

# Introducing the Second Bounded Context



Vladimir Khorikov

@vkhorikov | [www.enterprisecraftsmanship.com](http://www.enterprisecraftsmanship.com)

---

# In This Module

Bounded Contexts  
vs Sub-domains

Boundaries

Context Mapping

Types of isolation

Communication

Code Reuse

# Problem Description



- Dispense cash
- Charge the user's bank card
- Keep track of all money charged

# Bounded Contexts

Separation of the model into smaller ones

# Bounded Contexts

- 
- Boundary for the ubiquitous language

# Bounded Contexts

## Snack Machine

CompositeElement

- ☐ Attribute 1
- ☐ Attribute 2
- ☐ Method 1

## ATM

CompositeElement

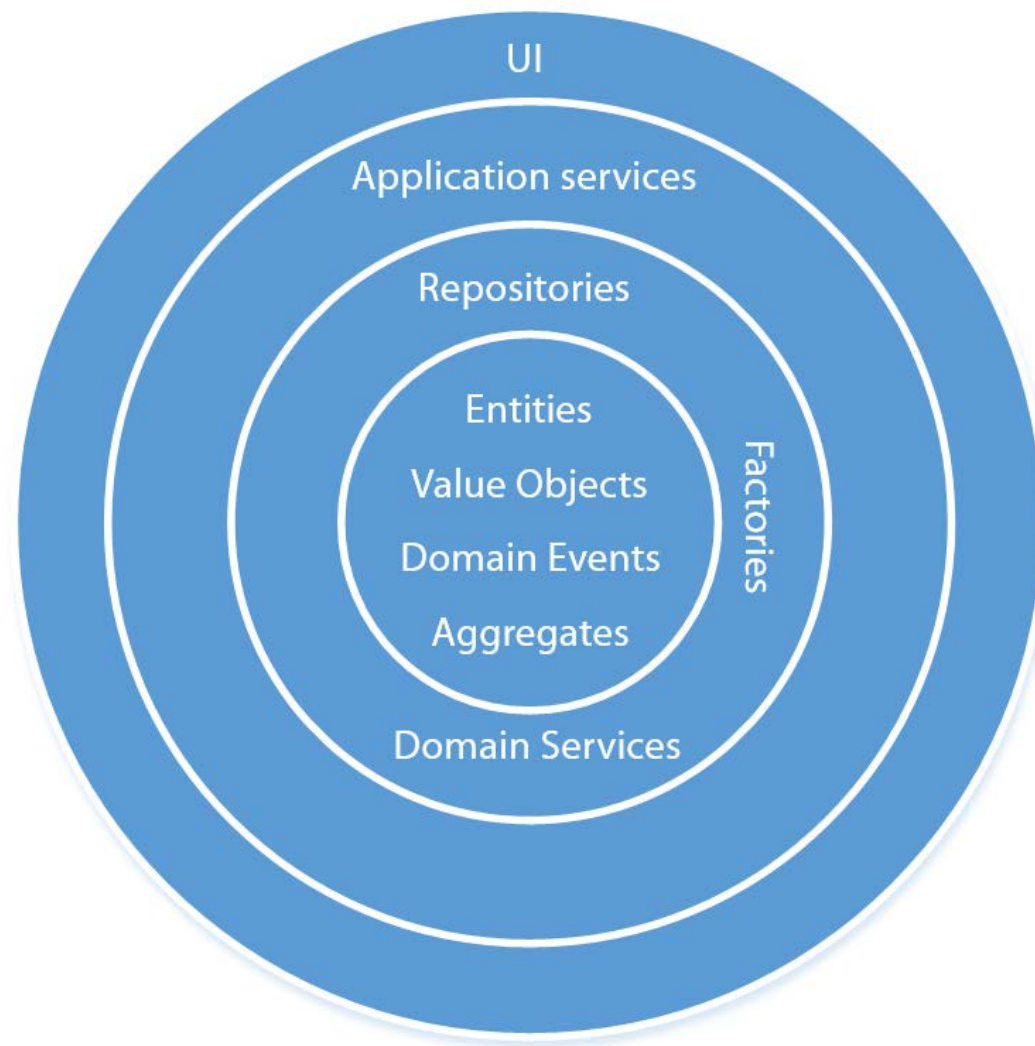
- ☐ Attribute 3
- ☐ Attribute 4
- ☐ Method 2

# Bounded Contexts

- 
- Boundary for the ubiquitous language

- 
- Span across all layers in the onion architecture

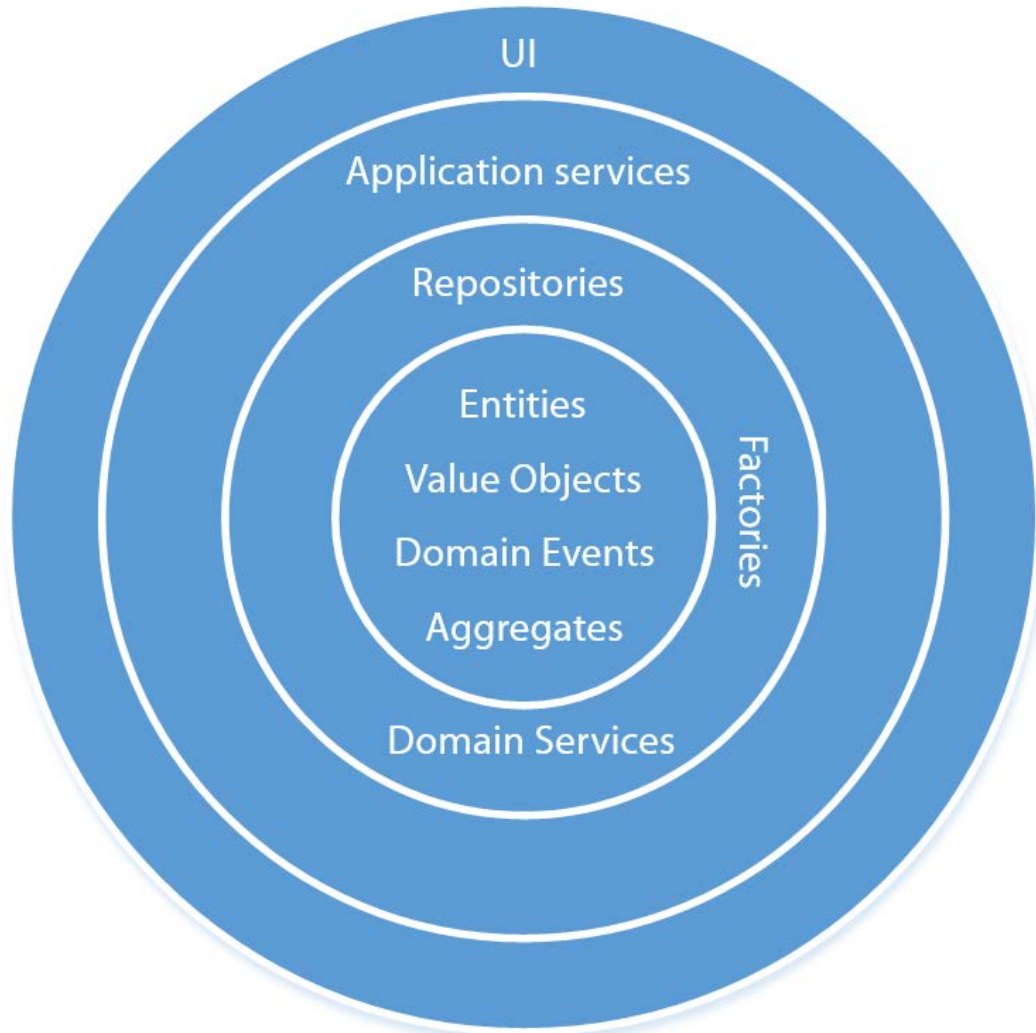
# Bounded Contexts



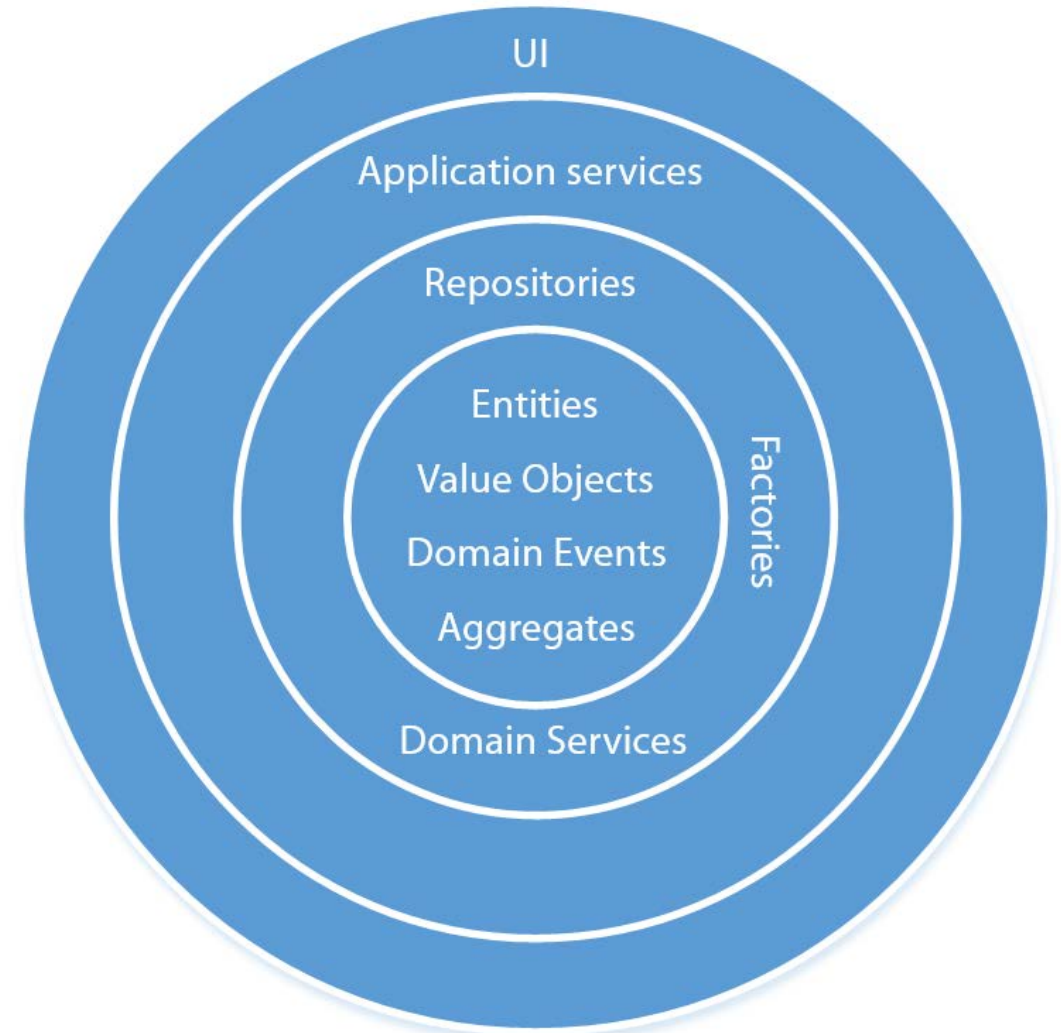


# Bounded Contexts

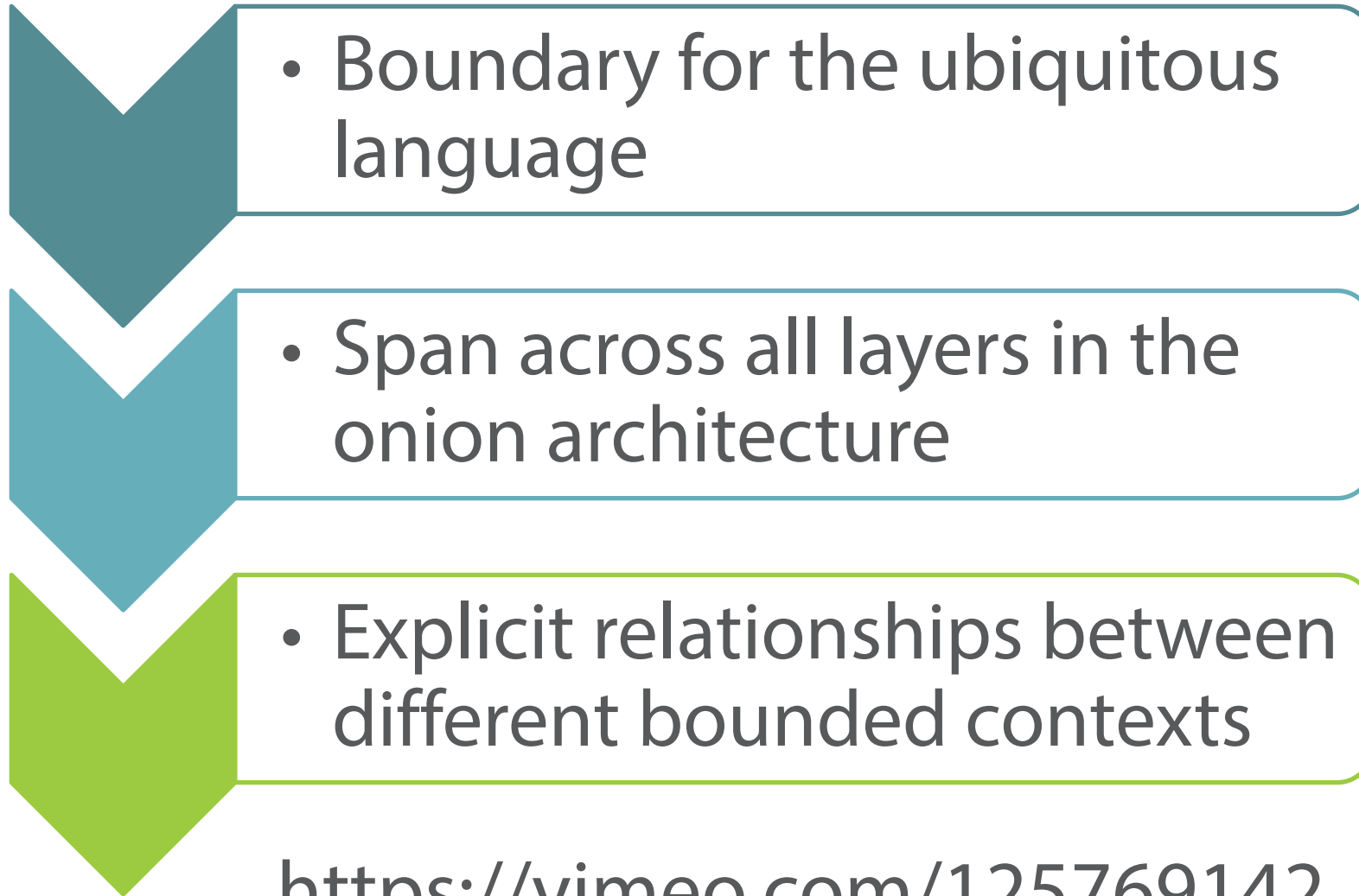
Bounded Context 1



Bounded Context 2

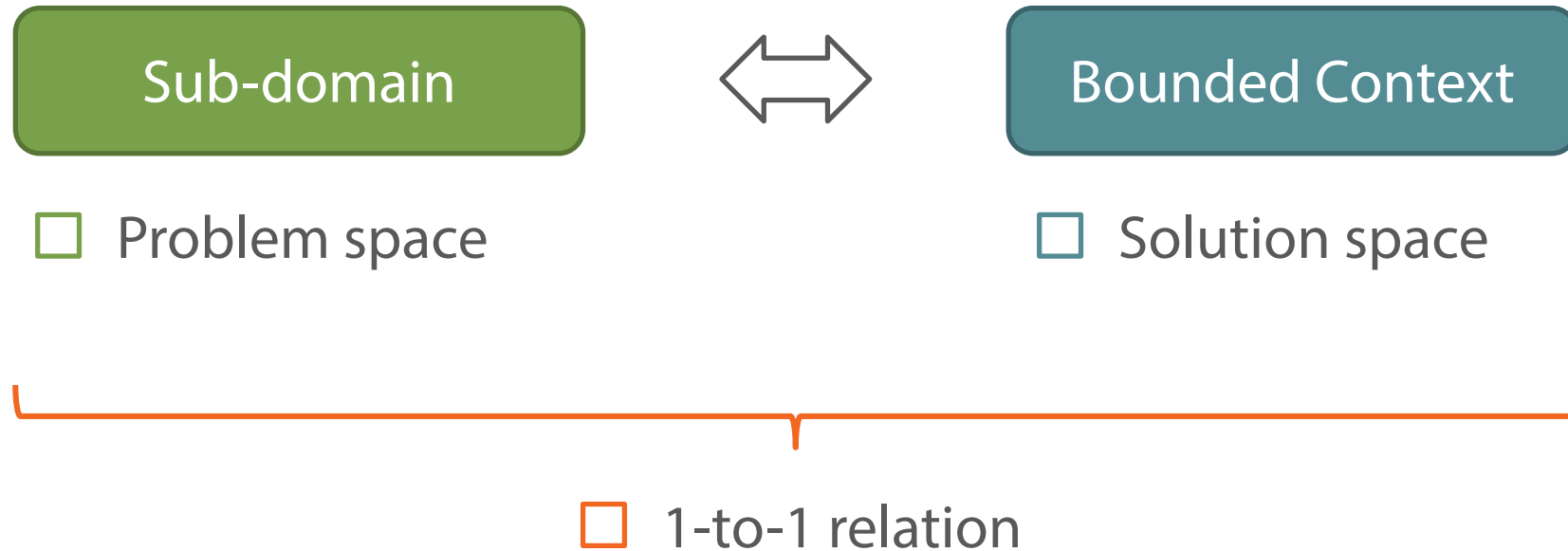


# Bounded Contexts

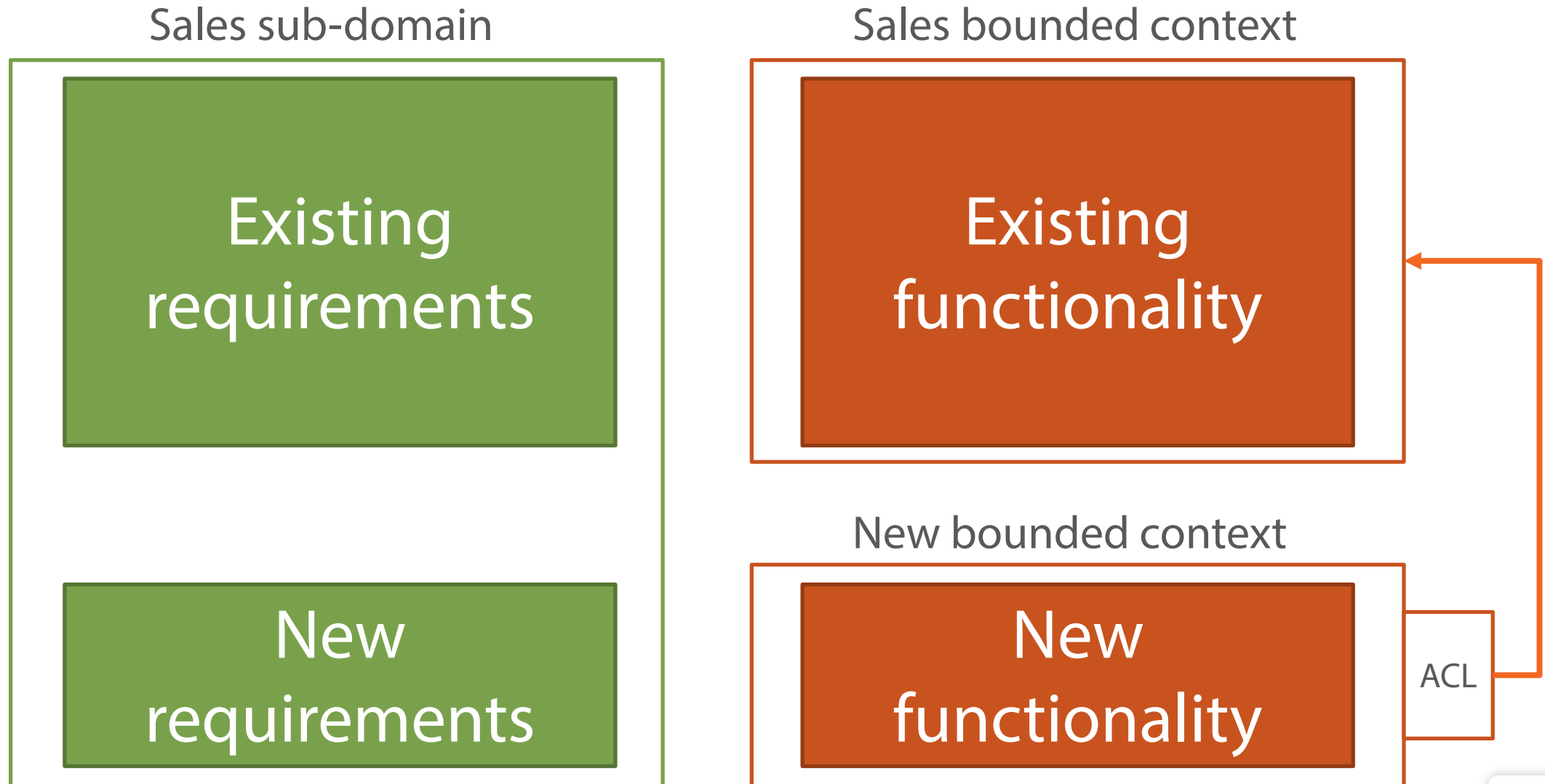


<https://vimeo.com/125769142>

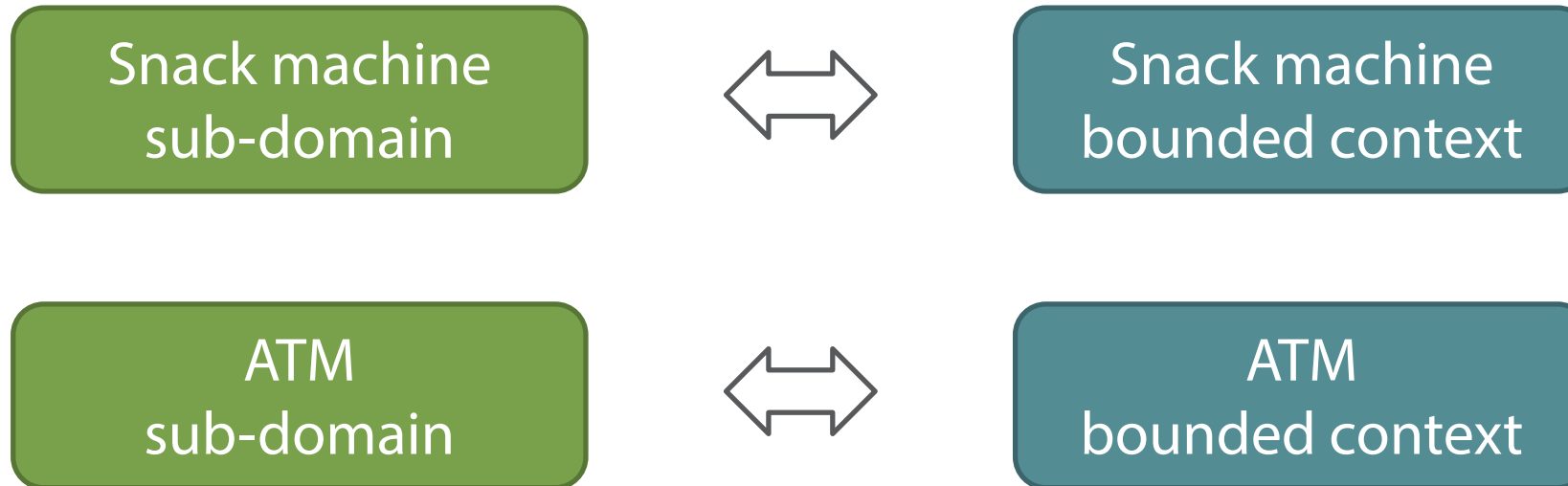
# Bounded Contexts and Sub-domains



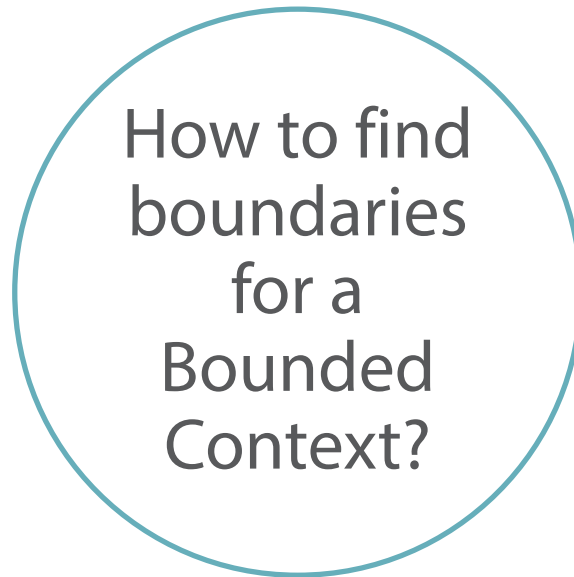
# Bounded Contexts and Sub-domains



# Bounded Contexts and Sub-domains



# Choosing Boundaries for Bounded Contexts



Talk to domain experts

# Choosing Boundaries for Bounded Contexts

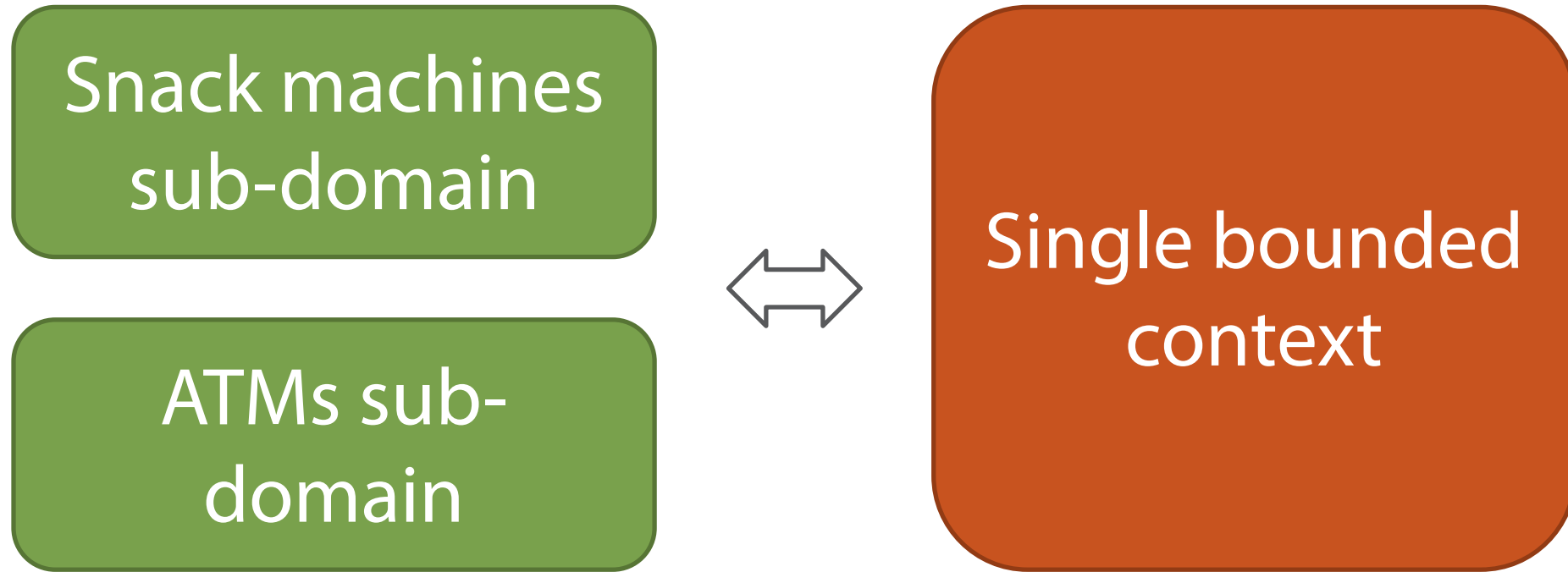
## Team size

- ☐ 6-8 developers is a max

## Code size

- ☐ Code should "fit you head"

# Choosing Boundaries for Bounded Contexts



Don't cover multiple sub-domains with a single bounded context



# Teams and Bounded Contexts



One team working on several bounded contexts



Two team working on one bounded context

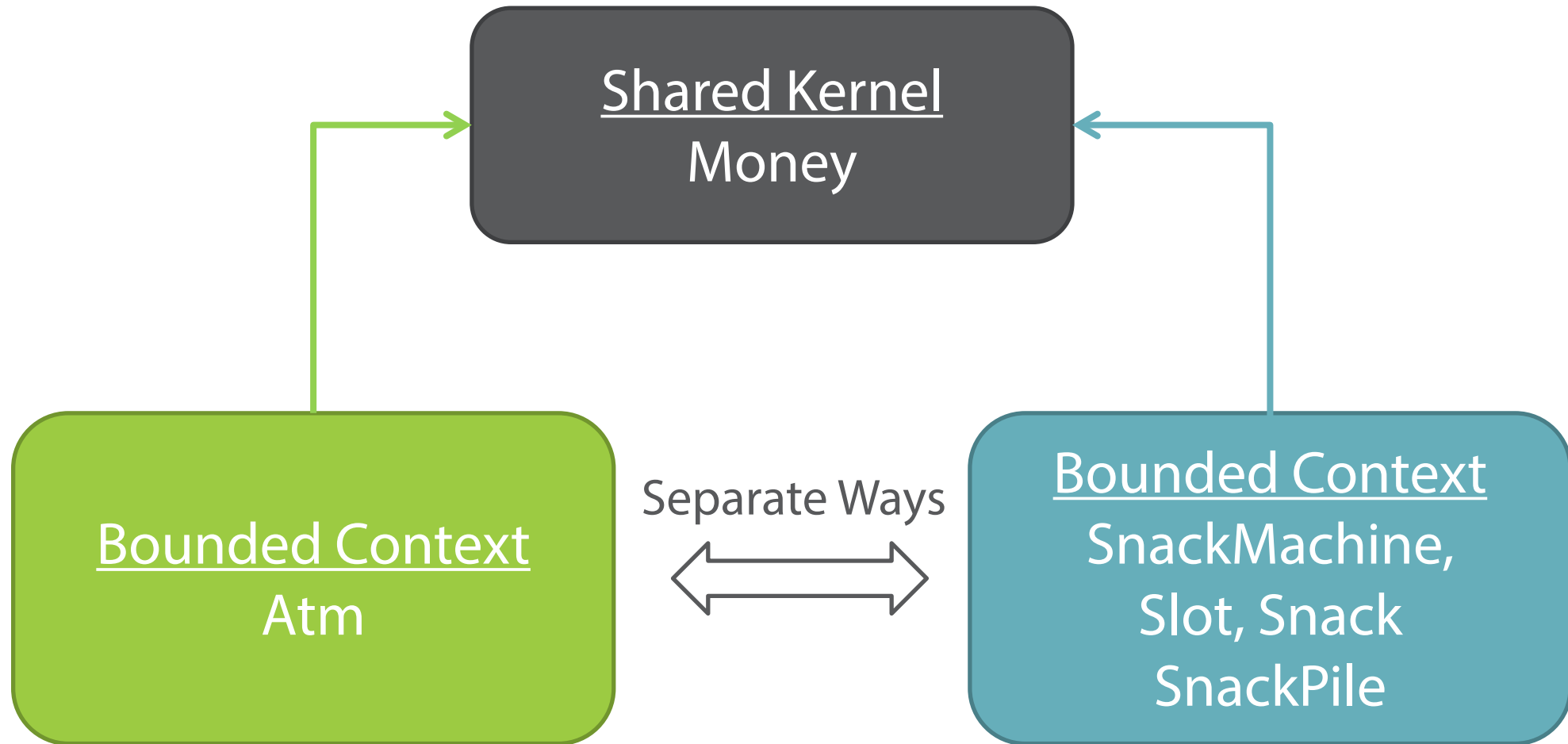
# Drawing a Context Map

Atm

MoneyInside : Money

MoneyChardged : decimal

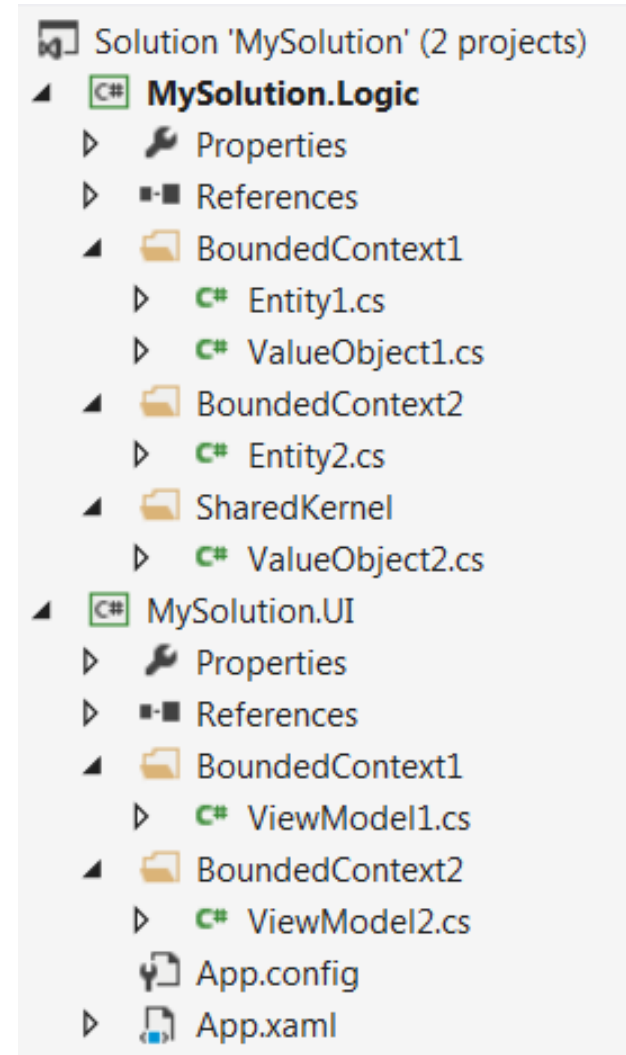
# Drawing a Context Map



# Types of Physical Isolation

## Type #1:

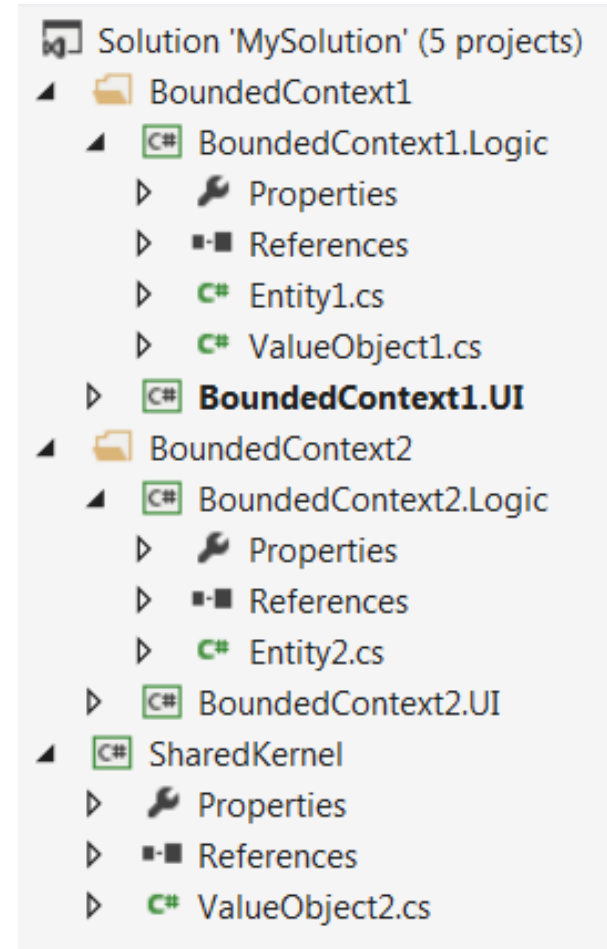
- ☐ Same assemblies
- ☐ Shared database instance



# Types of Physical Isolation

## Type #2:

- Separate assemblies

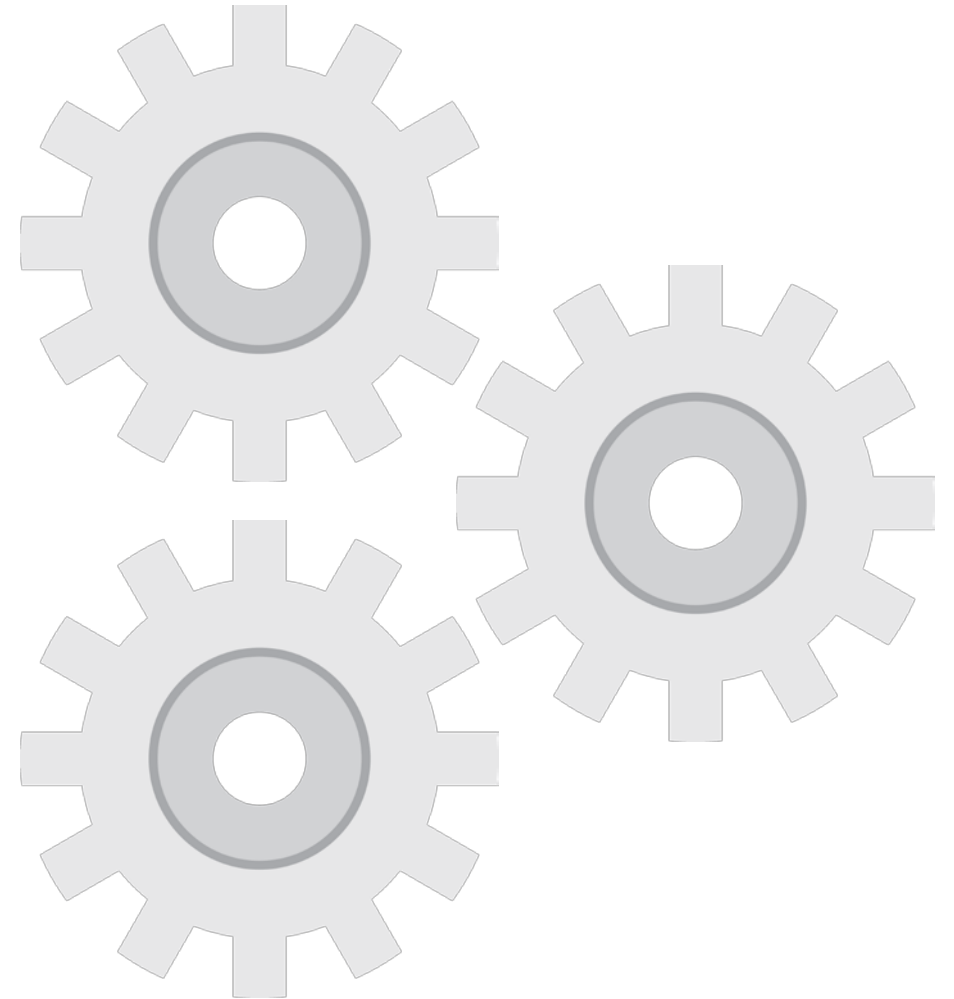


# Types of Physical Isolation

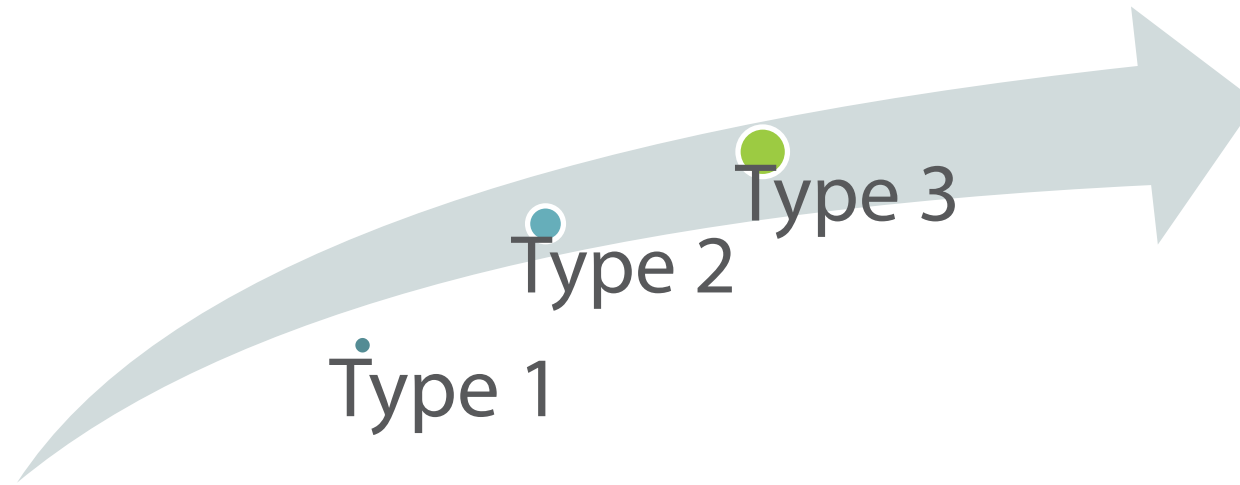
## Type #3:

- ☐ Separate deployments
- ☐ Run in separate processes
- ☐ Microservices

<http://bit.ly/1dI7ZJQ>



# Types of Physical Isolation



Easier to maintain proper isolation



Bigger maintenance overhead

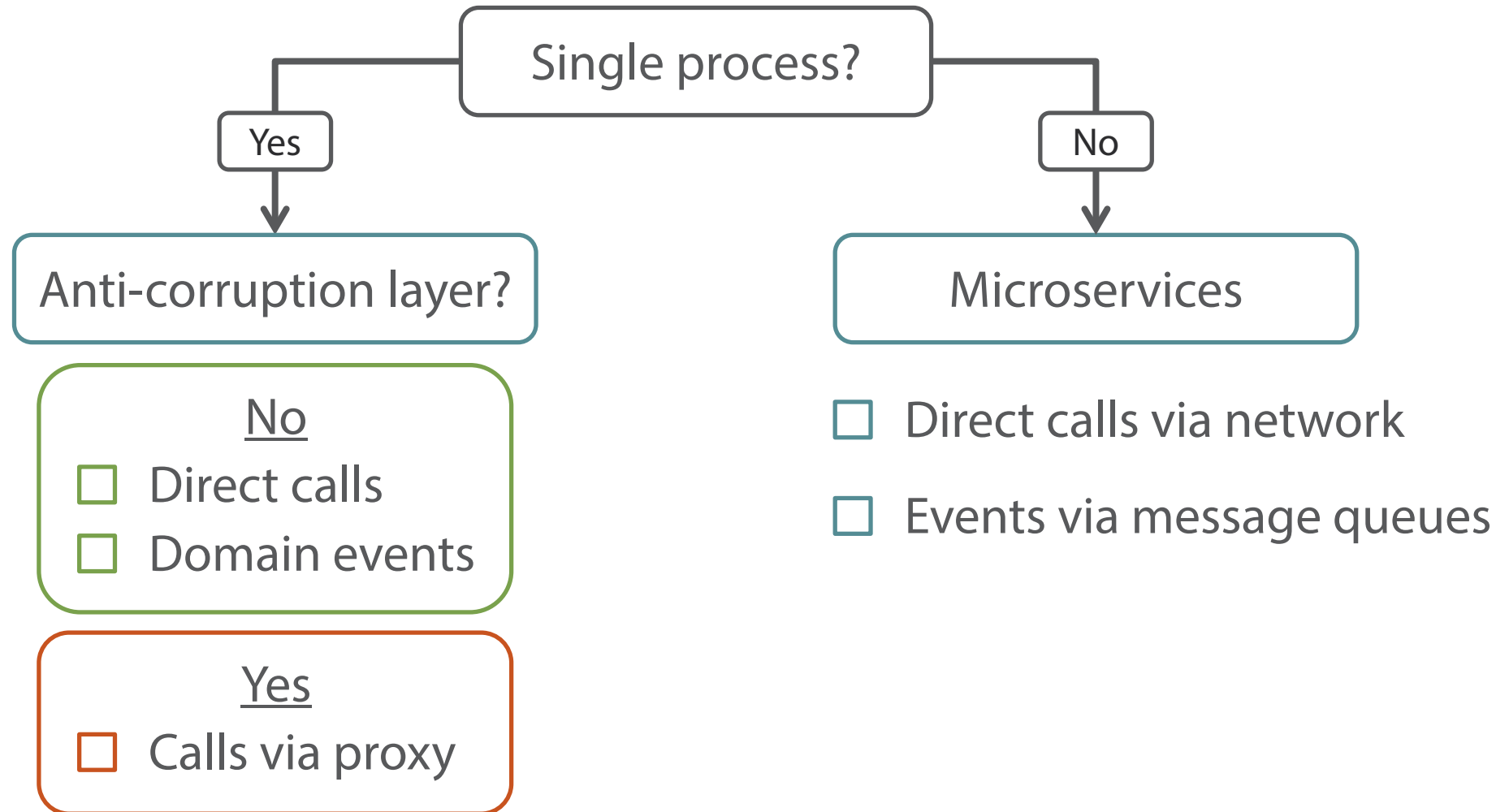
# Types of Physical Isolation

Be pragmatic

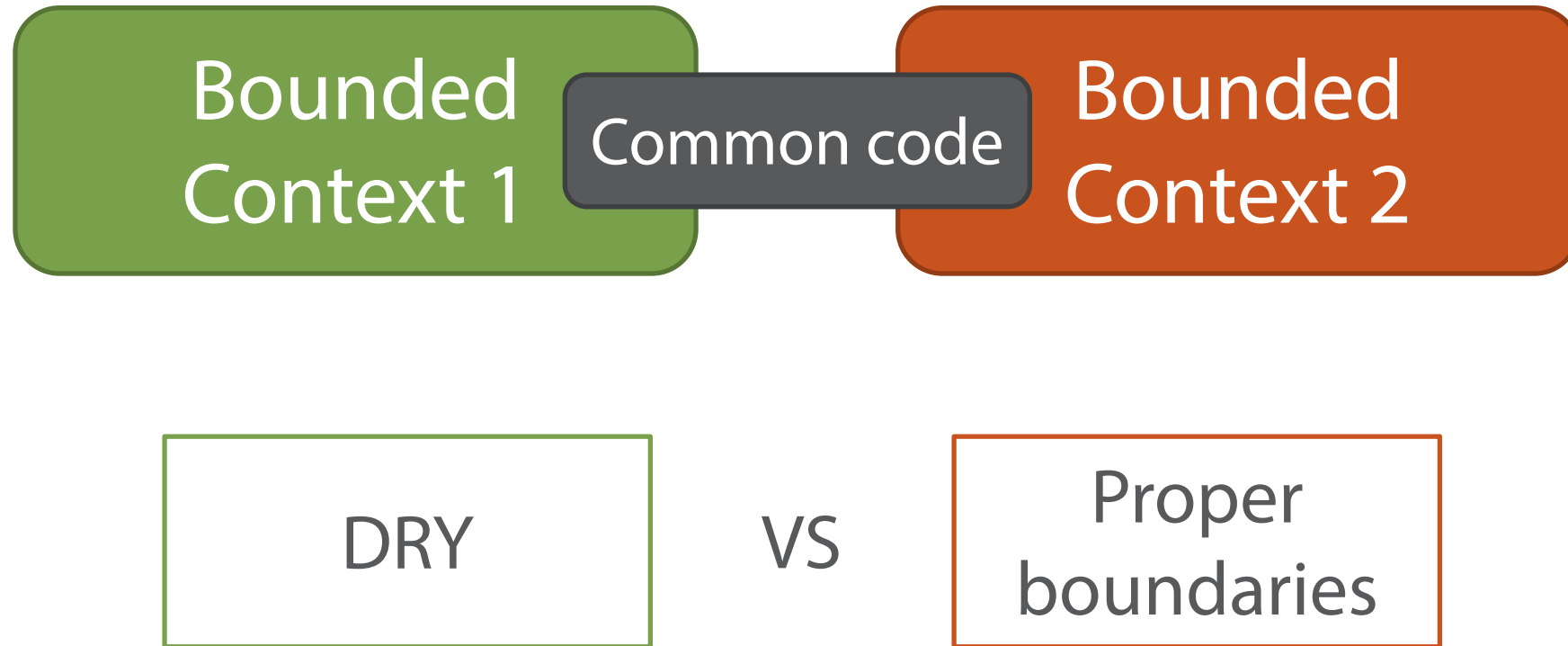
- ☐ Start with Type 1 isolation
- ☐ Move further only if necessary



# Communication Between Bounded Contexts



# Code Reuse Between Bounded Contexts



# Code Reuse Between Bounded Contexts

## Business logic

- Shouldn't be reused in most cases
- Extract to a shared kernel

## Domain base classes

- Reuse within a single team only

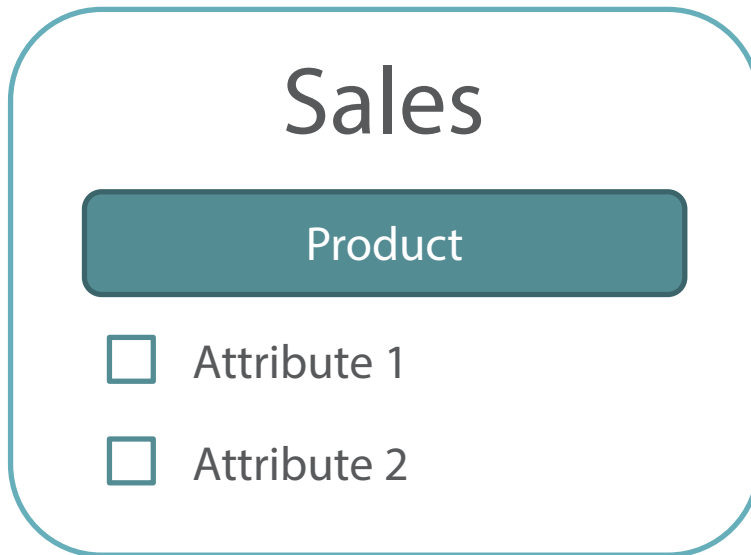
## Utility code

- Reuse within a single team
- Reuse across teams only if provides a lot of value

# Code Reuse Between Bounded Contexts

## Business logic

- Shouldn't be reused in most cases

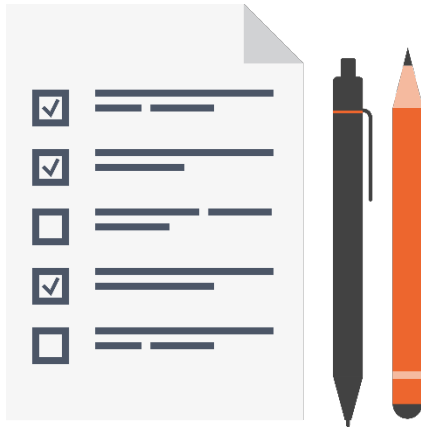


# Summary



- Bounded contexts
  - Provide boundaries for different models
  - Provide boundaries for ubiquitous language
- Bounded contexts vs sub-domains
  - 1-to-1 relation ideally (not always possible though)
- Context mapping
  - Reflects the current state of affairs, not a desired one

# Summary



- Types of physical isolation
  - Start with the weakest isolation
  - Move forward only if necessary
- Communication between bounded contexts
- Code reuse
  - Avoid reusing domain classes

# In the Next Module

Domain events

