Getting Started With DDD

Sander Mak

Software developer & architect

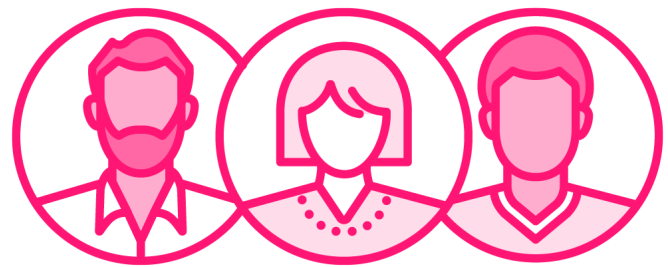@sander_mak

Looking Back

Domain model

Ubiquitous Language
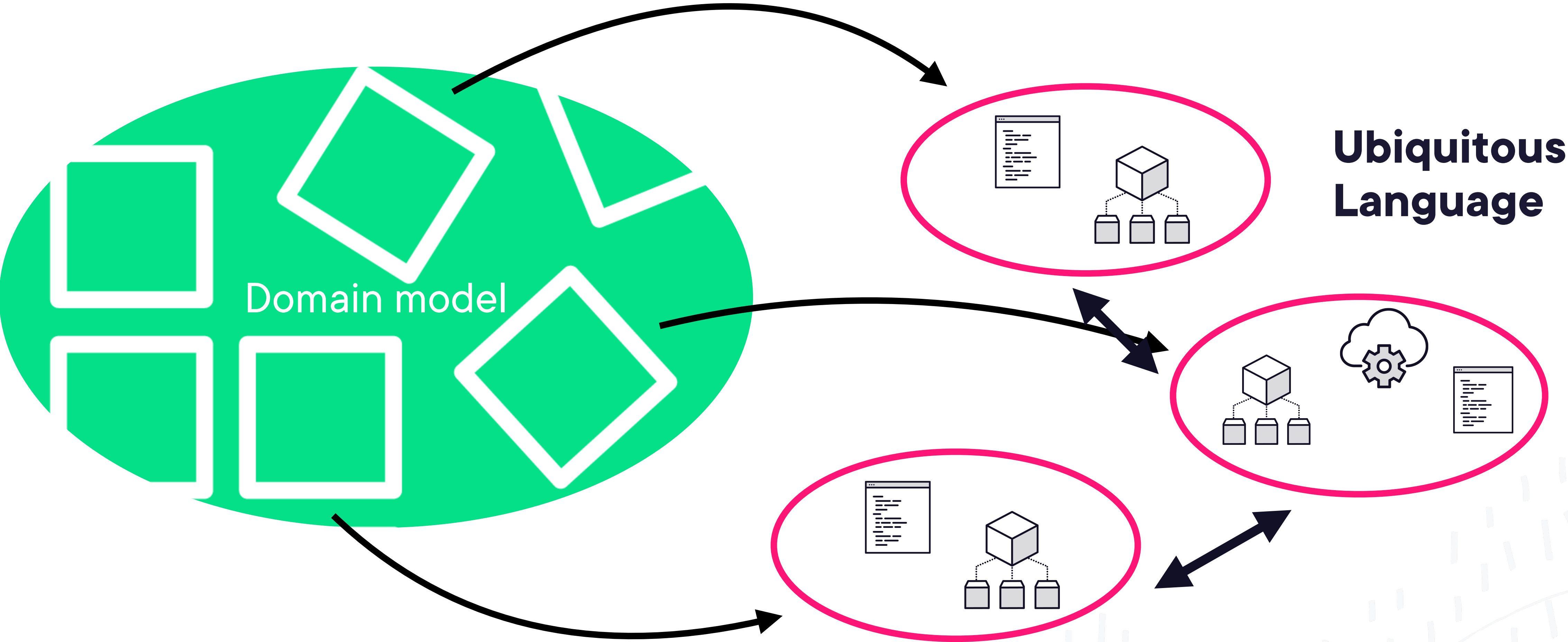
Adopting DDD

Adopt incrementally, start small

Work with the business
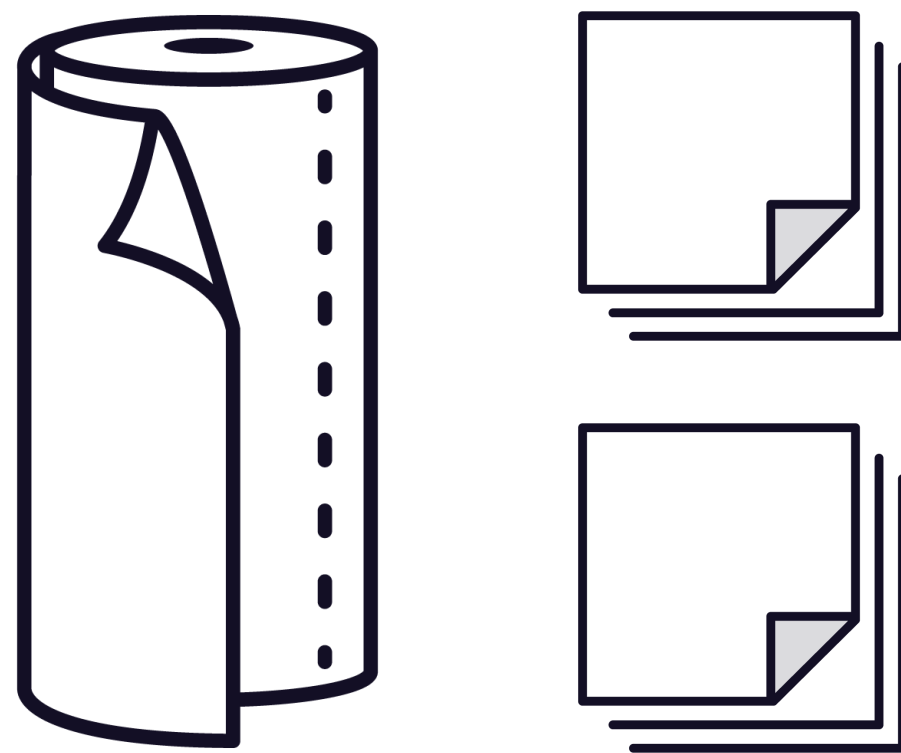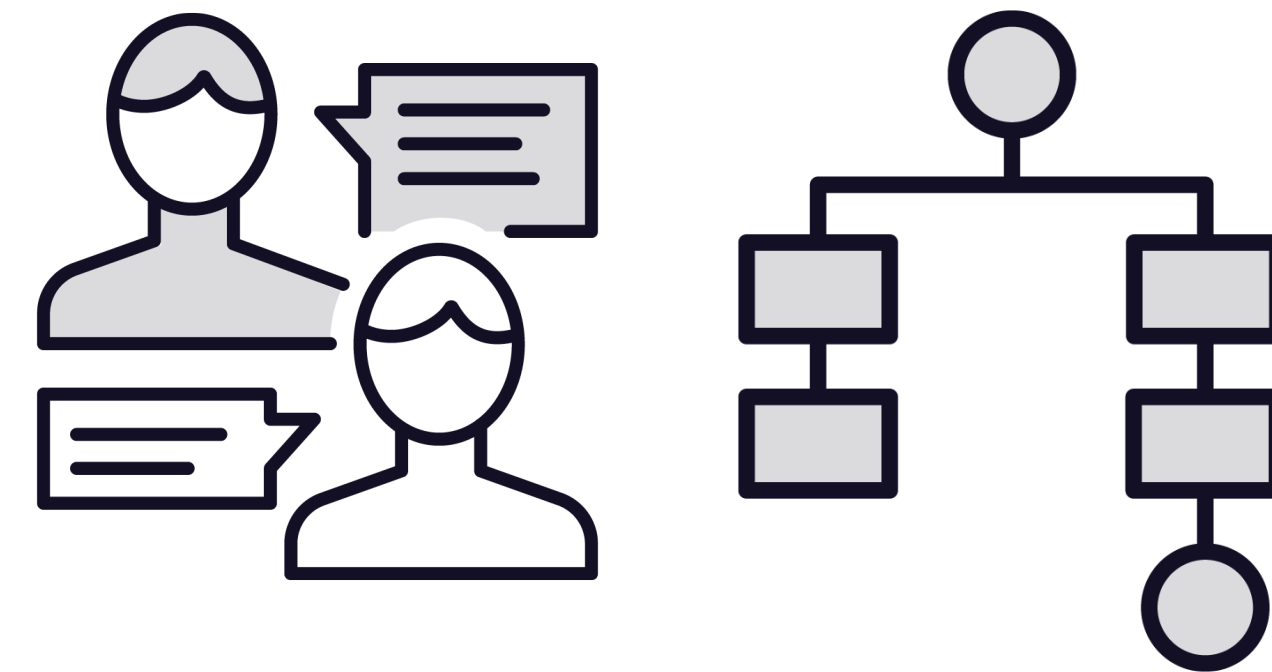
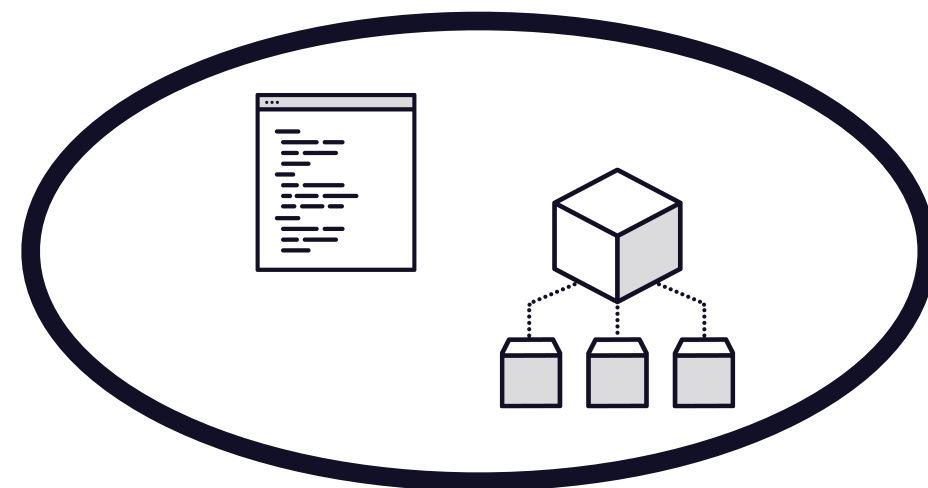Use DDD as toolbox

The Broader DDD Ecosystem

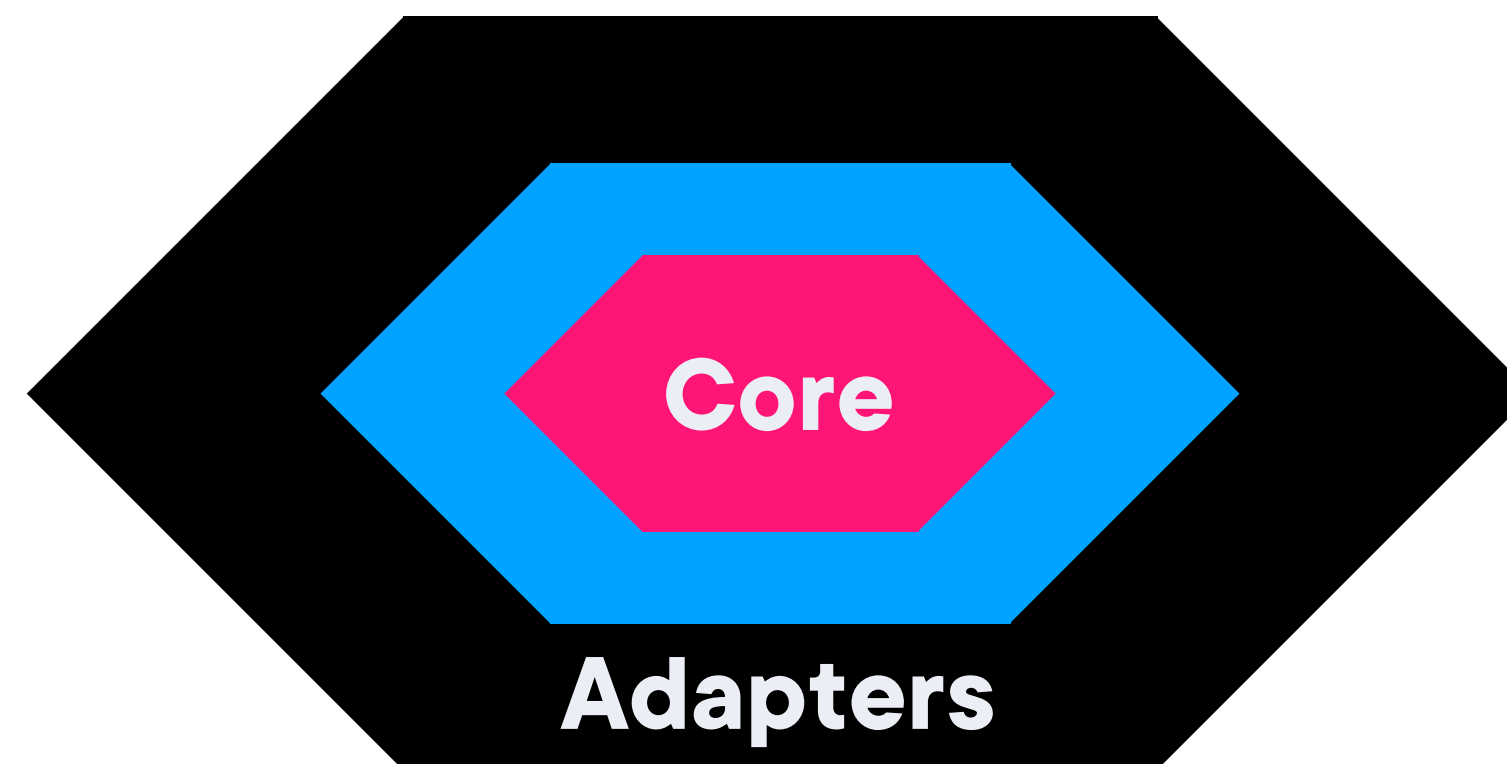**Event storming**

Domain storytelling

Requirements & analysis

The Broader DDD Ecosystem

Microservices
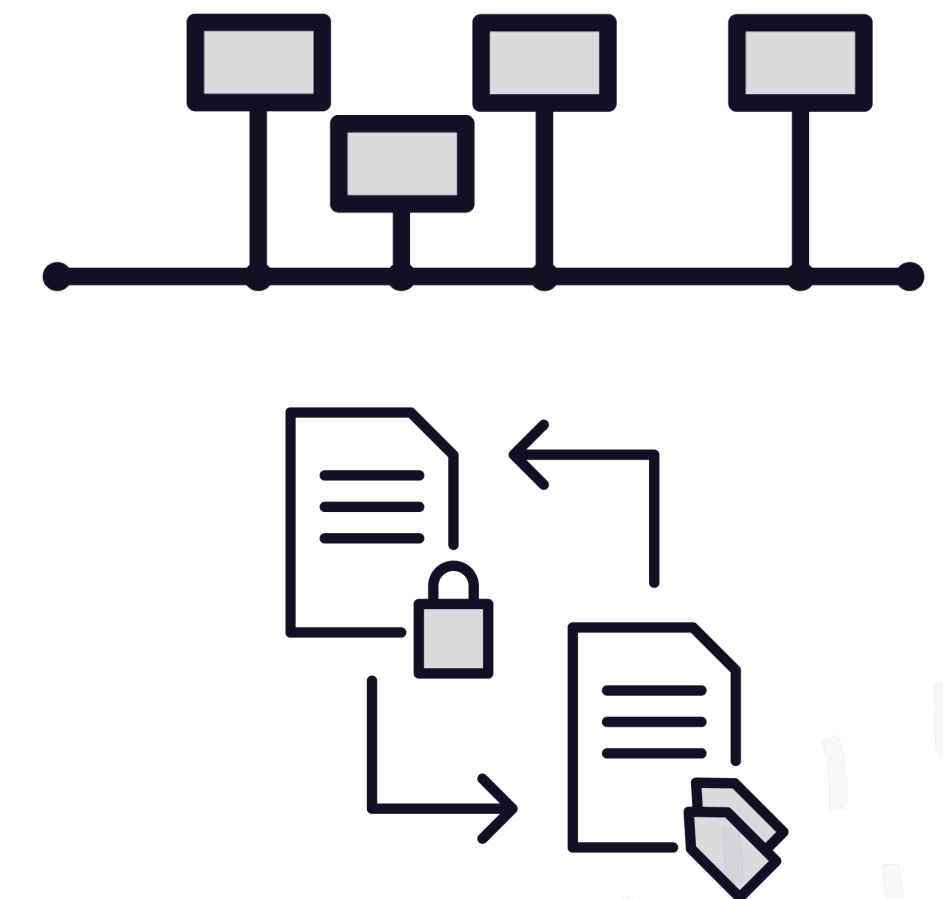
Onion / Hexagonal Architecture

Core

Adapters

Event sourcing CQRS

Architectural styles

Command Query Responsibility Segregation

Additional Resources

Name:		V5 github.com/ddd-crew/bounded-context-canvas		CC BY

Purpose

What benefits does this context provide, and how does it provide them? Describe the purpose from a business perspective

Strategic Classification

Domain	Business Model	Evolution
- core	- revenue	- genesis
- supporting	- engagement	- custom built
- generic	- compliance	- product
- other?	- cost reduction	- commodity

Domain Roles

Role Types
- draft context
- execution context
- analysis context
- gateway context
- other

Inbound Communication

Collaborator   Messages

`<Query>` `<Command>` `<Event>`

Ubiquitous Language

Context-specific domain terminology

`<Domain Term>`
`<definition>`

Business Decisions

Key business rules, policies, and decisions

`<Decision>`

Outbound Communication

Messages   Collaborator

`<Query>` `<Command>` `<Event>`

Assumptions

Describe which currently unverified assumptions went into this bounded context design. Make those assumptions explicit by documenting them here

Verification Metrics

Describe metrics which can be used to (in)validate the current structure of this bounded context?
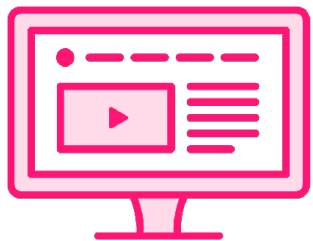
Open Questions

github.com/ddd-crew
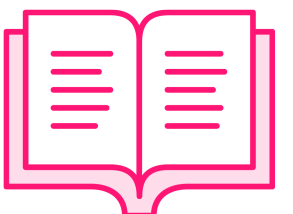**canvases**

Additional Resources

Domain-Driven Design Fundamentals
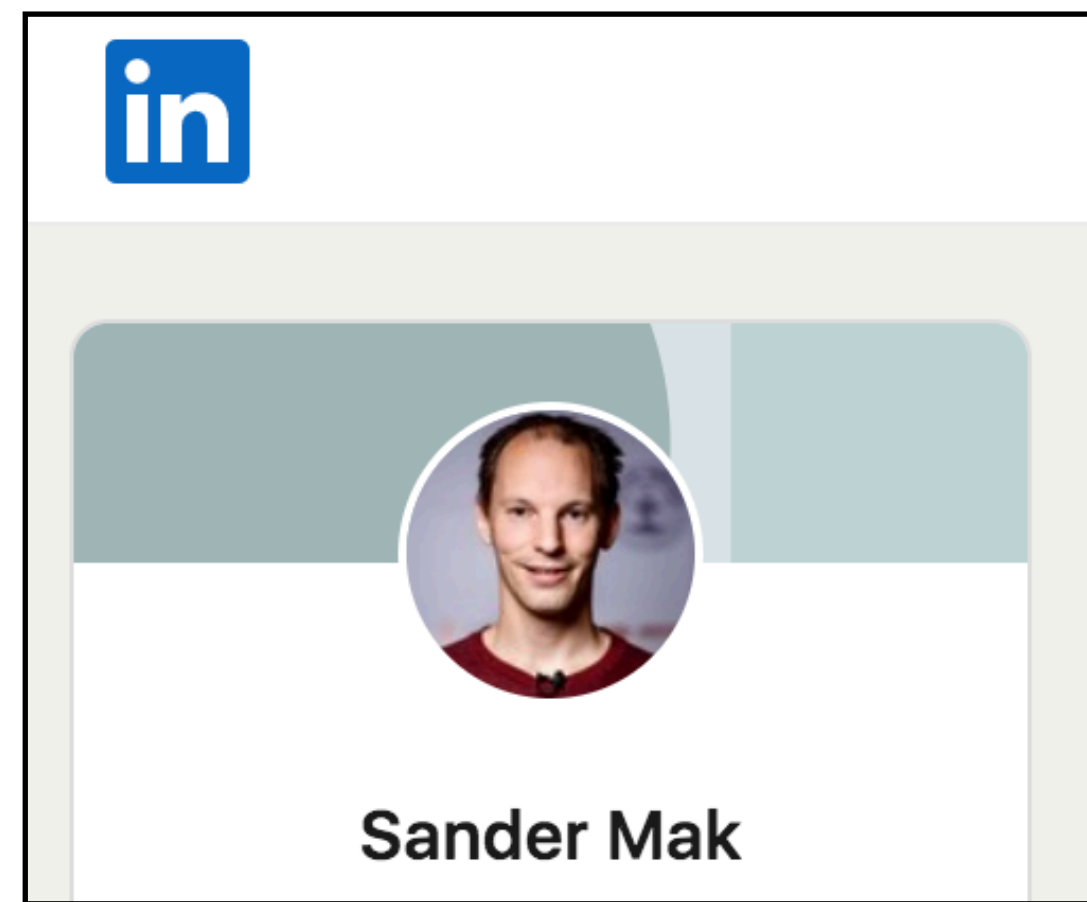
Domain-Driven Design in Practice

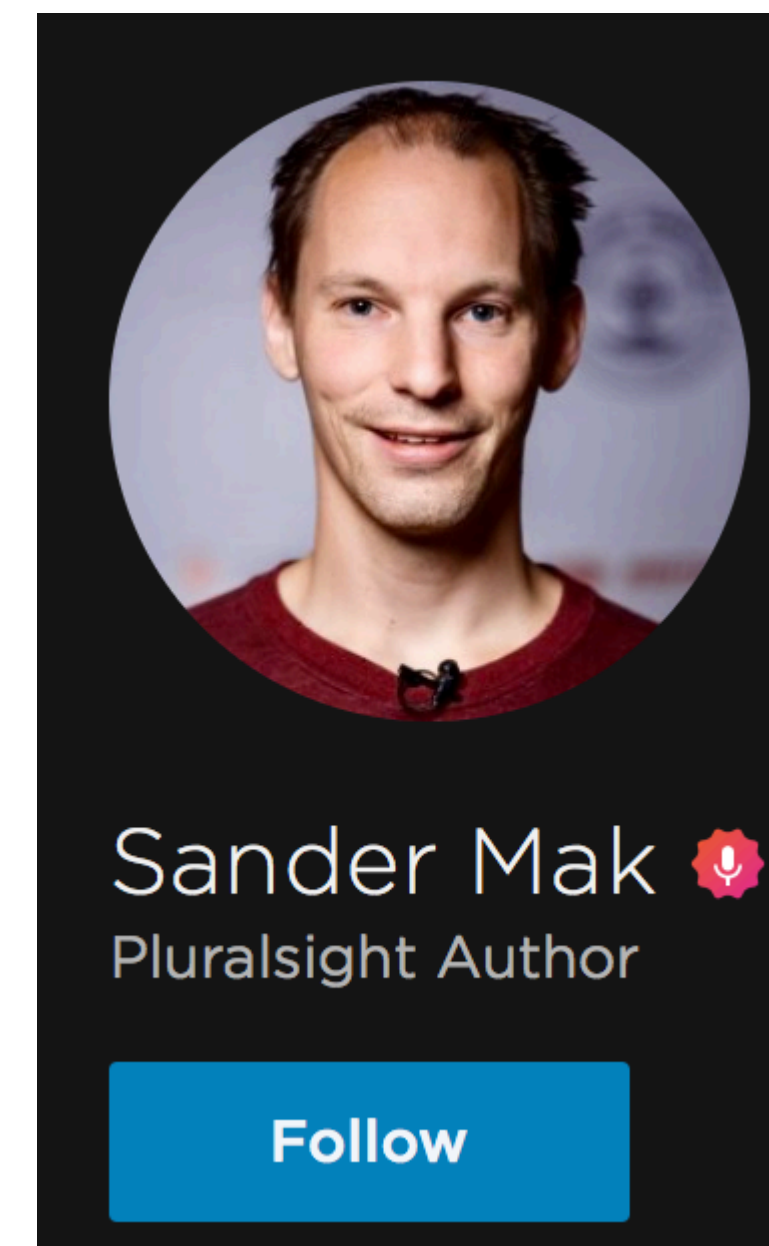Learning Domain-Driven Design, Vlad Khononov

Implementing Domain-Driven Design, Vaughn Vernon

Additional Resources

Follow for course updates

LinkedIn

bit.ly/ps-sander