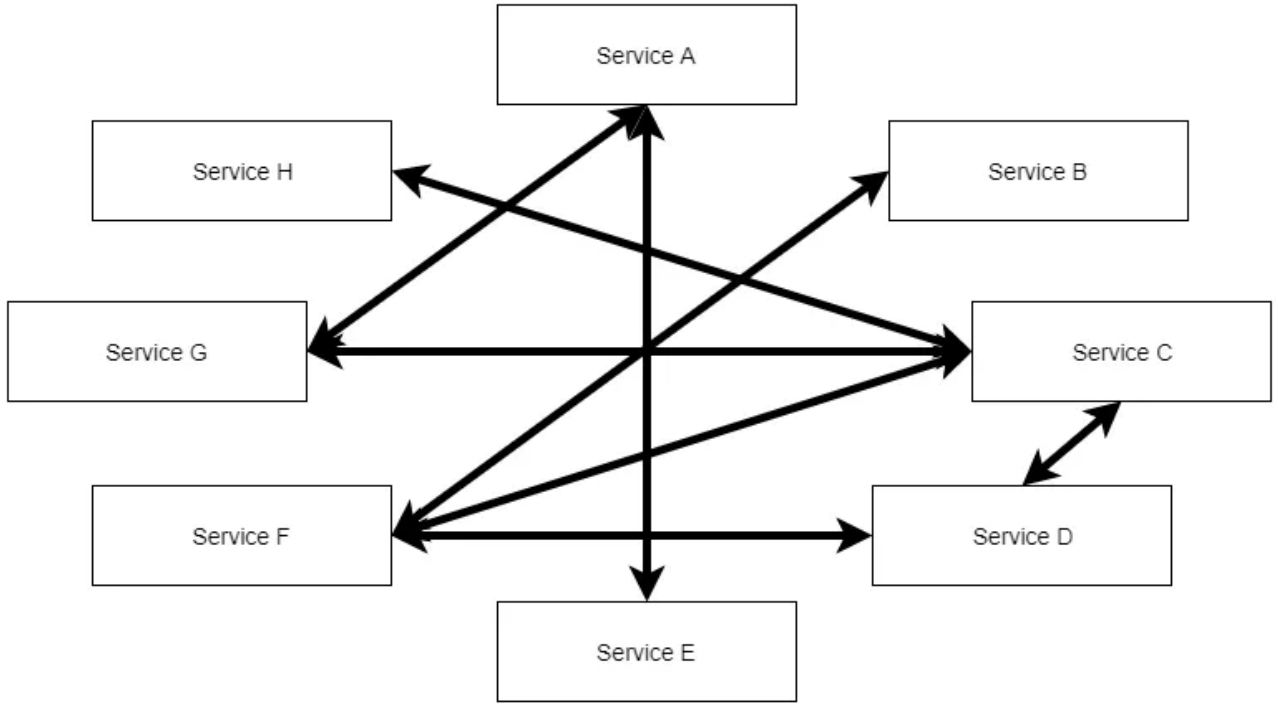
DOMAIN DRIVEN DESIGN
Tackling complexity in the heart of software

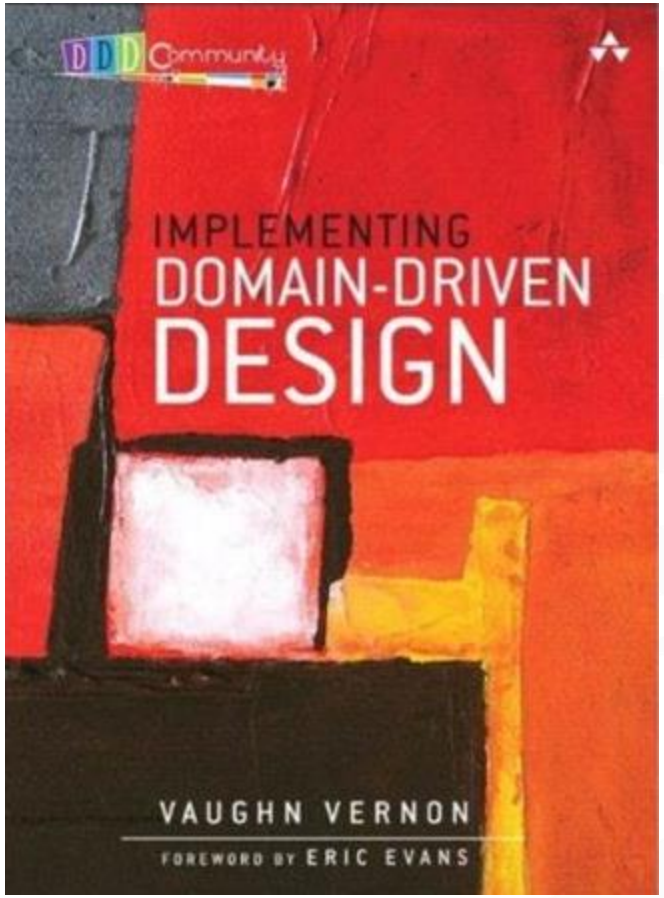
September 2020

Khó khăn gặp phải

- Nghiệp vụ thay đổi
- Logic nghiệp vụ không có sự tập trung
- Phụ thuộc vào database
- Dữ liệu không đảm bảo và nhất quán

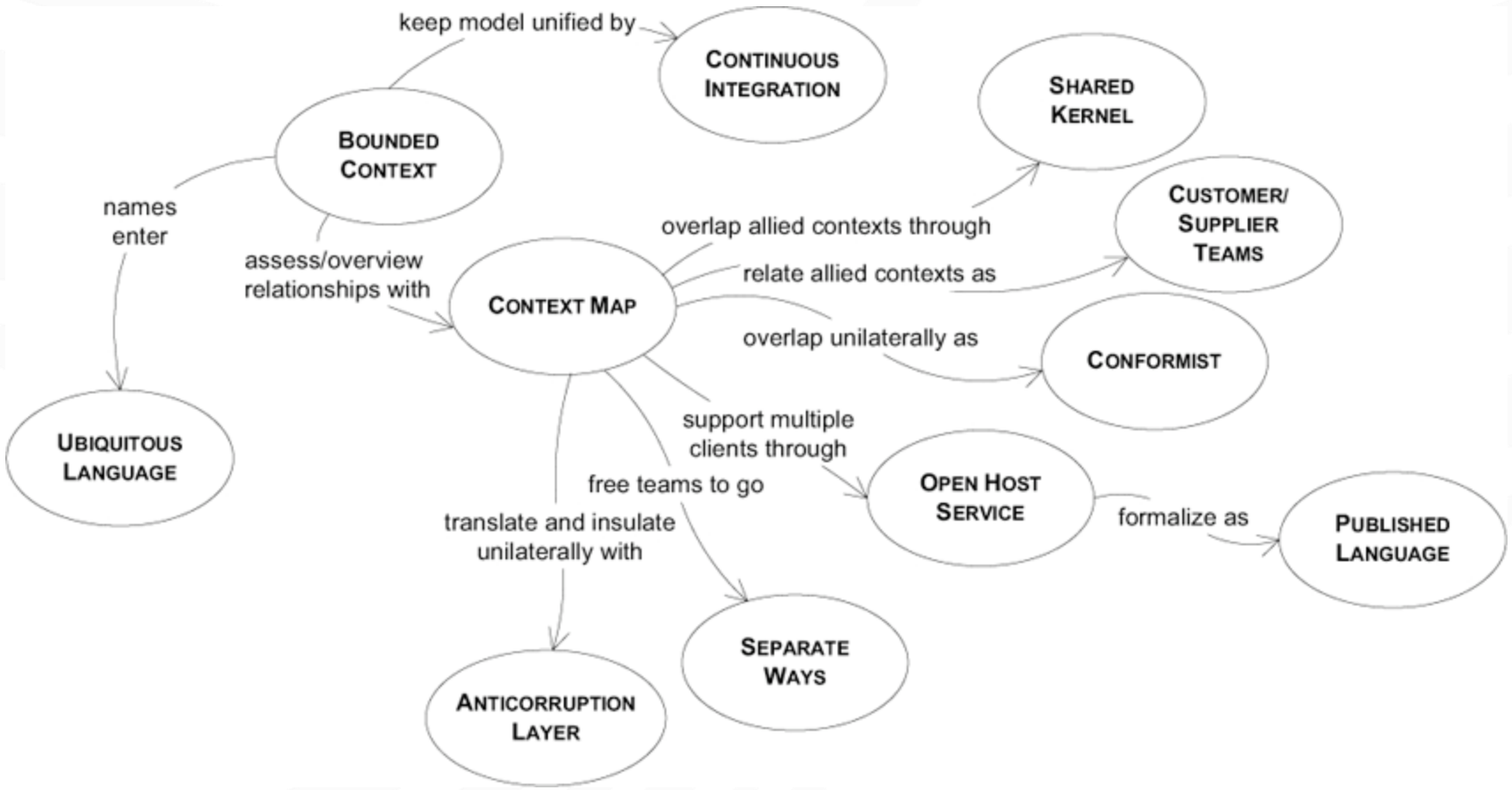


Domain Driven Design là gì



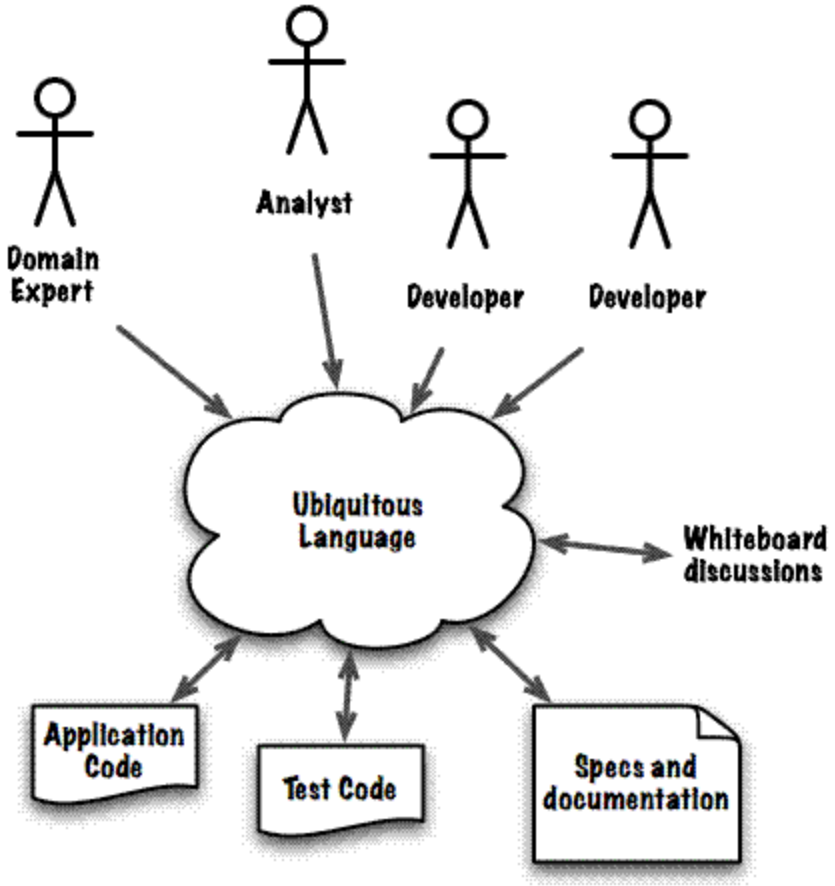
- Phương pháp thiết kế phần mềm lấy nghiệp vụ (domain) làm trung tâm
- Việc thiết kế domain model là trọng tâm
- Tăng cường sự cộng tác giữa các nhóm kỹ thuật (developers) và các chuyên gia nghiệp vụ (domain expert)

Stragetic Design



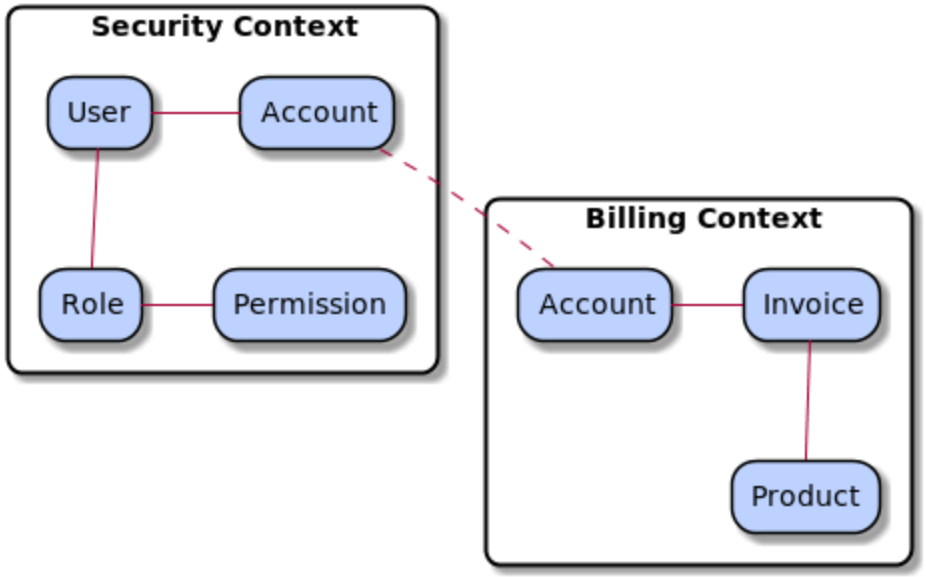
Ubiquitous Language - Ngôn ngữ chung

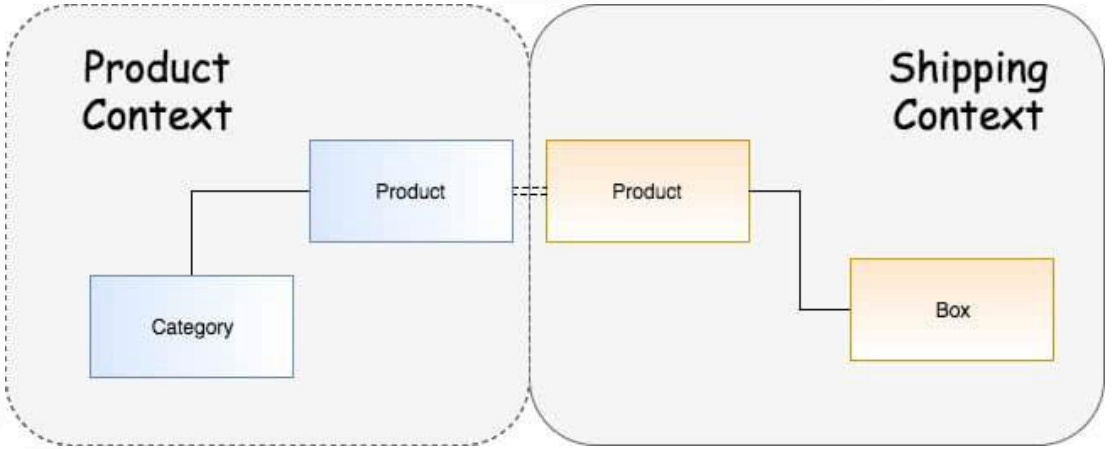
- Ngôn ngữ được sử dụng trong cả kĩ thuật lẫn trong nghiệp vụ
- Được phản ánh trong code
- Được phản ánh trong tất cả tính năng của hệ thống



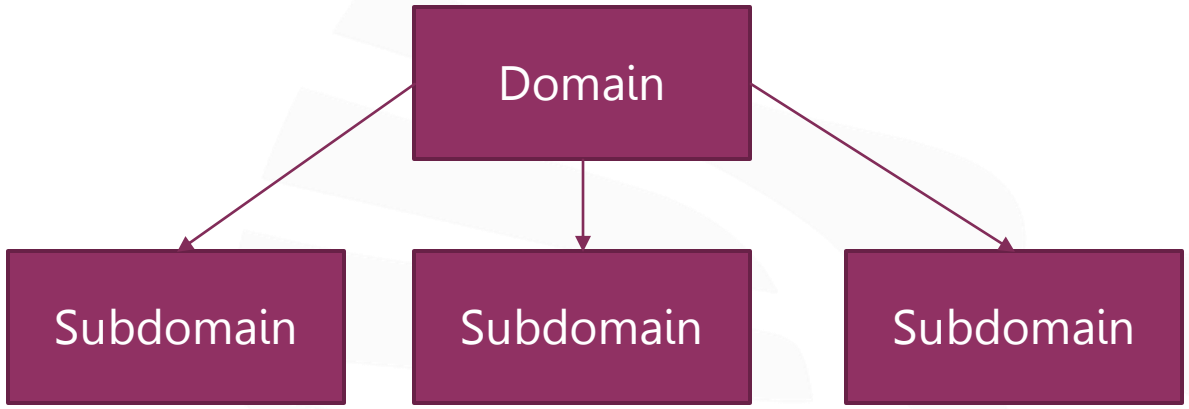
Bounded Context - Ngữ cảnh giới hạn

- Domain độc lập
- Được kết nối với nhau thông qua context map
- Phù hợp để triển khai microservice

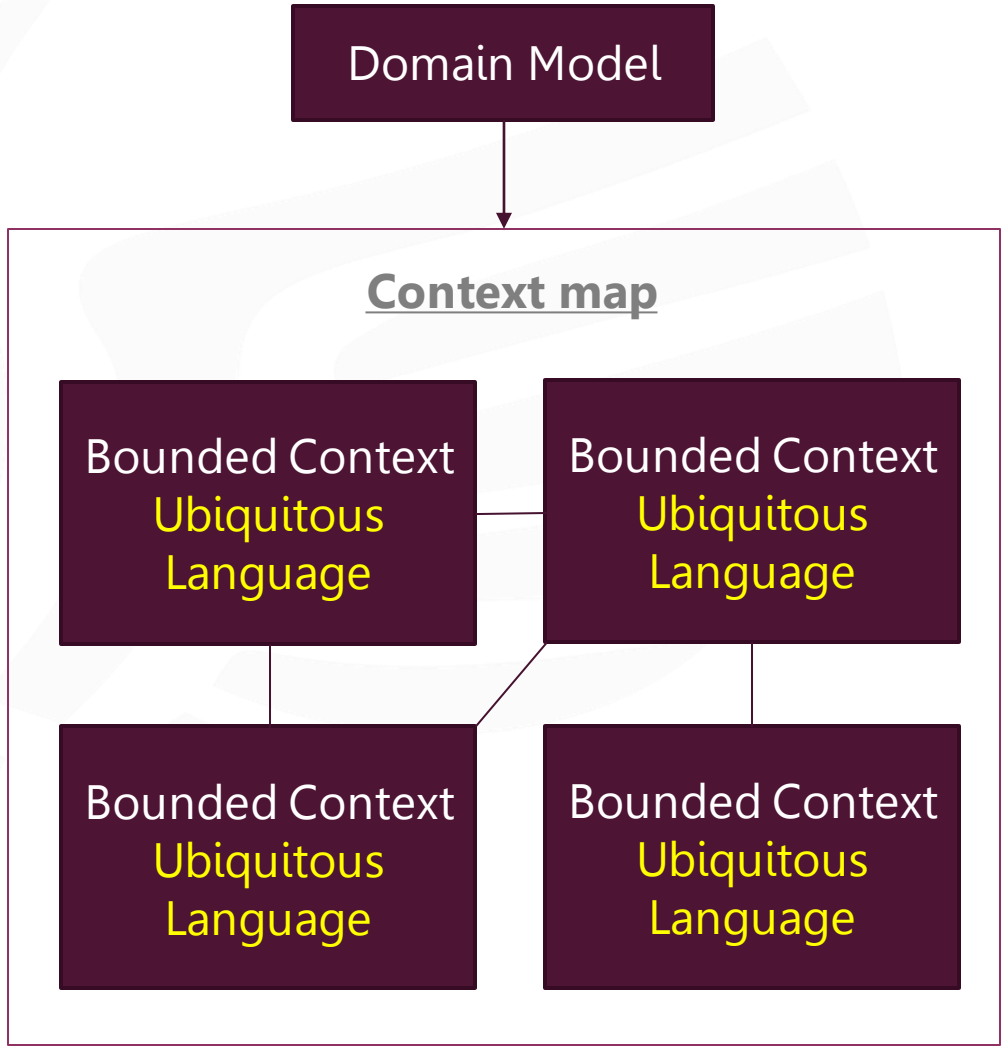




Problem Space

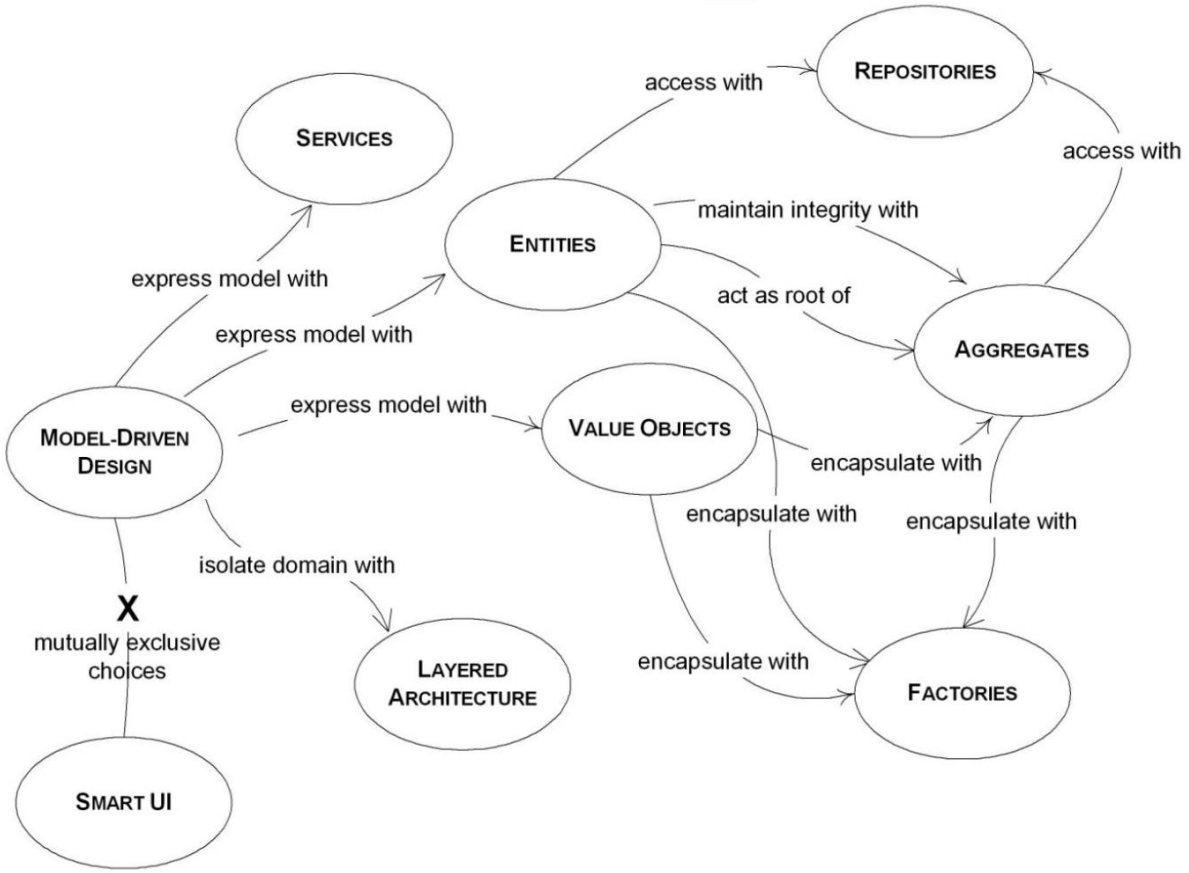


Solution Space



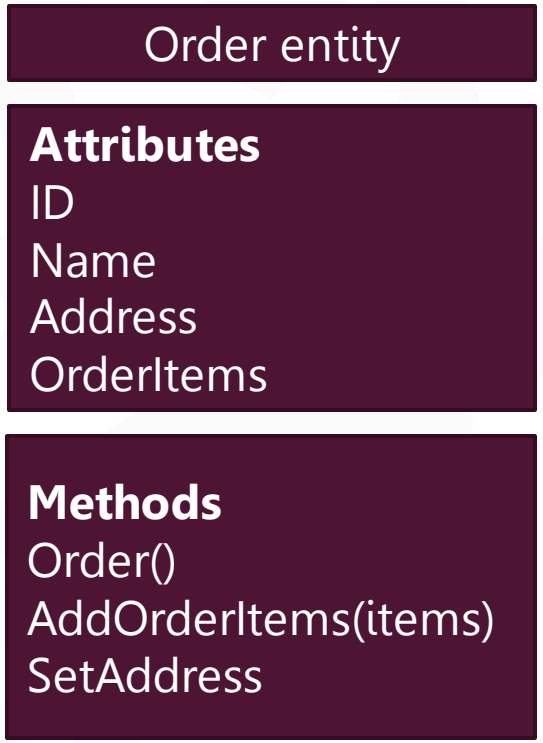
Context Map : Bản chất của các quan hệ Bounded Contexts

Tactical Design



Entity – Thực thể

- Có định danh bất biến và duy nhất
- Chứa các logic riêng
- Cần nên cân nhắc xem đối tượng có phải là một Entity



Value Object

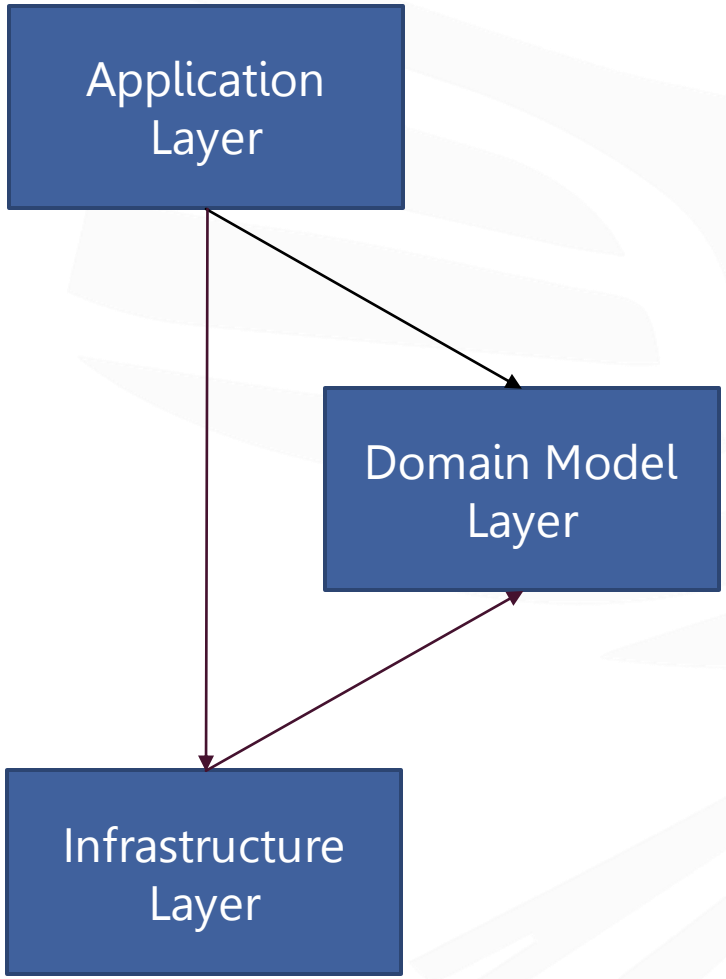
- Không có định danh
- Value object có tính bất biến, không khả chuyển (Immutable)
- Nếu các thuộc tính đều có cùng giá trị thì là các object như nhau.

Aggregate

- Một tập hợp các thực thể
- Các xử lý đều thông qua Aggregate Root
- Aggregate phải chứa behavior của các entity

Order entity	OrderItem child entity	Address Value Object
Attributes ID Name Address OrderItems	**Attributes** ProductId Price Quantity	**Attributes** City District State Road
Methods Order() AddOrderItems(items) SetAddress	**Methods** OrderItems() Increase(productId) SetAddress	

Layered Architecture – Kiến trúc phân lớp



- **Application Layer**
  - Đóng vai trò như controller trong MVC
  - Phụ thuộc toàn bộ vào 2 tầng dưới
- **Domain Layer**
  - Trái tim của hệ thống
  - Không phụ thuộc vào bất cứ tầng nào
- **Infrastructure Layer**
  - Chứa các thao tác xuống cơ sở hạ tầng
  - Phụ thuộc vào entity của tầng Domain và các thư viện, framework, ORM, .v.v

Vài khái niệm khác

- Façade && Repository pattern
- Specification pattern
- Rich vs Anemic Domain Model
- Domain service
- Domain event
- CQRS (ES)

THANK YOU