

Domain Driven Design 이란?

1210 이서준

첫 인상 무슨 말인지 모르겠다

...

DDD = 도메인이 중심이 되는 개발방식.

도메인이란?

사건이 발생하는 집합!

급식실을 예시로 들면

급식을 먹는 학생도메인 (Eat Domain)

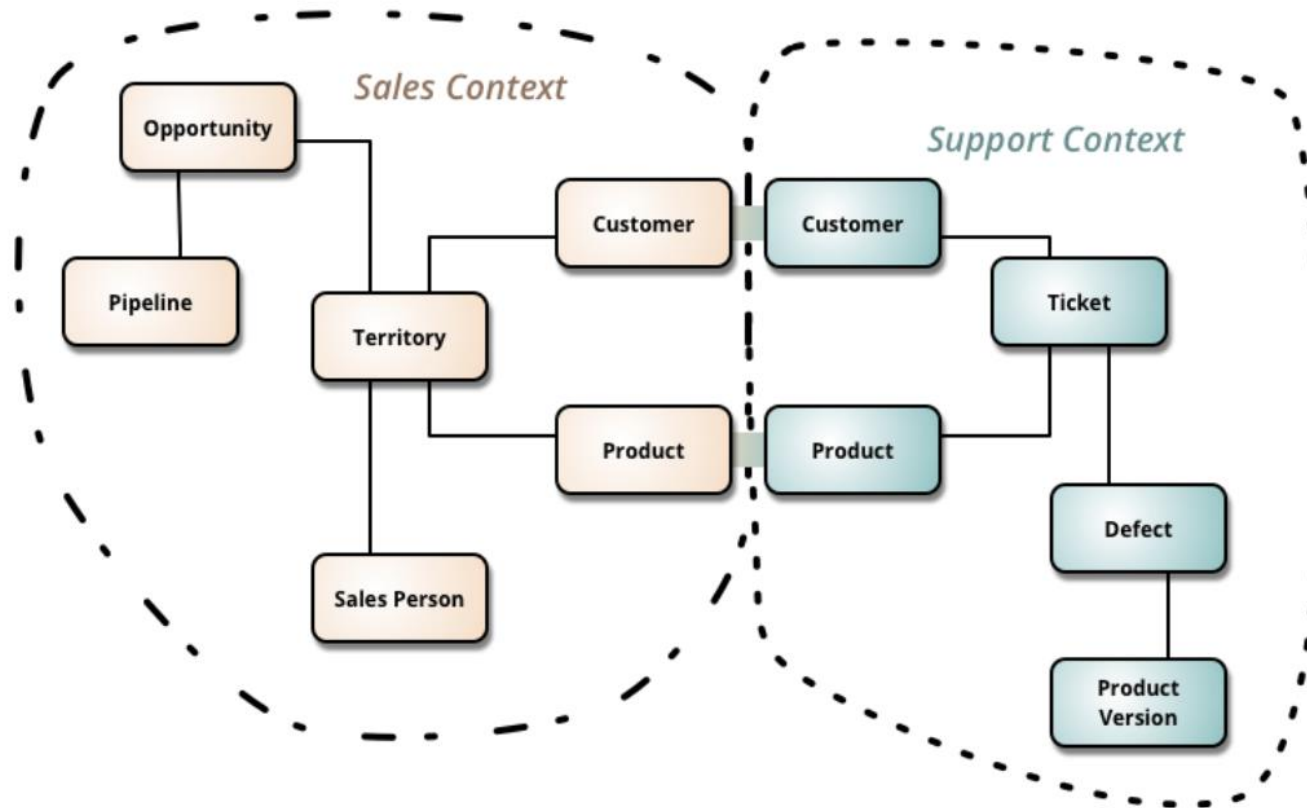
음식의 질을 관리하는 영양사도메인(Manage Domain)

음식을 만드는 요리사도메인(Cook Domain)

이러한 여러가지 도메인이 서로 상호작용하며, 설계하는 것이
도메인주도설계이다.

Bounded Context

또한 같은 객체가 여러 개 존재할 수 있음
(문맥에 따라 객체의 역할이 바뀜)



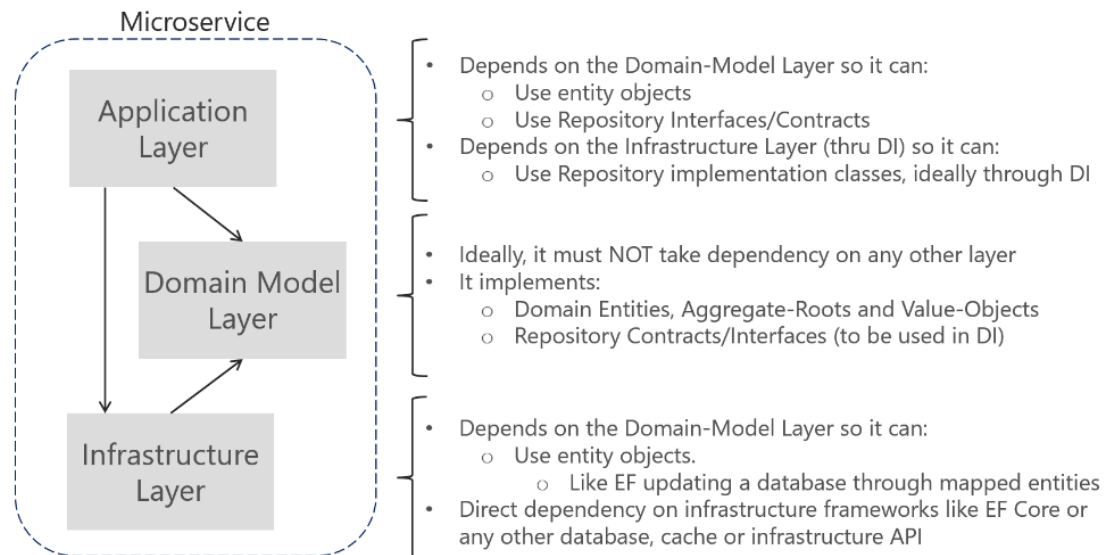
이런 관점을 서비스에 직접
적용시킨 것이 바로

마이크로 서비스.

서로 다른 도메인 영역에 영향을 끼치기
위해서는 API호출로 해야함!!

설계 방식

Dependencies between Layers in a Domain-Driven Design service



크게 3가지 레이어로 구분.

1. Application(응용) Layer : 도메인 서비스를 호출하고, 도메인에 필요한 구현을 직접 수행하지는 않으며, 도메인 영역의 객체에 이를 위임함.
2. Domain(도메인) Layer : 도메인에 필요한 데이터를 표현하고, 도메인과 관련된 구현을 담당함.
3. Infrastructure(인프라) Layer : 실제 구현을 담당, 구현 기술과 아주 밀접한 연관이 있는 레이어이다. 도메인 레이어의 구현기술과 관련된 부분은 인프라 레이어에 의존하게 됨.

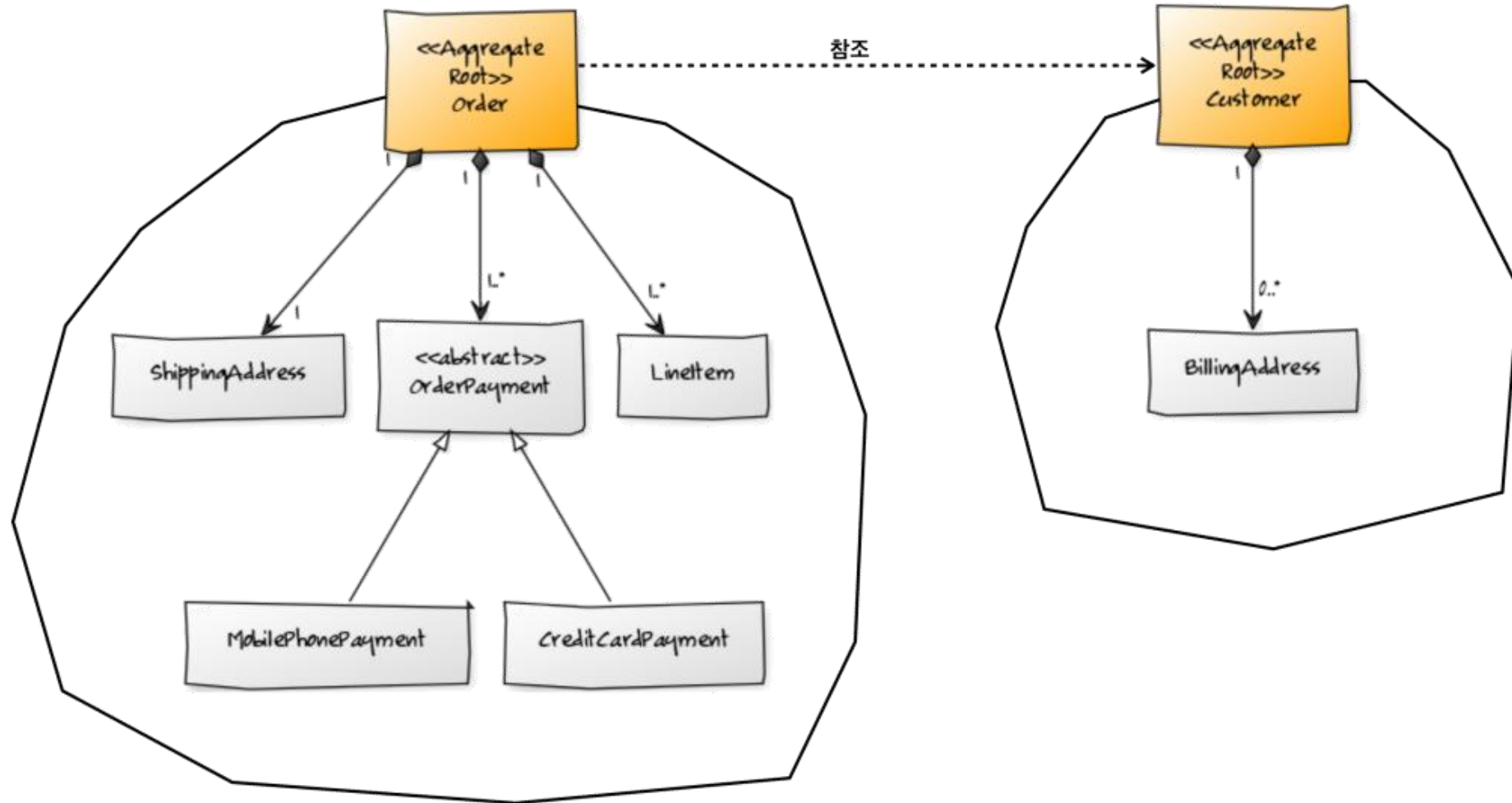
정리하면.

각각의 도메인들을 레이어로 철저히 분리해서 만든 것이
DDD의 핵심 설계 방식.

설계한 도메인들을 모듈별로 분리하는 것이 마이크로 서비스.

집합(Aggregate)

도메인 영역을 대표



단점

작은 여러 서비스들이 분산 되어있기 때문에 모니터링하기 힘들.
모놀리식 아키텍처에 비해 통신관련 오류가 잦음.