언어 제약 없이 사용할 수 있는 오픈소스 고성능 원격 프로시저 호출 프레임워크

요즘은 gRPC 말고도 Kafka를 이용한 통신도 사용한다고함.

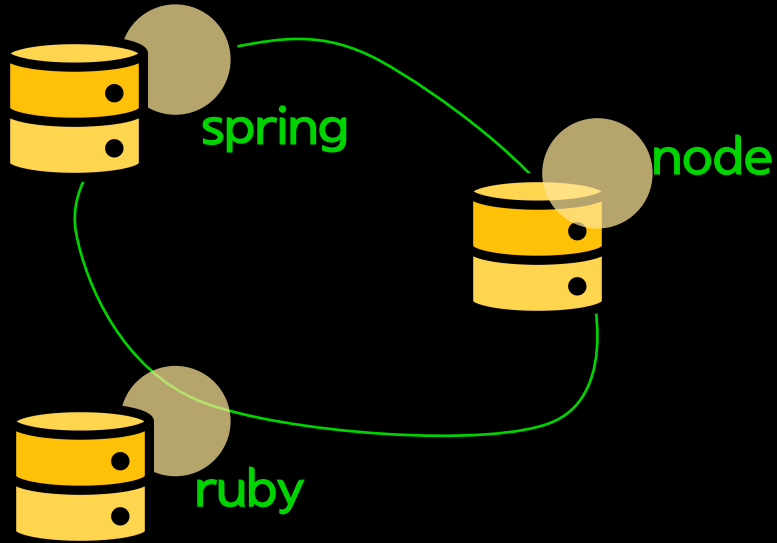
더 자세한 건 Nest 심화 발표 때 ><



MicroService Architecture

장점

- 기술 이기종성



- 회복성

하나의 시스템에 장애 발생 → 장애가 전파 x → 장애 격리

모놀리식은 한 서비스가 고장나면 모든 것이 멈추는 경우 발생

BUT...(주저리,,)

마이크로서비스도 서비스가 나누어져 있는 만큼

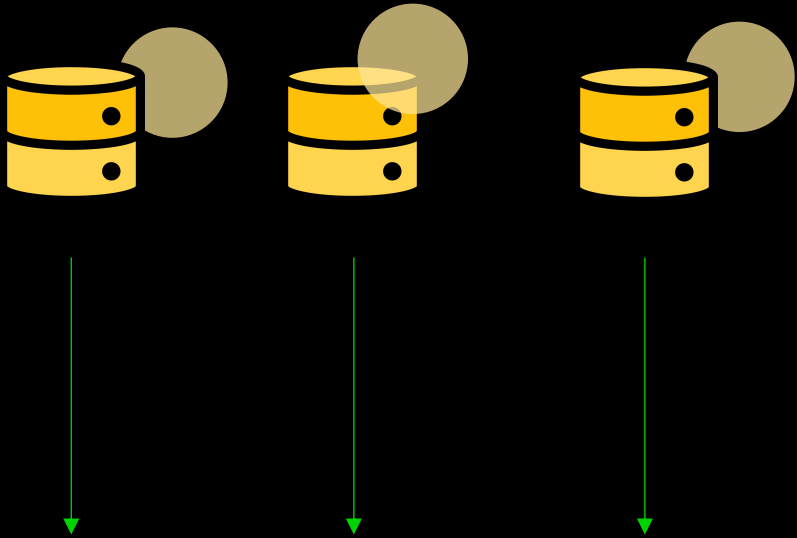
네트워크 장애가 발생할수도..

모든 서비스가 멈추지 않지만 어떠한 장애가 발생했을 때
회복이 수월할 수 있다.

MicroService Architecture

장점

- 배포 용이성



모놀리식에 비해 배포 주기 down (독립적으로 배포가능)

병렬 처리 가능

- 재사용성



다른 프로젝트 하는데 인증 서버 또 필요?

기존에 만들어진 인증 서비스를 재사용할 수 있다.

MicroService Architecture

그럼 언제 Monolithic에서 MSA로 언제 넘어가야할까?

코드 복잡도
서버 리소스

