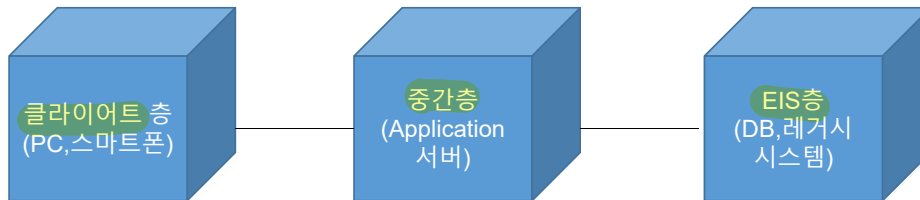


Application Architecture : Layered Architecture

(Layer형 Architecture)

SK(주) C&C 한정현

Tier와 Layer

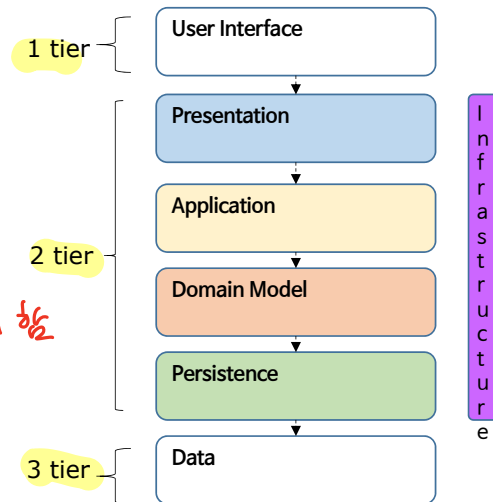


- Tier : 물리층 *물리적인 장비, 서버 컴퓨터*
 - Layer : 논리층 *Tier 내부의 논리적인 분할*
- ✓ Layer는 클라이언트 층에도 일부 있지만 . 기본적으로는 중간층에 있는 WEB Application을 논리적으로 분류한 것
 - ✓ Layered Architecture : 서로 인접한 Layer 단방향 액세스
 - ✓ 일반적인 Layer
 - Presentation Layer
 - Business Logic Layer
 - Data Access Layer

Layered Architecture

각 layer의 표준화, 모듈화

- 설계자들이 복잡한 시스템을 분리할 때 흔히 사용하는 패턴 중 하나로
- 상위는 하위를 호출하여, 다양한 서비스 이용
- 하위는 상위를 알지 못하게 구성함



1) 상위 layer가 하위 layer의 변경이 영향을 받지 않게 함

2) 어플리케이션이 쉽게 변경되거나 확장될 수 있게 함

(layer는 Tier로 묶이며 Tier 단위로 하나의 장비를 구성한다)

다양한 Layer 정의 유형

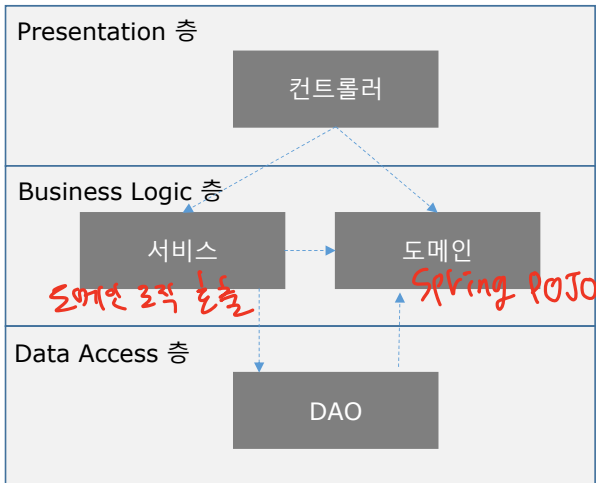
다양한 Layer 정의의 유형

자극 사항

Layer1	Layer2	Layer3	Layer4
Presentation	Presentation	Presentation	클라이언트 층에 있는 Presentation
		Application컨트롤러	서버 층의 Presentation
비즈니스	서비스	도메인	비즈니스
	도메인		
Data Access	퍼시스턴스	인티그레이션	인티그레이션
非Layer			리소스

일반적인 3Layer Layer

Spring에서 일반적으로 사용되는 구조



Layer	용어	정의
Presentat ion 층	컨트롤러	화면전환,버튼 눌렀을 때 동작제어,세션관리
Business Logic 층	서비스	유스케이스로 표현되는 특정 업무나 특정부서 처리 통합,Stateless클래스
	도메인	서비스로부터 비즈니스를 실행하는데 반드시 사용하는 고객과 주문 같은 클래스의 집합, 자신이 무엇인지를 나타내는 값과 그 값을 이용한 처리실현
Data Access층	DAO	데이터 저장소 처리

Layer 간의 호출 원칙

- Layer 사이의 낮은 결합도를 깨뜨리지 않도록 설계 필요
- 당연히 Layer 사이의 호출은 인터페이스를 통해 이루어져야 함.
- 인터페이스를 하나 더 만드는 것이 번거롭다고 그냥 클래스를 이용해서는 안됨.
- 인터페이스를 사용한다는 것은 Layer의 경계를 넘어서 들어오는 요청을 명확히 정의하겠다는 의미
- 구현에 의존하지 않음으로써 오브젝트 사이의 약한 결합을 유지(쉽게 확장, 변경, 개발 효율 ↑)

