

Quick Tutorial

5 patterns que tout CTO doit connaître

Les architectures qui scalent sans exploser
ton équipe



Maxime Lenne

CTO & Leader Produit Tech

→ 3 raisons principales

Pourquoi ton **monolithe** ralentit



Couplage fort entre modules

Un changement = cascade de régressions



Déploiements tout-ou-rien

Impossible de shipper en continu



Scaling horizontal impossible

Tu scales tout même quand 1 service est le goulot



Charge cognitive maximale

L'onboarding dev passe à 3 semaines

→ Pattern #1 : Event-driven

Du couplage direct à l'event bus

●●● order_service.rb

❌ Avant – couplage fort

```
def create_order(params)
```

```
  order = Order.create!(params)
```

```
  EmailService.send_confirmation(order)
```

```
  StockService.decrement(order.items)
```

✅ Après – event bus

```
def create_order(params)
```

```
  order = Order.create!(params)
```

```
  EventBus.publish("order.created", order)
```

```
# → chaque service écoute indépendamment
```

→ 4 étapes pour migrer

Migrer vers les **microservices** sans downtime

1

Identifier les bounded contexts

Cartographie tes domaines métier — Orders, Users...

DDD

2

Extraire d'abord les lectures

Commencer par les API read-only réduit le risque

Low risk

3

Strangler Fig pattern

Proxy qui route le trafic vers le nouveau service

Pattern

4

Feature flags + rollback plan

Basculer 5% → 50% → 100% avec kill switch

Prod safe

”

**Les microservices ne résolvent pas
les problèmes organisationnels.
Commence par aligner ton équipe,
pas ton architecture.**

– Conway's Law, appliqué en 2024

Mes voix préférées (et celles de mes enfants)

Zarvox	Voix alien/robot très distincte	utilisée actuellement
Trinoids	Voix extraterrestre grésillante	
Organ	Voix d'orgue, grave et musicale	
Whisper	Voix chuchotée, discrète	
Boing	Voix rebondissante cartoonesque	

// FIN DU THREAD

Sauve ce post pour plus tard →

Si ça t'a aidé, abonne-toi.

Je publie chaque semaine sur la tech & le
leadership produit.



Prenons rendez-vous

// LinkedIn

[/in/maxime-lenne](#)

// Site

[maxime-lenne.fr](#)

→ Composants de mise en valeur

Annotations & tags

code-annotation : Met en valeur un résultat, une règle ou une explication clé après un bloc de code

Variante verte : Utilise `border-color` + `background` pour différencier le ton



notif système native



message dynamique



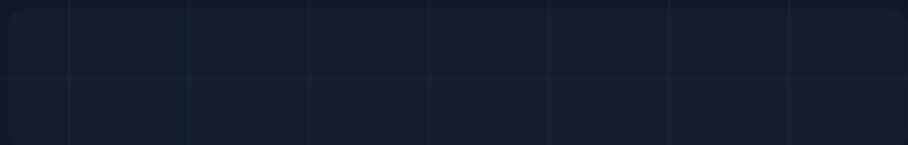
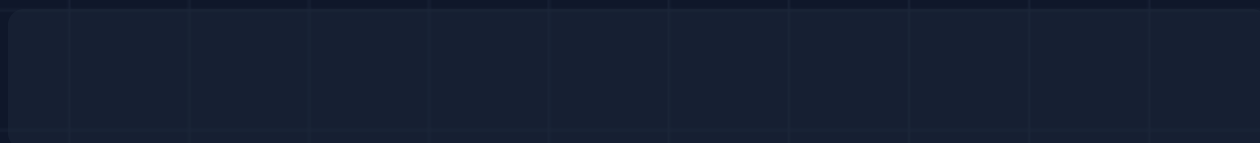
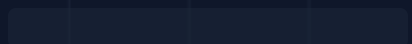
plug & play



moins de charge mentale



impossible à rater



Zone de contenu

