



CHUNILINC

M

SOFTWARE😊 ARCHITECTURE

소프트웨어 아키텍처





CONTENTS LIST😊

—

- 1 소프트웨어 아키텍처
- 2 모노리틱 아키텍처
- 3 MSA



CHUNILINC

M

SOFTWARE ARCHITECTURE



01





Software Architecture

소프트웨어 아키텍처



- ▶ 소프트웨어 구성요소(컴포넌트)들과의 관계를 정의한 시스템 구조
- ▶ 구성요소(컴포넌트)들 간의 상호 관계를 설계하고 전개하기 위한 지침과 원리



MONOLITHIC ARCHITECTURE



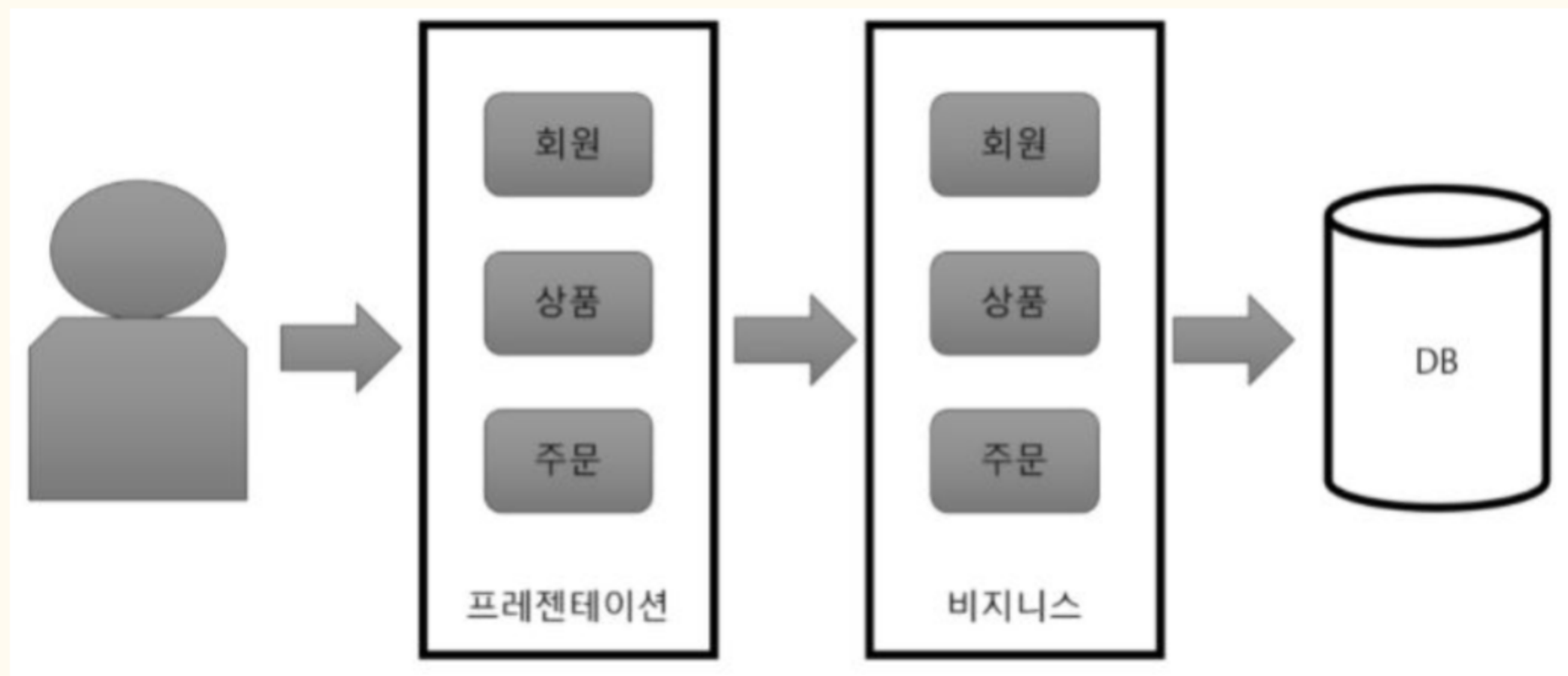
02





Monolithic Architecture

모노리틱 아키텍처



- ▶ 한 덩어리의 구조
- ▶ 장점
 - 하나의 어플리케이션만 수행하면 되서 환경설정이 간단
 - 컴포넌트들이 함수로 호출되기 때문에 성능의 제약이 덜하고 운영관리가 용이
- ▶ 시스템이 커지면서 문제가 발생
- ▶ 단점
 - 빌드 및 테스트 시간이 길어진다
 - 선택적 확장이 불가능(한개의 컴포넌트의 사용량이 증가할때 프로젝트 전체를 확장)
 - 하나의 서비스가 죽으면 전체 서비스가 죽는다



CHUNILINC

M

**MICROSERVICE
ARCHITECTURE**

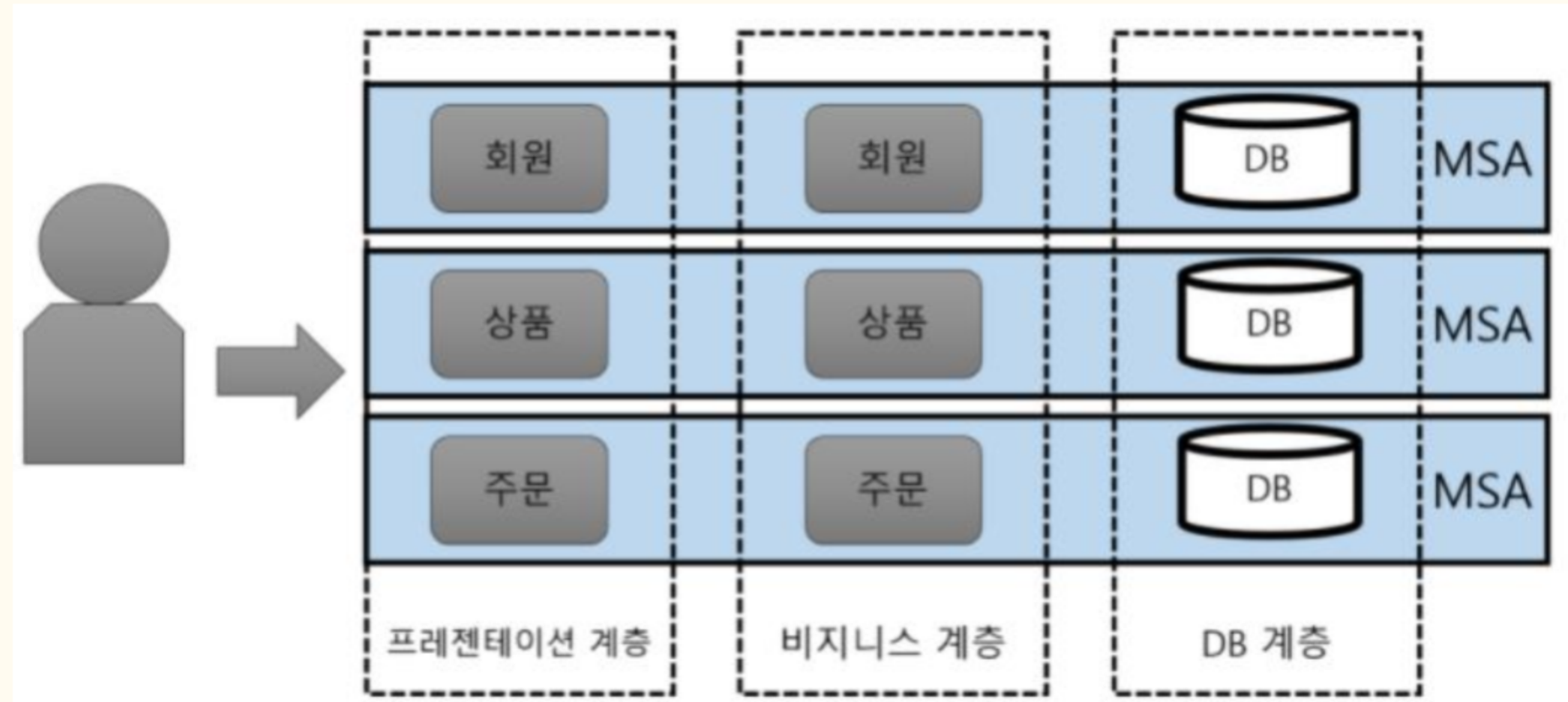


03



Microservice Architecture

마이크로서비스 아키텍처

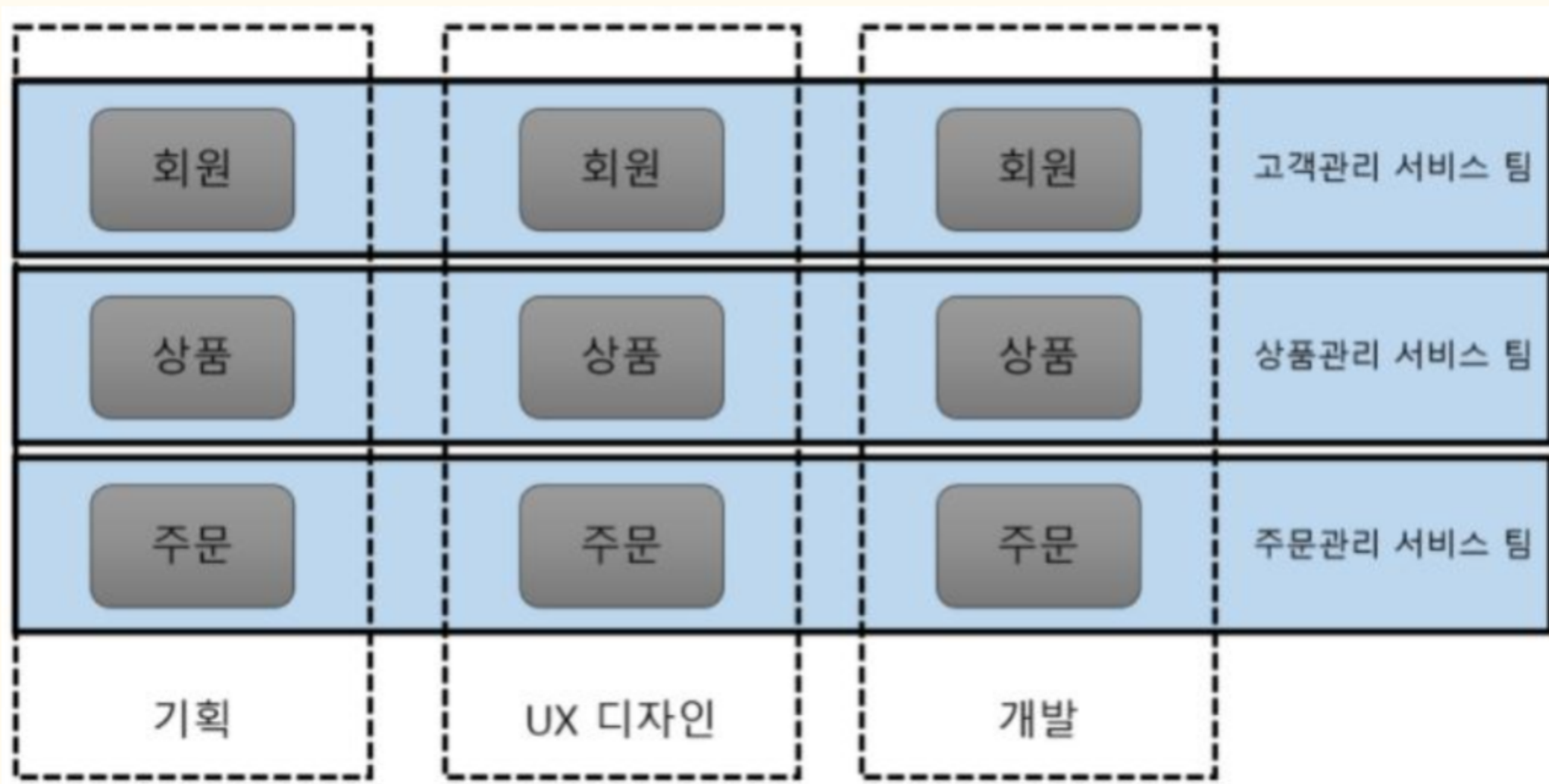


- ▶ 컴포넌트 별로 나누어 작은 서비스의 조합으로 구축
- ▶ API를 이용해서 타 서비스와 통신
- ▶ 데이터 분리 - 서비스별 DB 사용
- ▶ API Gateway - 모든 API서버를 묶어주는 역할, 라우팅, 로드밸런싱, 인증
- ▶ 장점
 - 독립된 서비스로 의존성이 없기 때문에 독립된 배포(부분적인 확장 가능)
 - 재조립을 통한 새로운 서비스 구축시 유용
- ▶ 단점
 - 서비스간 호출을 API통신을 이용하기 때문에 속도가 느리다



Team Change

팀의 변화



- ▶ 서비스마다 팀을 나눔
- ▶ 장점
 - 협업시 용이함(요청과 피드백이 빠르다)
 - 다른 팀에 대한 의존성이 낮아짐
- ▶ 단점
 - 인력관리가 어렵다
 - 팀의 역할 담당자들은 기본적인 업무 능숙도 필요



CHUNILINC

M

THANK 😊 YOU!

