

GROSSE FONCTIONNALITÉ PETITES LIVRAISONS

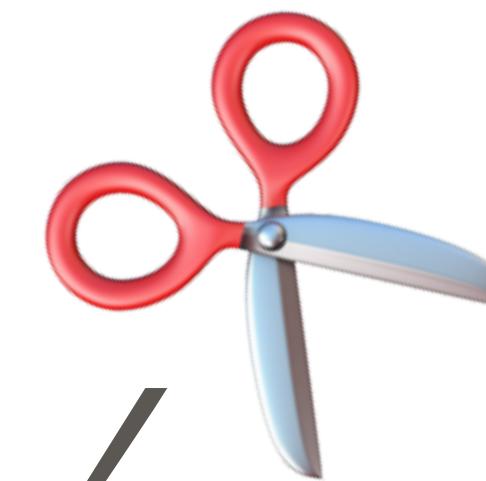
L'art du Découpage

GROSSE FONCTIONNALITÉ PETITES LIVRAISONS



L'art du Découpage

GROSSE FONCTIONNALITÉ PETITES LIVRAISONS

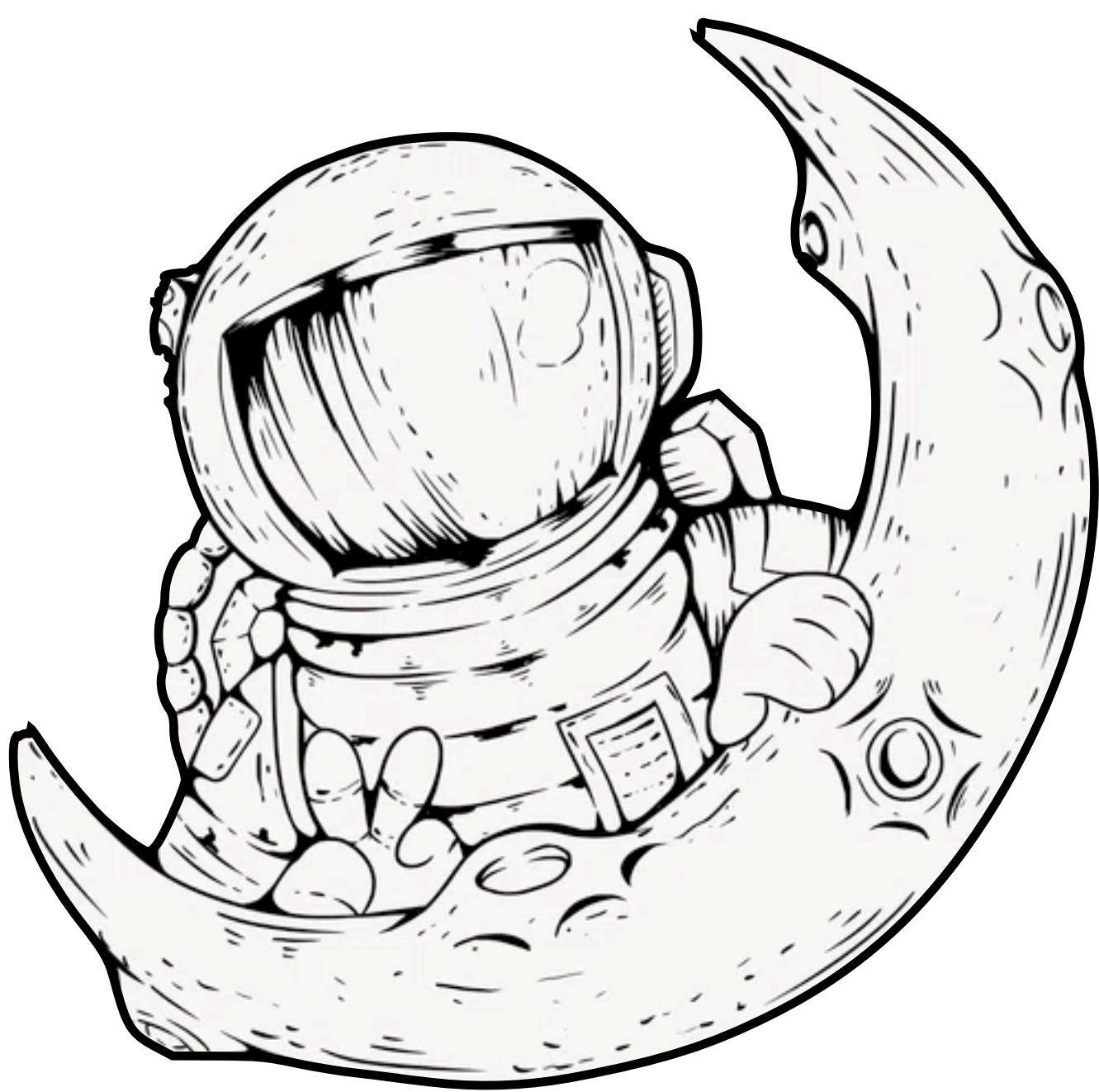


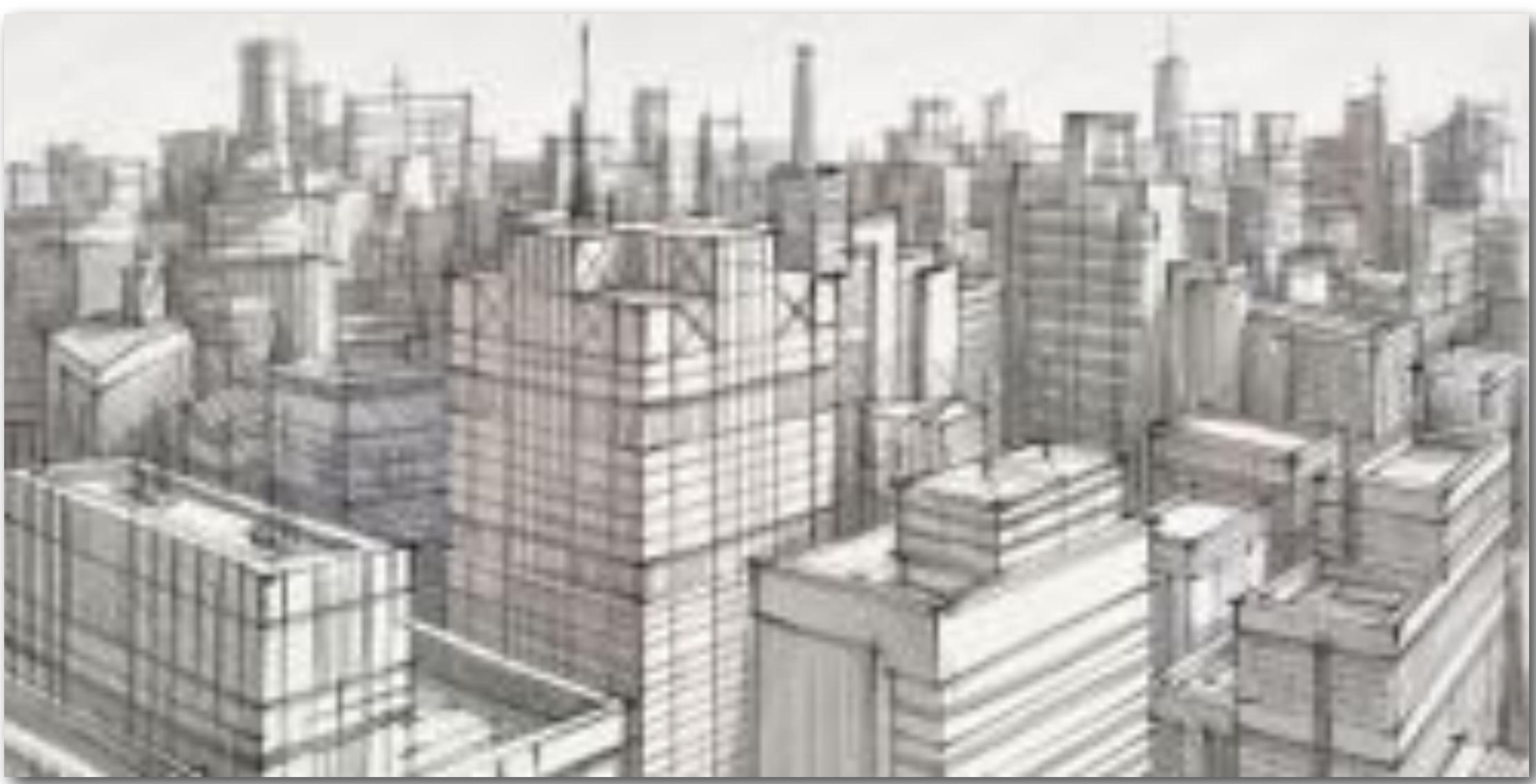
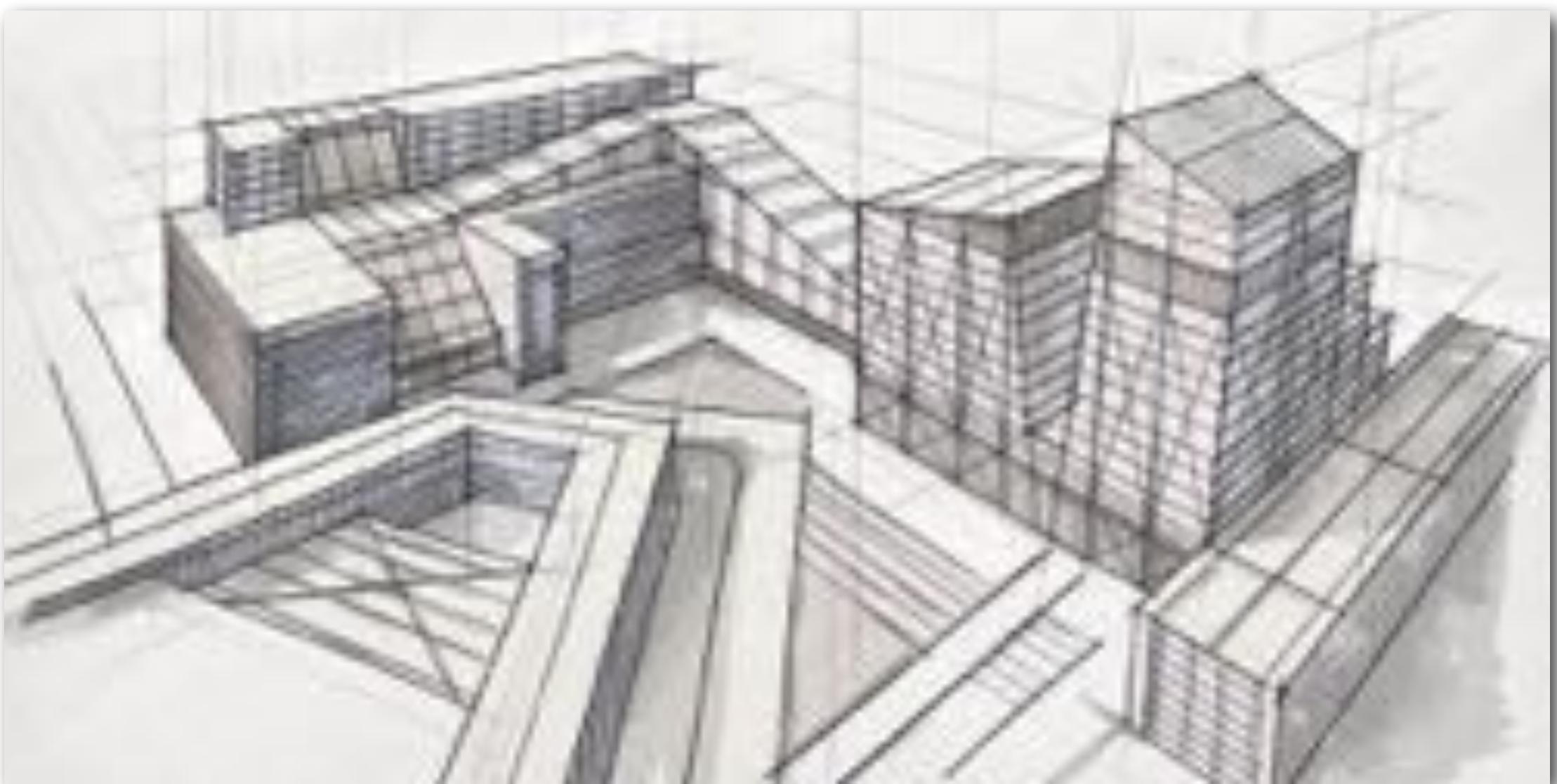
L'art du Découpage

GROSSE FONCTIONNALITÉ PETITES LIVRAISONS



L'art du Découpage





DÉCOUPER
FINEMENT

Apporter régulièrement de la valeur

Échouer rapidement

Réfléchir à la conception

DES LIVRAISONS SANS CRAINTE
DÉVELOPPER EN INCRÉMENTS
AFFINER SES LOTS

FLUX DE VALEUR

Bonne idée

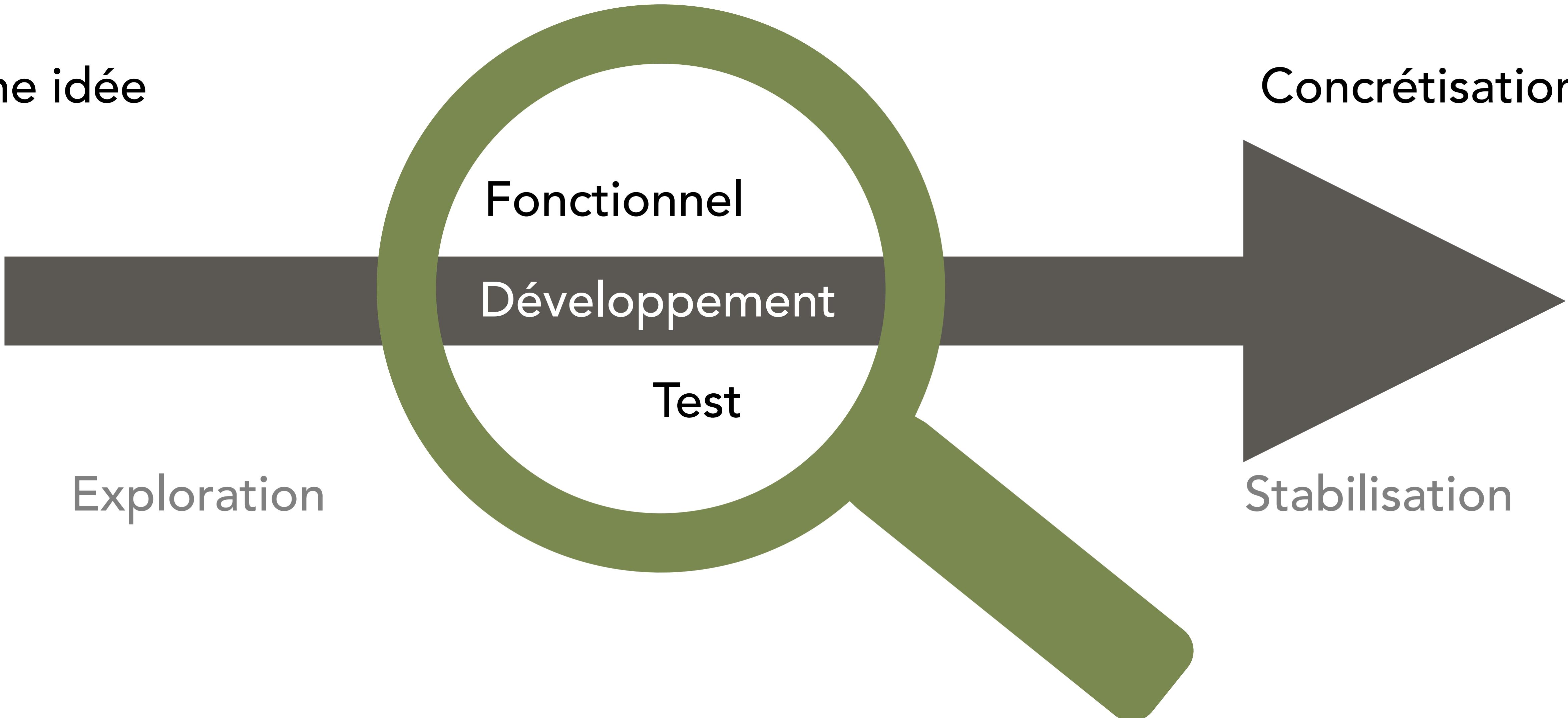
Concrétisation

Fonctionnel
Développement

Test

Exploration

Stabilisation





Yann Bertrand

(SuperPikaBoy)



@_YANNBERTRAND

DES LIVRAISONS SANS CRAINTE



Au fil de l'eau



AU FIL DE L'EAU



CORRECTIF



AMÉLIORATION GRAPHIQUE



PROTOTYPAGE



La fonctionnalité cachée



LA FONCTIONNALITÉ CACHÉE



FONCTIONNALITÉ COMPLEXE SUR 1 ÉCRAN



NOUVEL ÉCRAN OU REFONTE,
CRÉATION ENDPOINT



LA FONCTIONNALITÉ CACHÉE

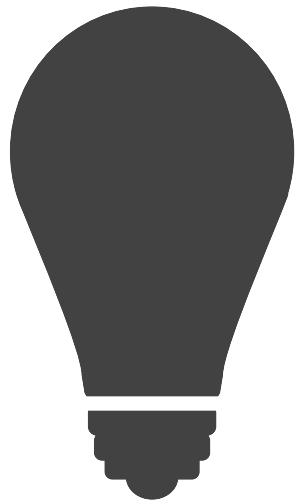
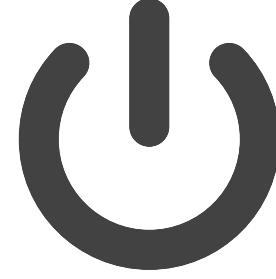




Le Feature flag

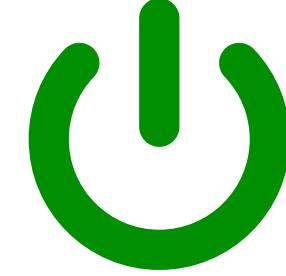


LE FEATURE FLAG





LE FEATURE FLAG





LE FEATURE FLAG



GROSSE FONCTIONNALITÉ



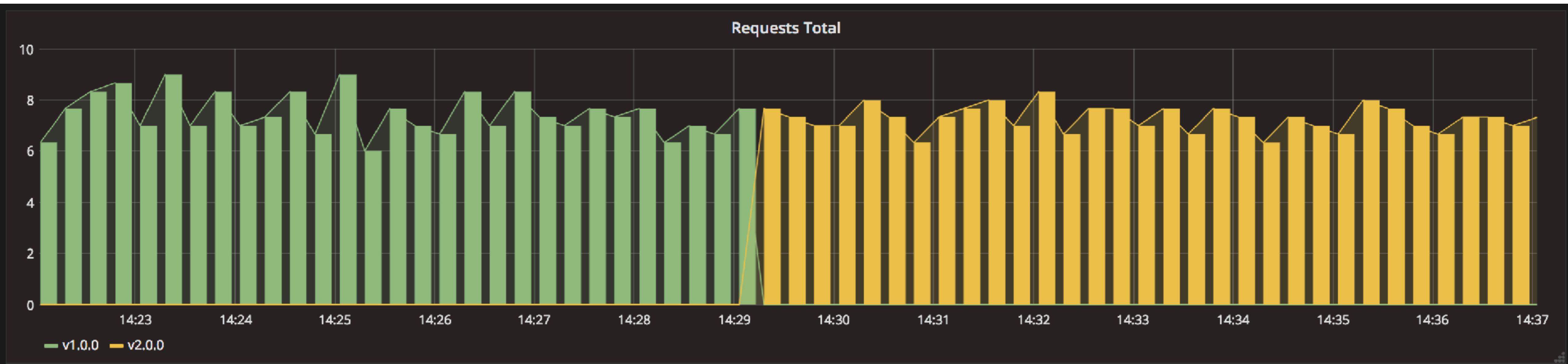
ATTENTE FORTE ET/OU DATE D'ACTIVATION IMPORTANTE



L'accès restreint



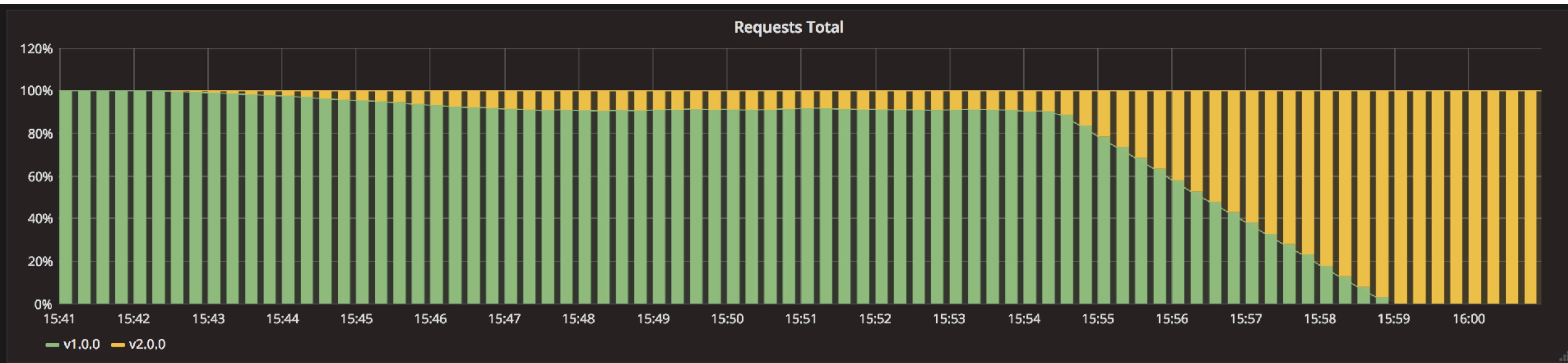
L'ACCÈS RESTREINT



DÉPLOIEMENT “BLUE/GREEN”



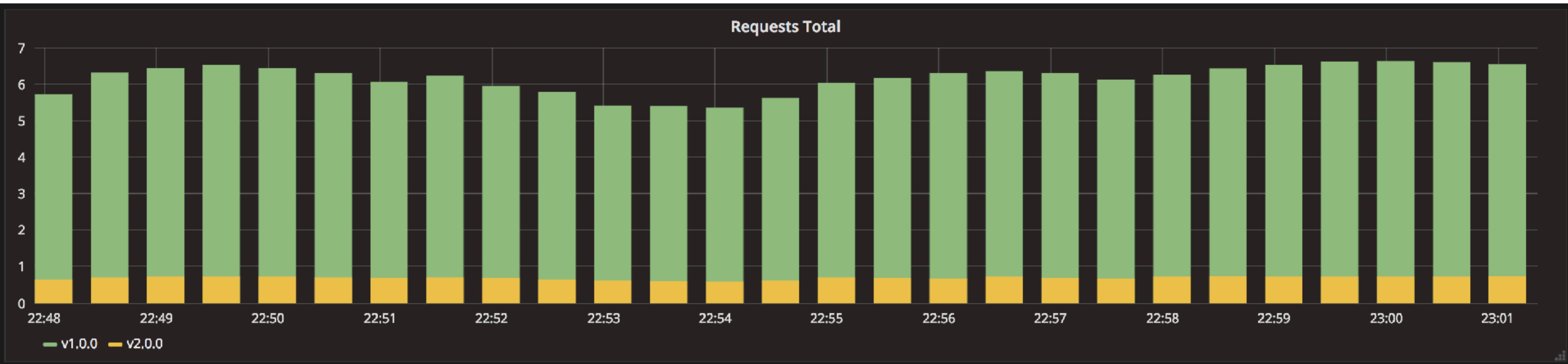
L'ACCÈS RESTREINT



DÉPLOIEMENT “CANARY”



L'ACCÈS RESTREINT



TESTS A/B



L'ACCÈS RESTREINT



**INCERTITUDE FONCTIONNELLE OU
TECHNIQUE**

DÉVELOPPER EN INCRÉMENTS

LE TEMPS INFORMATIQUE
N'EST PAS
UNIDIRECTIONNEL



Découpage rétrocompatible



DÉCOUPAGE RÉTROCOMPATIBLE

Comment ?

Exemple :

Récupération articles de journaux via une API HTTP en lecture seule

`article` contient un champ `date` au format timestamp (ex : 1685202389)

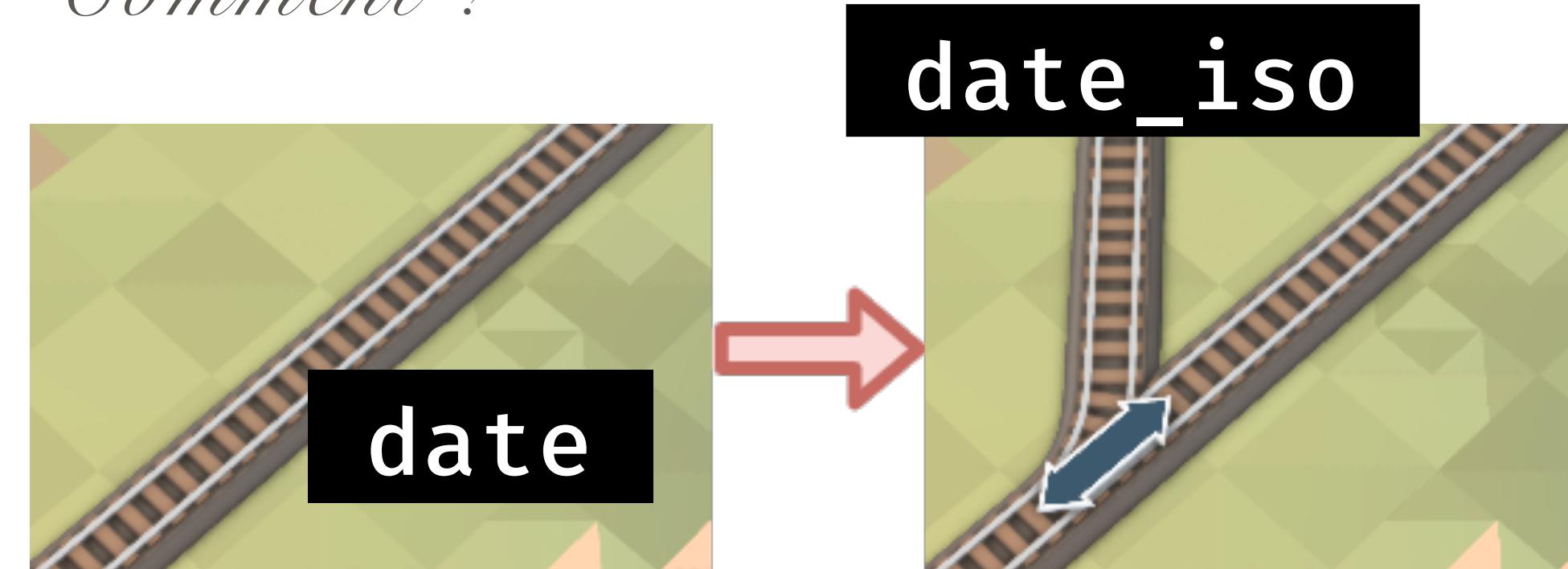
Migration souhaitée vers un format gérant les fuseaux horaires



DÉCOUPAGE RÉTROCOMPATIBLE

Comment ?

1. DUPLICATION

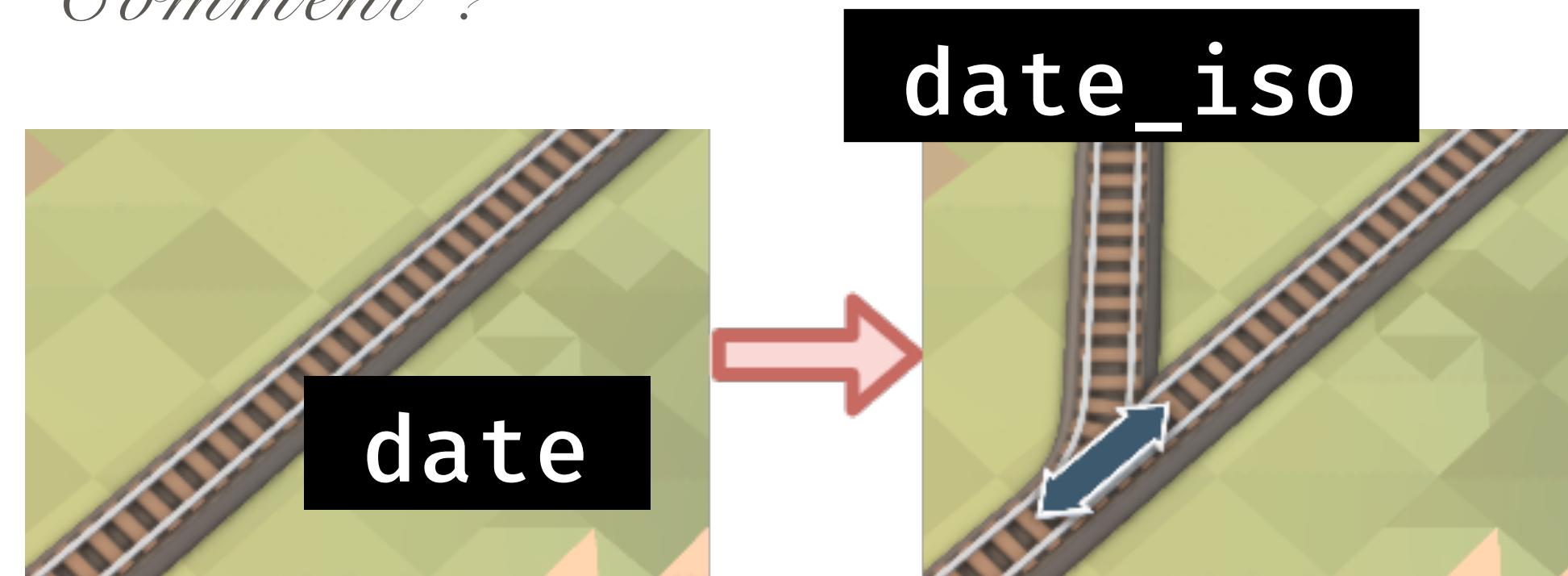




DÉCOUPAGE RÉTROCOMPATIBLE

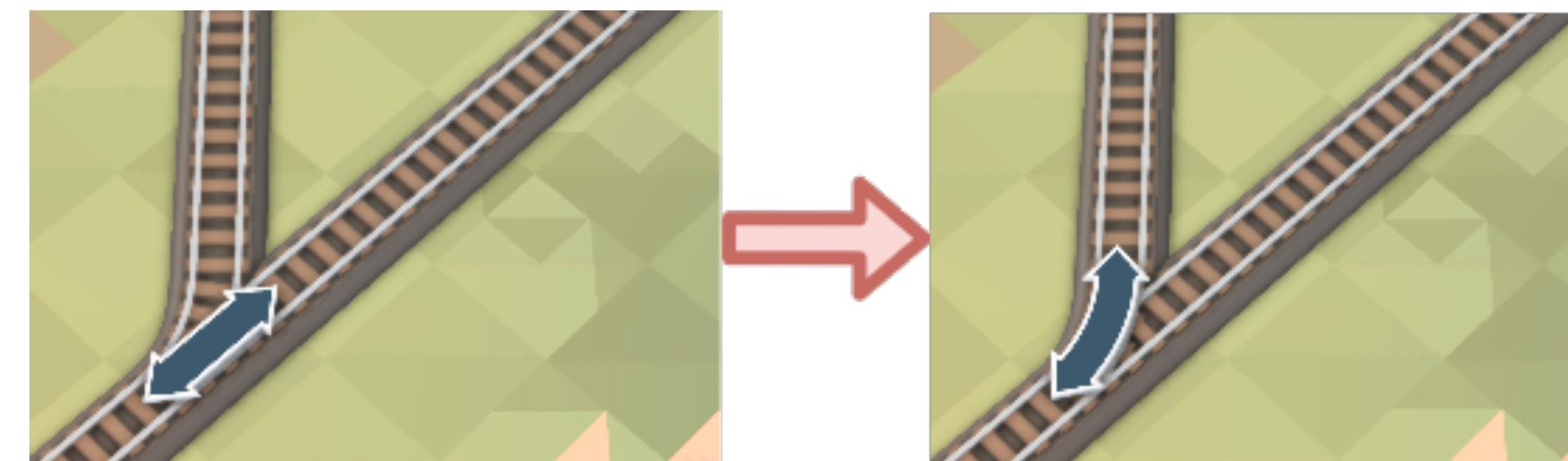
Comment ?

1. DUPLICATION



`date_iso`

2. BASCULE

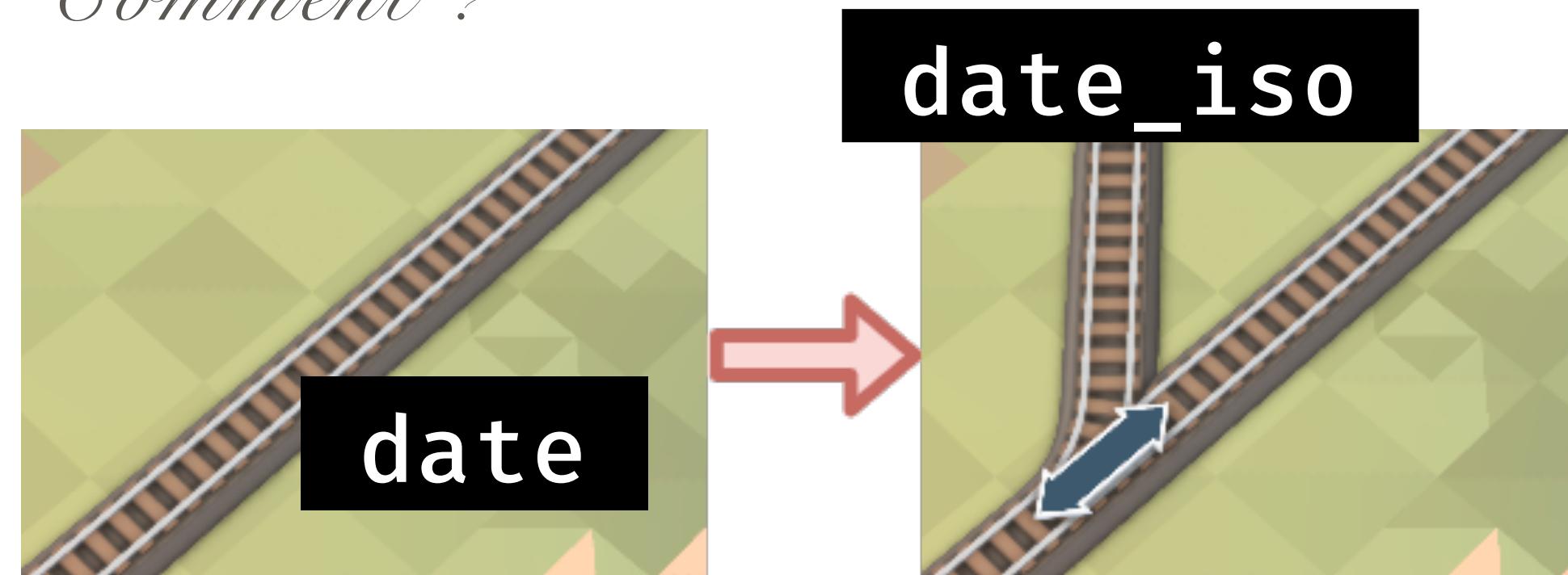




DÉCOUPAGE RÉTROCOMPATIBLE

Comment ?

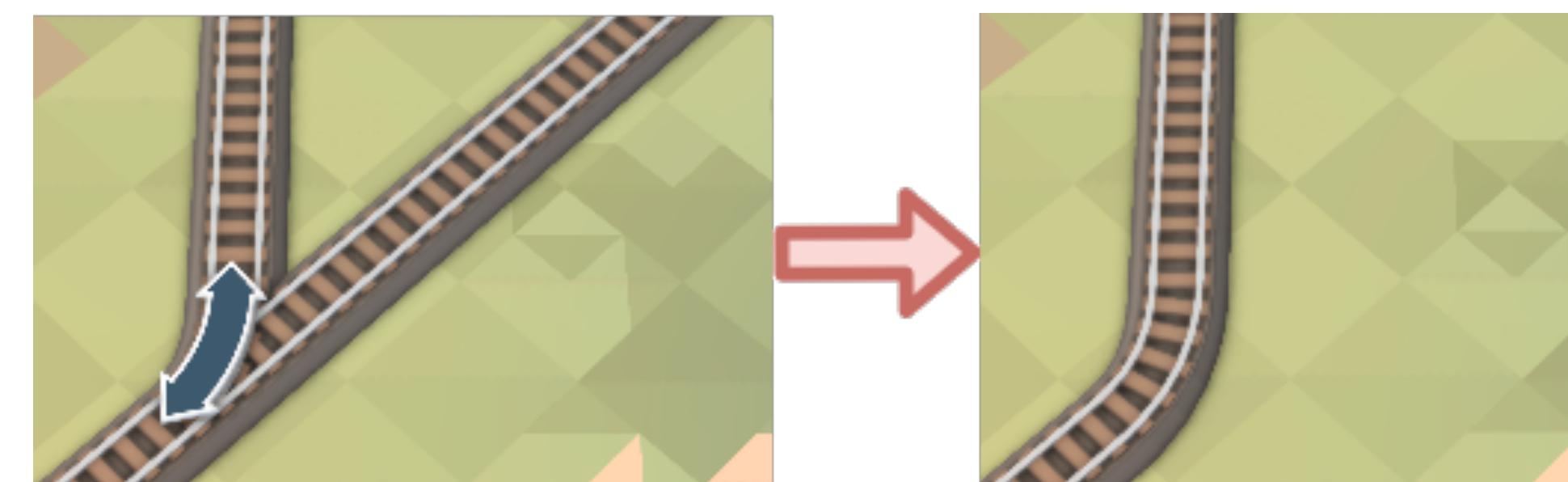
1. DUPLICATION



2. BASCULE



3. NETTOYAGE





Découpage par techno

 DÉCOUPAGE PAR TECHNO

 ATTENTION AUX CONTRATS D'INTERFACES



Découpage par tranche fonctionnelle



DÉCOUPAGE DE TRANCHE FONCTIONNELLE



NÉCESSITE RECAL POUR
CONSTRUIRE LES TRANCHES



Découpage fonctionnalité endettée



DÉCOUPAGE FONCTIONNALITÉ ENDETTÉE

STRATÉGIE 1 : NOUVEAUTÉ => REFACTO

AVANTAGES

- Valeur utilisateur arrive plus vite

INCONVÉNIENTS

- Travail sur du code non maîtrisé
- Le développement prend plus de temps
- Risque de laisser de côté le refacto



DÉCOUPAGE FONCTIONNALITÉ ENDETTÉE

STRATÉGIE 2: REFACTO => NOUVEAUTÉ

AVANTAGES

- Reprise de contrôle sur le legacy

INCONVÉNIENTS

- Pas de valeur utilisateur avant la dernière étape

3

AFFINER SES LOTS

TAILLE DE LOT IDÉALE

I Am Devloper · Follow @iamdevloper · Follow

10 lines of code = 10 issues.

500 lines of code = "looks fine."

Code reviews.

10:58 AM · Nov 5, 2013

6.6K Reply Copy link

Read 109 replies

TAILLE DE LOT IDÉALE

Étude Cisco 2006

- 10 mois
- 50 devs à distance
- 2 500 revues de code
- 3,2 millions de lignes de code



TAILLE DE LOT IDÉALE

Étude Cisco 2006

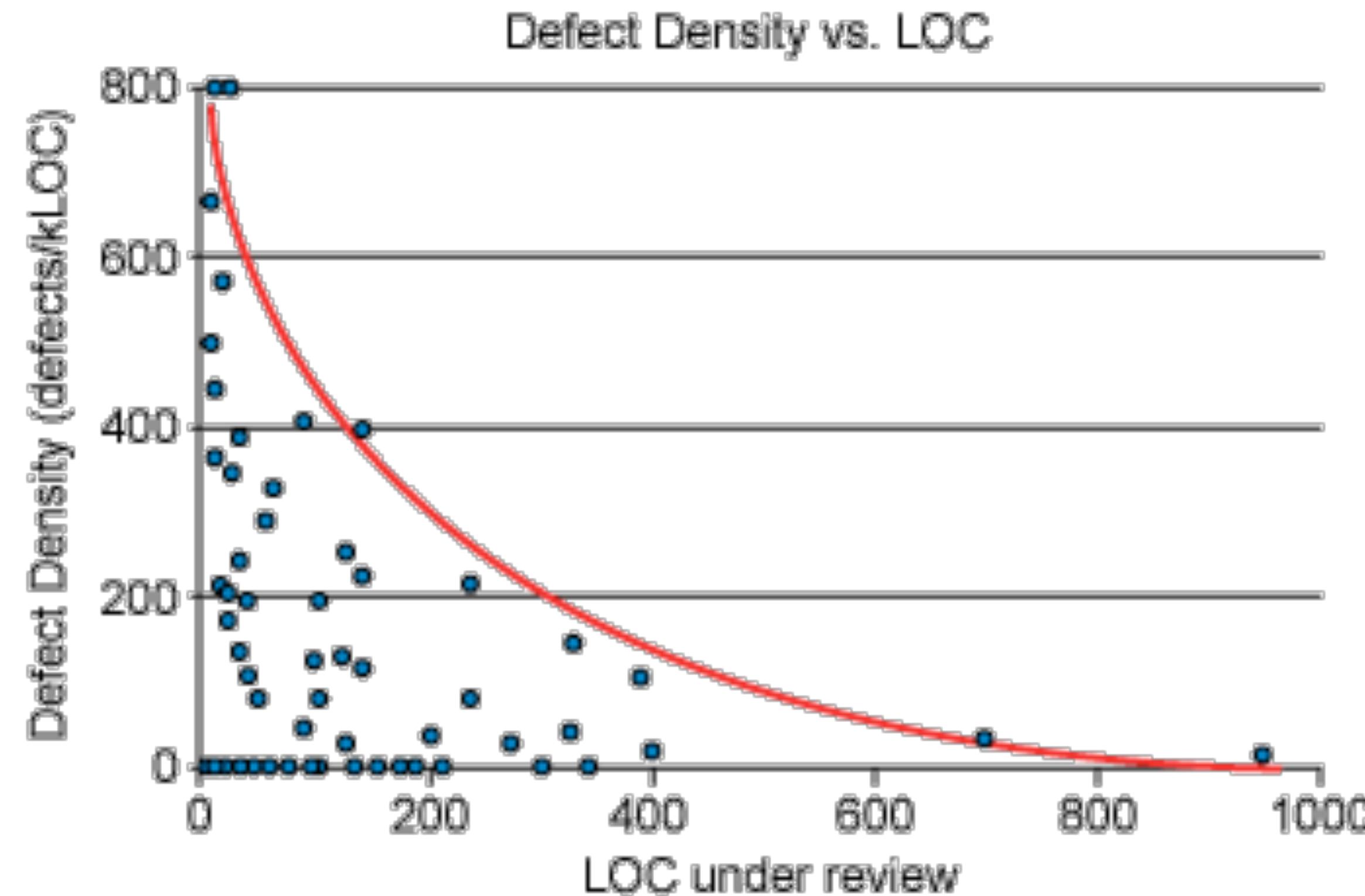
3 mesures

- ★ Temps de revue
- ★ Nombre de défauts détectés
- ★ Nombre de ligne de code



TAILLE DE LOT IDÉALE

Etude Cisco 2006



TAILLE DE LOT IDÉALE

Etude Cisco 2006

- ★ Maximum 400 lignes de code par revue
- ★ Devrait correspondre à 1h30 de revue
- ★ ~70-90% des défauts détectés

<https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>

TAILLE DE LOT IDÉALE

[TECH] Suppression d'une méthode non utilisée #3940

[Edit](#)[↳ Code](#)

Merged by pix-service... dev ← fix-remove-dead-code-compare-db-object on Jan 14, 2022 v3.155.0

[Conversation 3](#)[Commits 1](#)[Checks 13](#)[Files changed 1](#)

+0 -7

[all commits](#) [File filter](#) [Conversations](#) [Jump to](#)

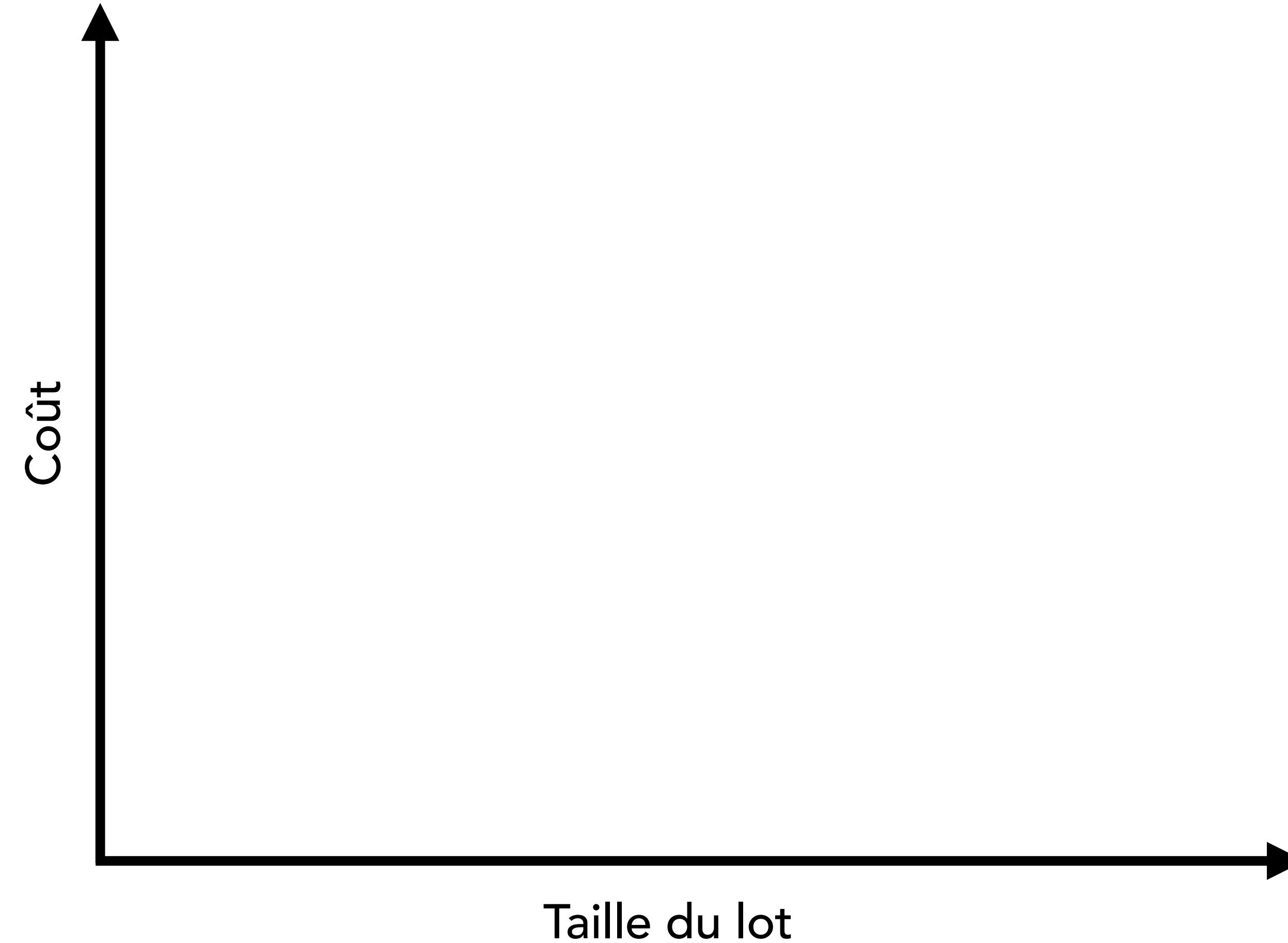
0 / 1 files viewed

[Review in codespace](#)[Review changes](#)[7 api/tests/test-helper.js](#)[Viewed](#) [...](#)

```
@@ -155,12 +155,6 @@ function catchErr(promiseFn, ctx) {  
155 155      };  
156 156    }  
157 157  
158 158    function compareDatabaseObject(evaluatedObject, expectedObject) {  
159 159      return expect(_.omit(evaluatedObject, ['id', 'createdAt', 'updatedAt'])).to.deep.equal(  
160 160        _.omit(expectedObject, ['id', 'createdAt', 'updatedAt'])  
161 161      );  
162 162    }  
163  
164 158    chai.use(function (chai) {
```

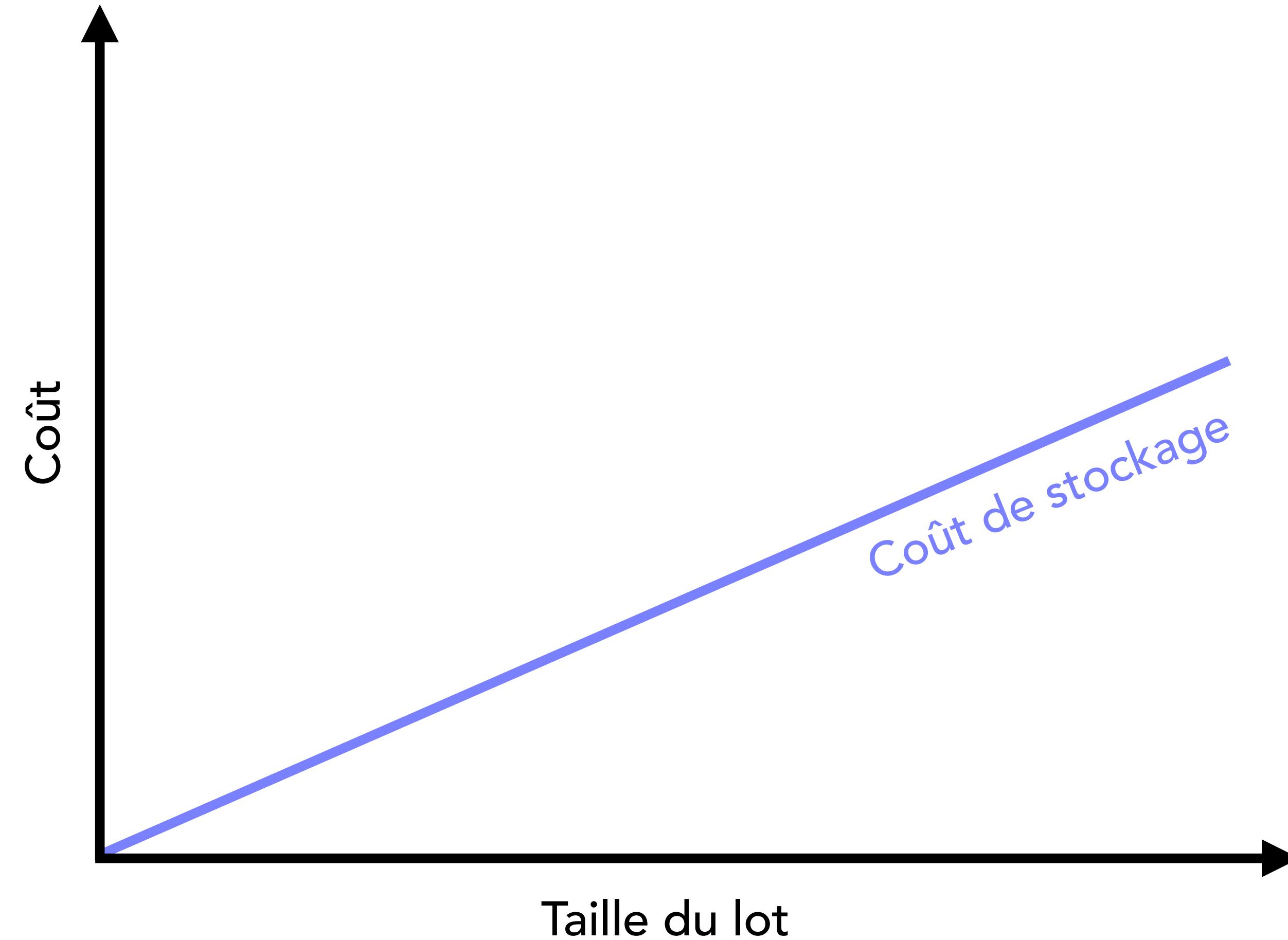
TAILLE DE LOT IDÉALE

Les coûts des lots



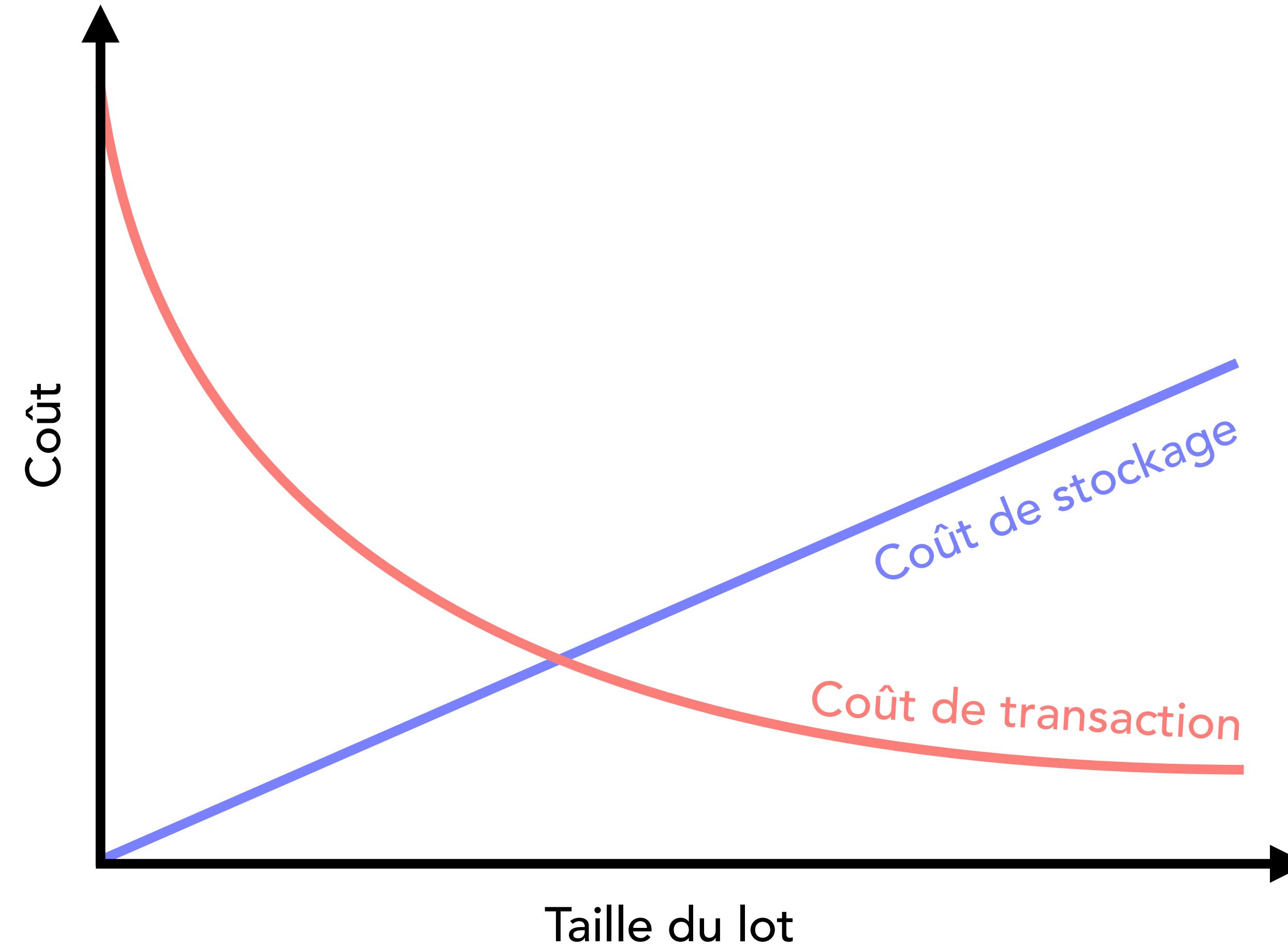
TAILLE DE LOT IDÉALE

Les coûts des lots



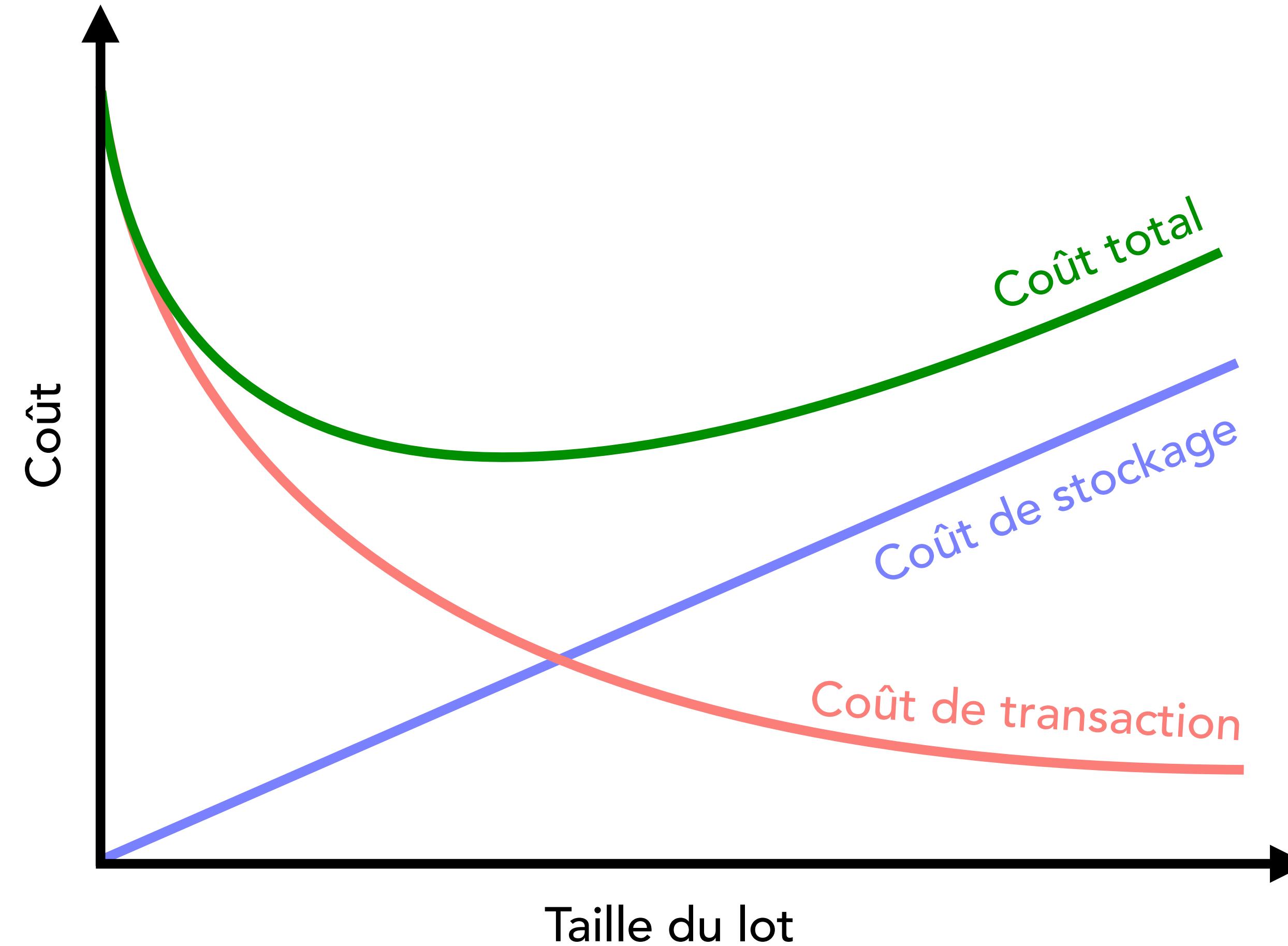
TAILLE DE LOT IDÉALE

Les coûts des lots



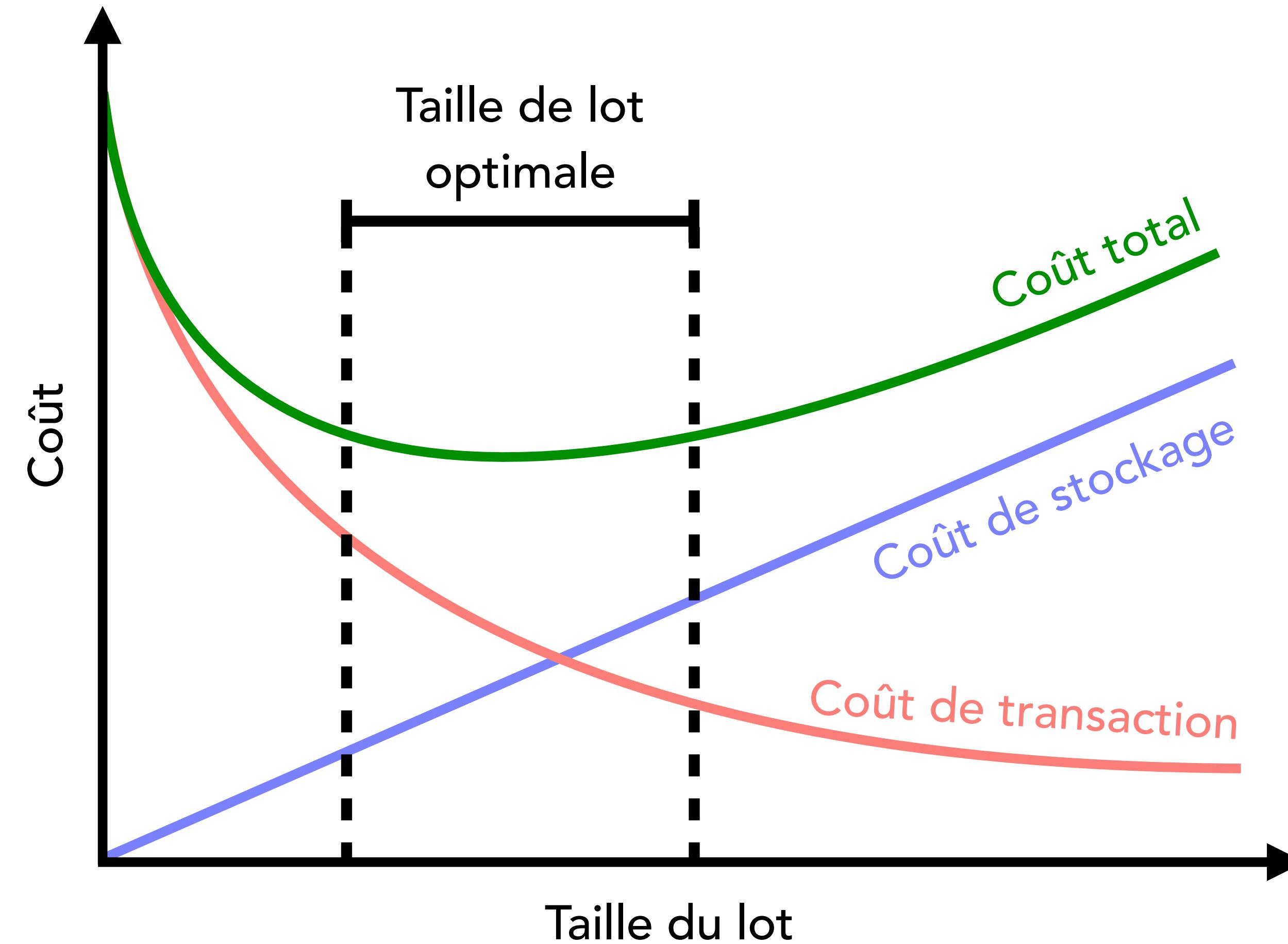
TAILLE DE LOT IDÉALE

Les coûts des lots



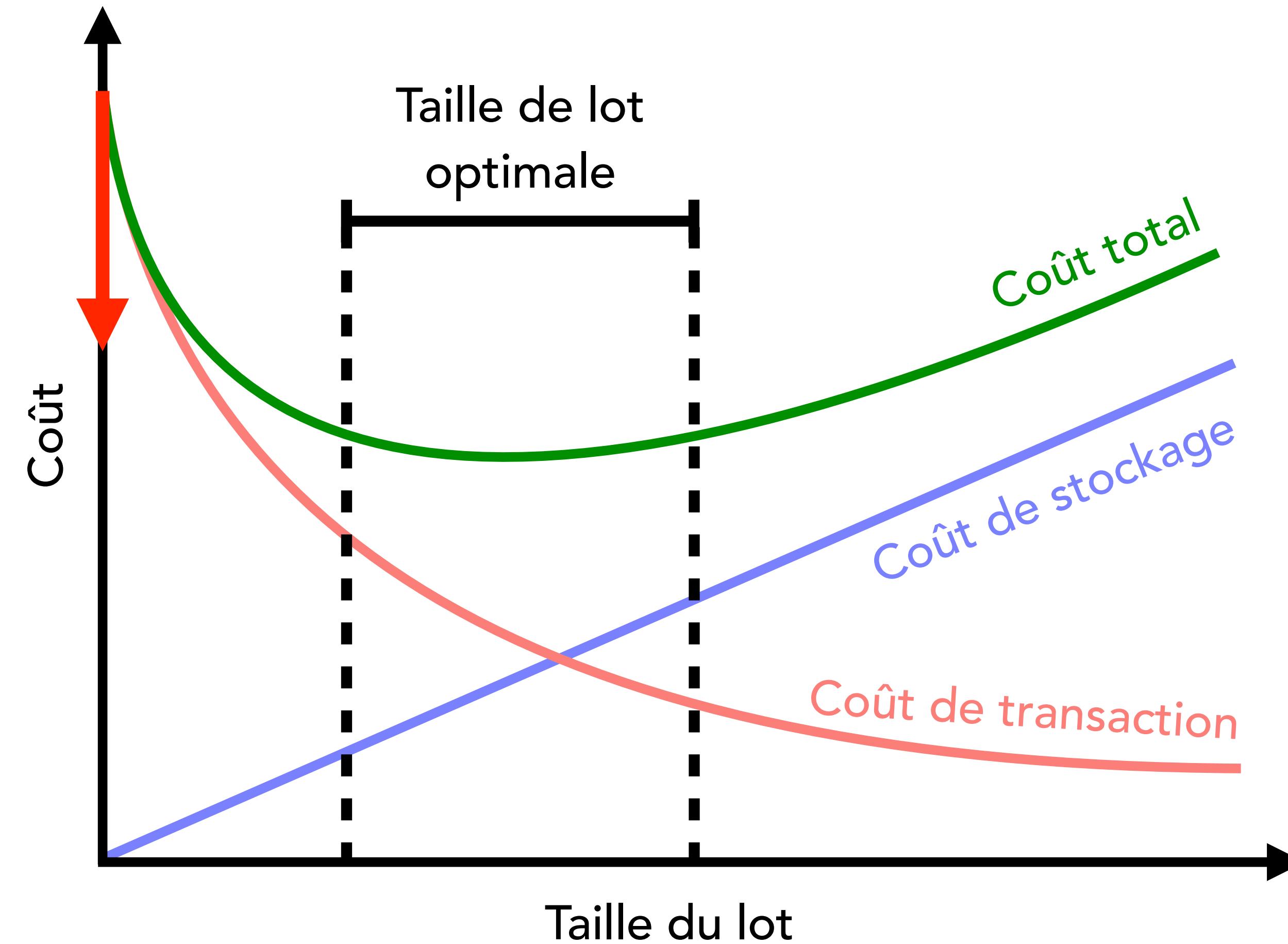
TAILLE DE LOT IDÉALE

Les coûts des lots



TAILLE DE LOT IDÉALE

Les coûts des lots



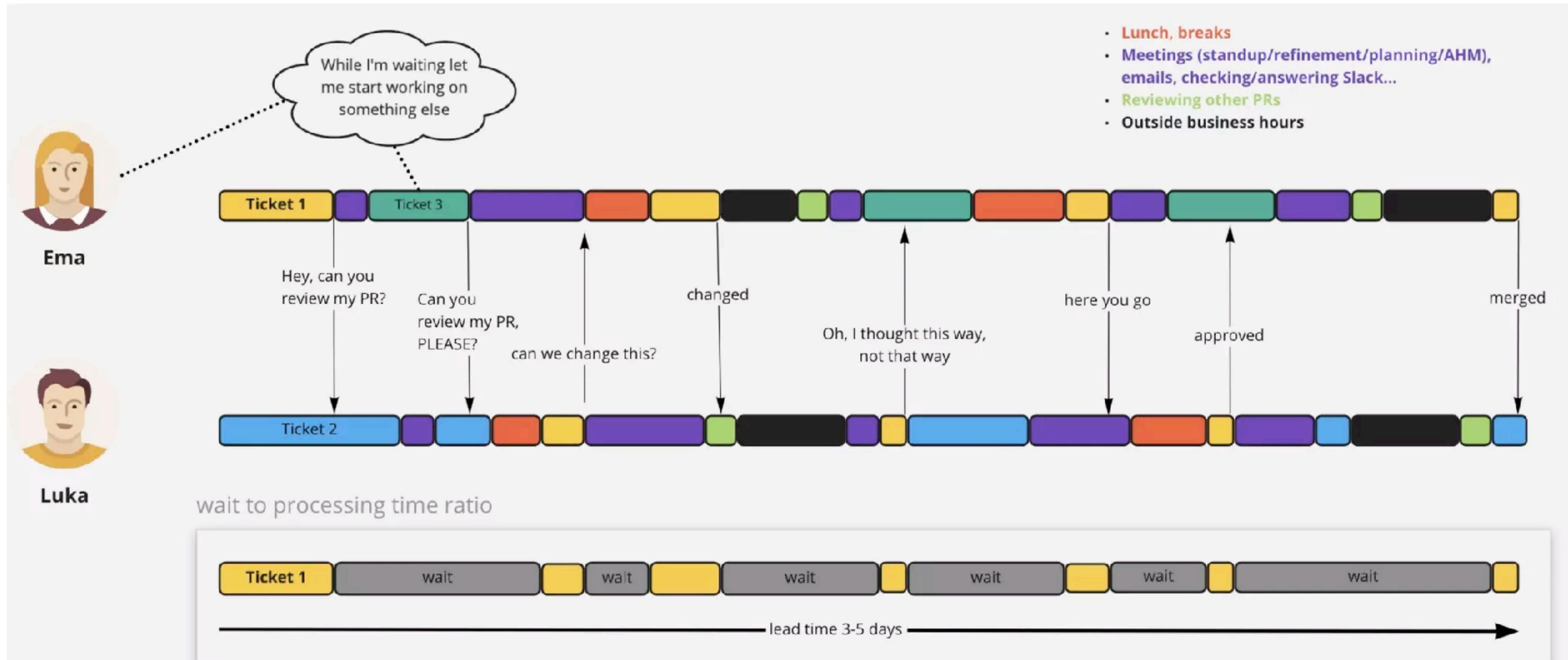


Limiter les revues asynchrones



LIMITER LES REVUES ASYNCHRONES

Loi de Little



<https://www.youtube.com/watch?v=fYFruezJEDs>



LIMITER LES REVUES ASYNCHRONES

“

1 SEULE LIGNE DE CODE, REVUE INSTANTANÉMENT

C'EST CE QU'ON APPELLE DU
PAIR/MOB PROGRAMMING

— *Dragan Stepanović*

”

<https://www.youtube.com/watch?v=fYFruezJEDs>



Itérer plus vite

⚡ ITÉRER PLUS VITE

Un exemple fonctionnel

WIKIPEDIA
L'encyclopédie libre



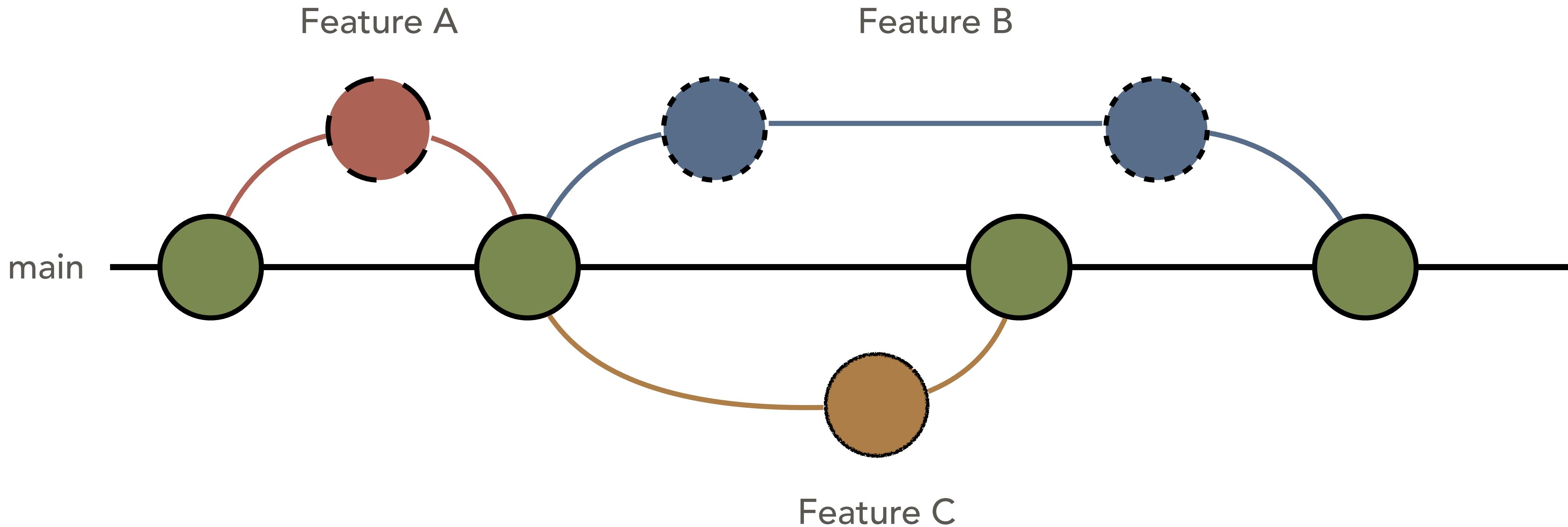
Langue	Nombre d'articles
Français	2 528 000+ articles
日本語	1 376 000+記事
Русский	1 921 000+ статей
Italiano	1 814 000+ voci
فارسی	965 000+ مقاله
English	6 668 000+ articles
Español	1 869 000+ artículos
Deutsch	2 808 000+ Artikel
中文	1 360 000+ 条目 / 條目
Português	1 103 000+ artigos

FR ⚖

— 文 A Lire Wikipédia dans votre langue —

⚡ ITÉRER PLUS VITE

Vers le Déploiement Continu



CONCLUSION

CONCLUSION

Méthodes de livraison

 Au fil de l'eau

 La fonctionnalité cachée

 Le Feature flag

 L'accès restreint

Méthodes de découpage

 Découpage rétrocompatible

 Découpage par techno

 Découpage de tranche fonctionnelle

 Découpage fonctionnalité endettée

LE PRODUIT ET LA BASE DE CODE SONT VIVANTS

CONCLUSION

-

GAGNONS EN CONFIANCE
ITÉRONS DAVANTAGE



-

SOURCES

- Pourquoi et comment gérer la compatibilité ascendante ?
 - <https://techblog.deepki.com/retrocompat/>
- The anatomy of the perfect Pull Request
 - <https://hugoodias.medium.com/the-anatomy-of-a-perfect-pull-request-567382bb6067>
- Optimal Pull Request size
 - <https://smallbusinessprogramming.com/optimal-pull-request-size/>
- Async code reviews are killing your company's throughput
 - <https://www.youtube.com/watch?v=fYFruezJEDs>
- Continuous delivery vs. SCRUM : est-ce compatible ?
 - <https://www.youtube.com/watch?v=OCe-6X5MT9c>
- Le code de Pix
 - <https://github.com/1024pix/pix>
- The ideal PR is 50 lines long
 - <https://graphite.dev/blog/the-ideal-pr-is-50-lines-long>

Feedbacks

