

# DDD não é Arquitetura em Camadas

Eduardo Pires  
Microsoft MVP



# Eduardo Pires

Sou desenvolvedor ;)



[www.eduardopires.net.br](http://www.eduardopires.net.br)

# Já parou para pensar ou ouviu alguma vez?

- Como faço para persistir uma entidade com Entity Framework no DDD?
- Como faço para popular um DropDownList seguindo o padrão DDD?
- Estou criando um back-end em WebAPI + DDD
- Onde coloco uma camada de cache num projeto DDD?



# DDD não é Arquitetura em Camadas!

O DDD é uma abordagem de modelagem de software que segue um conjunto de práticas com objetivo de facilitar a implementação de complexas regras / processos de negócios que tratamos como **domínio**.



# DDD não é Arquitetura em Camadas!

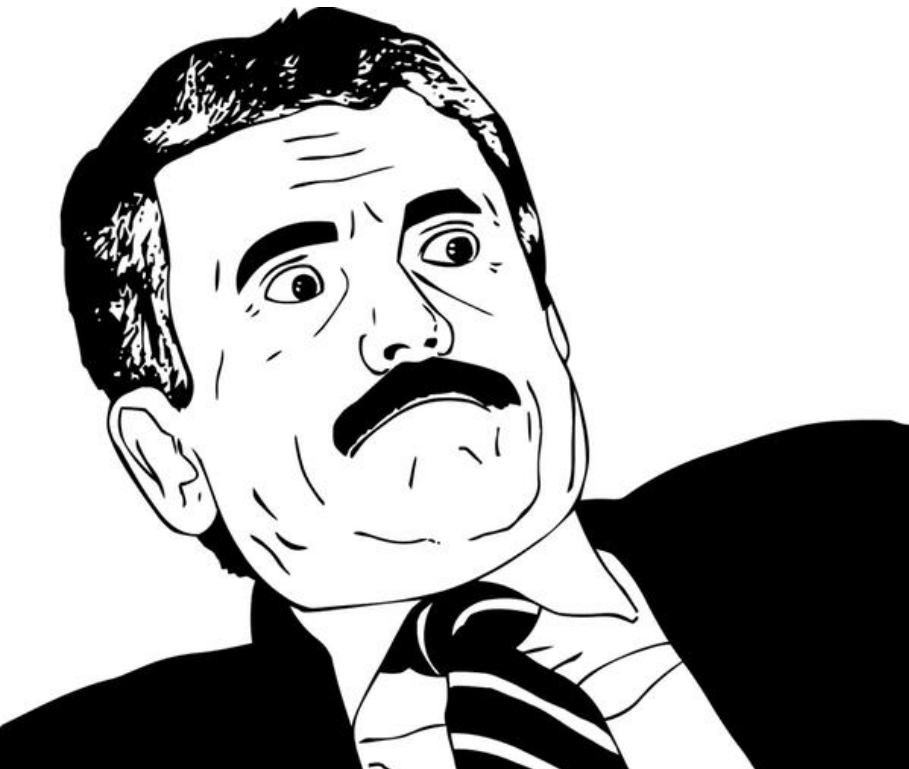
Domain Driven Design como o nome já diz é sobre design. Design guiado pelo domínio, ou seja, uma modelagem de software focada em resolver problemas na complexidade do negócio.

“Toda arquitetura é design, mas nem todo design é arquitetura”  
Grady Booch



# DDD não é Arquitetura em Camadas!

O DDD não é uma receita pronta sobre como desenvolver uma arquitetura baseada em Presentation, Services, Application, Domain e Infra.



# DDD não é Tecnologia!

O DDD não depende da tecnologia que você irá utilizar.  
Seja Web, Mobile, SPA, Desktop!

O DDD não irá influenciar em diversas decisões:

- Como preencher um controle na camada de apresentação
- Como expor uma API REST
- Qual tecnologia usar para persistir os dados
- Como realizar o deploy da aplicação
- Como modelar seu banco de dados
- Como trabalhar com camadas de Infra (ex. Cache, Log, IoC)
- Qualquer outra decisão de tecnologia.

# Evite as Gafes!



Evite utilizar o termo DDD acompanhado de questões que não envolvem diretamente os conceitos do DDD, pois pode dar a entender que você não compreendeu sobre do que se trata o assunto.



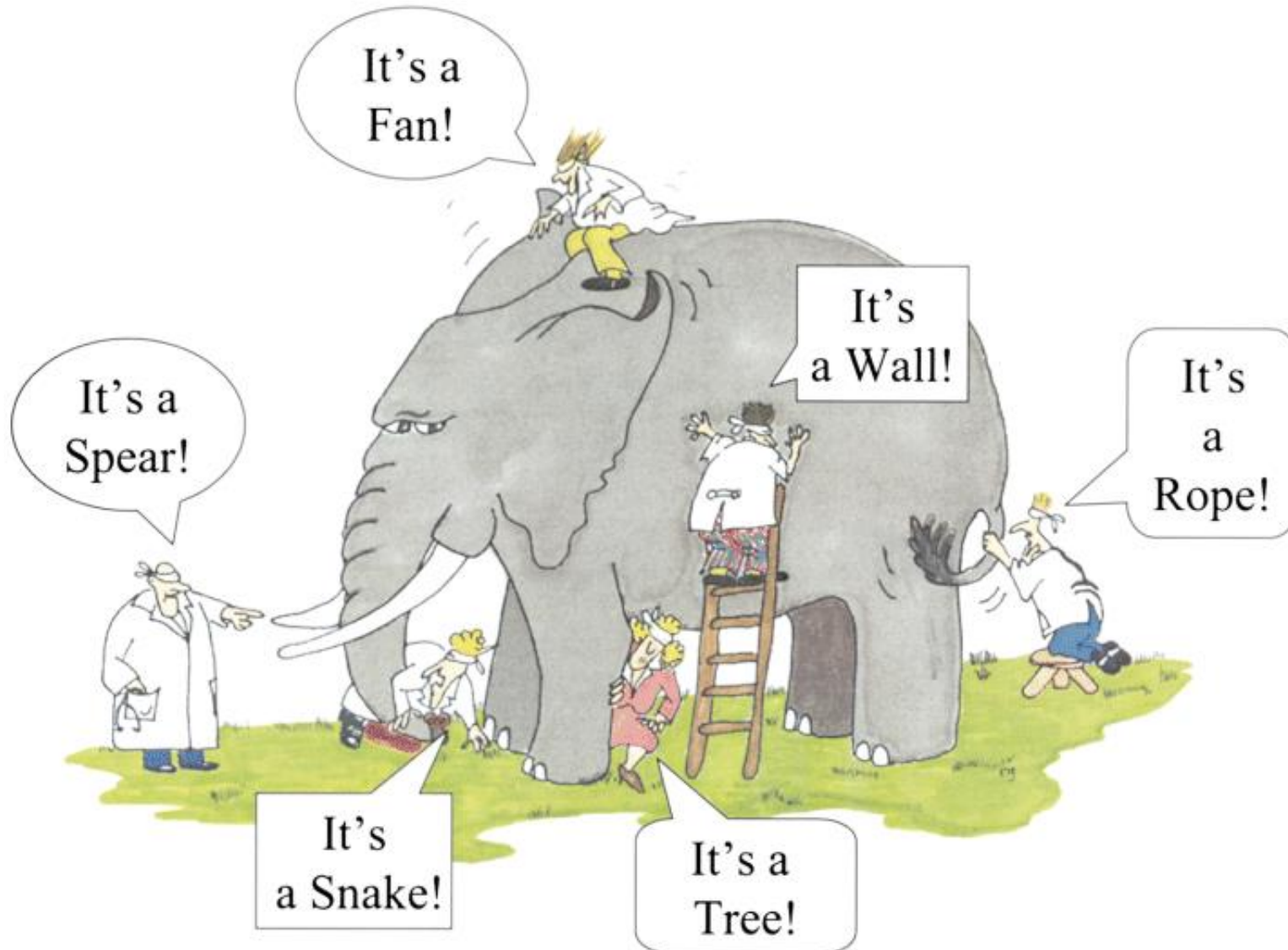
# Então sobre o que é o DDD?

DDD é apenas uma abordagem de modelagem de software, guiada pelas intenções de negócio e implementada seguindo uma série de padrões conhecidos.

# DDD passo a passo

## #1 – Entender o Negócio

# Entender o Negócio



# Entender o Negócio



## Domain Expert

- Conhece do negócio, os processos e seus termos
- Define novos termos, processos e regras
- Tem conhecimento da operação



## Time de Desenvolvimento

- Precisa do apoio do Domain Expert para entender o processo, e as terminologias do negócio

# DDD passo a passo

## #2 – Extrair a Linguagem Ubíqua

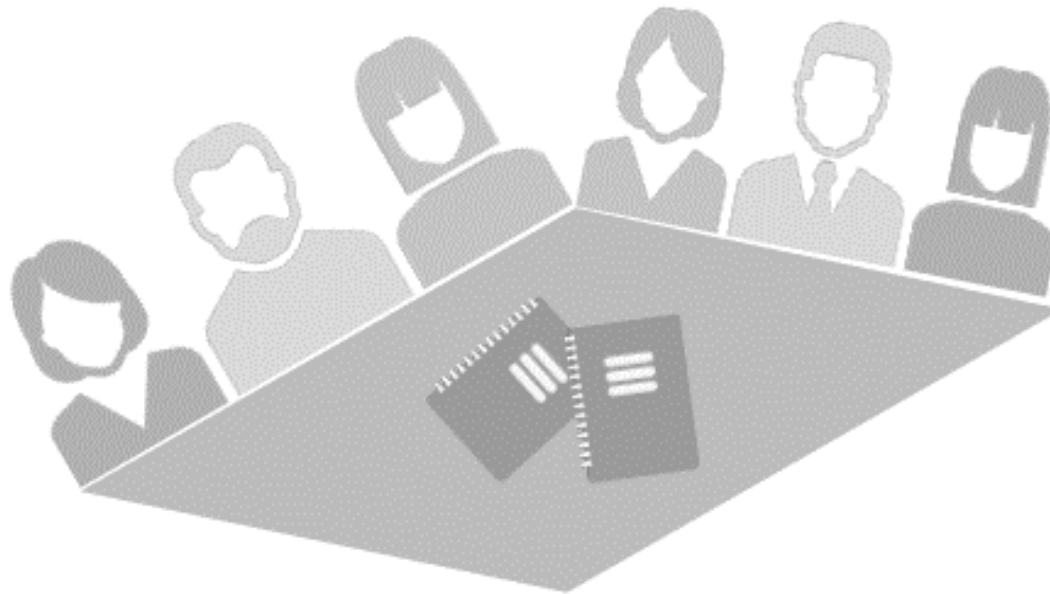
# Extrair a Linguagem Ubíqua

Linguagem  
natural, não  
artificial

Obtida em  
reuniões e  
brainstormings

Composta e  
refinada com o  
tempo

Fluente e única

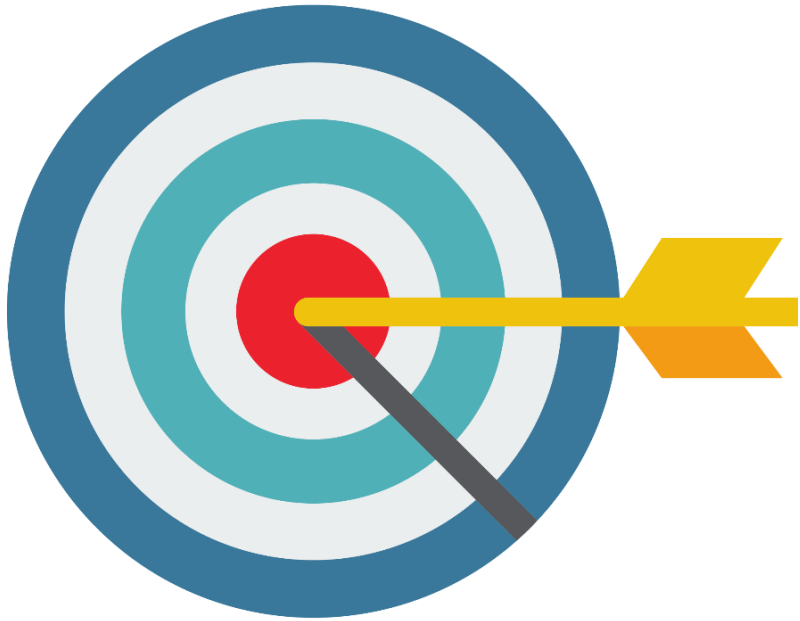


Atende ao  
entendimento  
do **Domain  
Expert**

Atende ao  
entendimento  
da **Equipe  
Técnica**



# Ubíqua = Usada em todo lugar

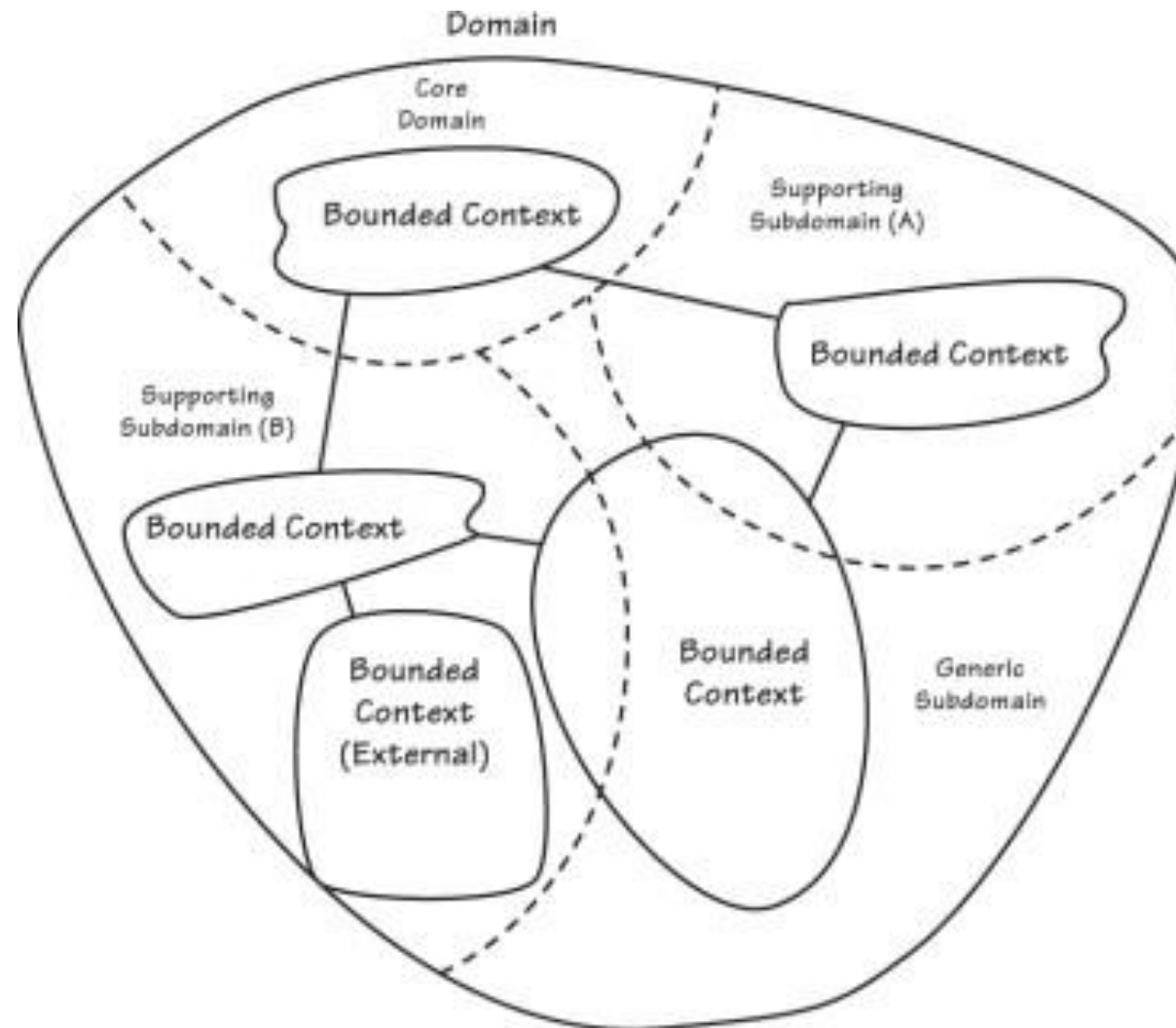


- User Stories e Requisitos Funcionais
- Reuniões
- E-mails
- Documentação técnica
- Documentação do cliente
- O cliente conhece e usa
- Os departamentos relacionados usam
- **O CÓDIGO FONTE usa** ←

# DDD passo a passo

## #3 – Modelagem Estratégica

# Modelagem Estratégica - Context Map

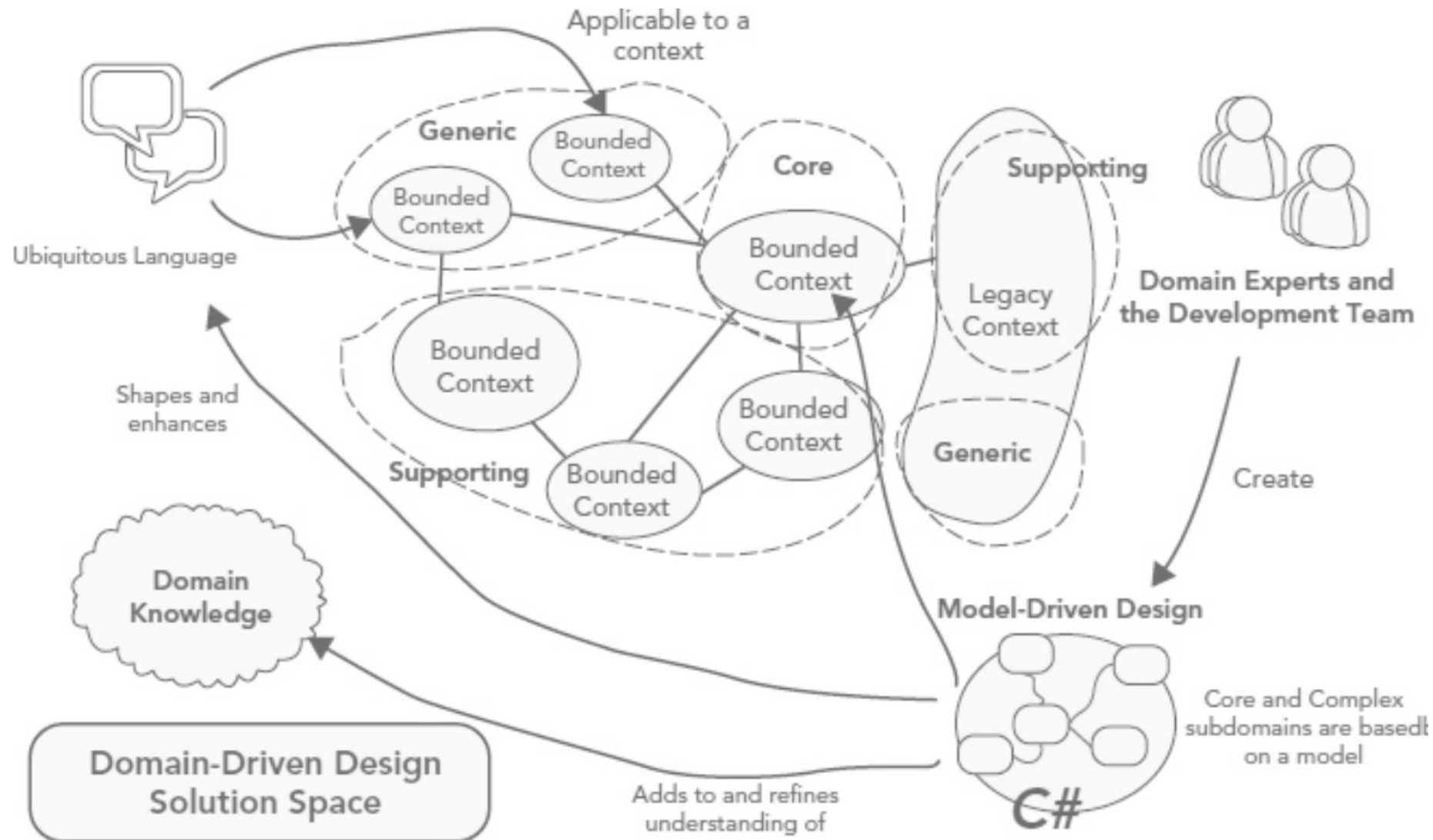


# Modelagem Estratégica



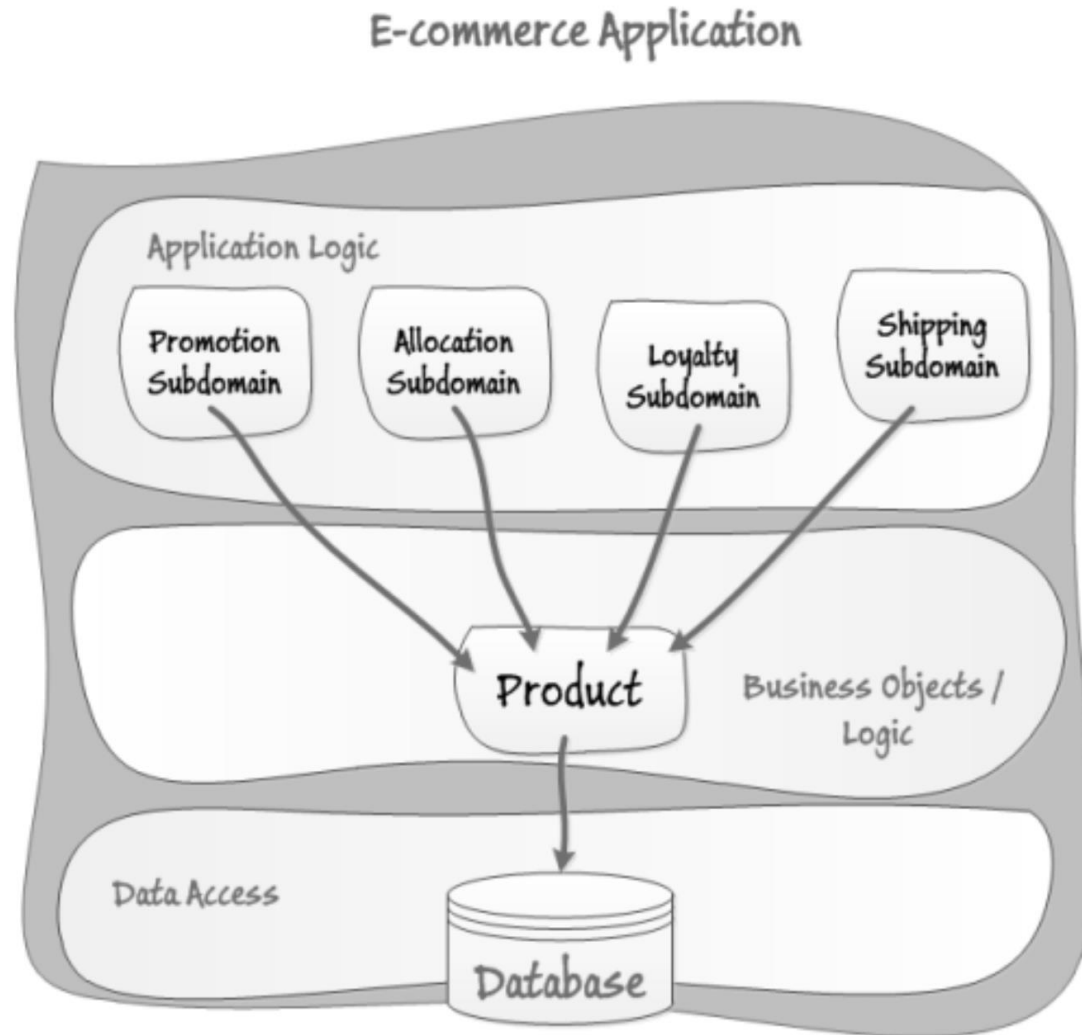
## Dividir para conquistar

# Modelagem Estratégica - Context Map



# Modelagem Estratégica

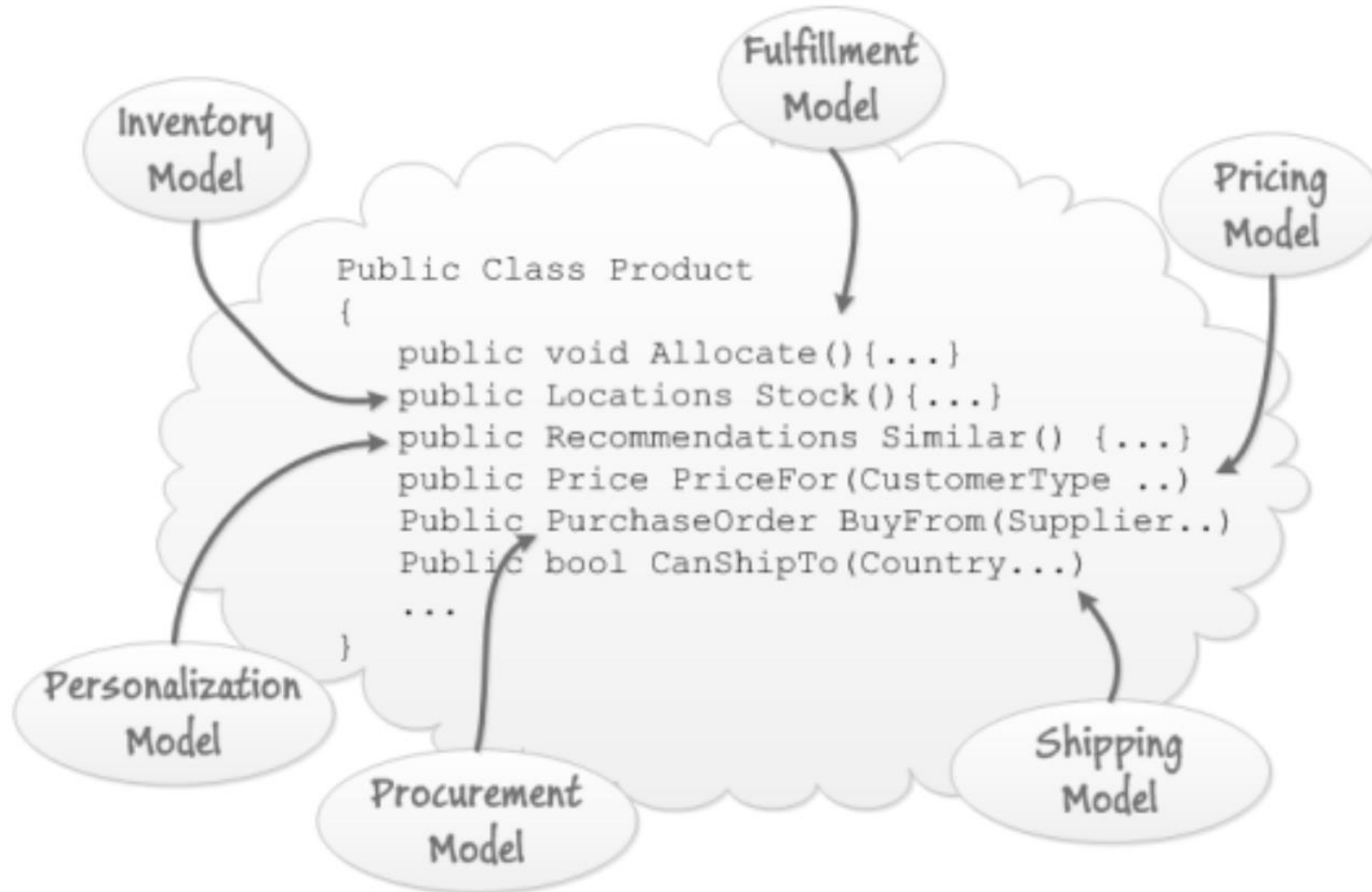
## Modelo de Negócio VS Modelo de Domínio





# Modelagem Estratégica

## Big Ball of Mud



# Modelagem Estratégica

## Bounded Contexts



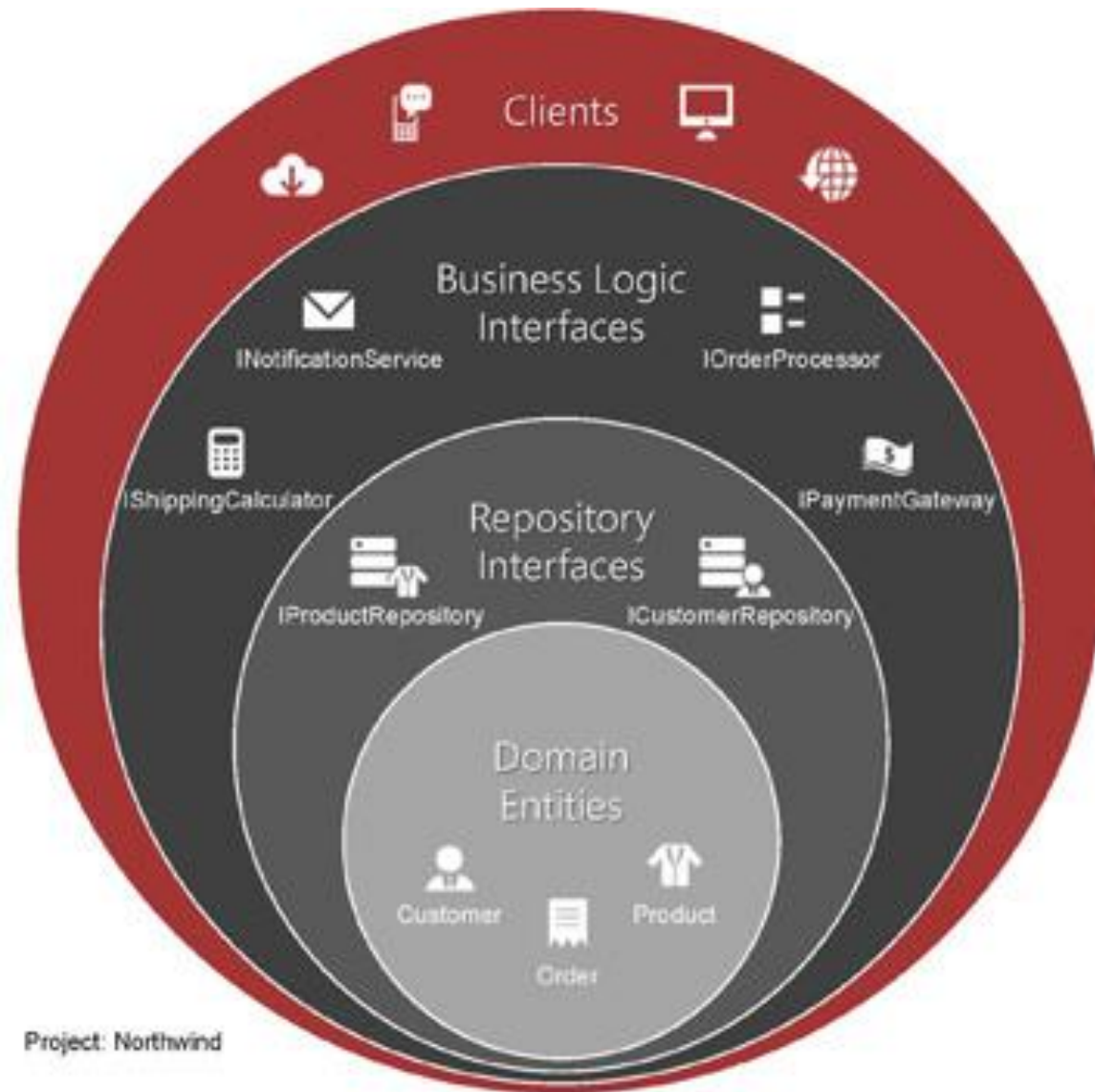
# DDD passo a passo

## #4 – Definir a Arquitetura

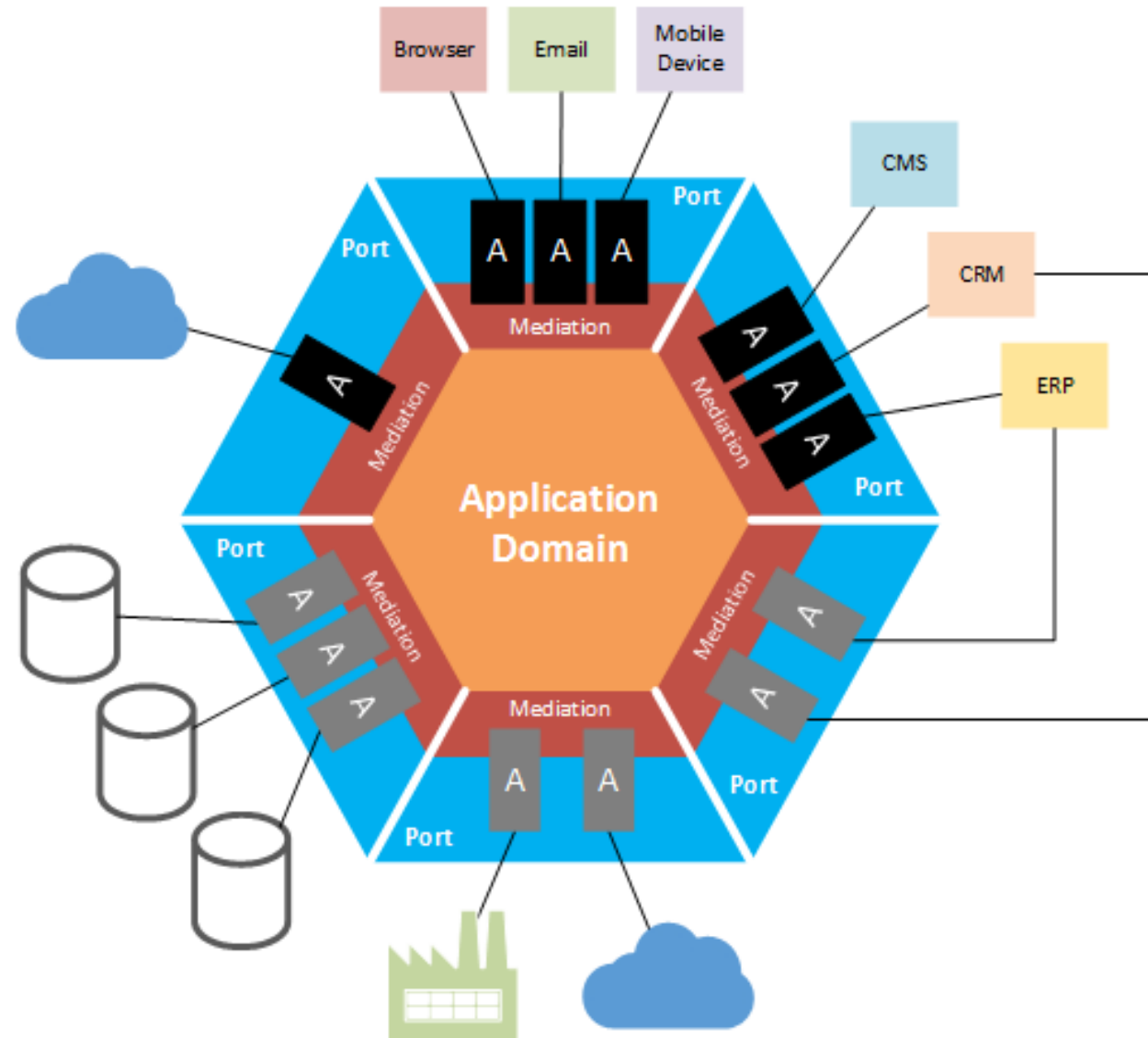
# Definir a Arquitetura

Estilo Arquitetural  $\neq$  Padrão Arquitetural

# Arquitetura Cebola

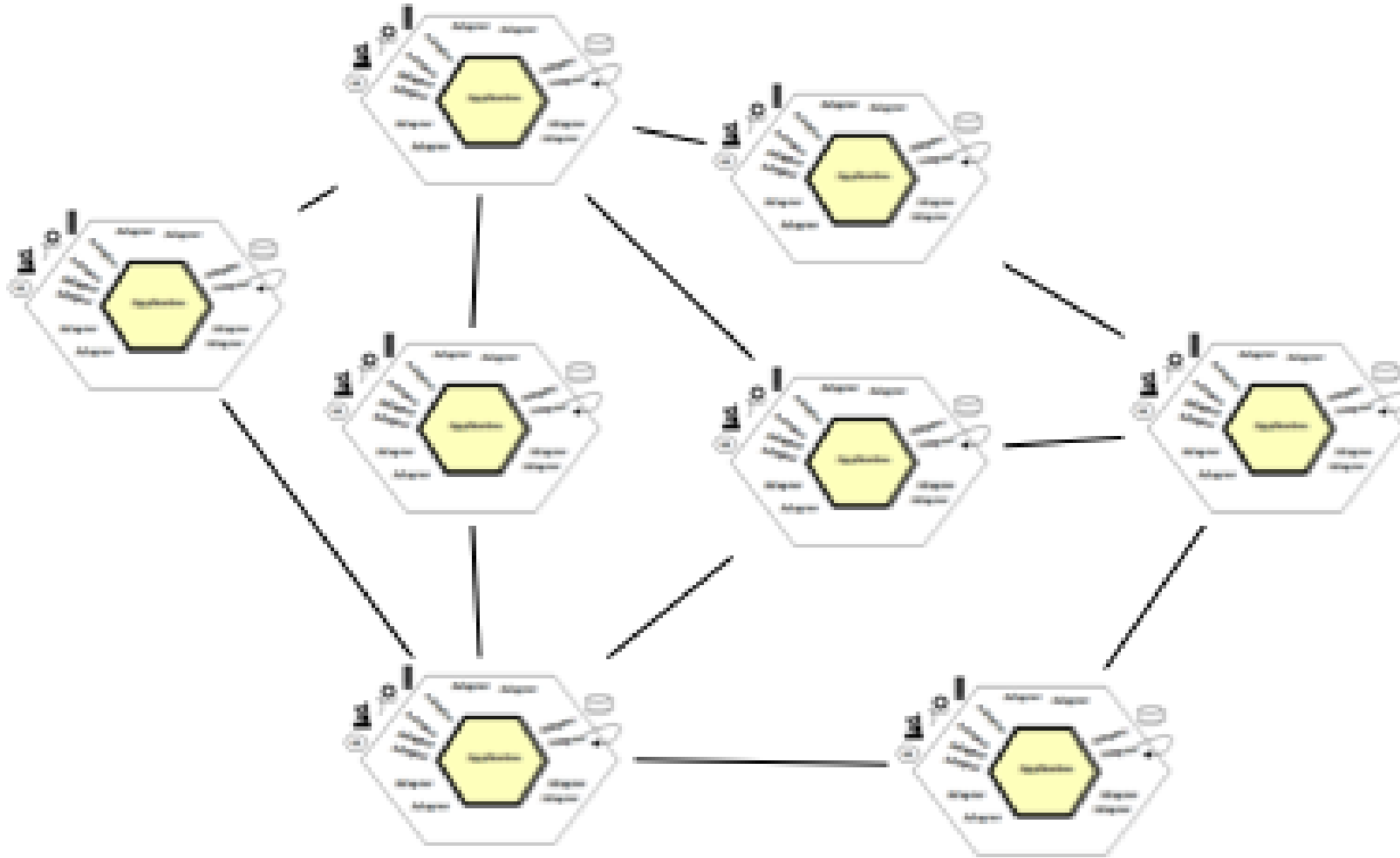


# Arquitetura Hexagonal

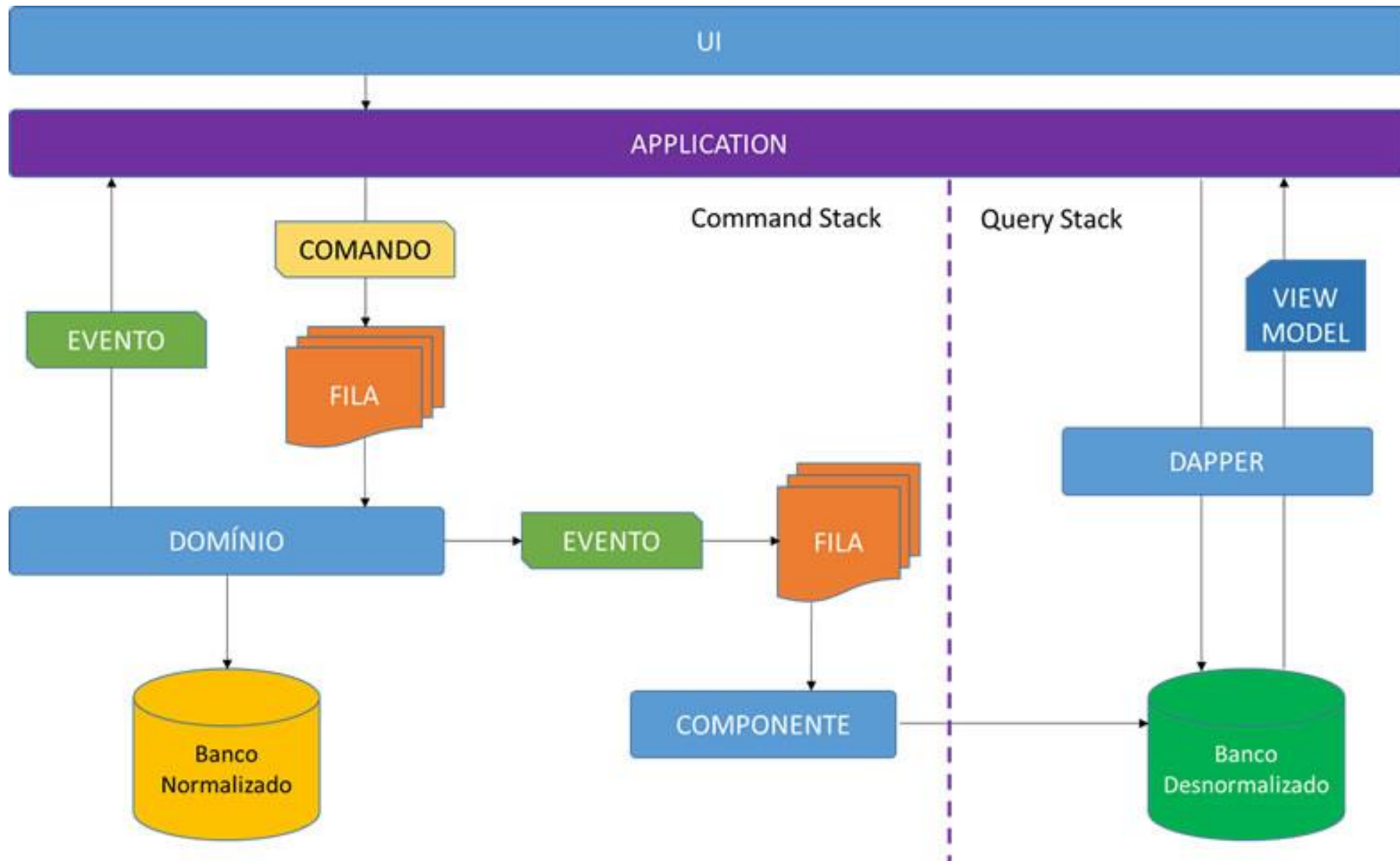




# Arquitetura de Micro-Serviços



# Padrões de Arquitetura - CQRS




# DDD passo a passo

## #5 – Modelagem Tática

# Modelagem Tática

- Aggregate Object
- Domain Model
- Value Object
- Domain Service
- Command Handlers
- Application Service
- Repository
- External Service

A man with a beard and a dark shirt is shown from the chest up, looking upwards and to the right with a thoughtful expression, his hand resting on his chin. A large thought bubble is positioned above his head, containing the text 'DDD é necessário para essa Aplicação?'. Three smaller circles lead from the man's head to the main thought bubble.

DDD é necessário  
para essa  
Aplicação?

# Erros Comuns:

#1 - Tornar-se escravo do ferramental



# Erros Comuns:

#2 - Não se envolver com os Domain Experts

# Erros Comuns:

## #3 - Ignorar a Linguagem Ubíqua

# Erros Comuns:

#4 - Não identificar os limites dos contextos

# Erros Comuns:

#5 - Escrever entidades anêmicas

# Erros Comuns:

#6 - Assumir que toda lógica é lógica de domínio

# Erros Comuns:

#7 - Focar demais na infra-estrutura

# Muito Obrigado



[www.eduardopires.net.br](http://www.eduardopires.net.br)

<http://epires.me/DevXDDD>

