



DDD 산책

개발자 관점의 DDD 소개와 샘플 구경

KSUG 박성철 (gyumee@gmail.com)

通身翻卷完整协调，摆扣步
 进行，不可忽前忽后。
 (影子抱球) 由上式左脚踏尖
 转，右脚随之向左脚扣步，同时
 球由身球右半转，右脚前移半步
 球由左左腿转出，重心向上，左
 脚向下，两重心上下相对，眼着右

由左腋下随身体左转方向划一弧
 圆，如同抱物形状，不可伸直。
 完全相同(52, 53, 54, 55, 56, 57.)
 加着翻身动作，但初学须由慢而快
 已整协调。



图51

图52



图53

图54

기공?
 점공?

上绕至左
 腕腕左
 翻
 手从眼



是
砍
腕

五步按剑截腕
 腕是剑由身自所
 围隔的横截右腕
 左旁出插左手绕



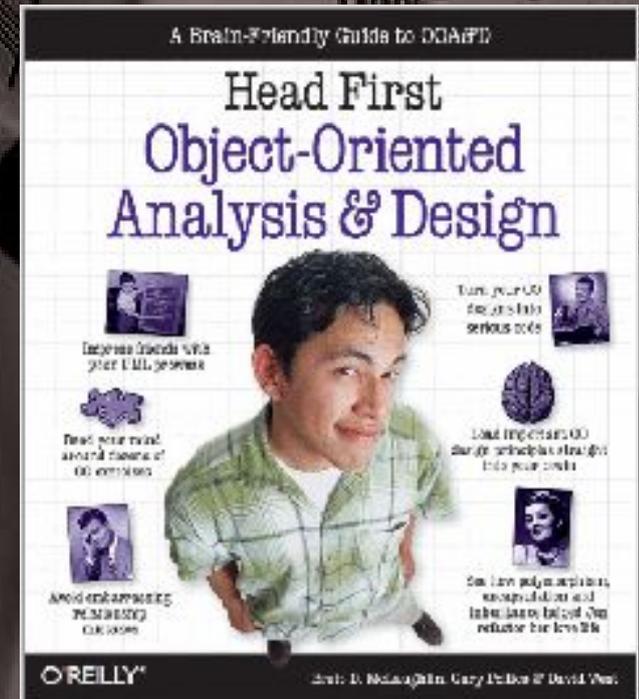
横
 剑
 腕
 图

左
 砍
 作

좋은 소프트웨어?

우리는 문제를 해결하려고 소프트웨어를 만든다. 사람들이 문제를 가지고 있으므로 우리는 사람을 위해 소프트웨어를 만든다.

좋은 소프트웨어는 눈 앞의 문제를 해결할 뿐 아니라 고객이 장차 요구할 피할 수 없는 변경에 대응하도록 유지보수와 수정이 쉬워야 한다.



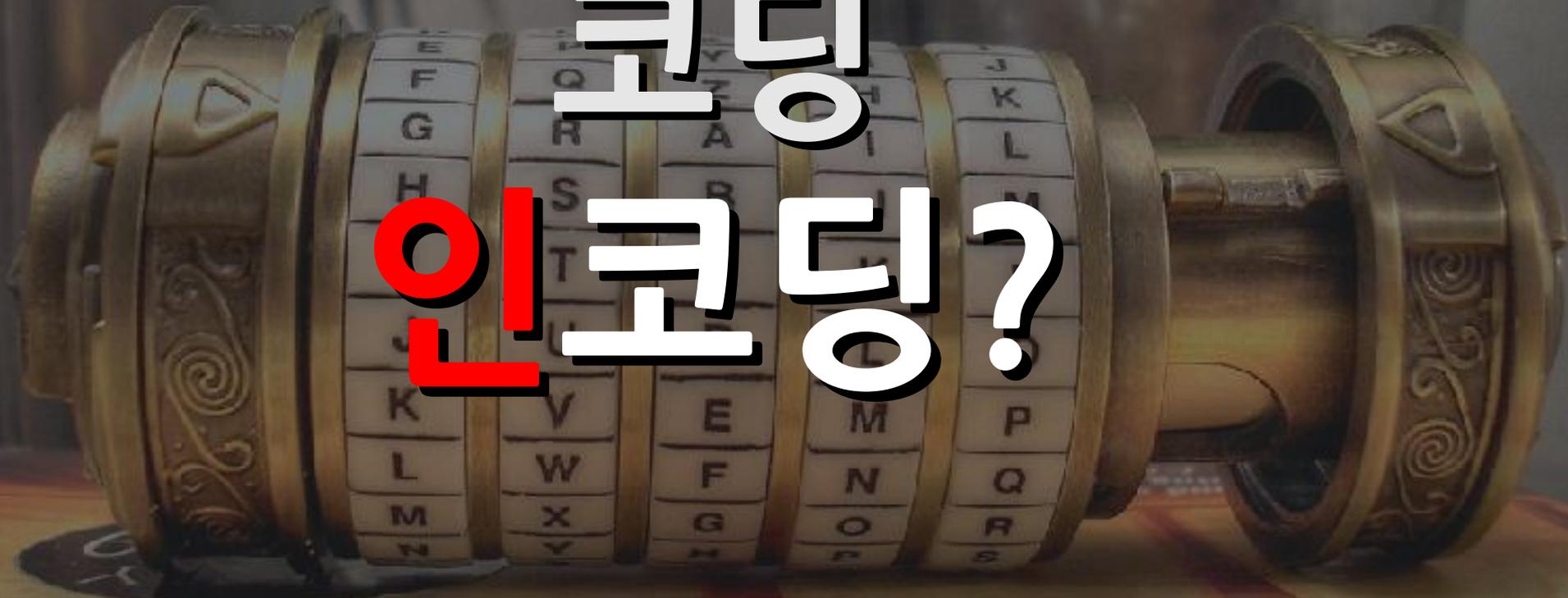


미녀리라고 하더니...

코드

코딩

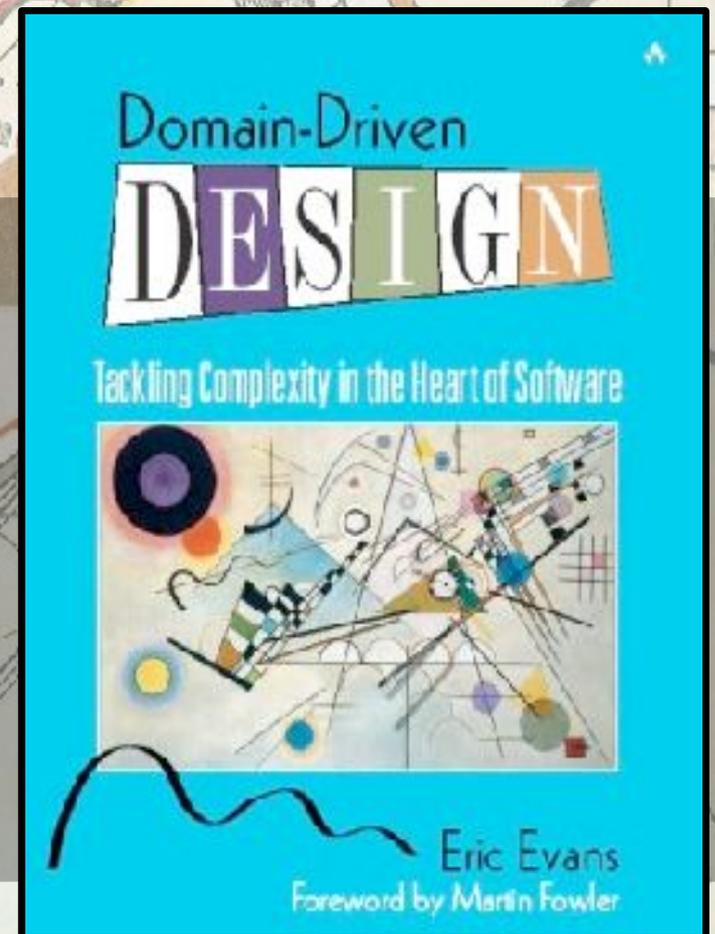
인코딩?



Domain-Driven Design

Tackling Complexity in the Heart of Software

Author: Eric Evans
Hardcover: 560 pages
Publisher: Addison-Wesley (August 30, 2003)
Language: English
ISBN-10: 0321125215
ISBN-13: 978-0321125217
Product Dimensions: 9.5 x 7.3 x 1.4 inches
Shipping Weight: 2.8 pounds
Average Customer Review: 4.2 out of 5 stars
Amazon Bestsellers Rank: #47,075 in Books
#14 in ... Design Tools & Techniques
#32 in ... Systems Analysis & Design

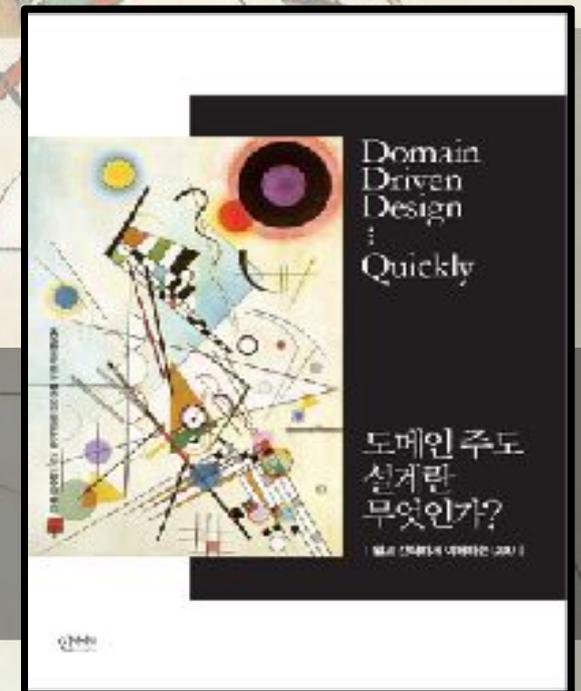


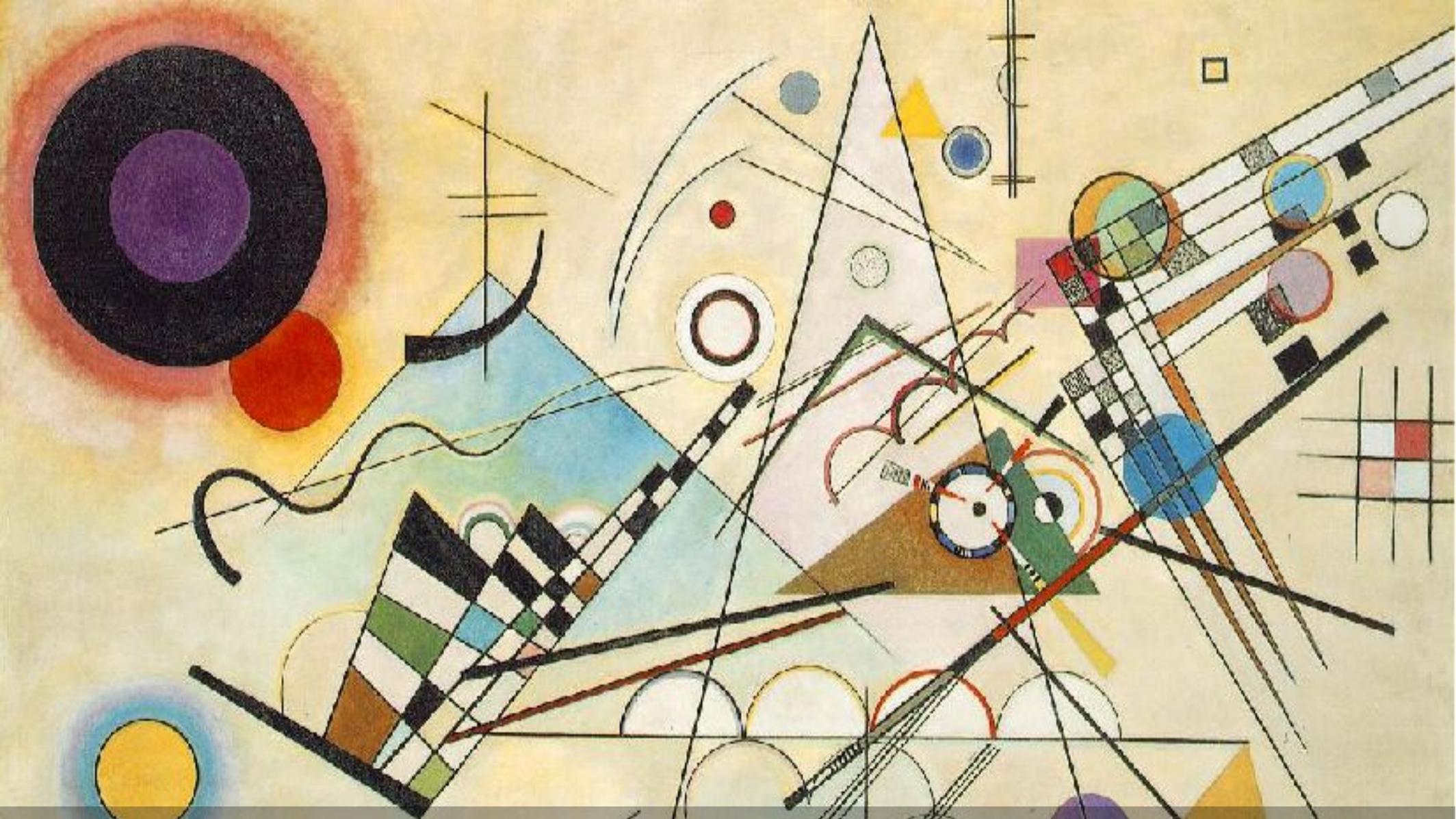


도메인 주도 설계

소프트웨어의 복잡성을 다루는 지혜

도메인 주도 설계란 무엇인가?





구성 8 - 칸딘스키

1923; 캔버스에 유화, 140 x 201 cm;
솔로몬 R 구겐하임 미술관, 뉴욕

혼돈과 복잡성 안에서 질서를 찾음
추상적 언어 창조
추상적 언어를 적용



도메인 주도 설계란?

소프트웨어의 복잡성을 다루는 **지혜**

단순화 { 도메인
모델
협력

도메인 주도 설계 구성

보편적 언어

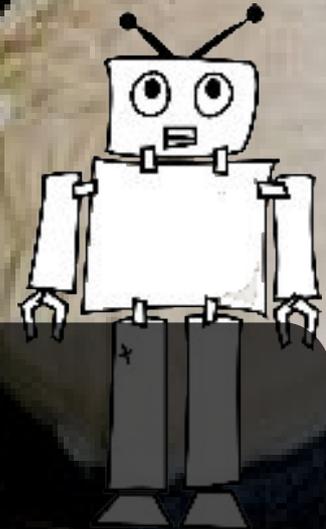
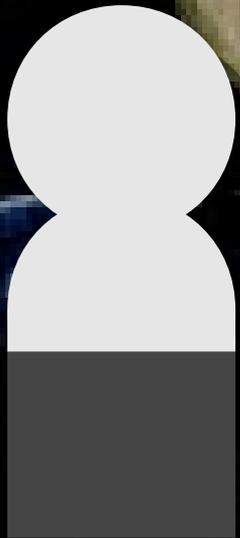
모델 주도 설계
리팩터링
전략적 설계



보편적 ubiquitous 언어

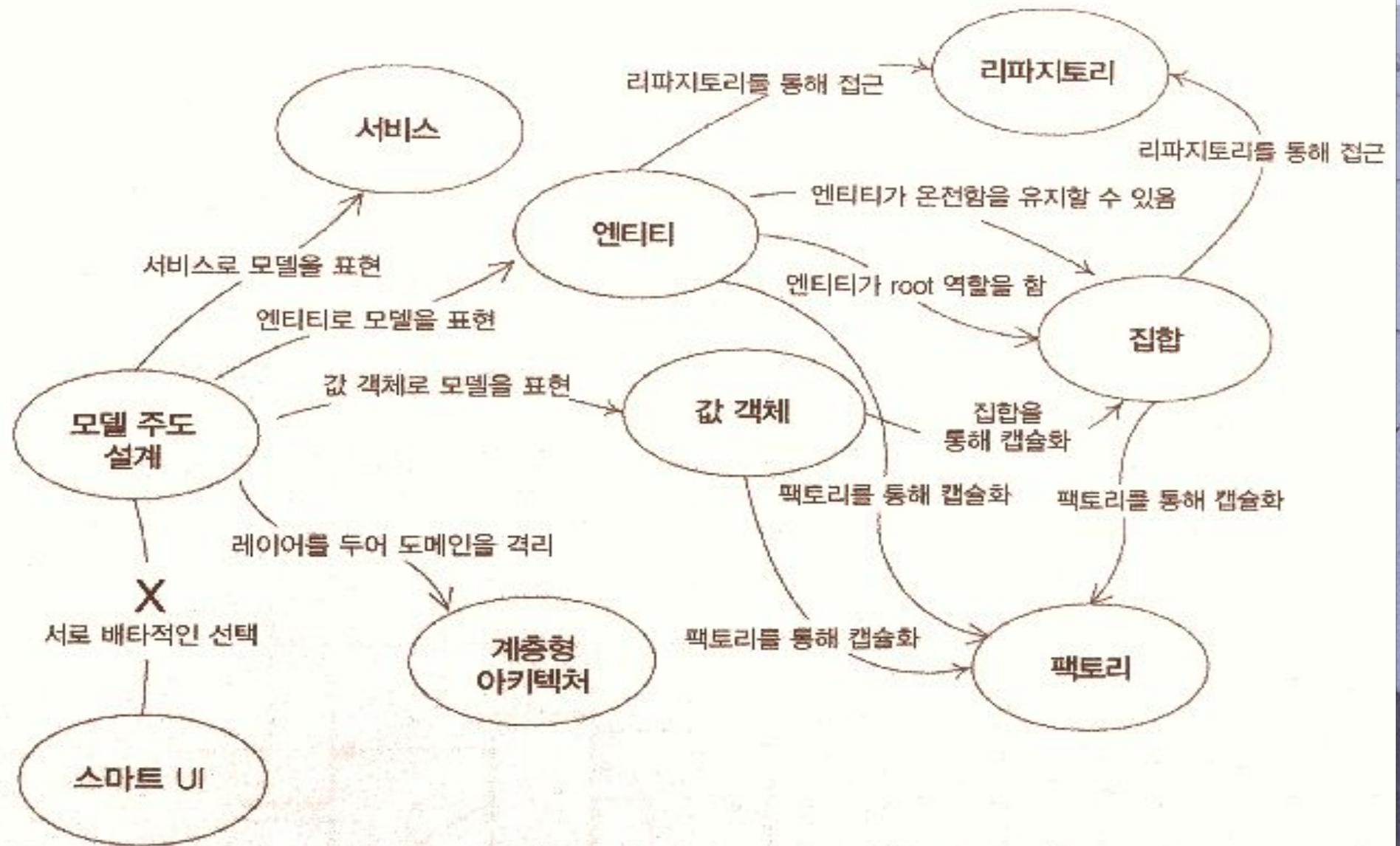
언어적 불일치: 번역 문제

공통 언어 개발
모델과 일치
탐구 활동



한 팀, 한 언어

모델 요소



계층형 아키텍처

도메인에게 자유를!!

사용자 인터페이스

애플리케이션

모델이 사는 곳

도메인 domain

기반구조 infrastructure



엔티티 Entity

고유 식별자를 가지고 있는 모델 객체

- 식별자는 유일하고 생애 동안 불변
- 비교는 식별자만 사용
- 값은 계속 변할 수 있음

연관 관계는 최대한 단순화

값 객체 Value Object

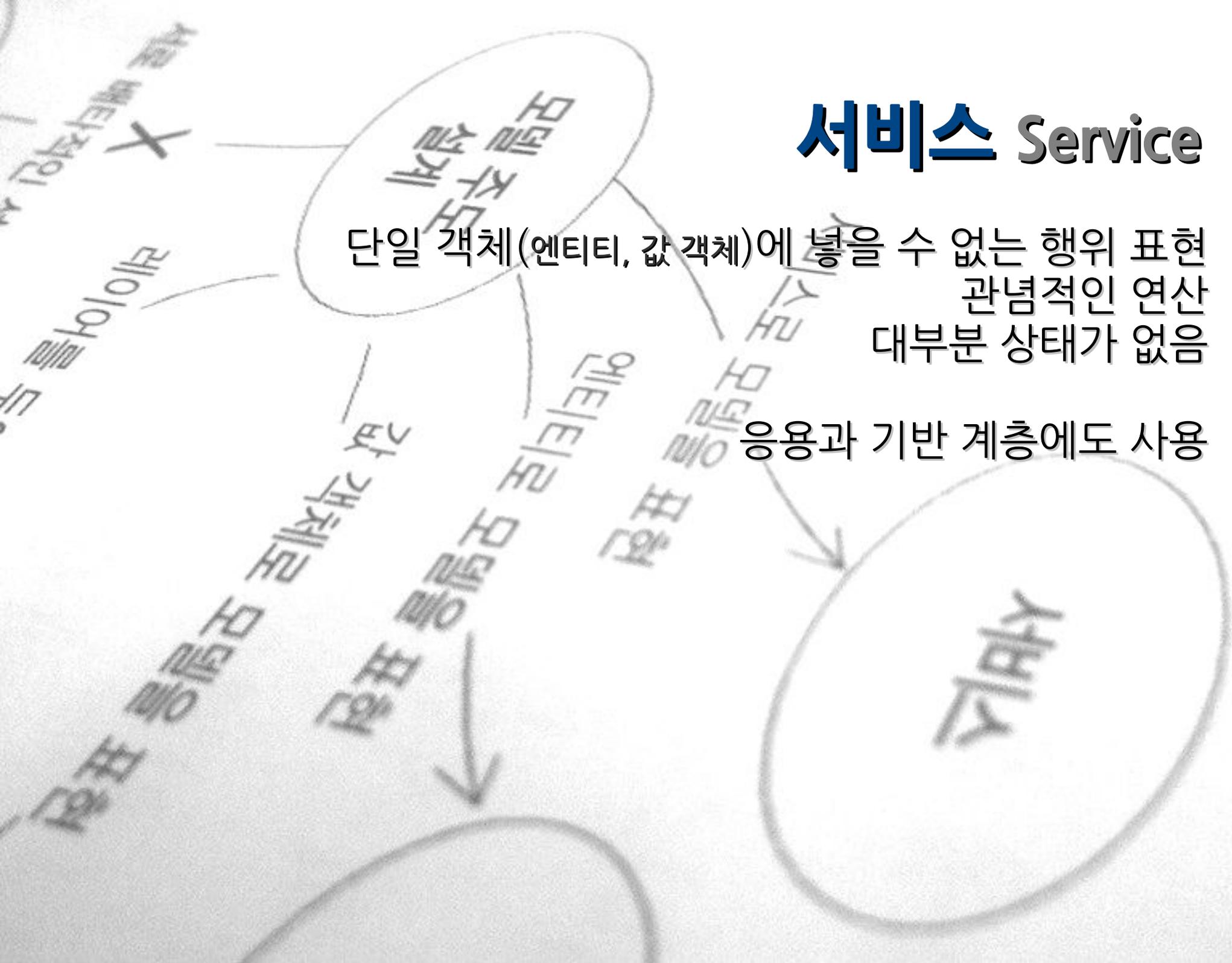
고유 식별자가 필요 없는 모델 객체
담고 있는 값으로 비교
불변 객체 / 부작용 방지 / 단순화

값 객체

서비스 Service

단일 객체(엔티티, 값 객체)에 넣을 수 없는 행위 표현
관념적인 연산
대부분 상태가 없음

응용과 기반 계층에도 사용



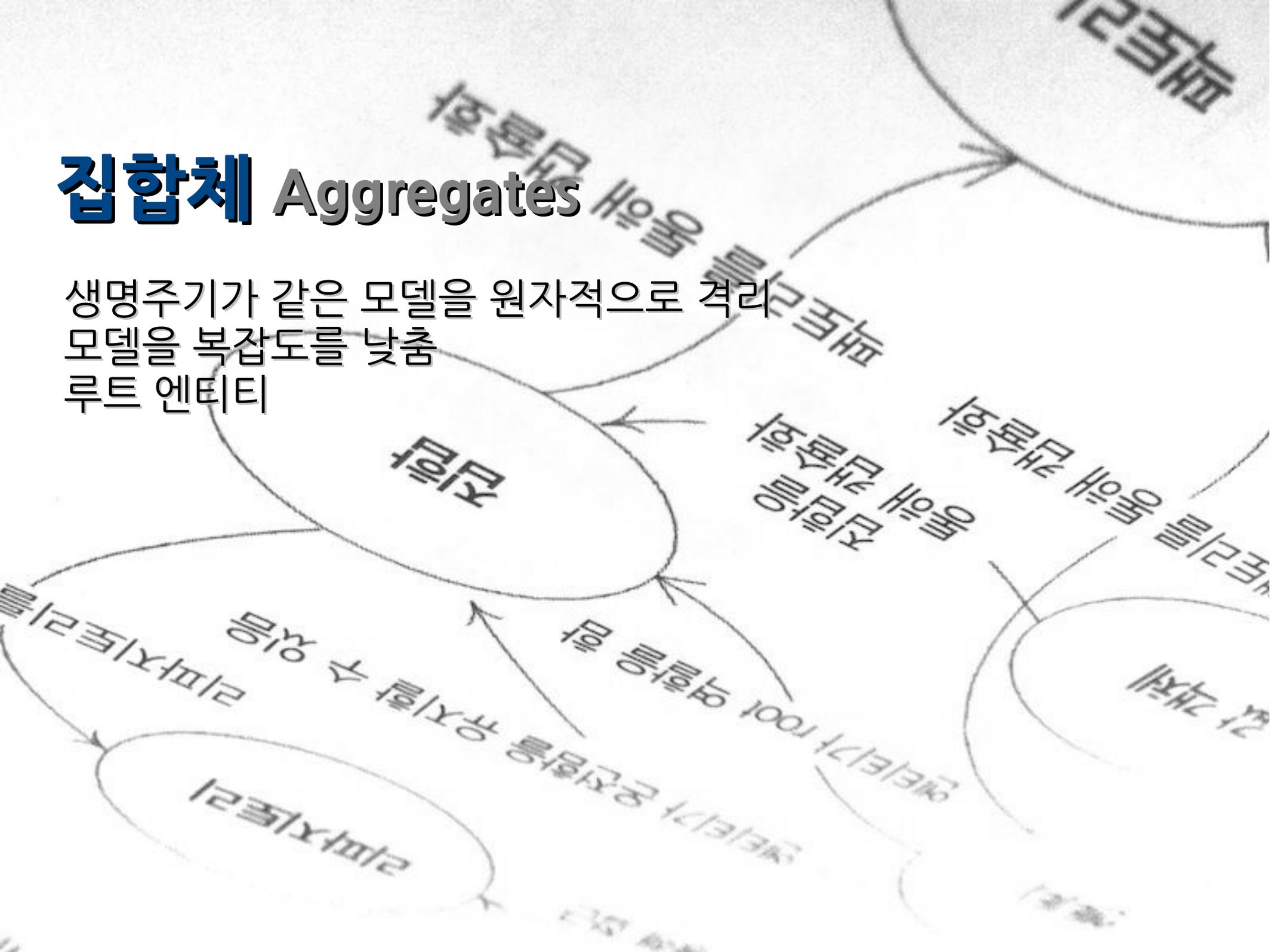


저장소 Repository

모델 객체를 보관하는 저장소
 보관 관련 기능을 인프라 분리하는 도메인 모델
 자바의 Map 인터페이스와 유사한 인터페이스 제공

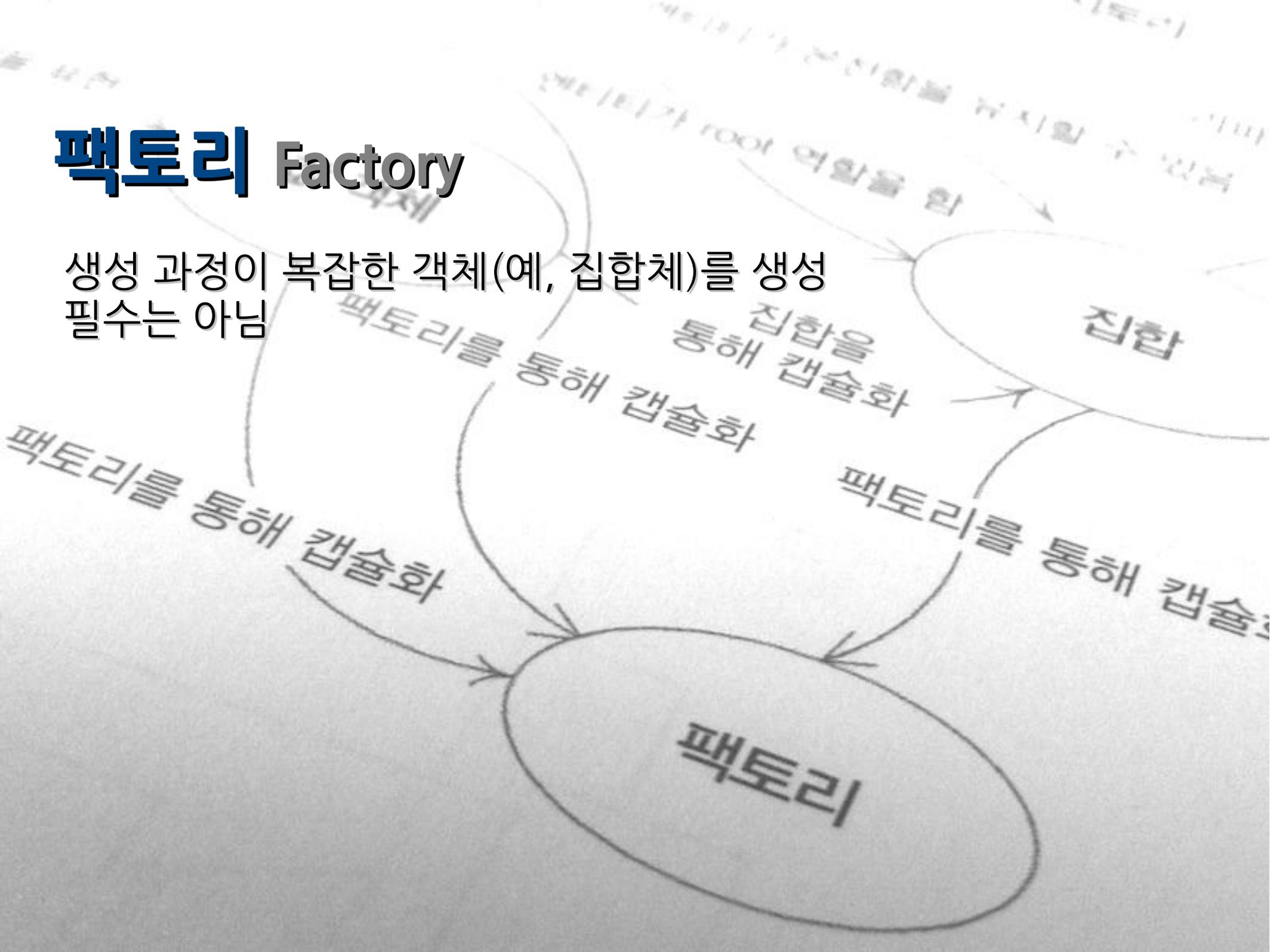
집합체 Aggregates

생명주기가 같은 모델을 원자적으로 격리
모델을 복잡도를 낮춤
루트 엔티티



팩토리 Factory

생성 과정이 복잡한 객체(예, 집합체)를 생성
필수는 아님



Talk is cheap.
**Show me
the
CODE!**

Linus Torvalds



DDDSample.sf.net

소개

에릭 에반스와 스웨덴 Citerus 사 제작

DDD 적용 예시 제공

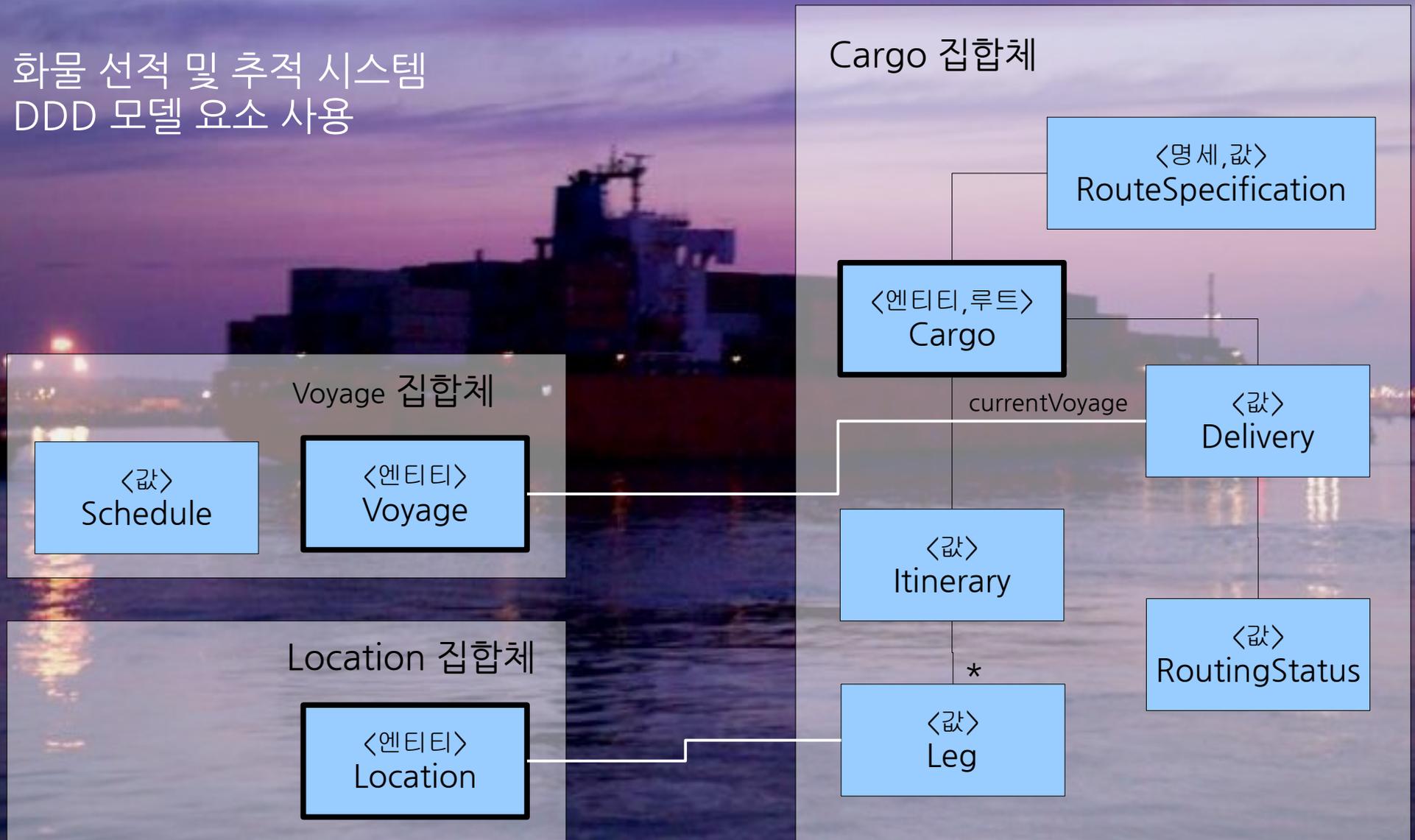
Cargo 예제 구현

다양한 시도와 토론 유도



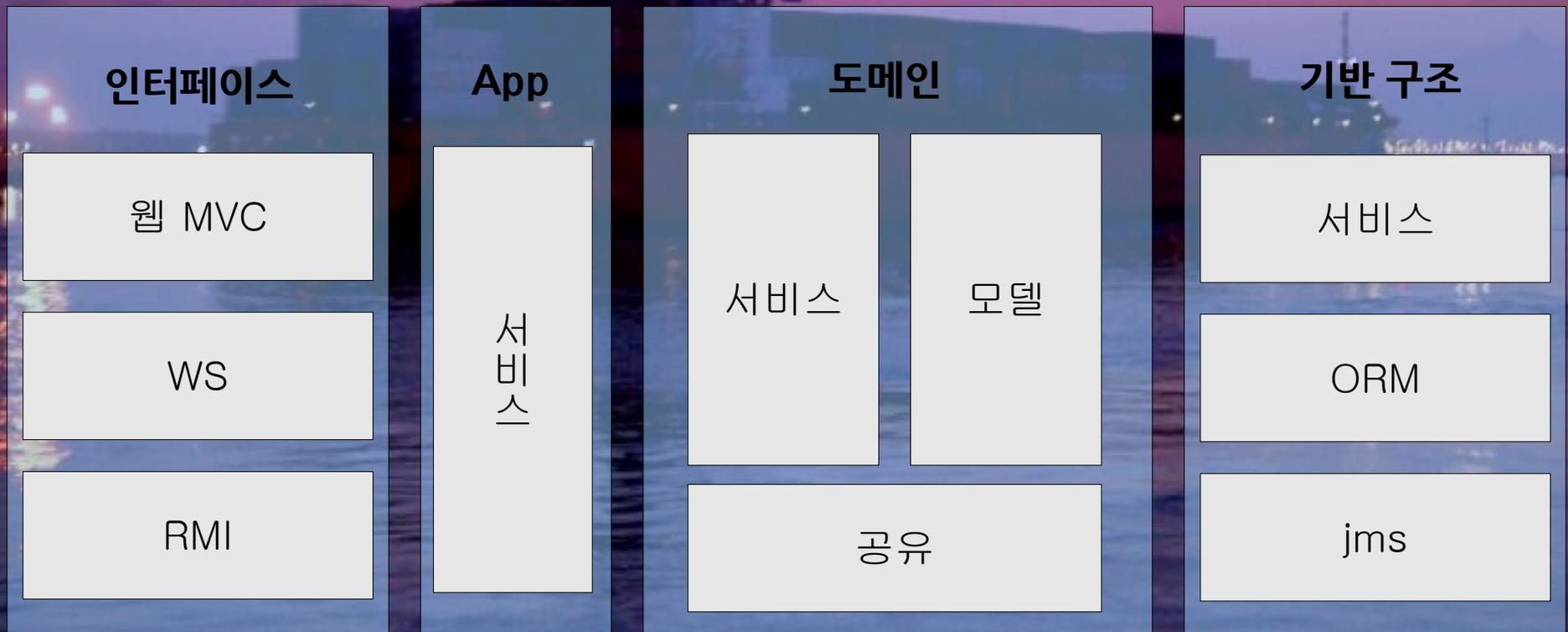
모델 구조

화물 선적 및 추적 시스템
DDD 모델 요소 사용



계층 구조

도메인을 기반 구조에서 분리
하부 계층에만 의존
계층마다 다른 의미의 서비스 존재



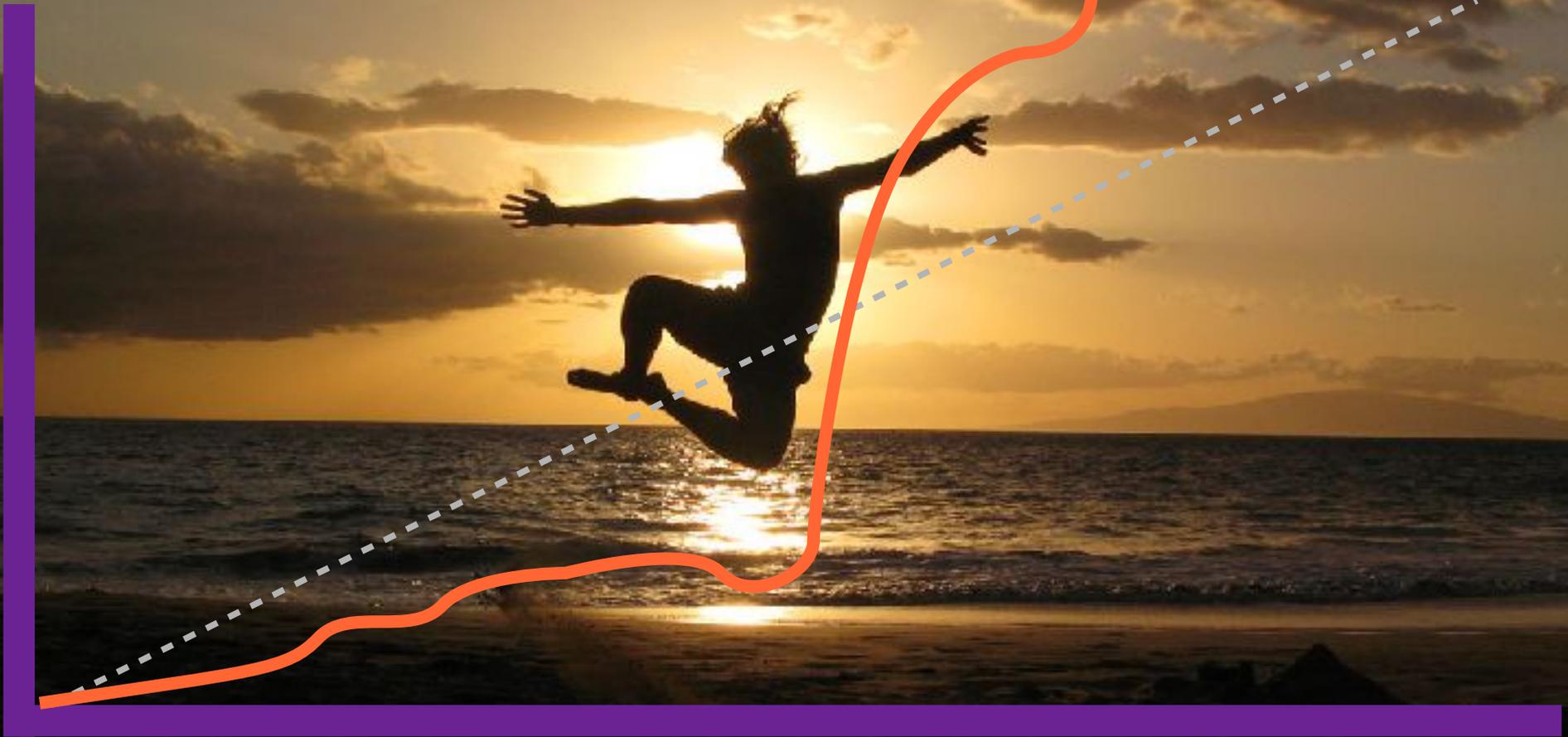
모델 리팩터링

설계 품질을 개선하는 리팩터링이 아닌 도메인에 대한 새로운 통찰을 반영하거나 모델의 표현력을 개선



도약!!

급진적인 리팩터링의 효과



A close-up photograph of a white ceramic coffee cup filled with a latte. The coffee has a thick, light-colored foam on top, decorated with a heart-shaped latte art design. The cup sits on a matching white saucer. A silver spoon is placed on the saucer in front of the cup. The background is dark with some colorful, abstract patterns.

자바는 DDD에 적합한가?

가장 많이 보급된 객체지향 언어
POJO 지원 프레임워크 발달
ORM 발달
객체지향, 패턴 보급
서술성 개선
동적 언어 지원 강화

관련기술

- 객체지향
- 패턴(설계, 분석, 아키텍처)
- 리팩터링
- 애자일(반복 개발)
- 테스트 자동화, TDD
- 지속적 통합
- DSL
- POJO/POCO/단순 객체
- 프레임워크
- ORM
- 함수형 프로그래밍

MDA?

한 독자의 부정적인 평가

The book is rather wordy, and a lot of common-sense ideas are repeated at length. I don't feel that the patterns in the book are much more than re-statements of basic principles of OO design

질문이나 토론은 **한국 스프링 사용자 모임**에서...

그룹스: <http://groups.google.com/group/ksug>

블로그: <http://www.ksug.org/>

