

# DDD in large product portfolios

Andreas Pinhammer



## Key takeaway n°1

Partir from scratch dans un environnement complet peut être un moyen simple et efficace d'expérimenter le DDD

## Key takeaway n°2

Des plateformes agnostiques d'un produit particulier permettent de limiter la charge mentale des équipes et la duplication

### Situation initiale



Une grande compagnie d'assurance allemande  
2.000+ assets techniques

### Élément perturbateur



Mauvais  
Time To  
Market

N'attire pas  
les jeunes



### Péripéties

Huit équipes

Expérimenter le DevSecOps

Cross-fonctionnelles

Construire un MVP

Expérimenter l'Agile

Produit responsabilité civile

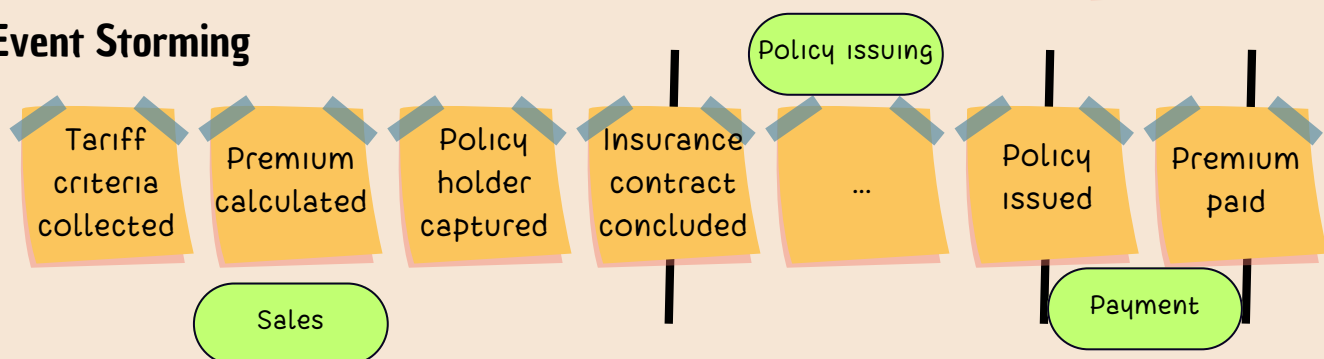
Expérimenter le DDD

En partant de zéro

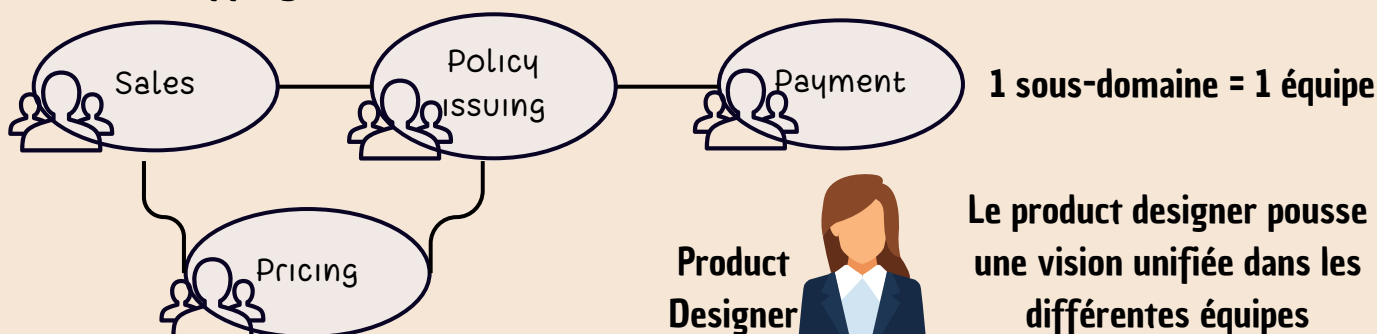
Application de l'Inverse Conway Maneuver  
aligner la structure sur le domaine souhaité

event | subdomain | pivotal event

#### 1. Event Storming



#### 2. Context Mapping

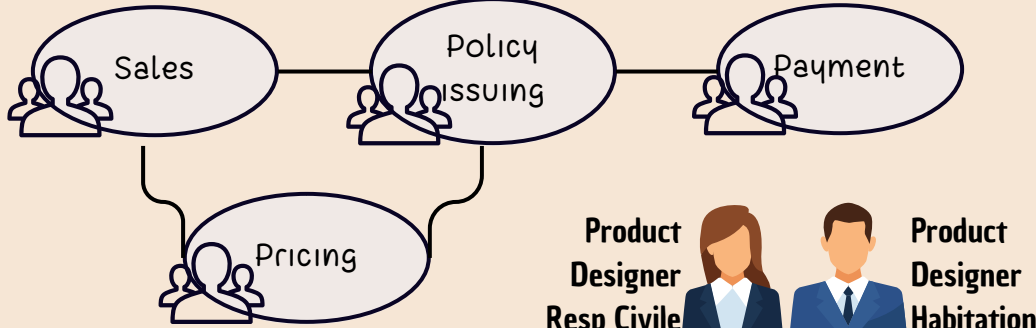


1 an plus tard, un MVP opérationnel

Choix de partir sur un second produit (assurance habitation),  
pour tester la scalabilité du modèle

Répétition des  
étapes 1 et 2

Processus similaire  
--> Ajout d'un product  
designer dédié



### 3 problèmes

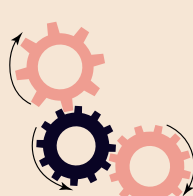
Backlog alignment  
backlog multi-produits  
compliqué à arbitrer

Feedback produit  
Designers loin des retours  
Impact faible des équipes

Charge mentale  
Chaque nouveau produit  
s'ajoute aux équipes

### 3. Retrospective

Pourquoi de nouveaux produits ajoutent de la charge mentale aux développeurs ?



Process similaires, mais...



Vocabulaires différents



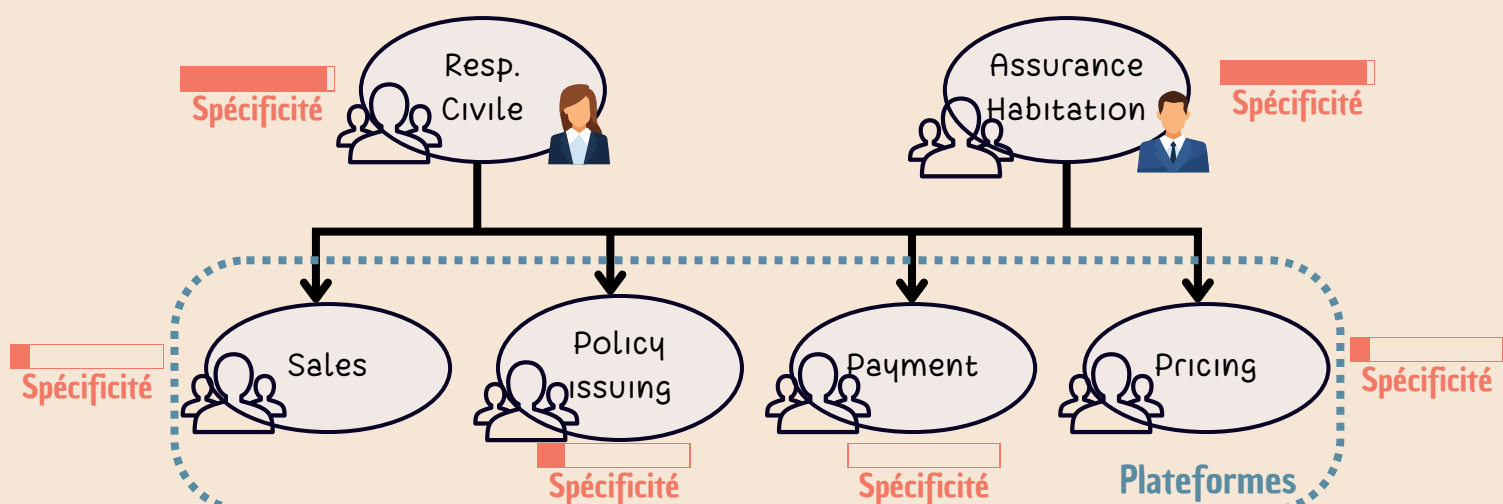
Besoins clients différents

Utiliser des plateformes génériques



Un produit est un sous-domaine

### Résolution



### Histoire bonus

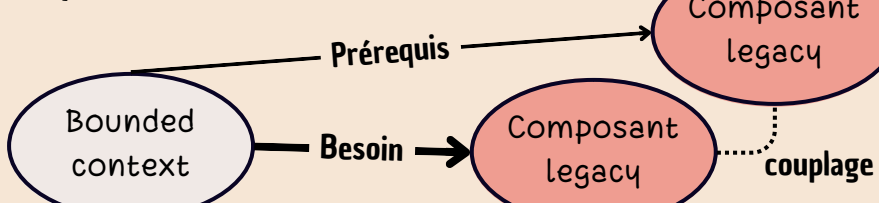


Comment ces  
nouveaux produits ...

... peuvent être intégrés  
à l'ancien legacy ?



Legacy contraignant • Fortement couplé  
• Dépendances externalisées (transitives)

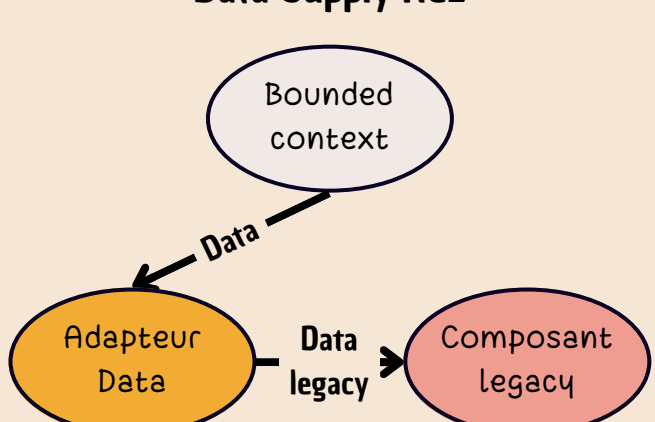


Mise en place d'un  
"Ice Wall"



Ensemble d'Anti Corruption Layers (ACL) entre le legacy et les  
composants DDD. 2 typologies d'adapteurs :

#### Data Supply ACL



#### Pseudo Bounded Context ACL

