

Practical DDD - Transforming theories into guidelines

Hila Fox



Key takeaway n°1

Lorsque les équipes prennent la responsabilité de leurs composants, les architectes doivent devenir des supports plutôt que des décideurs

Key takeaway n°2

Pour être efficace, le savoir doit faciliter la vie des équipes (via des partis pris), être partagé activement et évoluer dans le temps

AUGURY

Sondes physiques à fixer sur des machines industrielles lourdes, pour la surveillance et la maintenance préventive

IoT
API

Big Data
Mobile app

De 100 à 400 employés en un an et demi



Comment scaler efficacement tout en gardant...

Des équipes indépendantes
Responsables de leurs assets

Des périmètres claires
Cohérence archi/tech

1 Faire évoluer le rôle des architectes

Un architecte pour aider chaque Dev Group



Leader sans autorité

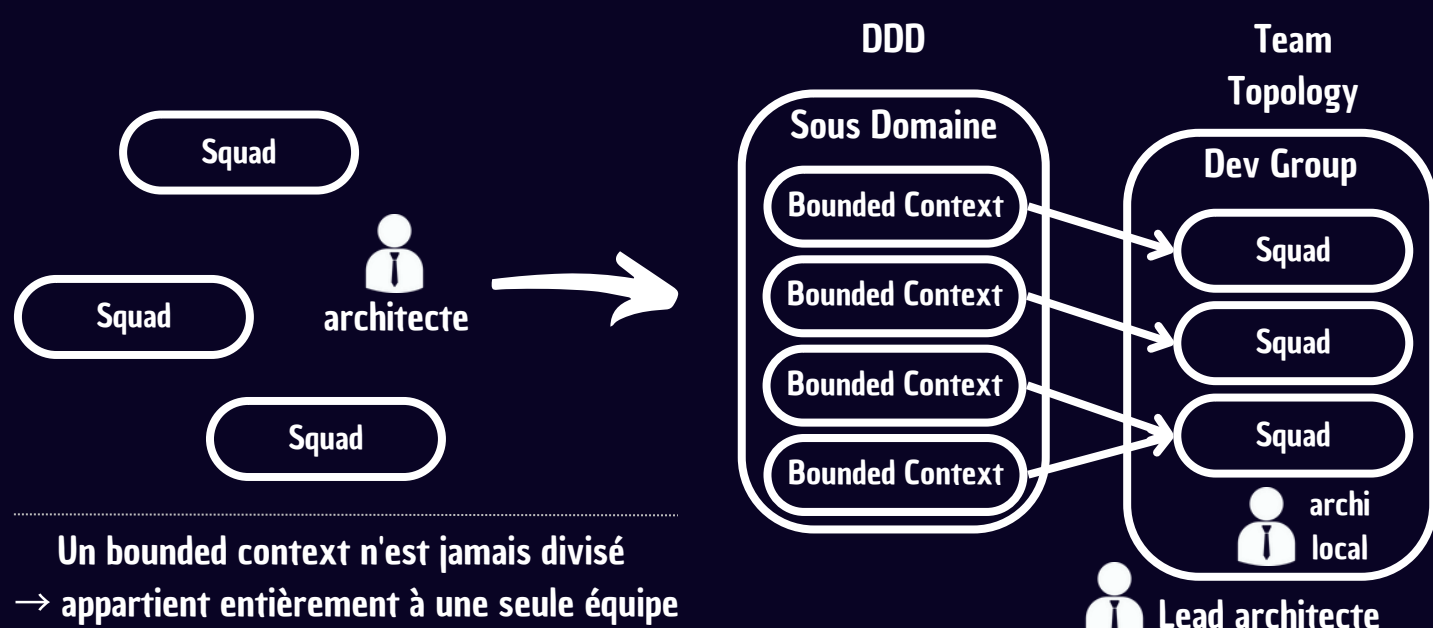
Aider à faire des choix



Propager le savoir

À disposition des équipes

2 Donner un cadre fonctionnel aux équipes



3 Donner un cadre architectural aux équipes

Paradigmes basés sur le DDD
Choix conceptuels déjà faits, adaptés au contexte de l'entreprise



6 mois de recherches des architectes concentrés dans un livre, distribué aux dévs
→ Les développeurs n'ont pas le temps de lire le livre
→ Ceux qui prennent le temps n'internalisent pas les infos trop denses

Livre découpé en

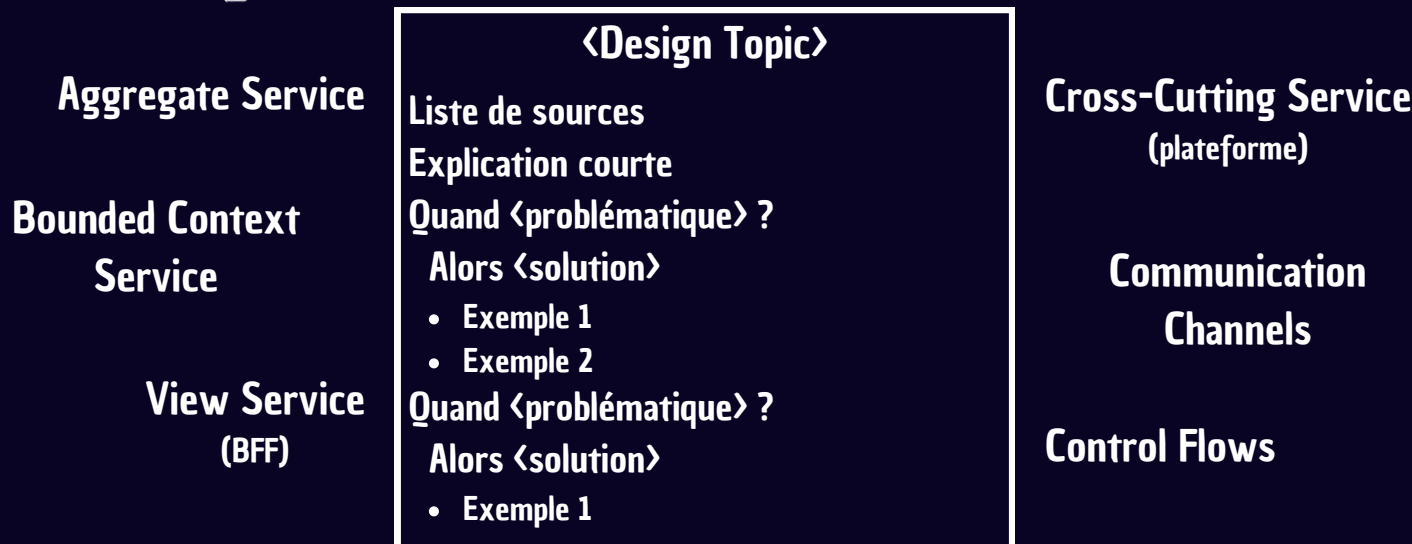


Un dictionnaire simplifié
Adapté au contexte

Des fiches actionnables
Les Designs topics

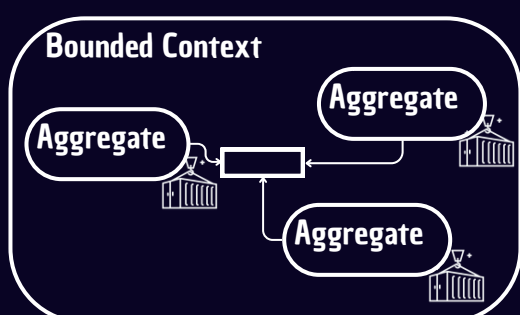


Savoir plus simple à appréhender et propagé par les architectes



Exemple: Design Topic Aggregate (micro) service

Choix technique: MicroService à la maille d'un aggregat



Plus simple pour

- Faire évoluer le périmètre d'une équipe
- Redécouper un bounded context



Mais Quand le domaine est récent / les boundaries non claires, Alors un microservice à la maille du bounded context est possible, en trajectoire

Exemple: Design Topic Communication Channels

Quand générer un nouveau domain event ?

Quand générer des service requests ?



Quand utiliser des commandes ?

Quand un agrégat doit-il consommer un domain event ?

Quand enrichir un event avec des données ?

Et pour la suite ? Continuer à propager de nouveaux Design Topics



Entités distribuées



Micro frontends



Context Mapping



Generic Domains