

Domain-Driven Design: The Big Picture

Introducing Domain-Driven Design



Sander Mak

Software developer & architect

@sander_mak

Domain-Driven Design



Course Overview

Introducing
Domain-Driven
Design

Strategic DDD

Tactical DDD

Getting Started
With DDD

Who Is This Course For?



Developer



Architect



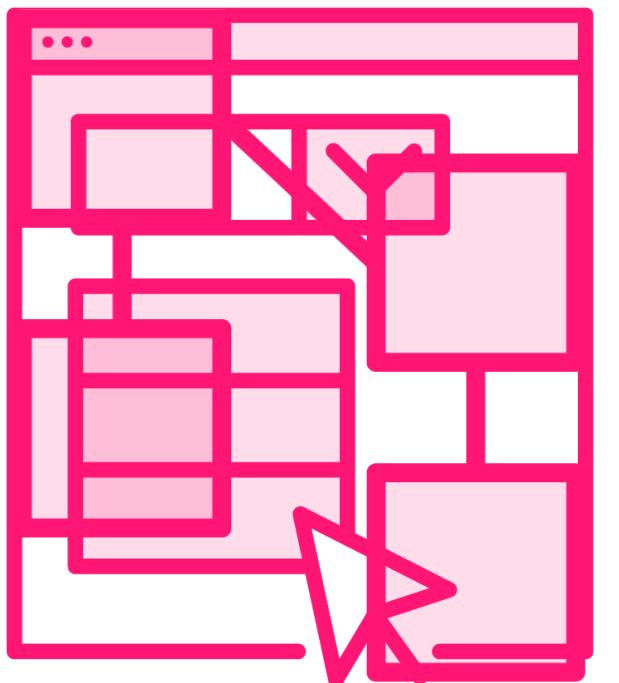
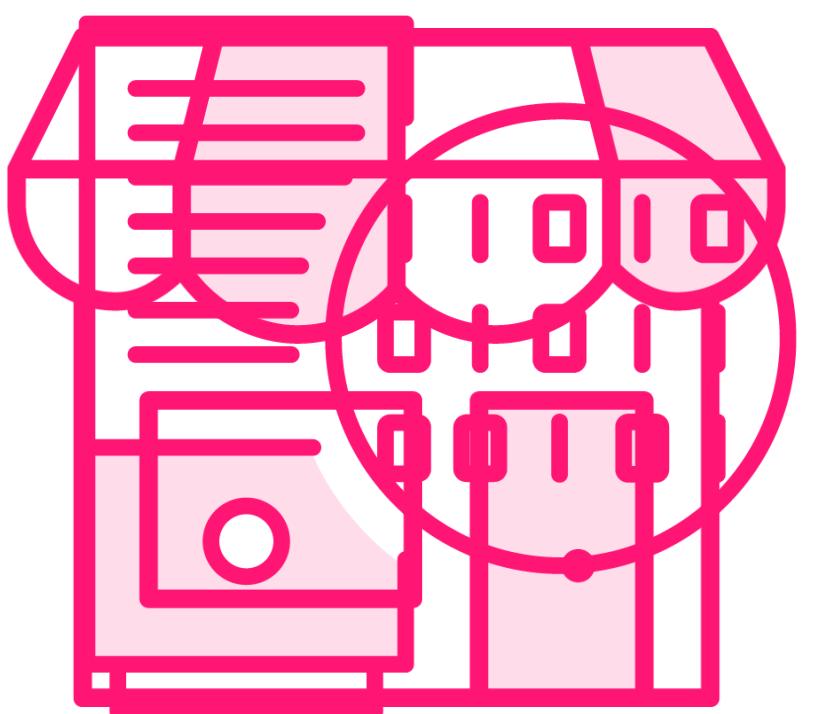
Manager

DDD is technology-agnostic

What is DDD?

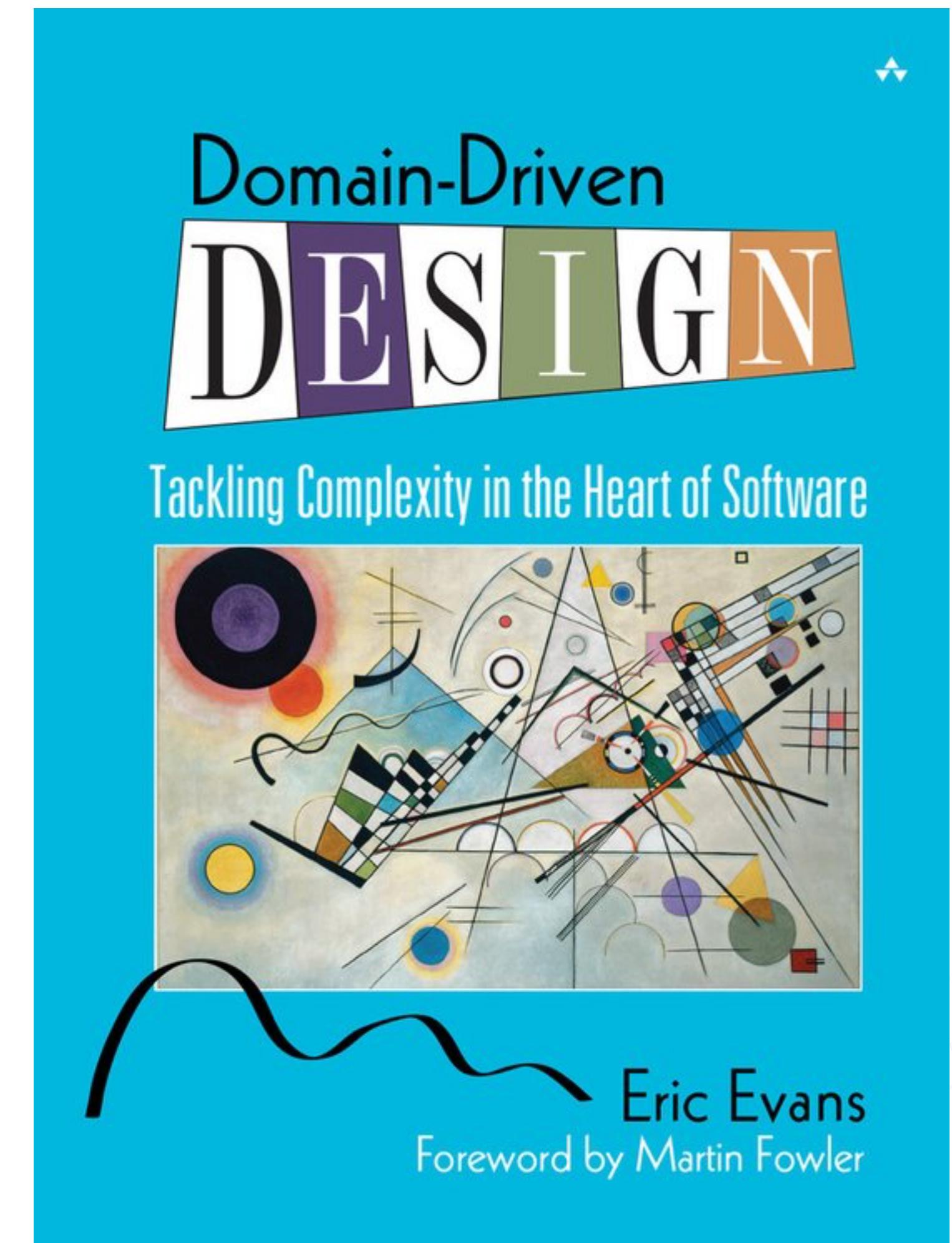
Domain-Driven

Design



Tackling complexity in the heart of software

"The blue book"

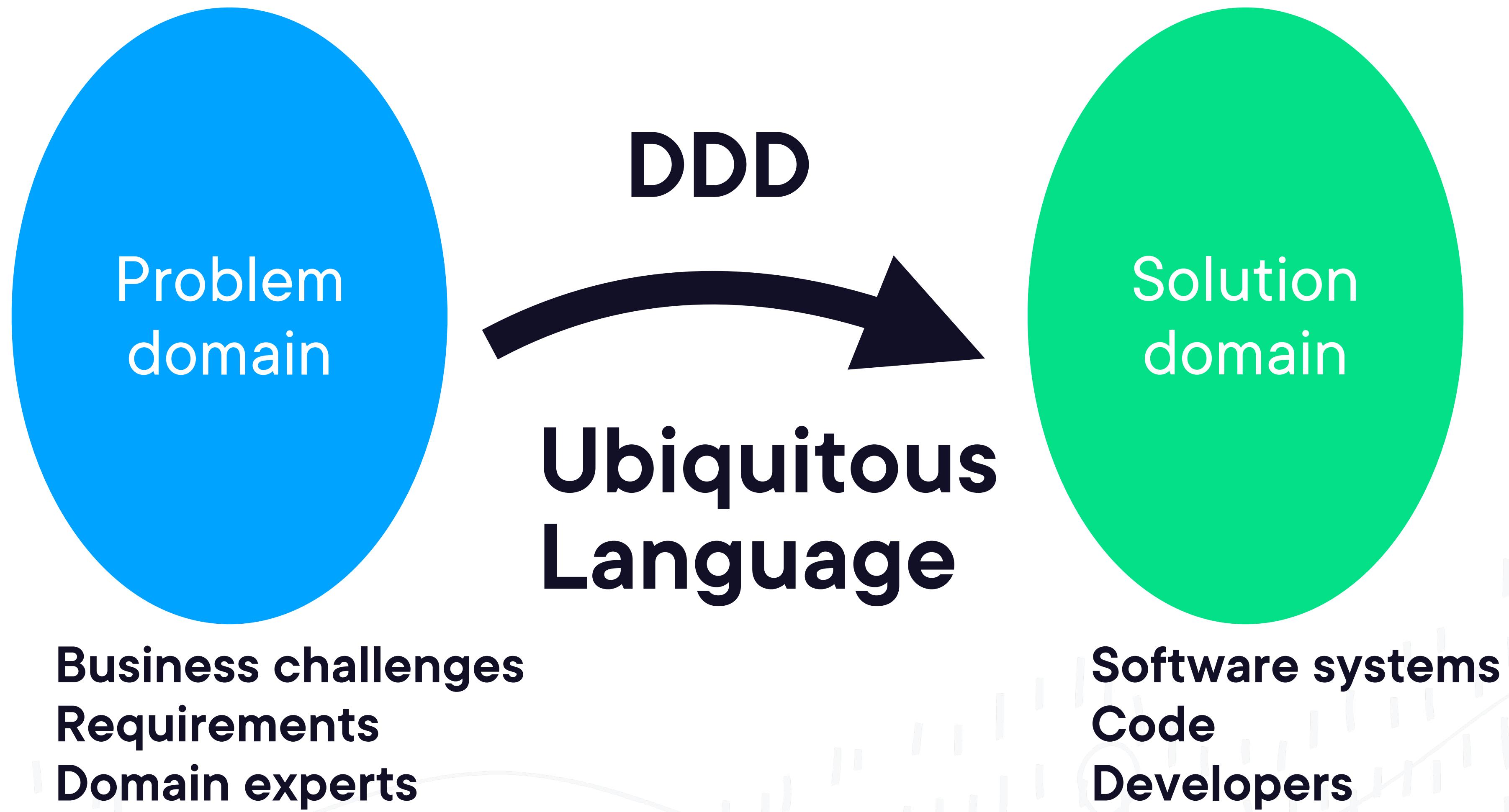


Eric Evans, 2003

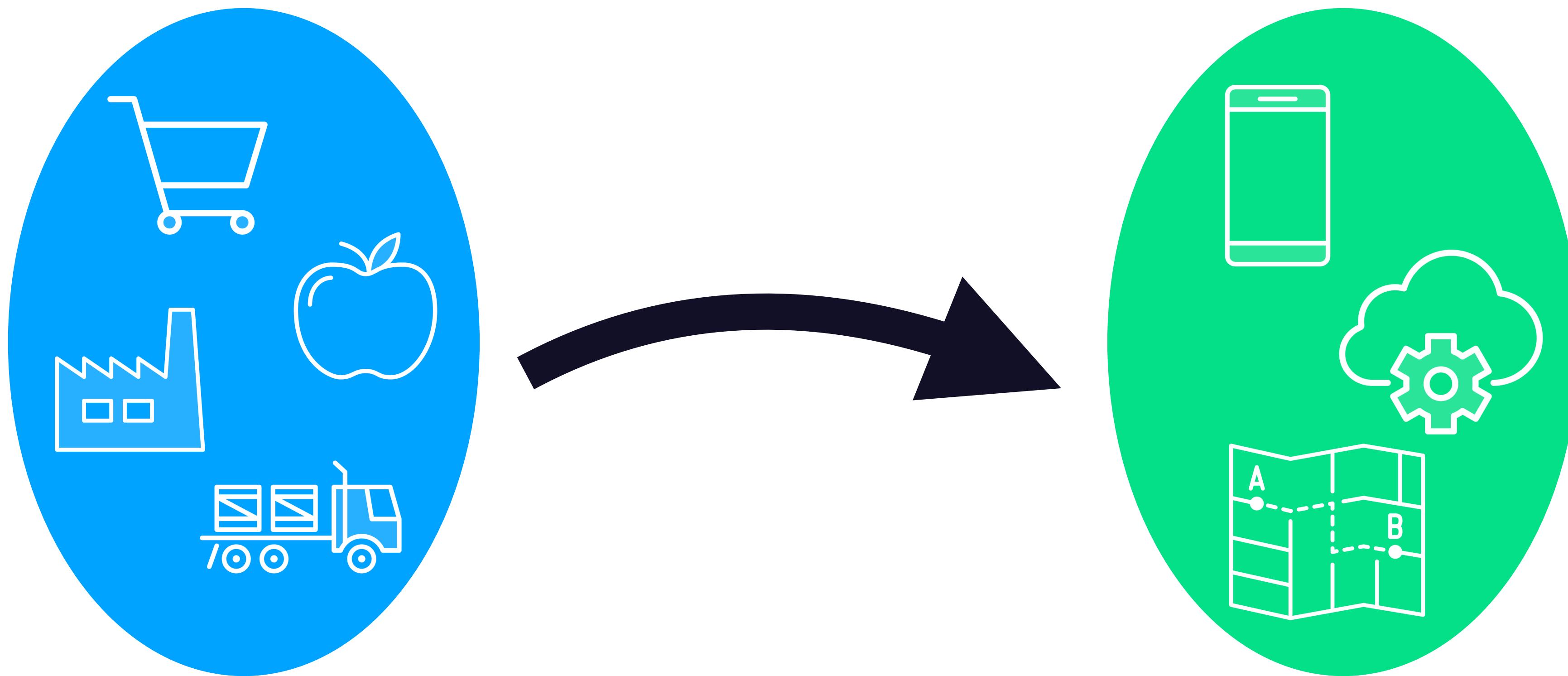
Martin Fowler

“Domain-Driven Design is an approach to software development that centers the development on programming a domain model that has a rich understanding of the processes and rules of a domain.”

What Is a Domain?



What Is a Domain?



Online groceries

Why Design?



Think before you code!

Uncover risks & assumptions

Avoid costly rework

Share knowledge

Why Design?



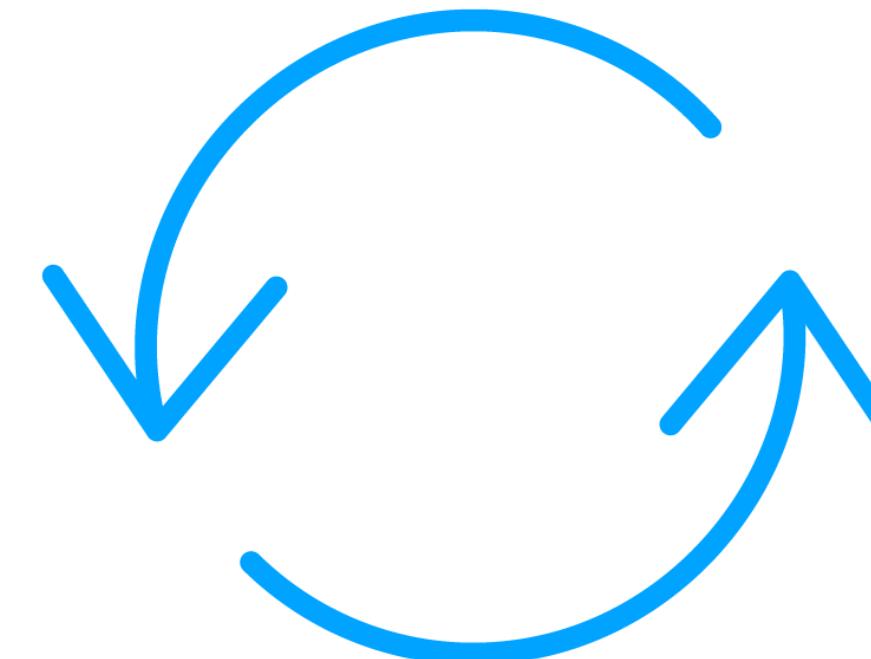
Big Design Upfront



Why Design?

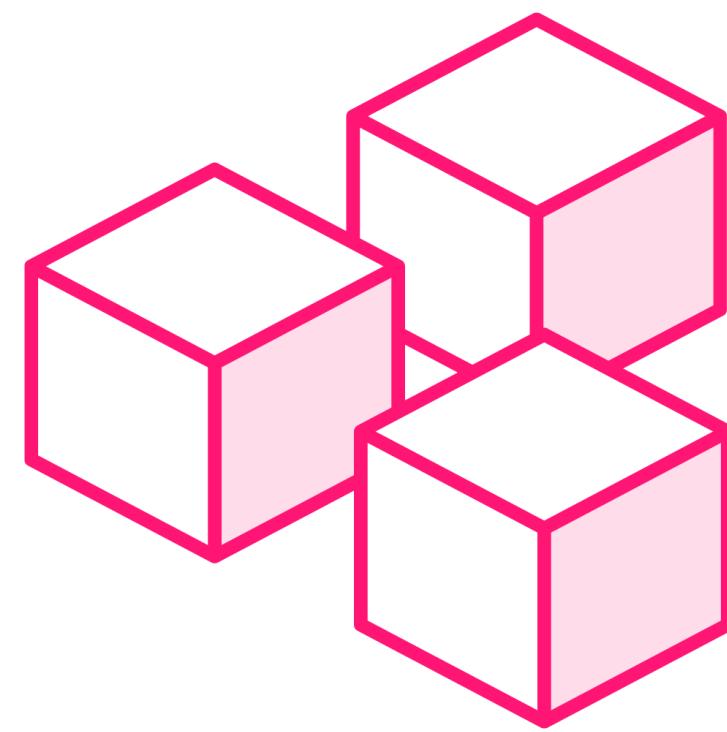


Incremental design

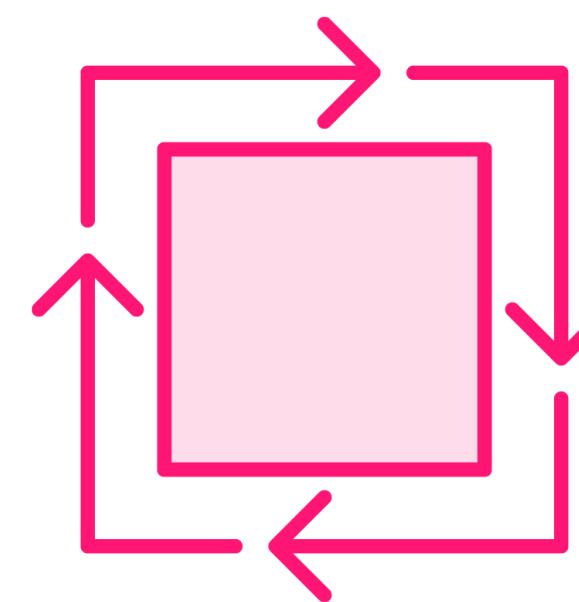


with domain experts

Why Do We Need Domain Models?



Abstraction

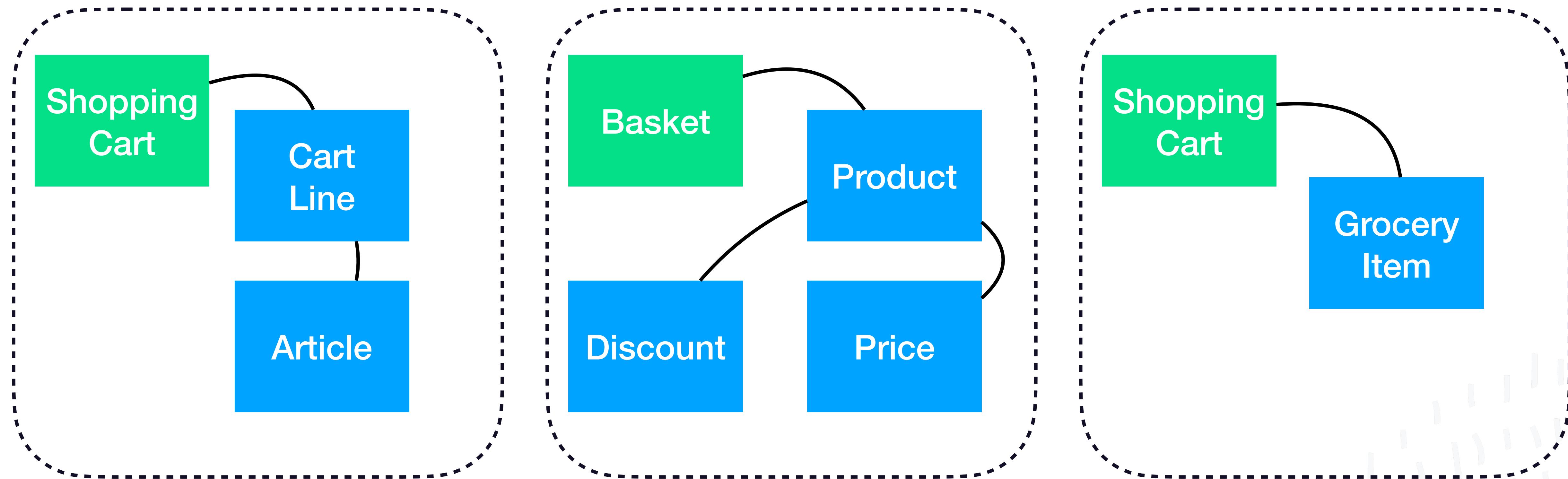


Boundaries



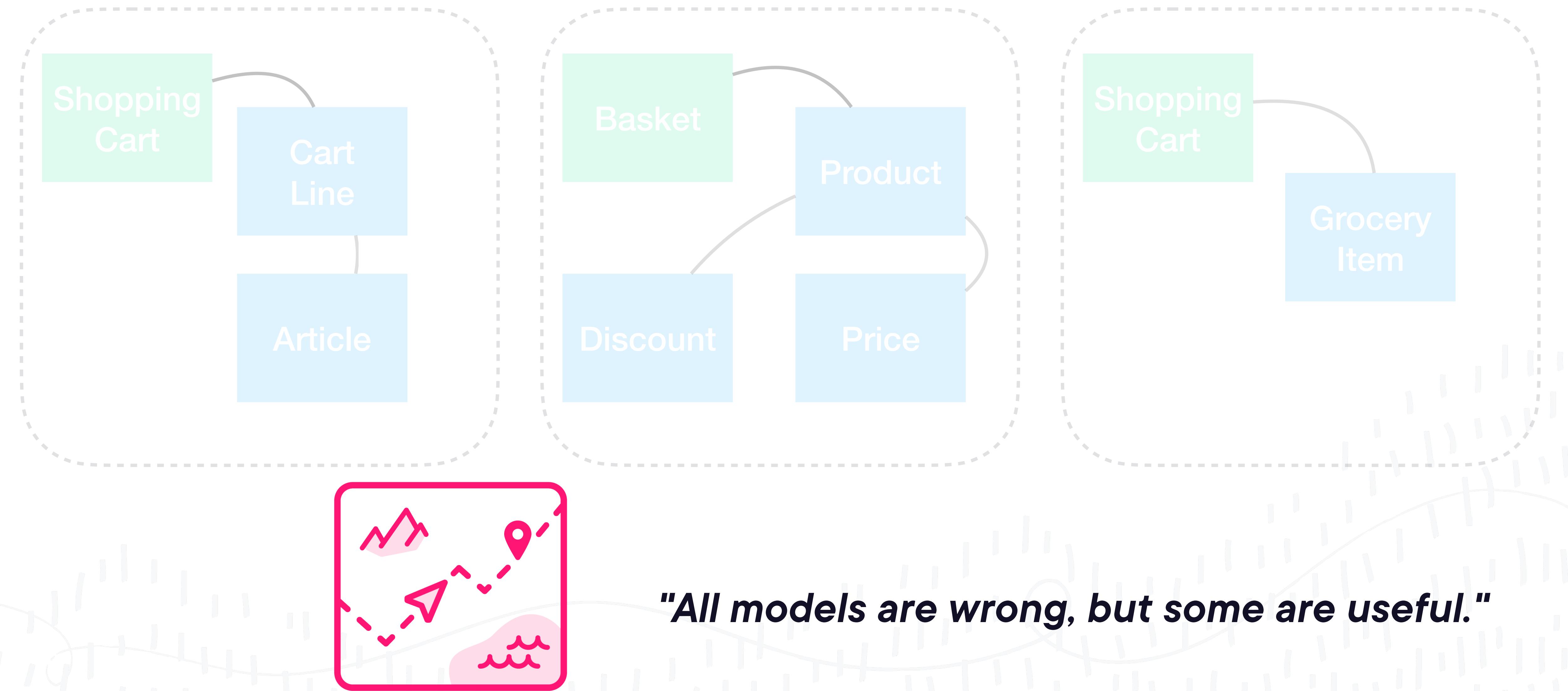
Shared understanding

Why Do We Need Domain Models?

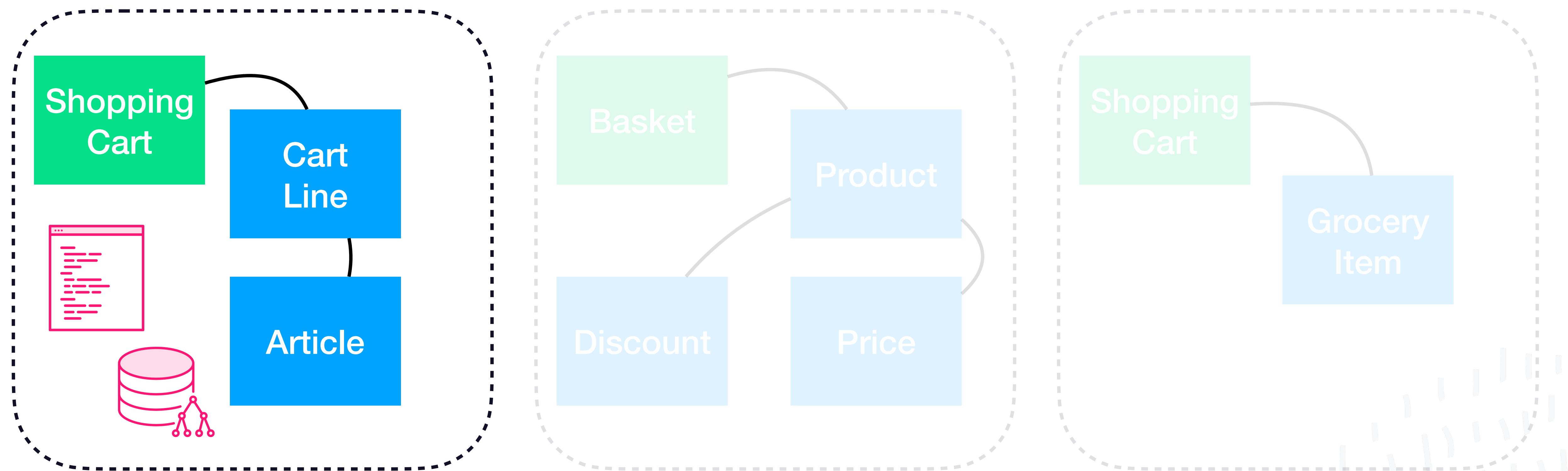


"All models are wrong, but some are useful."

Why Do We Need Domain Models?



Why Do We Need Domain Models?



DDD Concepts



DDD Concepts

Ubiquitous Language

Bounded Context

Strategic



DDD Concepts

Entities & Value Objects

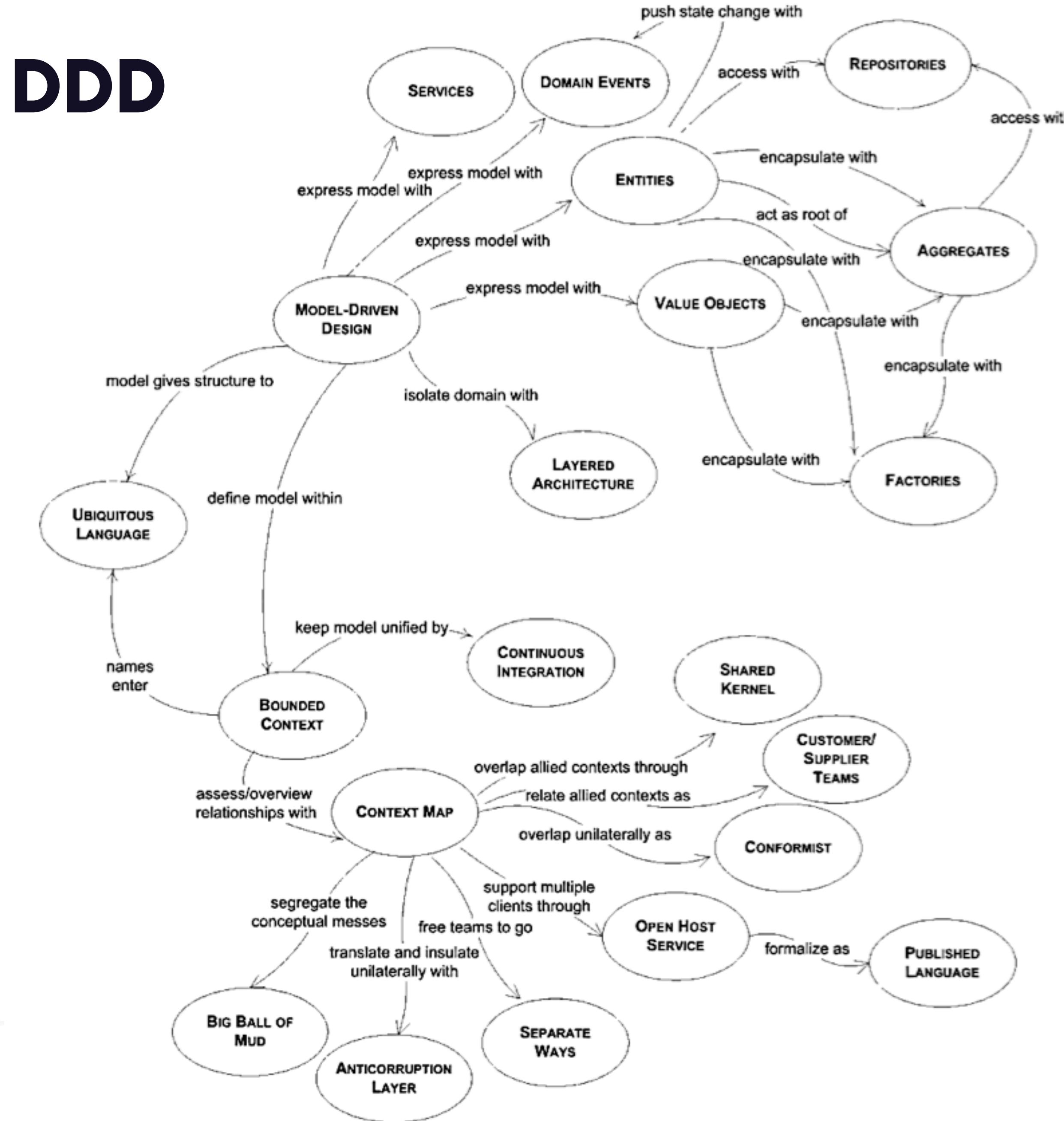
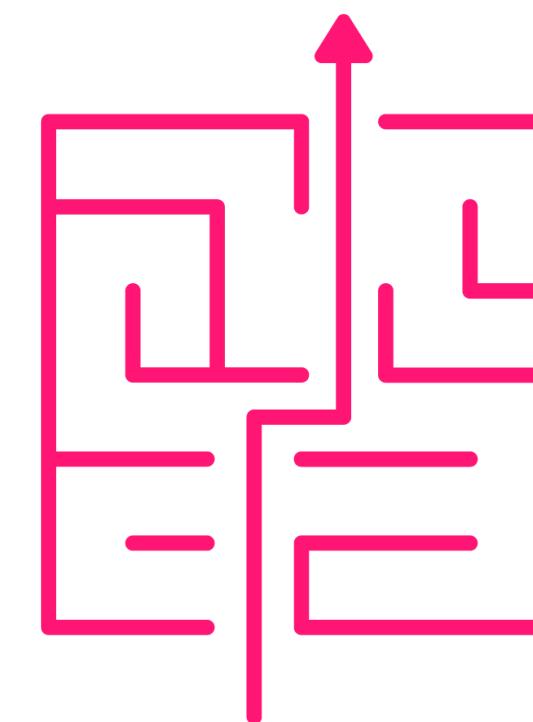
Aggregates

Repositories

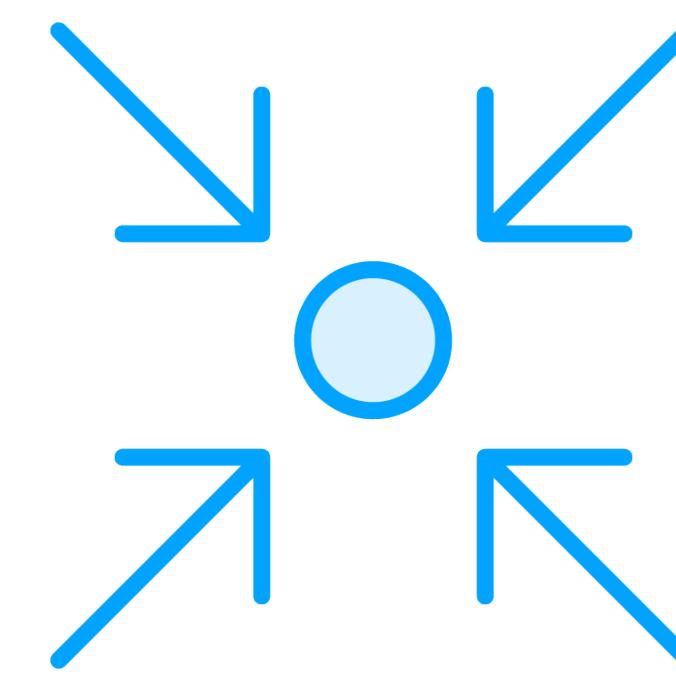
Services



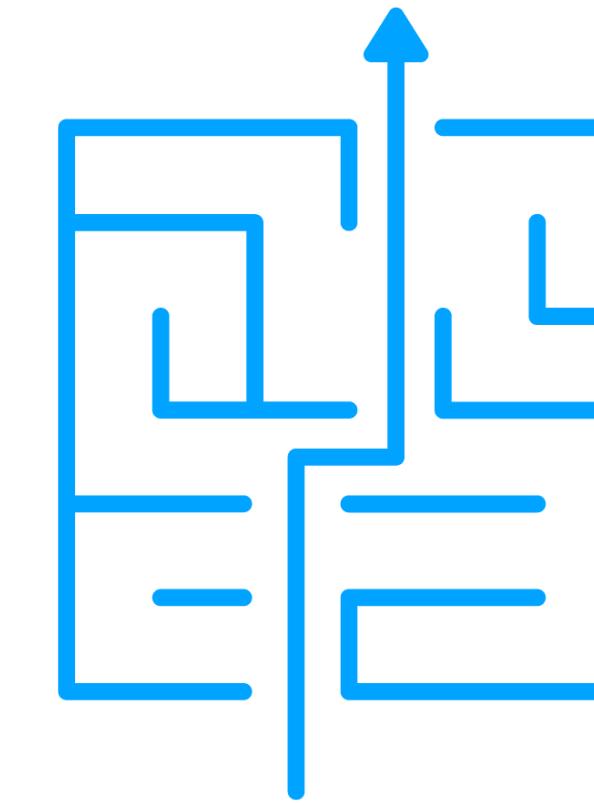
When (Not) to Use DDD



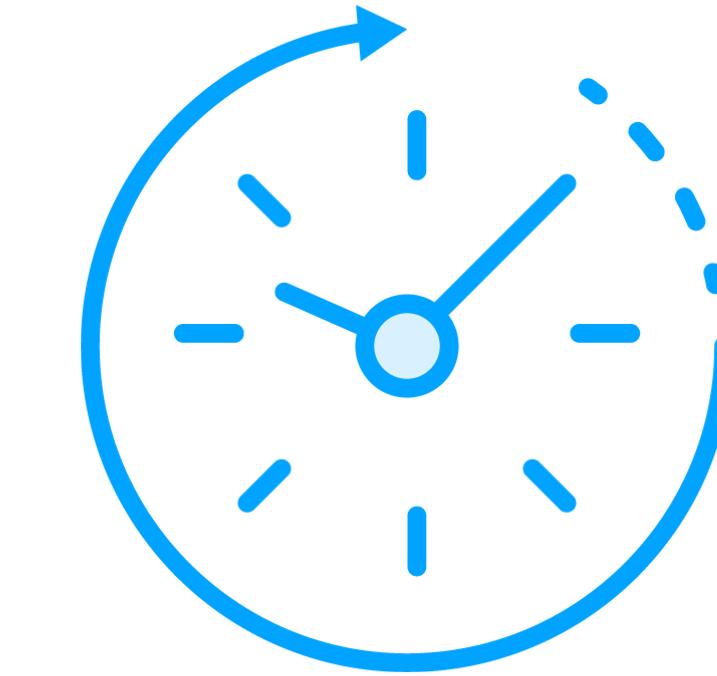
When to Use DDD



Core domain



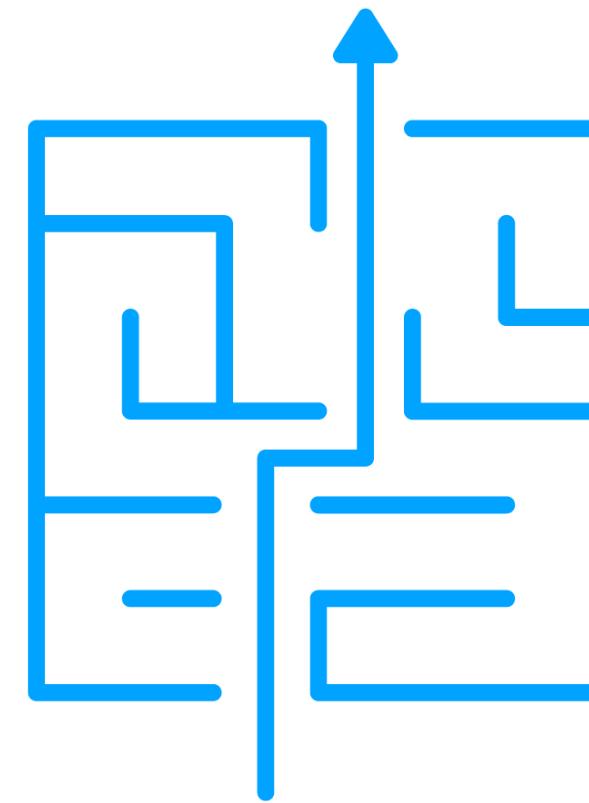
Complexity



Long-term evolution



When Not to Use DDD



**Complexity only
technical in nature**

