SK 주식회사

KAIST

Microservice inner Architecture

(Microservice 내부 Architecture)

SK(주) C&C 한정헌

SK 주식회사

KAIST

학습 목표

- Microservice inner Architecture 의미에 대해 이해할 수 있다.
- Layered Architecture에 대해 이해할 수 있다.
- Hexagonal Architecture에 대해 이해하고, Hexagonal Architecture에 의해 Layered Architecture가 어떻게 변경되는지 알수 있다.
- Hexagonal Architecture를 반영한 Layered Architecture 각 Layer의 설계에 대해 살펴본다.

SK 주식회사

KAIST

Application Architecture와 JAVA 엔터프라이즈 발전흐름

SK(주) C&C 한정헌

SK 주식회사

KAIST

어플리케이션의 내부 구조를 정의하는 활동

Application Architecture

정의

- 일반적으로 Application 전체의 구조, 공통된 방식(매커니즘)이라고 정의
- 시스템의 Application이 공통으로 이용할 수 있는 사용자인터페이스 구조나 데이터베이스 접근방식등 시스템의 기반이 되는 부분

목표

사용자

개발자

운영자

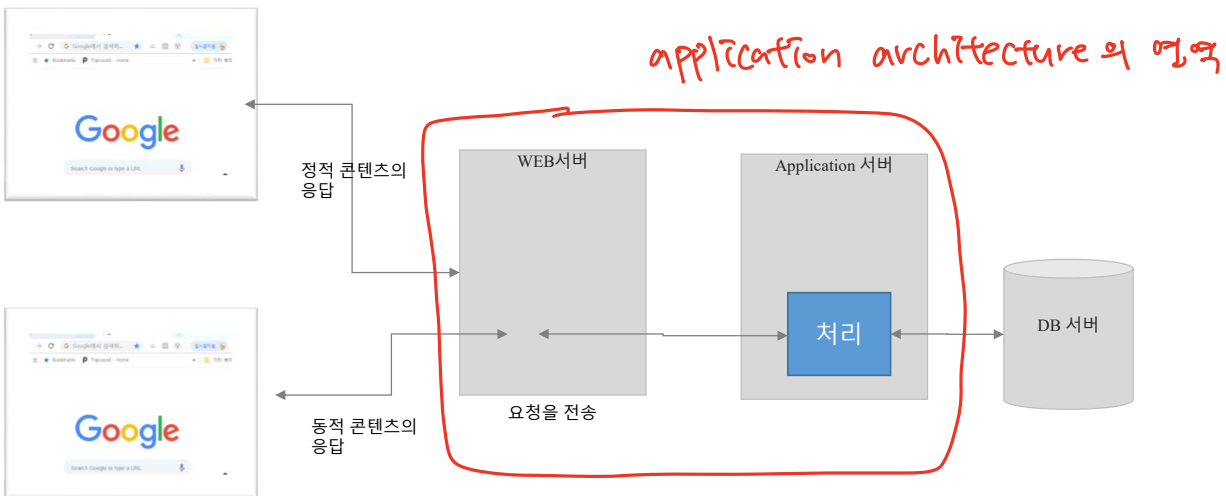
✓ 사용자의 요구(기능요구,비기능요구)

✓ 개발 효율
- 의도를 파악하고 쉽고 이해하기 쉬운 구조
- 테스트 하기 쉬운 구조

✓ 유연성
- 변경하기 쉽고 기능을 추가하기 쉬운 구조
- 미래의 환경 변화에 대응할 수 있는 구조

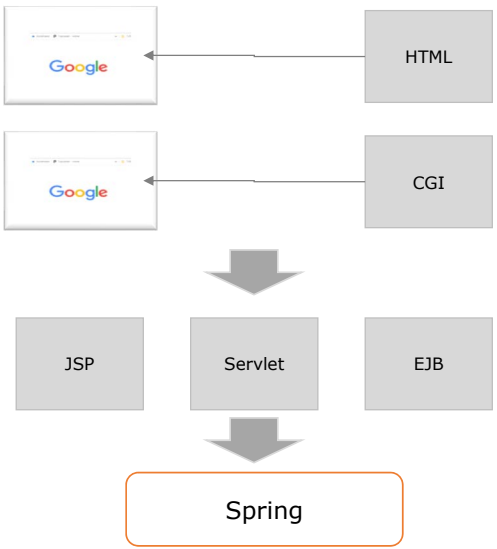
WEB Application이란?

WWW 의 호출 구조



JAVA 엔터프라이즈의 역사

JAVA 개발 Trend



1. • HTML
   정적 콘텐츠
2. • CGI
   동적 콘텐츠
   1요청 1 프로세스
   세션관리가 없음
   Perl등의 언어로 작성 → 세션 관리 기능 X
3. • 서블릿,JSP
   CGI의 문제점 해결
   JAVA언어 이용
4. • EJB 3.0 이전
   컴포넌트 모델 업계 표준
   분산 오브젝트
   화면과 비즈니스 로직을 분리
5. • Spring
   J2EE의 단점을 개선하려는 목적으로 만들어져,
   당시에는 "J2EE without EJB"라는 구호로 등장
   현재는 클라우드 시대를 주도

Spring의 등장

- J2EE는 시간이 지날수록 무거워지고 복잡해짐
- 로드 존스, 중량급인 J2EE 컨테이너를 대신할 경량 컨테이너로 DI,AOP 기능 가진 Spring 고안
- DI * AOP 컨테이너는 POJO(Plain Old Java Object)라고 부르는 컨테이너와 프레임워크등에 의존하지 않는 일반 오브젝트의 생명주기 관리나 오브젝트간의 의존관계를 해결하는 Architecture를 구현한 컨테이너
- 선언적 트랜젝션 관리를 POJO로 구현 ,DB접근은 OR매핑 프레임워크 사용
- 컨테이너에 의존하지 않은 단위테스트 쉽게 가능
- 사실상 Spring이 JAVA/JAVA EE의 표준 프레임웍
- 전자 정부 표준 프레임웍 기반 기술

출처: http://spring.io/