

Tactical Domain Driven Design

trendyol
learning

Fatih Furkan Has



PURPOSE



- DDD kavramı
- DDD bileşenleri
- Gerçek örnekler
- DDD zorluklarını tartışmak

WHY



- DDD'nin önemi artıyor
- Uzun vadede tasarruf
- Başarılı yazılım çözümleri
- Örnek bir uygulama ile konsepti anlamak
- Kariyer avantajı sağlamak





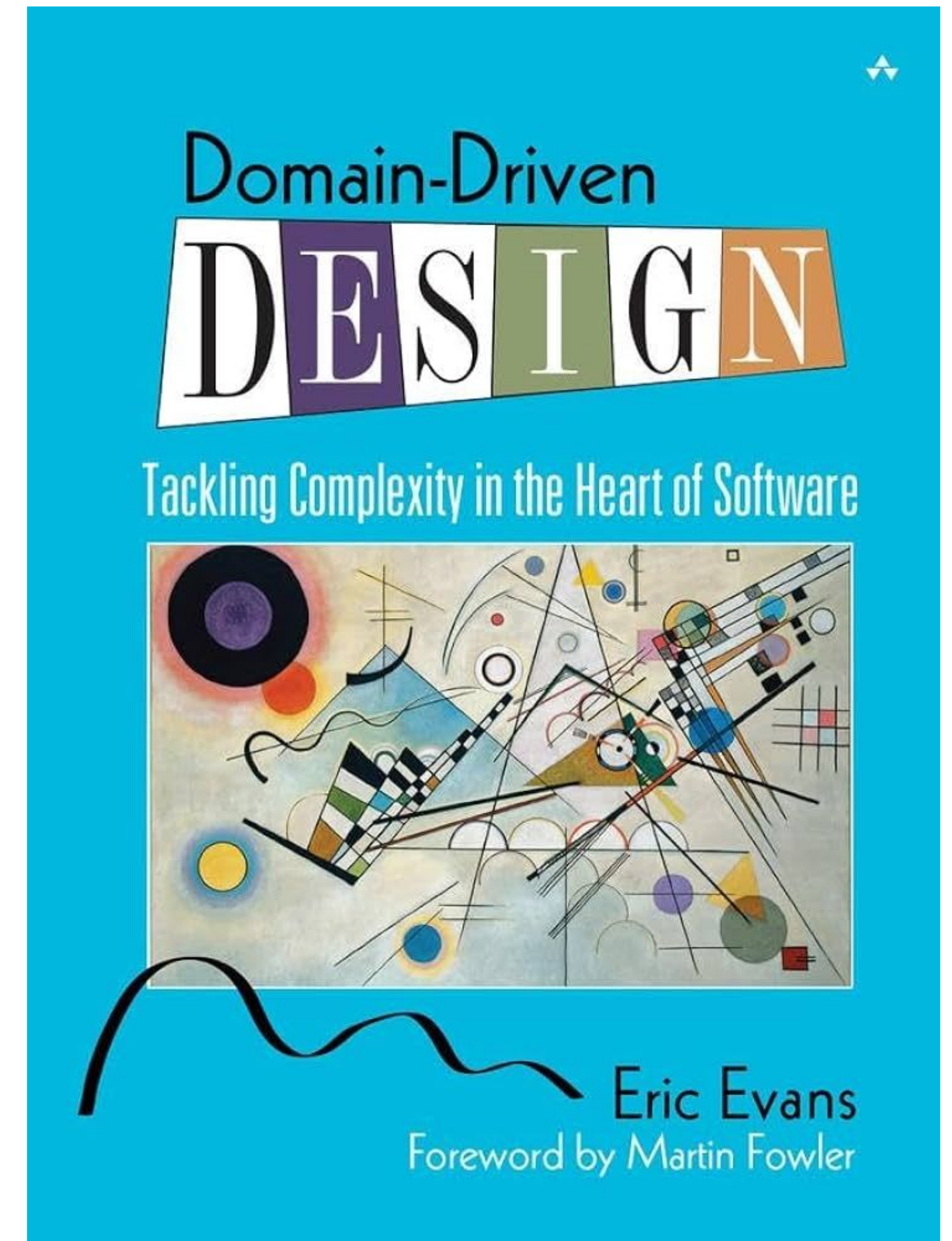
AGENDA

- 1 DDD
- 2 Tactical DDD
- 3 Örnek Uygulama
- 4 Özet ve Q&A



DDD

- Business Logic
- Ubiquitous Language
- Complexity
- Alignment
- Communication
- Flexibility
- Scalability
- Technical Debt
- Evolving Requirements



Fundamentals of DDD

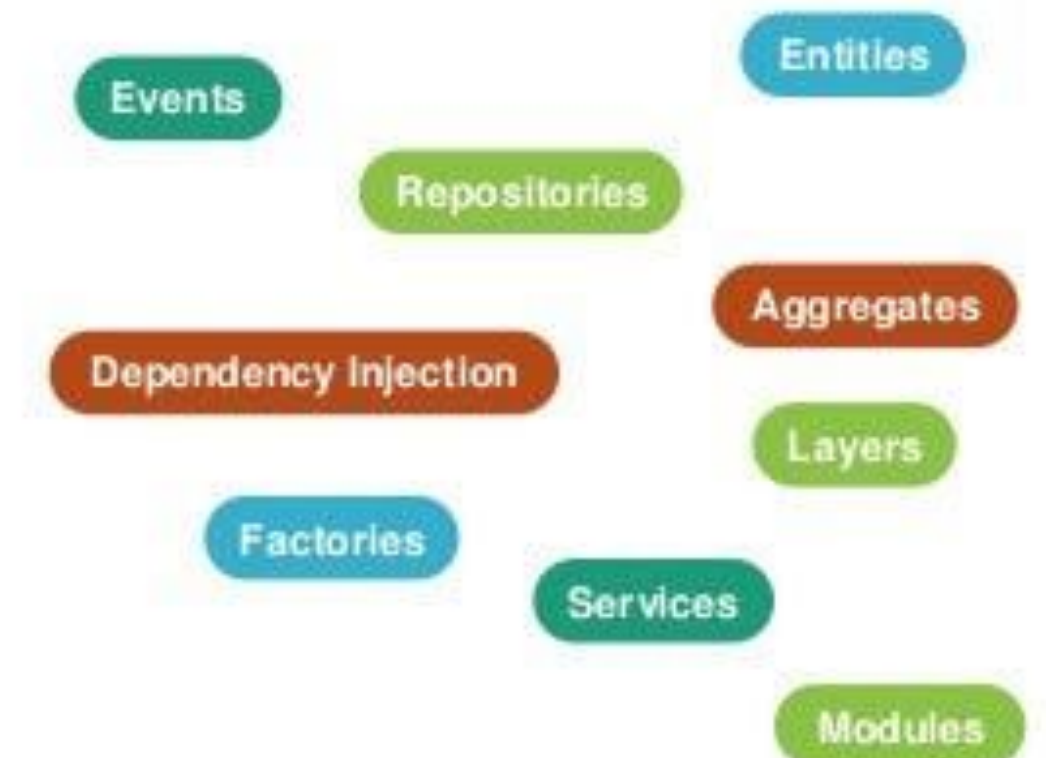
- Ubiquitous Language
- Bounded Context
- Strategic DDD
- Tactical DDD
- Complexity
- Business Objectives

Domain Driven Design

Strategic design

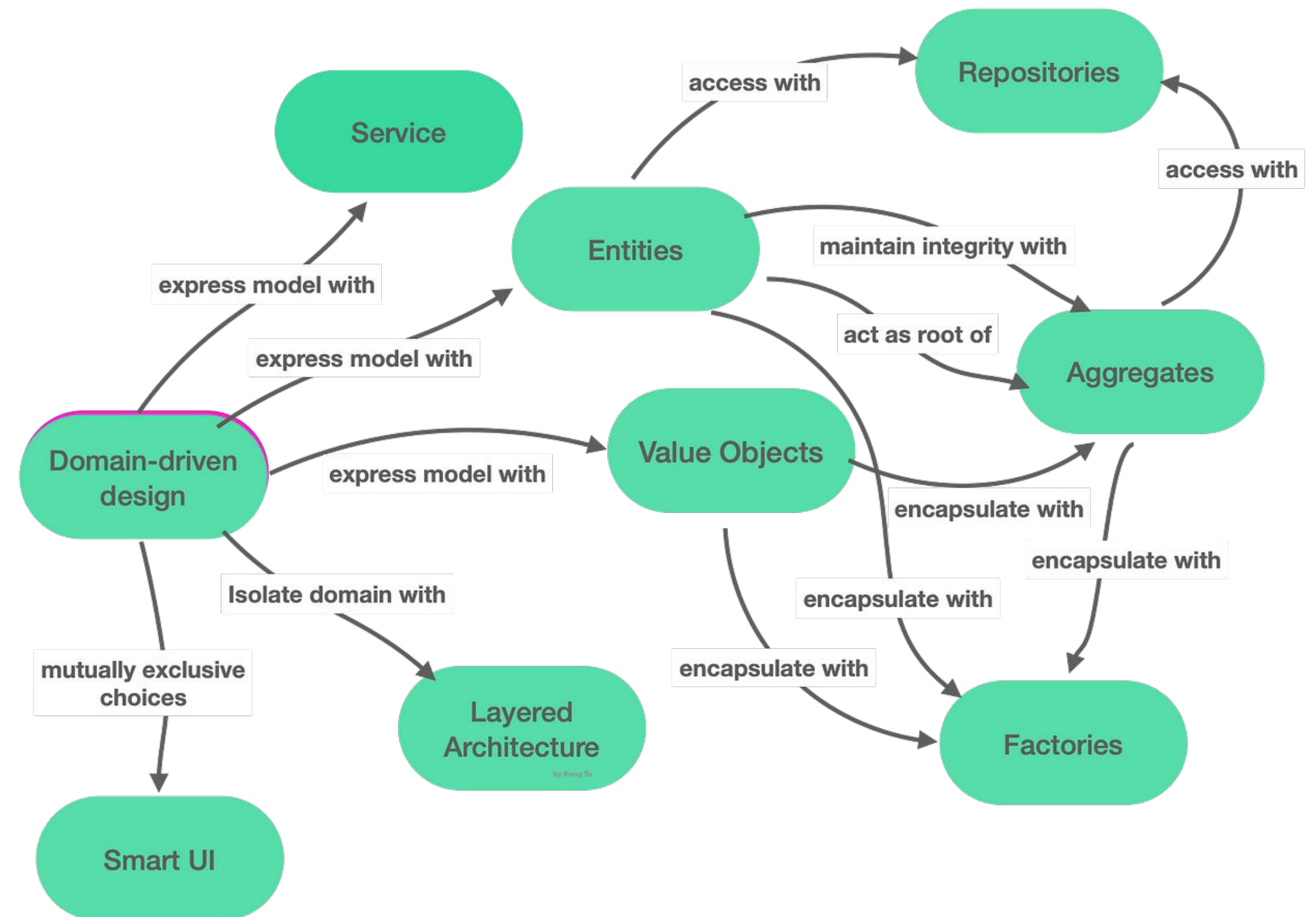


Tactical patterns



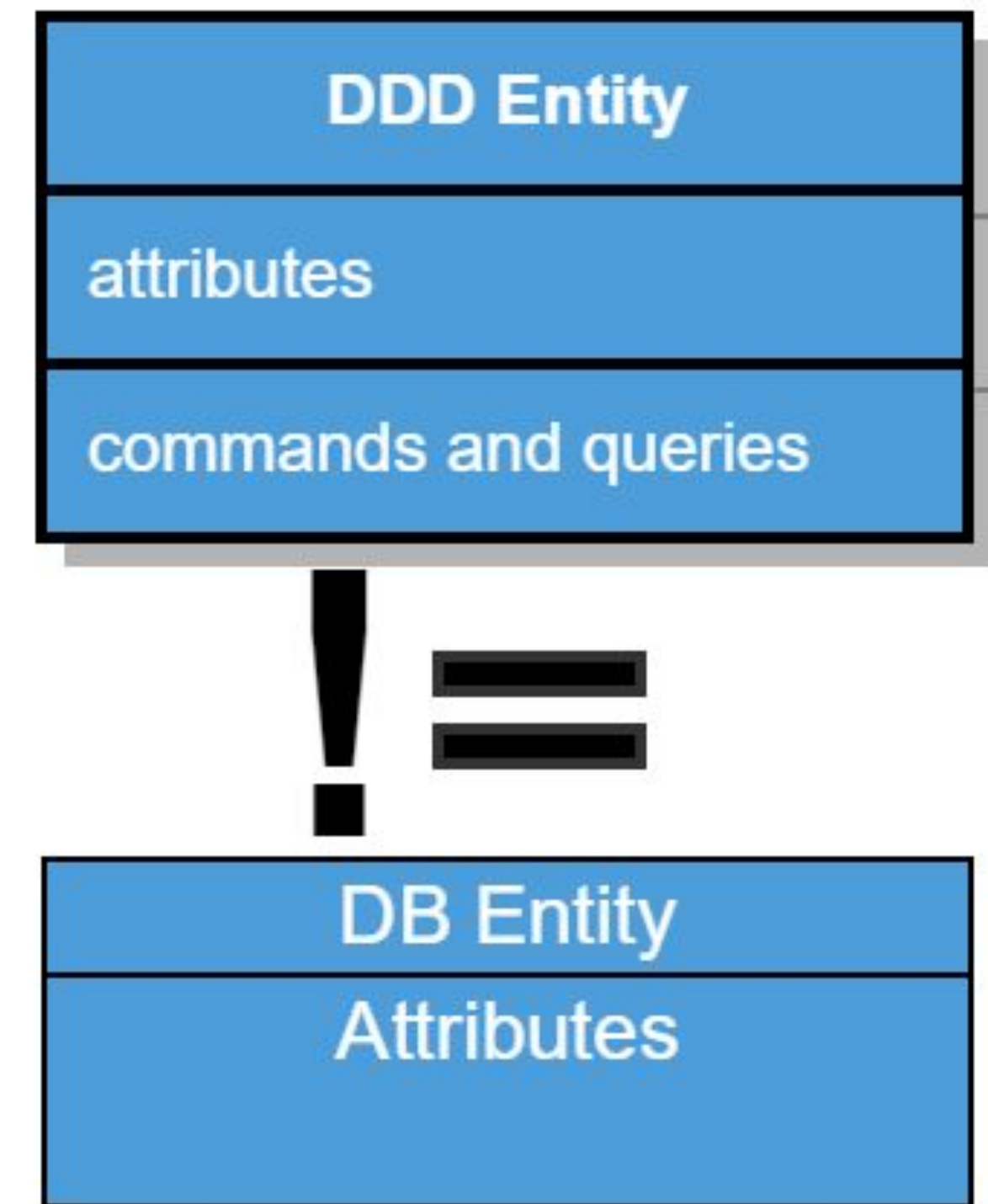
Tactical DDD

- Entity
- Value Object
- Aggregate
- Domain Service
- Application Service
- Event
- Repository



Entity

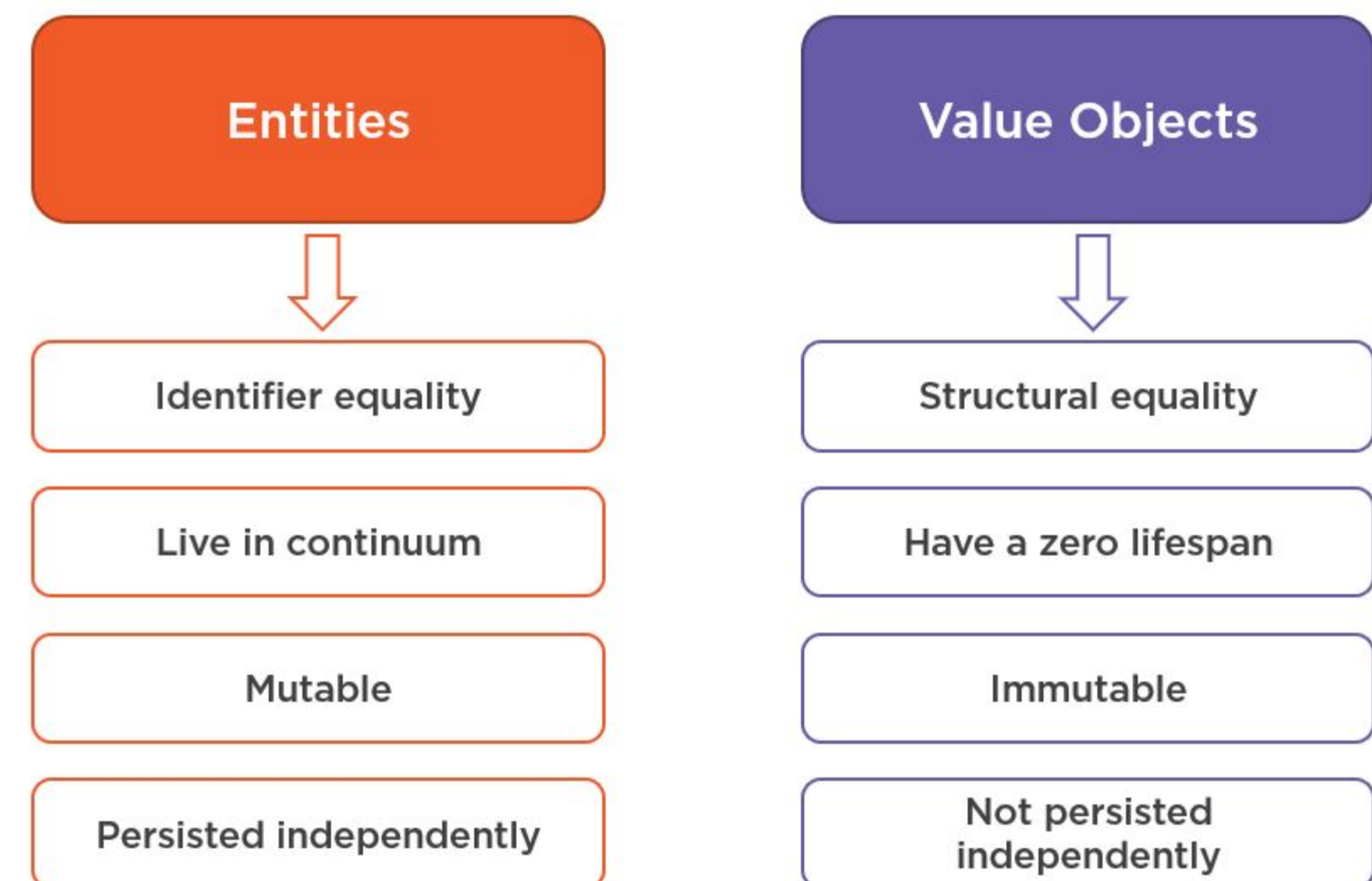
- Unique Identity
- Business Rules & Behaviors
- Long-lived Objects



Value Object

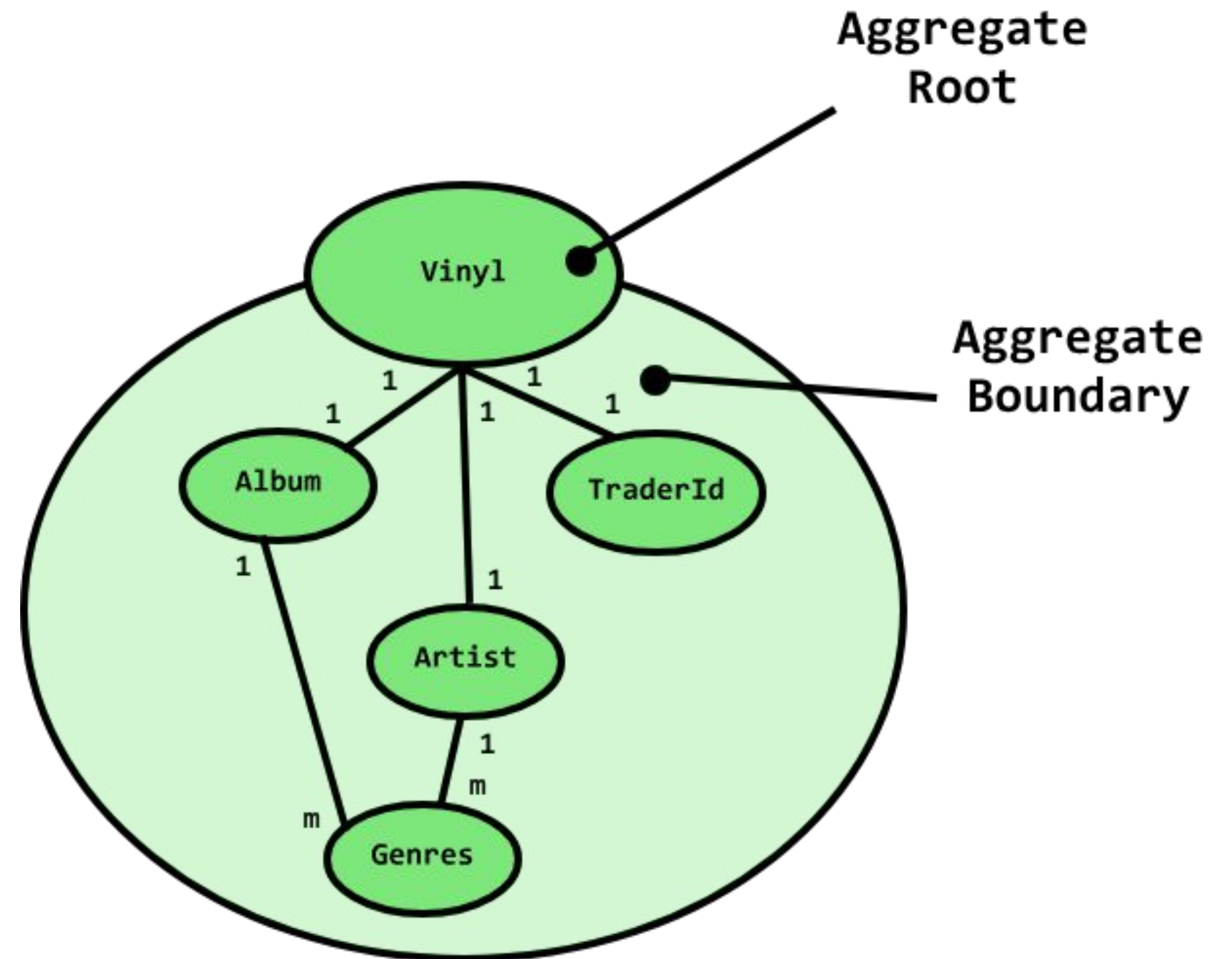
- Differences from entity
- Identity
- Immutability

Value Objects vs Entities at a Glance



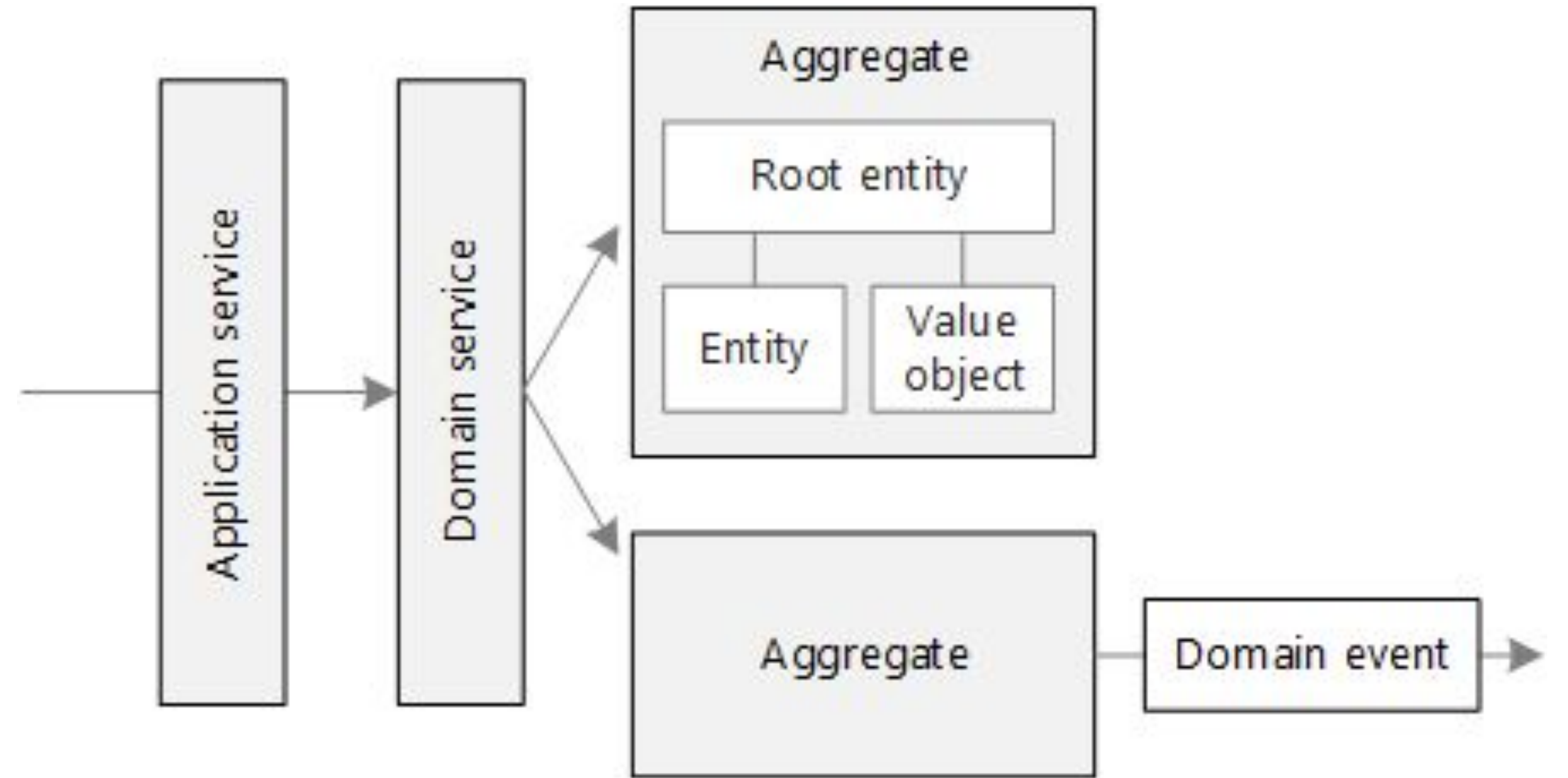
Aggregate

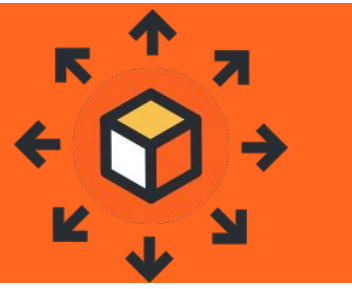
- Root Entity
- Consistency Boundaries
- Transactional Consistency
- Domain Model
- Business Rules
- Isolation



Domain Service and Application Service

- **Domain Services:** Domain içindeki iş kurallarını ve işlevselliği modellemek için kullanılır. Belirli bir Entity ya da Value Object'e net bir yer bulamayan işlevleri temsil eder. Aynı zamanda domain modelimizin saf ve bütünlüklü kalmasını sağlar.
- **Application Services:** Domain logic'inin orkestrasyonunu gerçekleştirir. Uygulamanın farklı bileşenleri arasında koordinasyon ve etkileşim sağlar. Uygulamanın genel işleyişini yönetirken, domain logic'ini dış etkenlerden koruma görevi de üstlenir.
- **Farkları:** Domain Services, spesifik iş kurallarını yönetir. Application Services ise uygulamanın genel işleyişini ve domain logic'inin orkestrasyonunu ele alır.





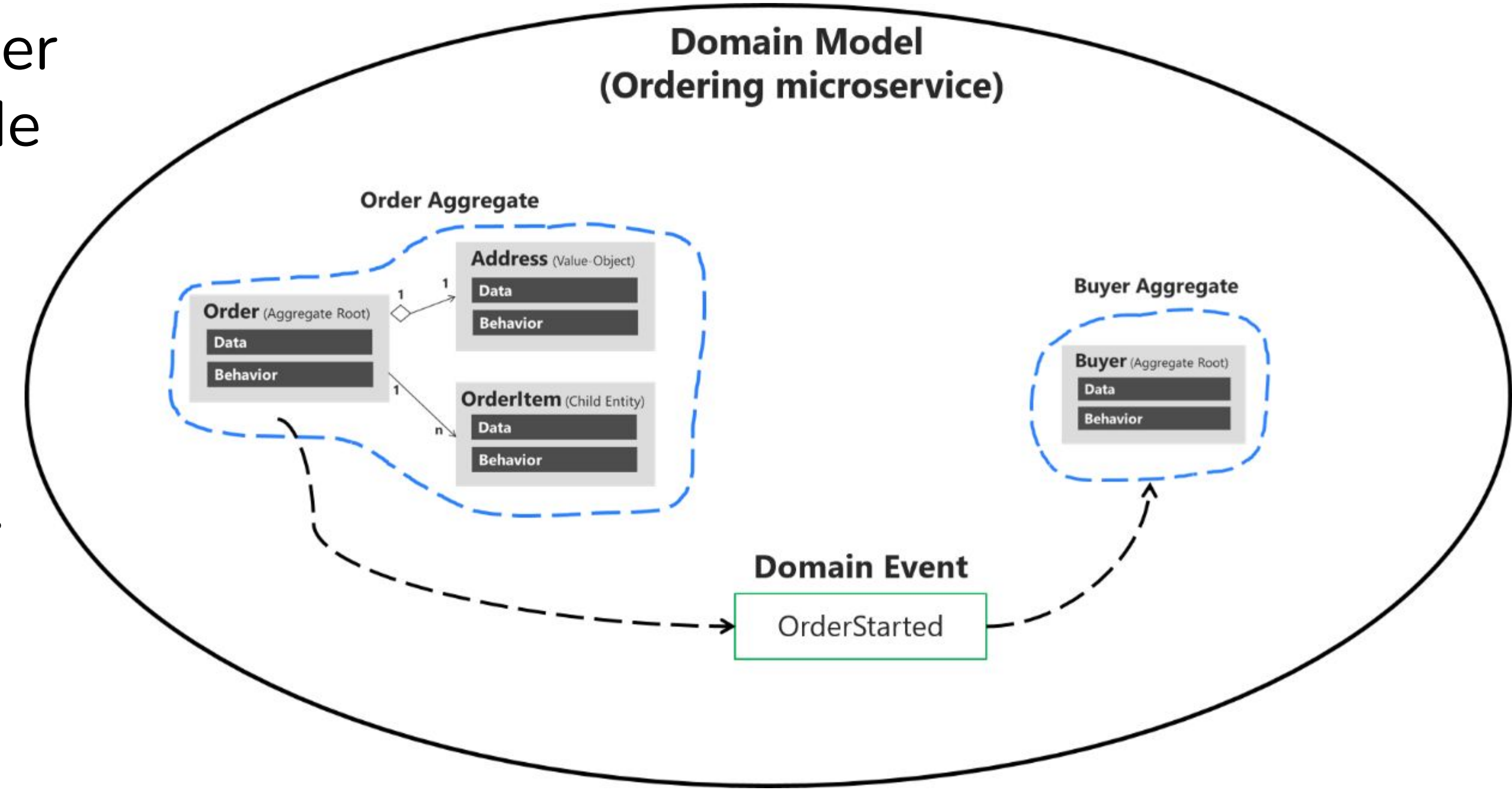
Repository

- Domain modelini veri katmanının teknik detaylarından etkilenmeden korur.
- İş mantığı ve uygulama kurallarını, kullanılan veritabanı teknolojisinden tamamen ayırır.
- Aggregate'lerin tutarlılık sınırları içerisinde kalmasını garantileyerek veri erişimini yönetir.
- Domain modeli ile veritabanı operasyonları arasındaki etkileşimi kolaylaştırır.



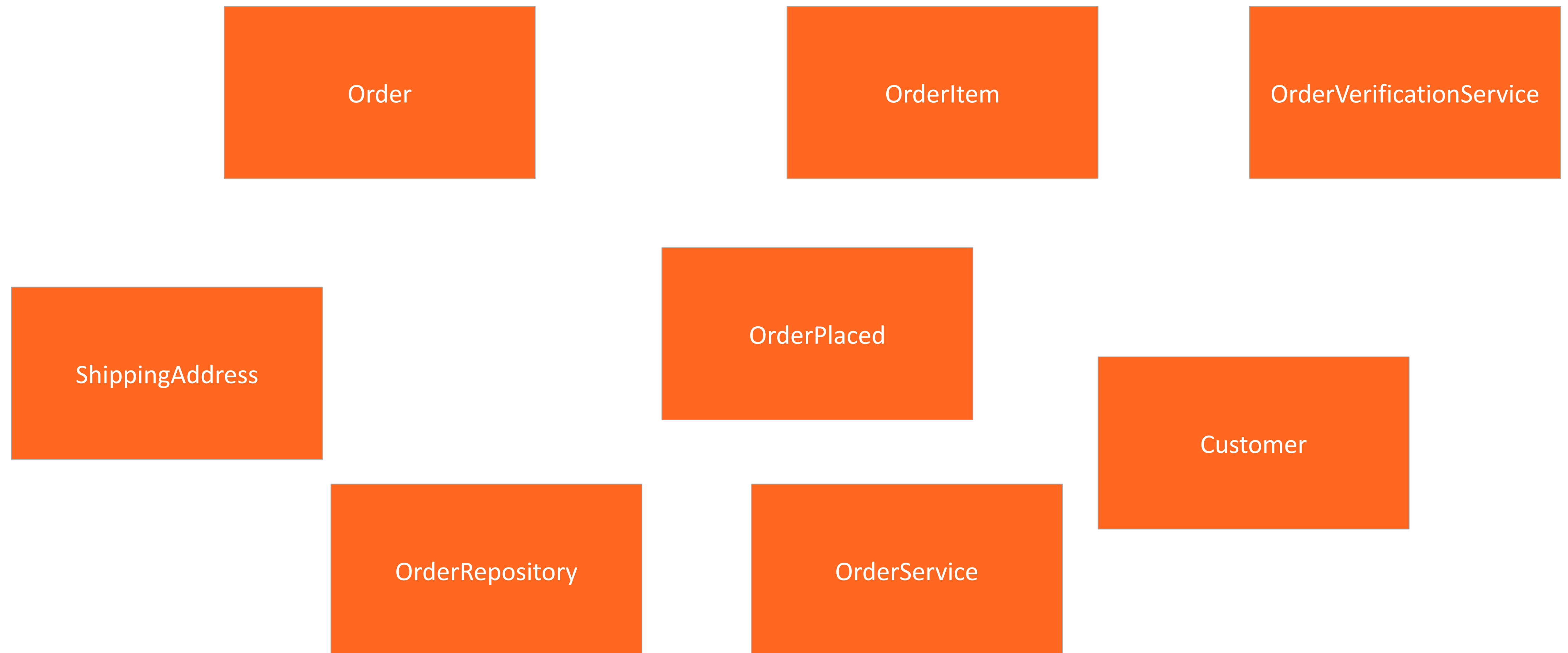
Event

- Domain'deki önemli deęişiklikleri temsil eder
- Genellikle gemiř zamanlı mesajlar řeklinde yayınlanır
- Bu olaylara tepki olarak sistem bileřenleri belirli eylemler gerekleřtirir.
- Sistemler arasındaki baęımlılıęı azaltır.
- Hızlı ve tutarlı bir kullanıcı deneyimi saęlar.





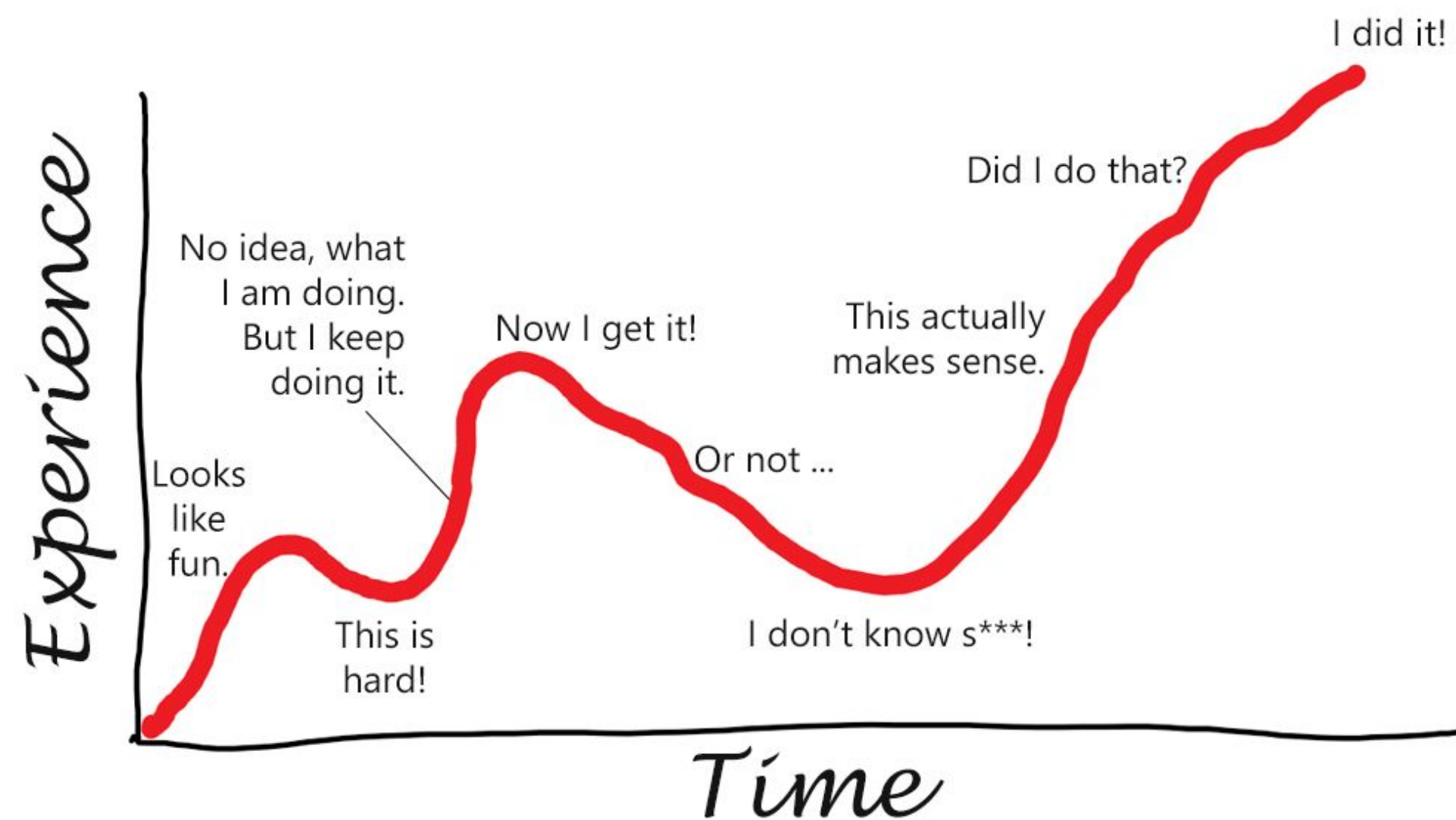
Örnek





Zorluklar

- Learning Curve
- Overengineering
- Yanlış Aggregate ve Entity Tanımı



Q&A

THANK YOU

trendyol learning 

