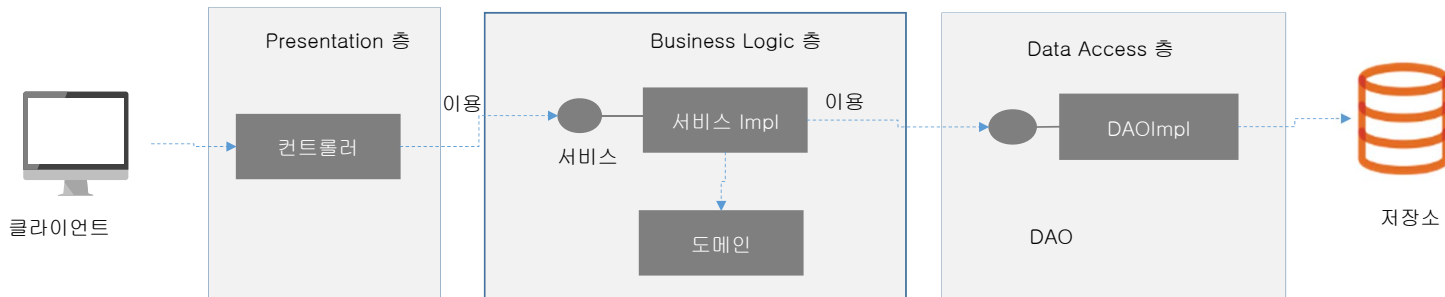


Application Architecture : Hexagonal Architecture

SK(주) C&C 한정현

Layer간의 의존성

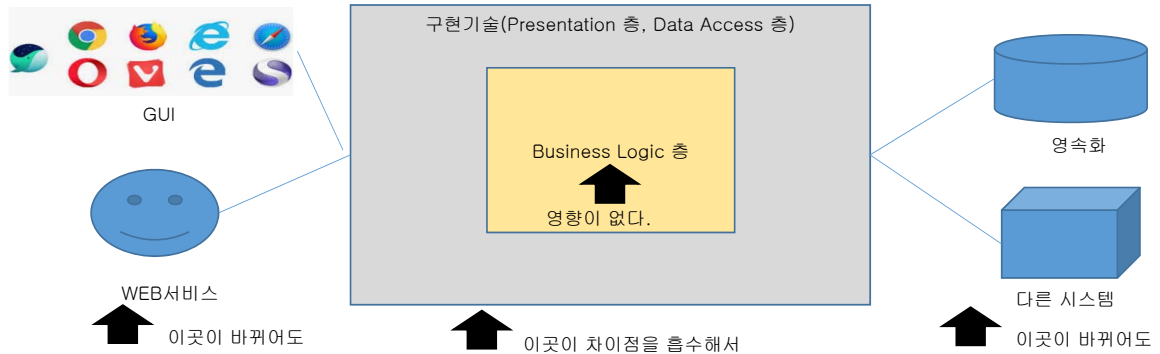
전향적인 아키텍처



↑
핵심 비즈니스 기능은
한번 정의되면 변경 불가능

헝사 고널

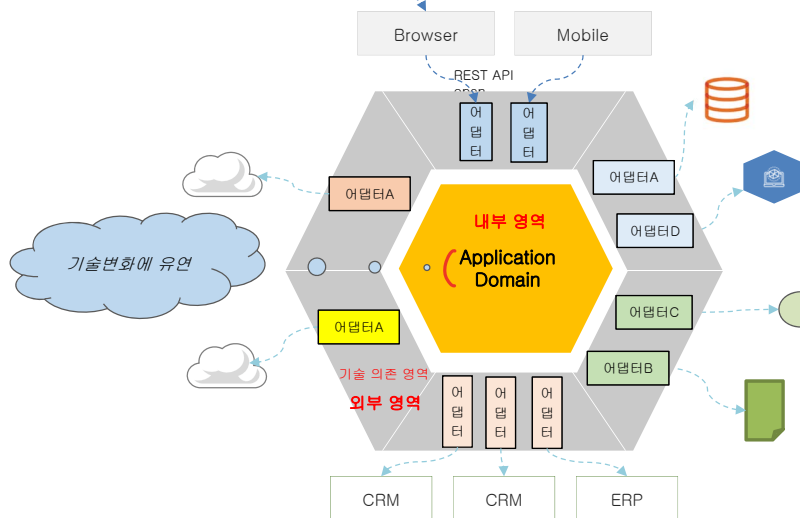
- WEB Application Layer는 크게 비즈니스와 관련 된 부분과 Business Logic의 결과를 처리하는 두 부분으로 나눌 수 있음
- Business Logic이야말로 제일 중요, Business Logic의 결과를 어떻게 다룰지 구현하는 기술(브라우저에 표현하는 기술,DB에 저장하는 기술)이 Business Logic에 영향을 미치지 않는 것이 좋은 설계



→ Port Adapter Pattern

hexagonal architecture

Alistair Cockburn이 정의한 다양한 이질적 Client가 동등한 지위에서 시스템과 상호작용하도록 하는 대칭성을 만들어주는 스타일의 Architecture



Hexagonal Architecture

외부영역

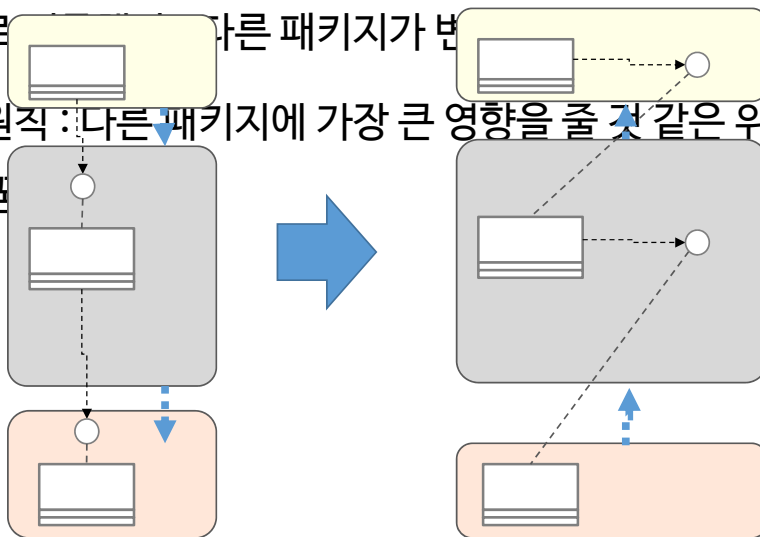
- 1) 이질적인 클라이언트가 압력을 보낼 수 있음
- 2) 영속 데이터를 가져오거나 내부의 결과를 DB에 저장
- 3) 다른 위치로 전송하는 메시지 메커니즘 제공
- 4) 변경되는 타입마다 이에 특화된 어댑터 존재
- 5) API를 통해 내부 영역과 이어짐

7억채지향 5가지 원칙 중

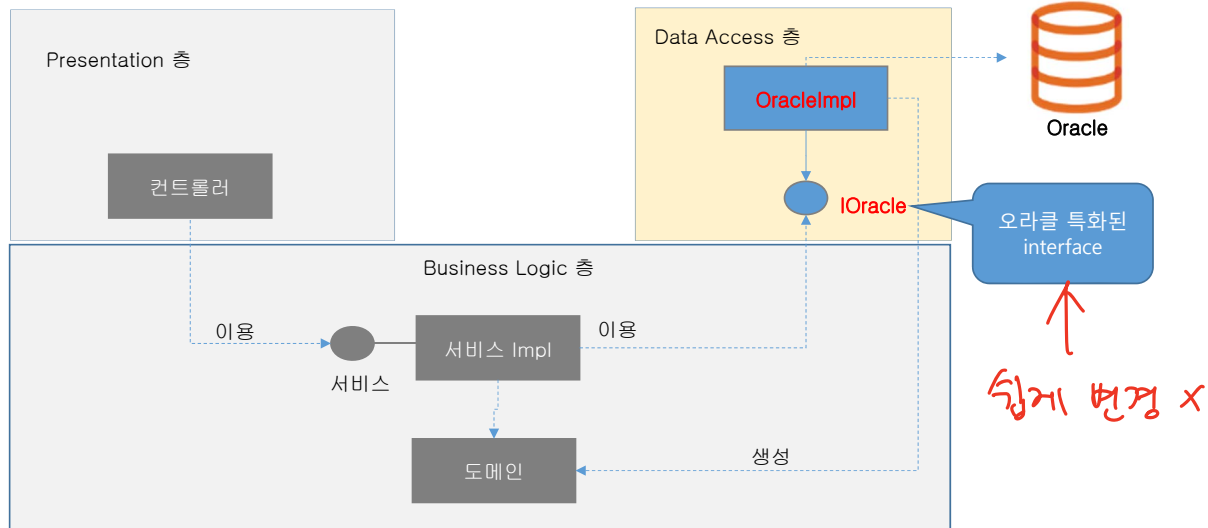
DIP (Dependency Inversion Principle)

의존관계의 역전 원칙 ➔ 의존하려면 잘 변경되지 않는 부분에 의존하라

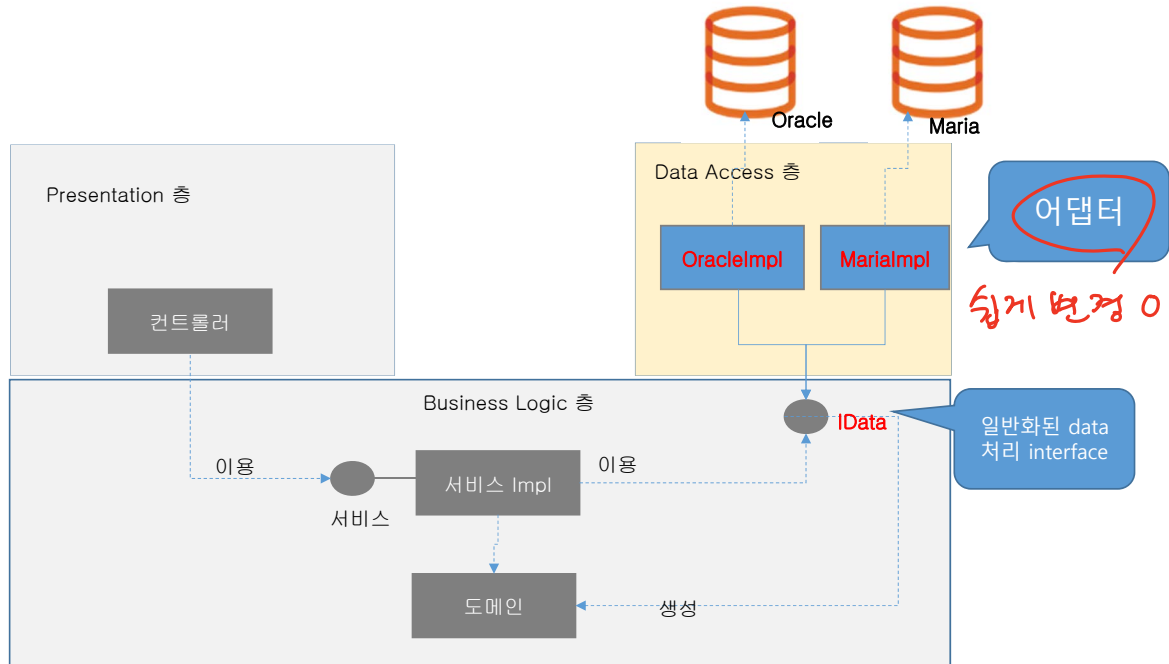
- 안정된 방향으로 다른 패키지가 변하지 않는다.
- 의존관계역전 원칙 : 다른 패키지에 가장 큰 영향을 줄 것 같은 위치에 있는 패키지의 방향을 바꾼다



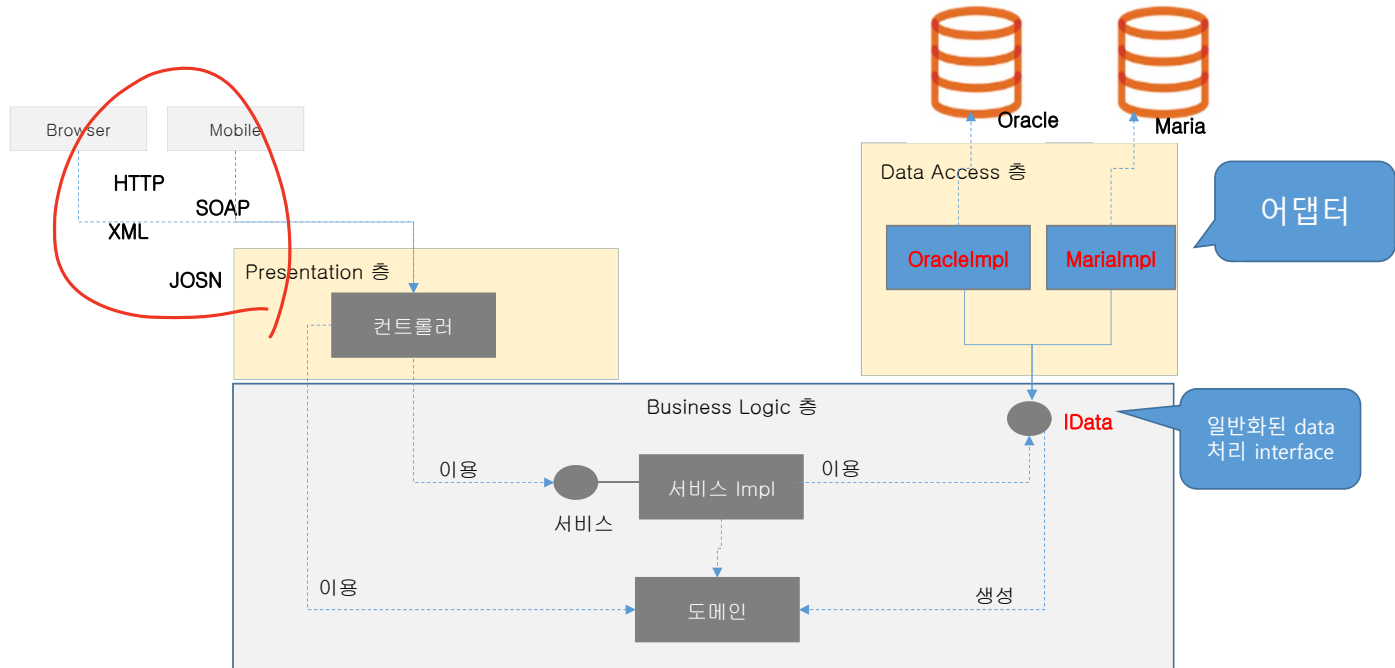
Layered Architecture의 변경



Layered Architecture의 변경



Layered Architecture의 변경



Layered Architecture의 변경

