



C'est l'histoire d'une app web



Qui rencontre le #serverless

Yves @Sogilis.com
Brissaud

 @_crev_
 eunomie

Chapitre 1

L'ignorance



Besoin d'une app web





Besoin d'une app web



A quoi sert l'app ?

A uploader et traiter des photos



Upload

Paperclip



Resize

1 utilisateur



2 utilisateurs

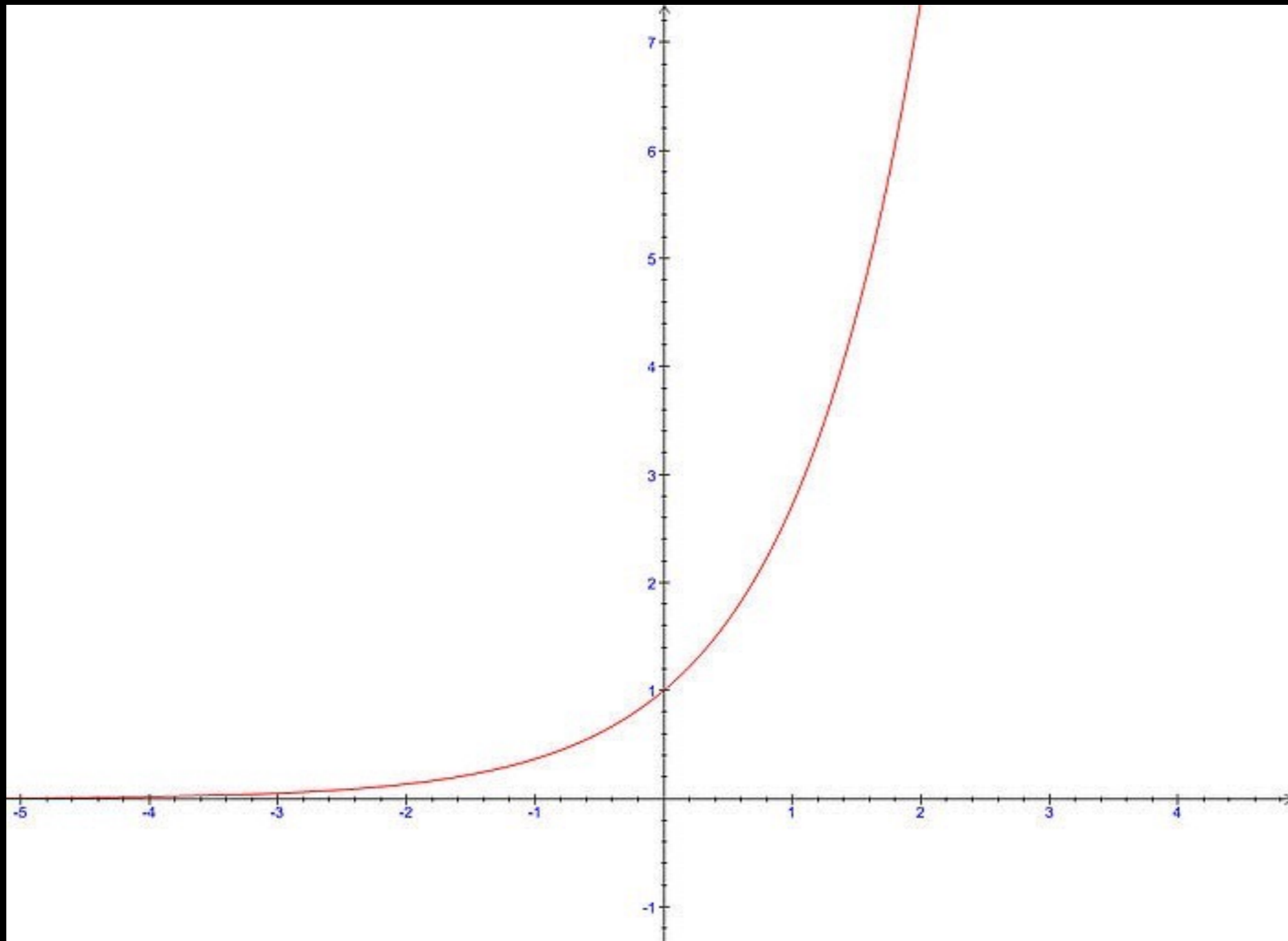


5 utilisateurs



wat

CPU Load





Upload

Paperclip



Resize

- 500+ photos
- 1080p minimum
- 3 resizes en sortie
(1080, 720, 360)

PAR UTILISATEUR

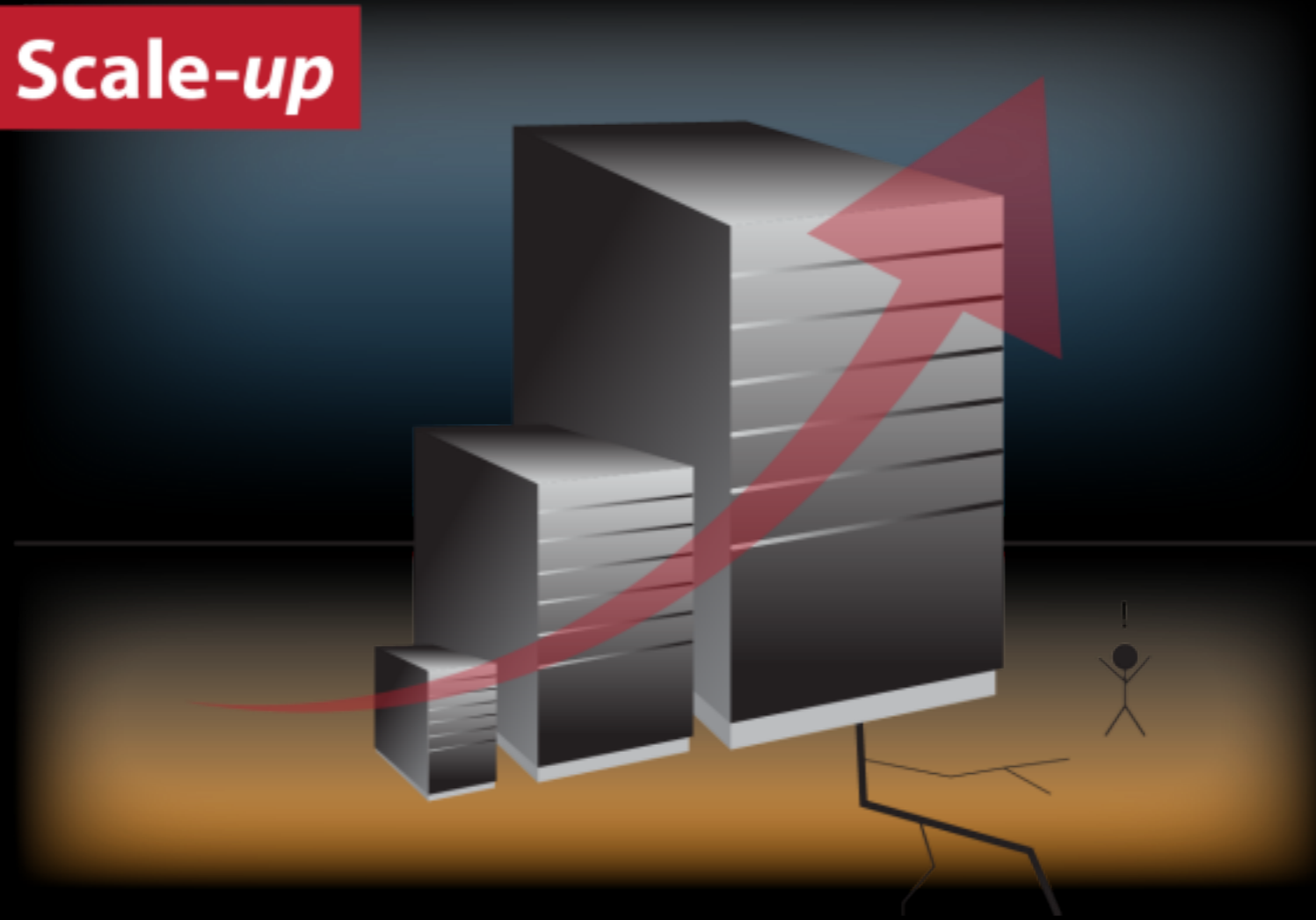
100% CPU dans
ImageMagick

Chapitre 2

La découverte

La solution classique

Scale-up



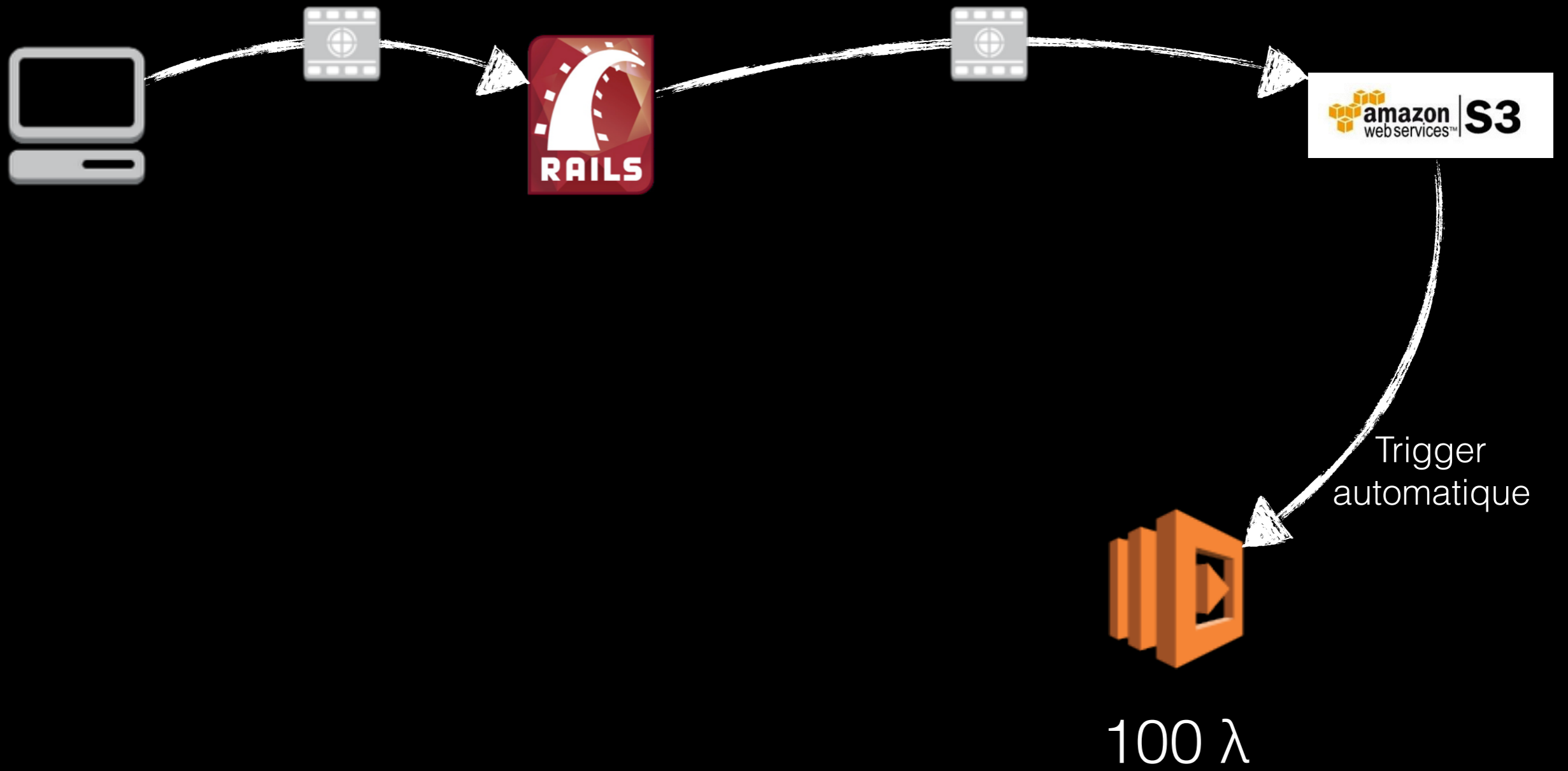
L'autre solution

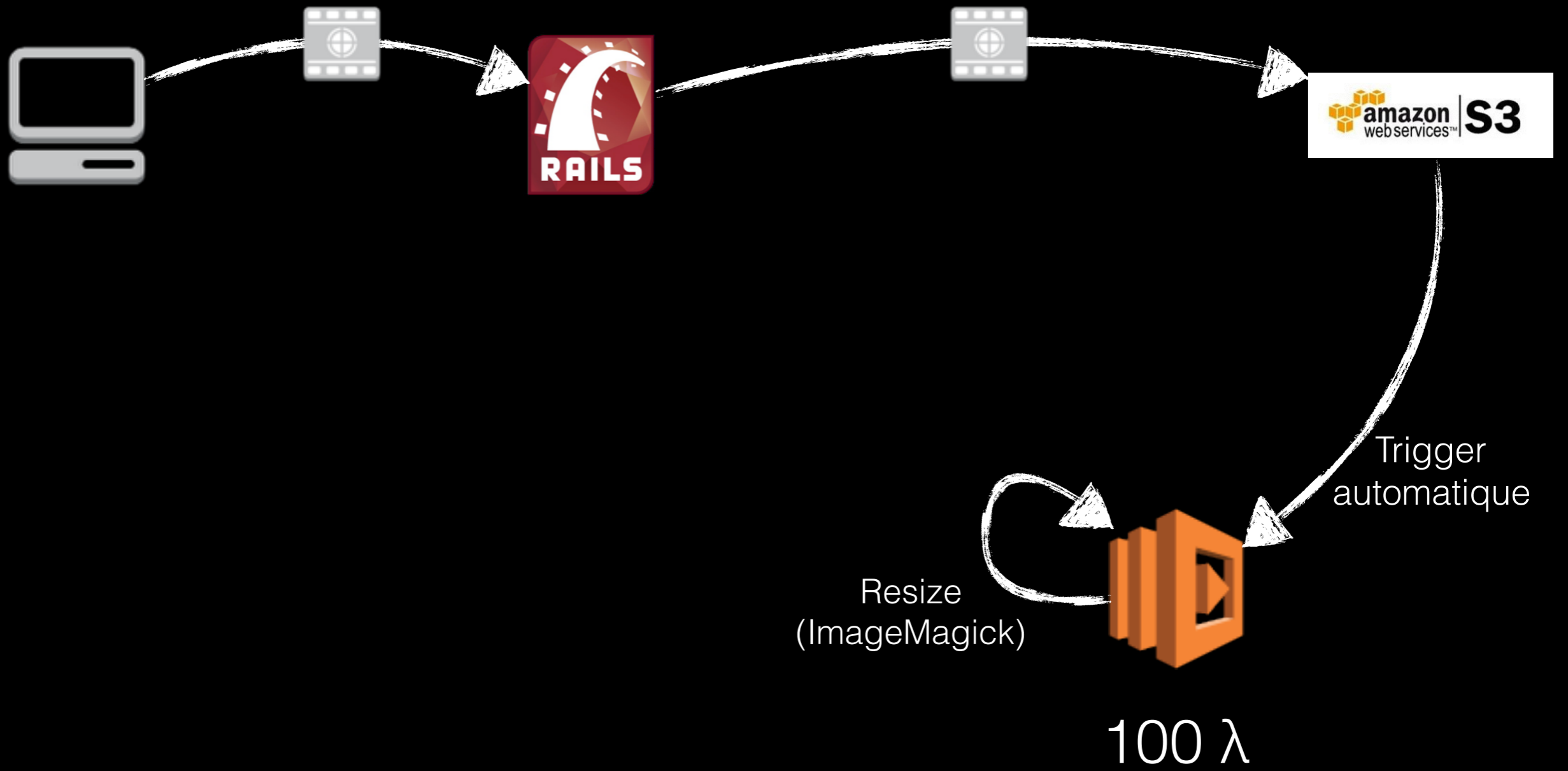
λ

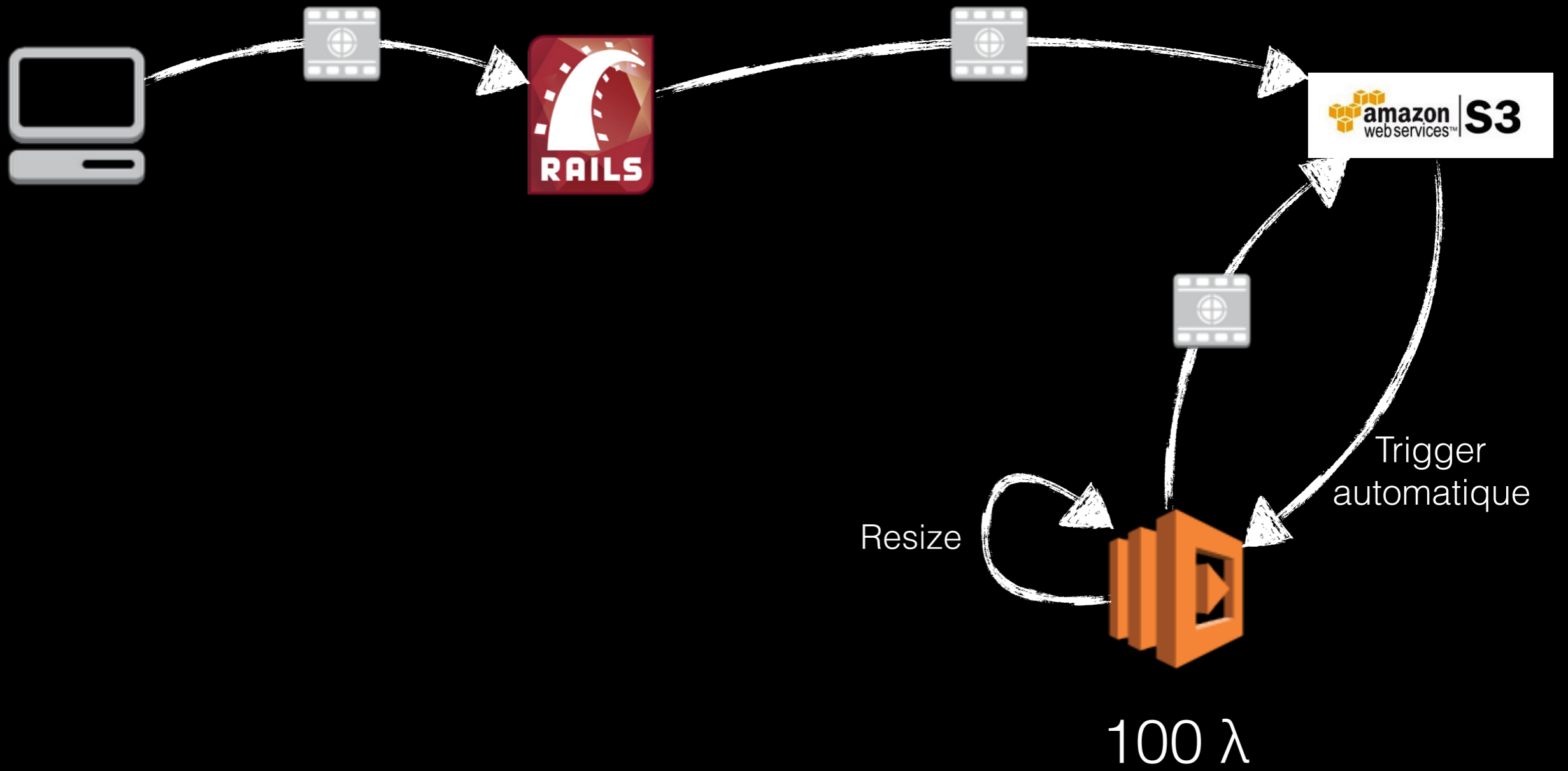


AWS Lambda est un service de calcul sans serveur qui exécute votre code en réponse à des événements et gère automatiquement les ressources de calcul sous-jacentes pour vous.









- Rails -> proxy vers S3
- Pas de calcul sur les serveurs
- Disponibilité (100 lambda)



That'll work

Chapitre 3

Vers l'infini et au delà

*« En tant qu'utilisateur, afin de préparer
ma vidéo, je veux découper et
redimensionner chaque photo. »*

— Da Backlog

ENTRÉE

± 1800 photos

$> 1080p$ & $\leq 4K$

Sur action utilisateur

ENTRÉE

± 1800 photos

$> 1080p$ & $\leq 4K$

Sur action utilisateur

SORTIE

± 3600 photos

$1080p + 720p$ pour chaque

Le plus **rapidement** possible

ENTRÉE

± 1800 photos

> 1080p & ≤ 4K

Sur action utilisateur

SORTIE

± 3600 photos

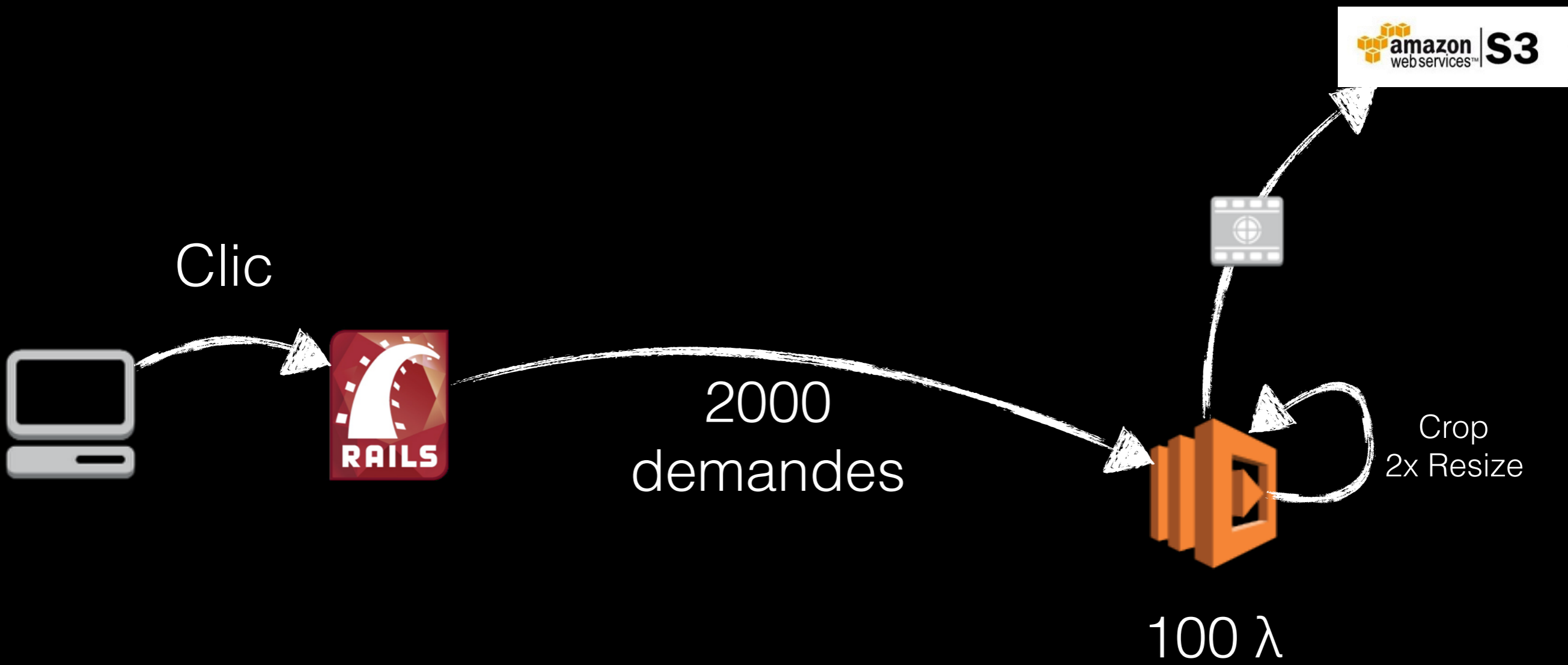
1080p + 720p pour chaque

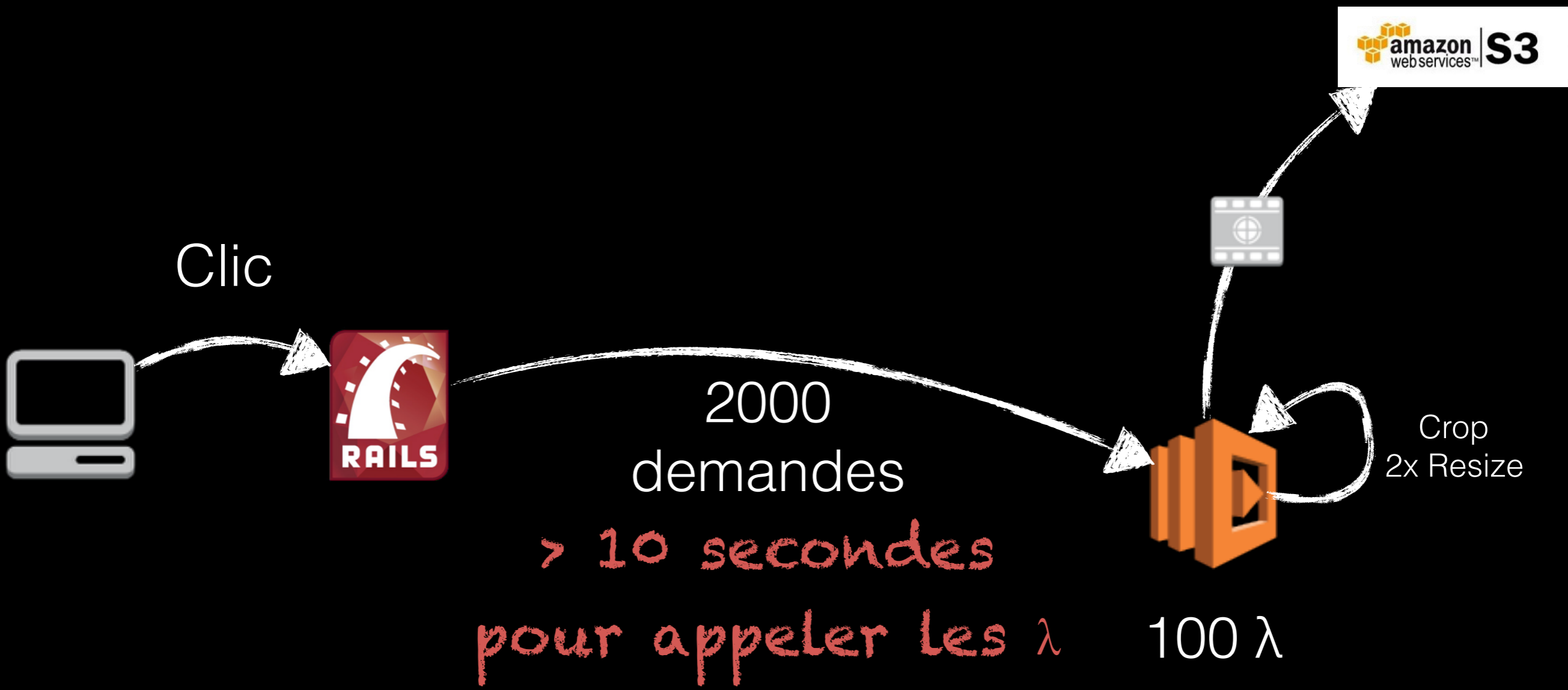
Le plus rapidement possible

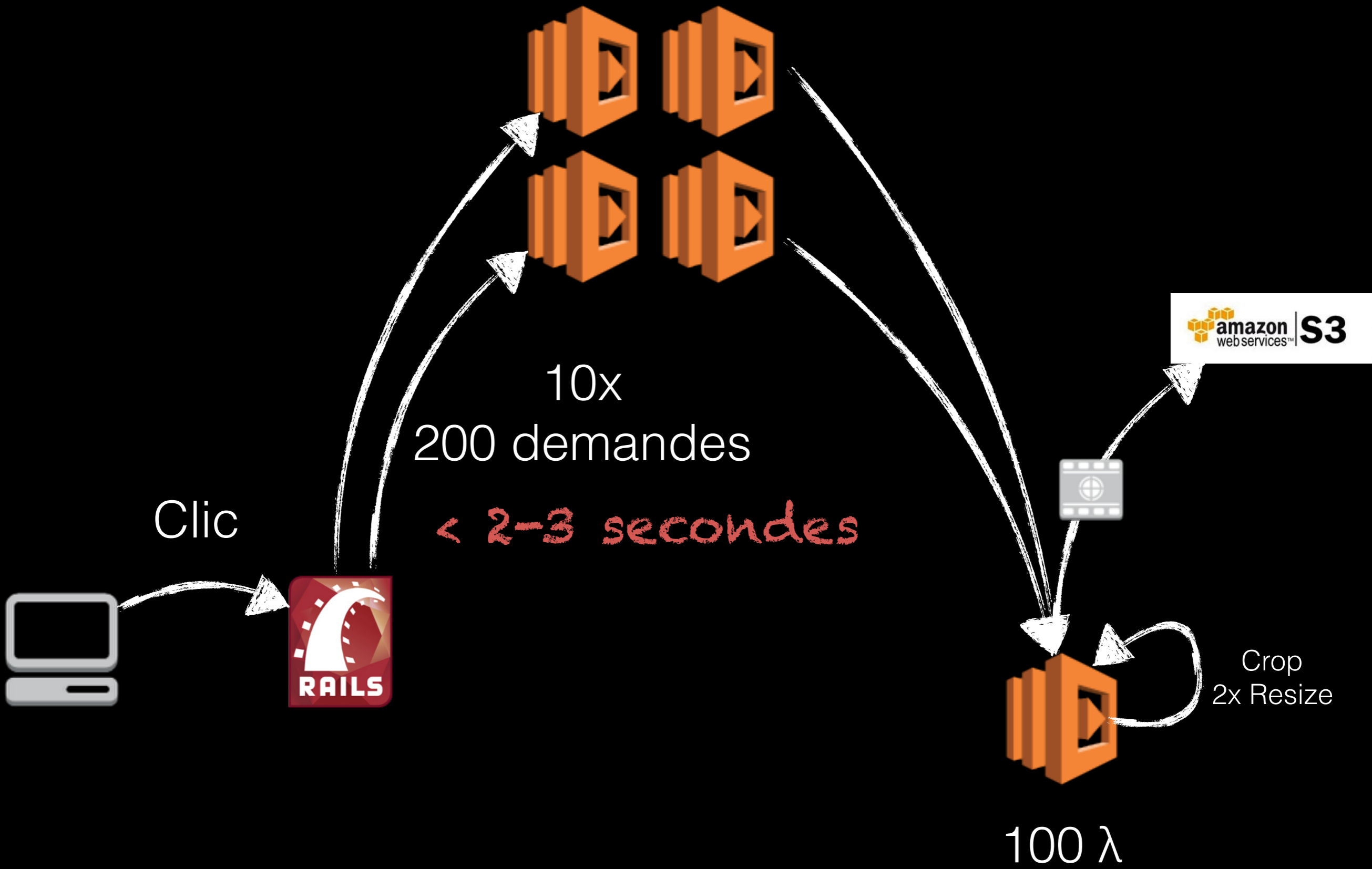
« 10 SECONDES

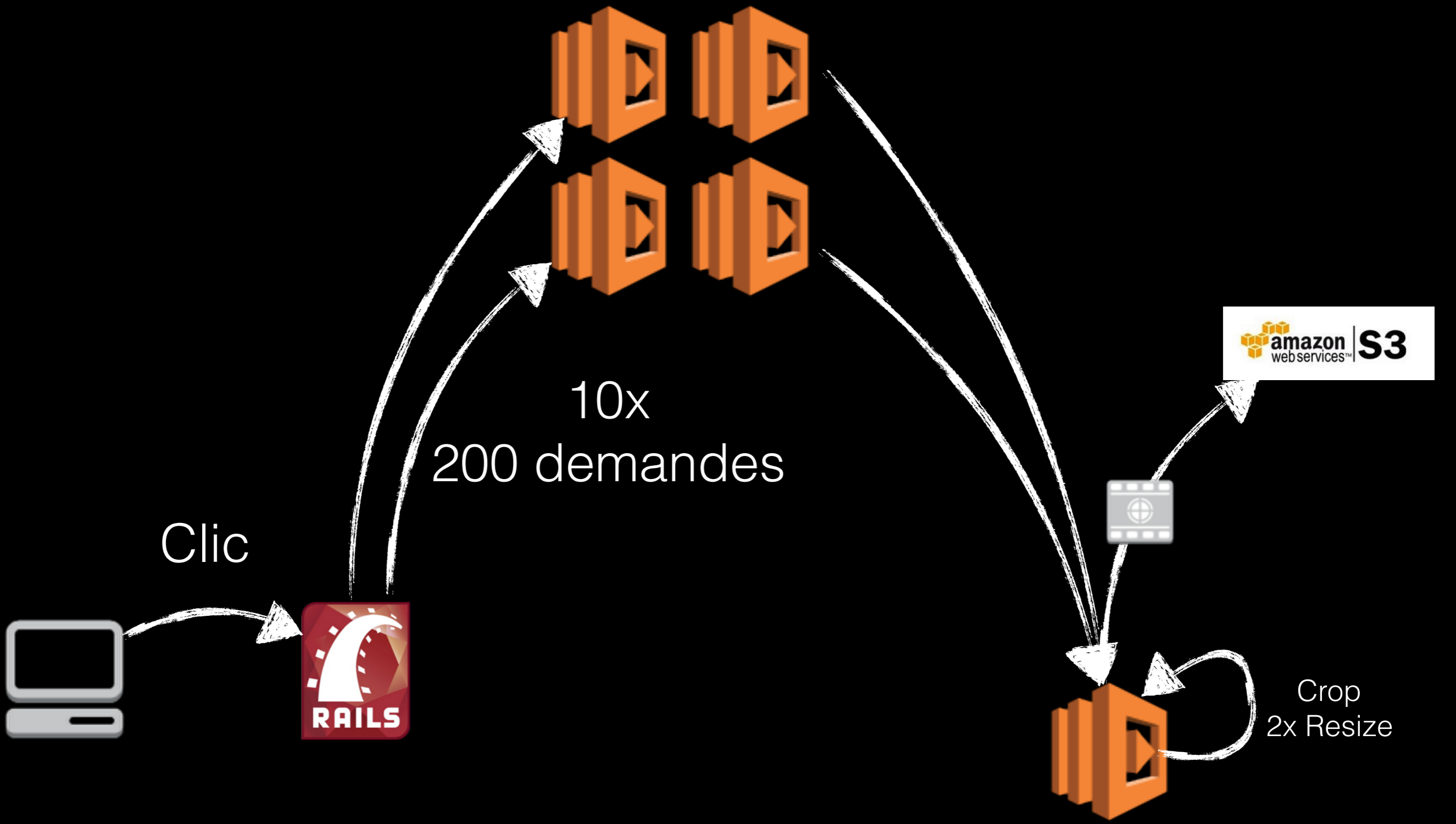
*« Et quand il y a
plusieurs utilisateurs ? »*

λ FTW





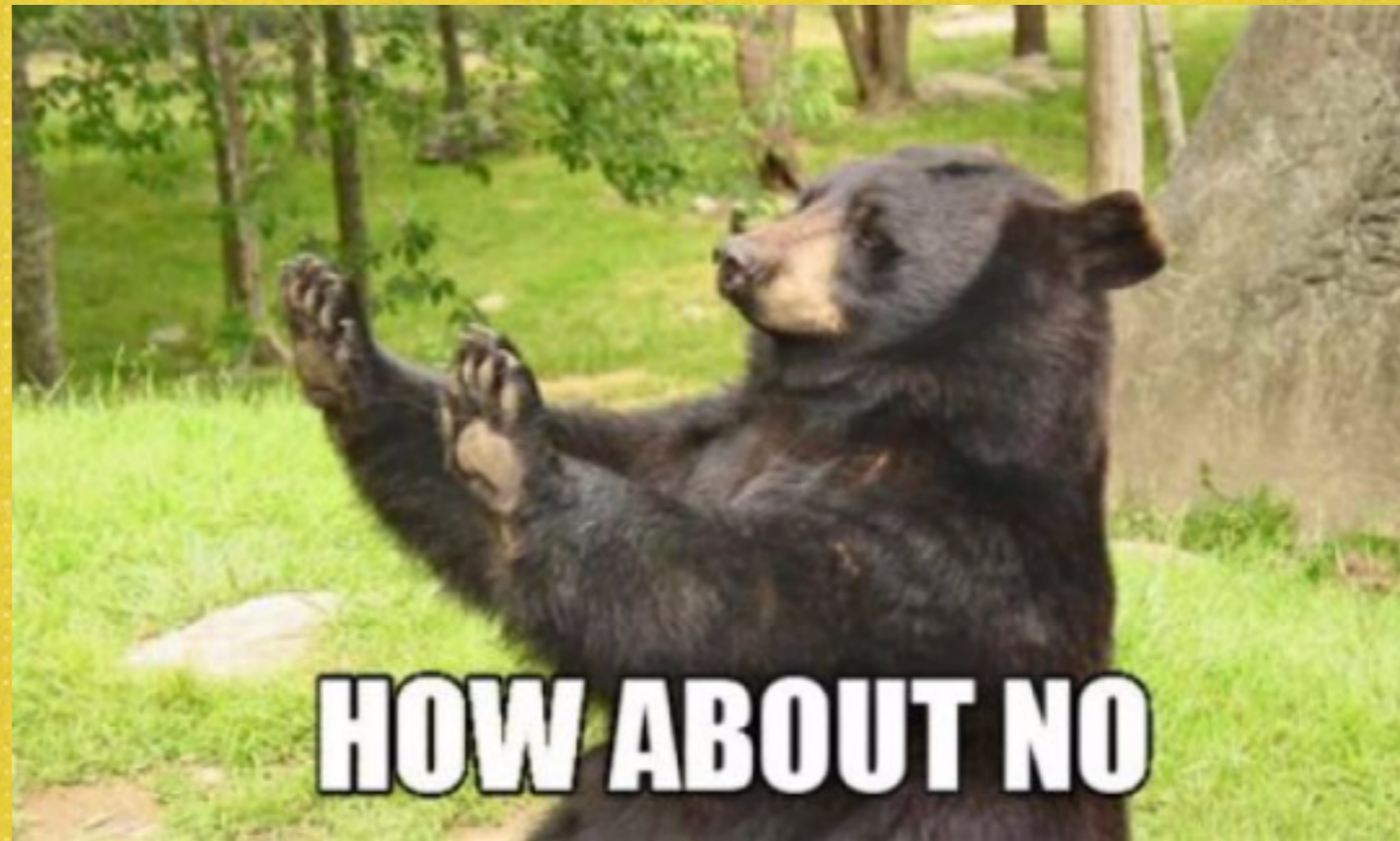




40 secondes mini
sans parler de latence



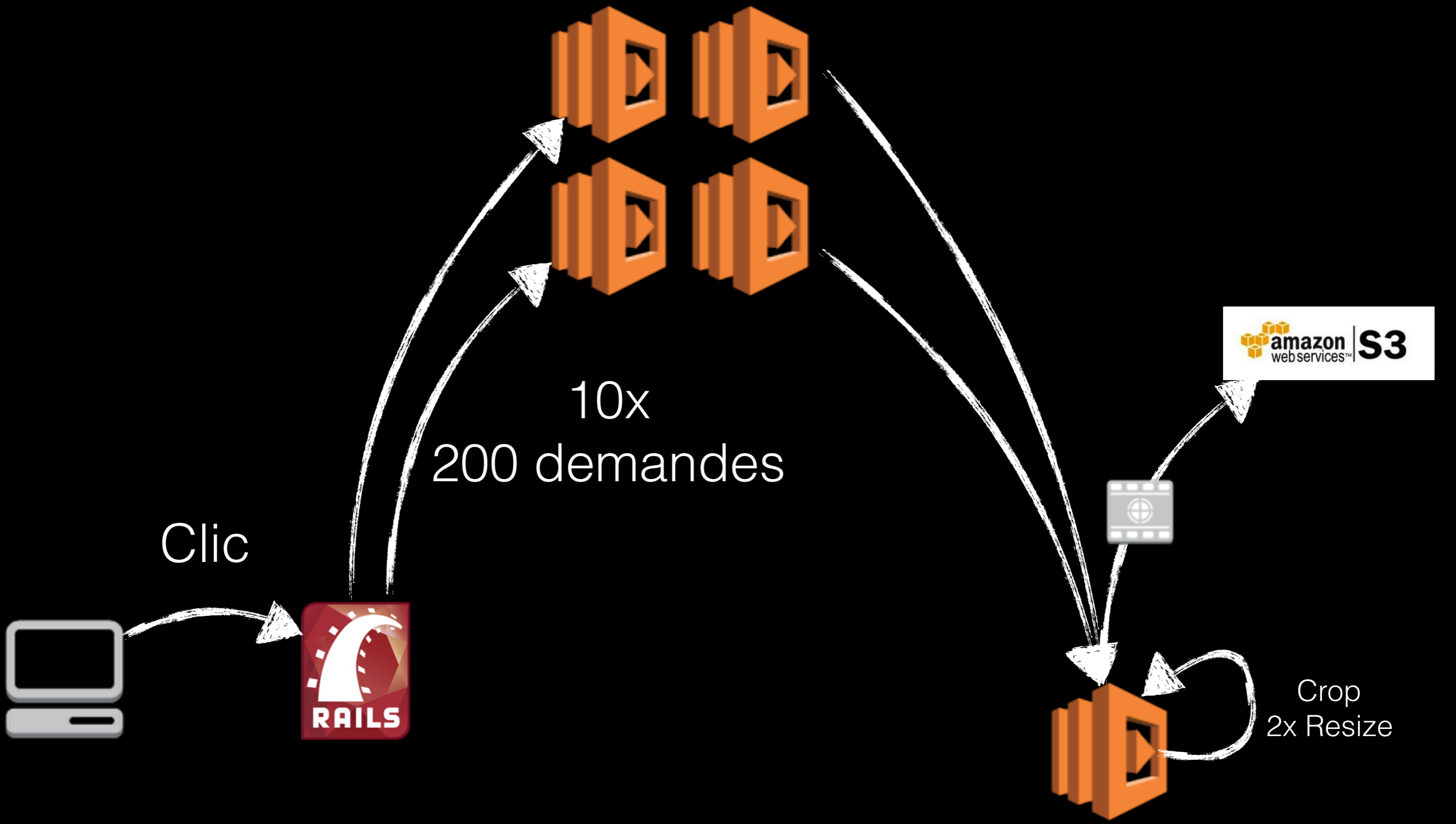
100 λ
 \approx 2 secondes
/ image



« Amazon, I Want Moar λ »



2000λ FTW



~~40~~ 2 secondes mini
sans parler de latence

⇒

2000 λ
≈ 2 secondes
/ image

≈ 3-4 secondes

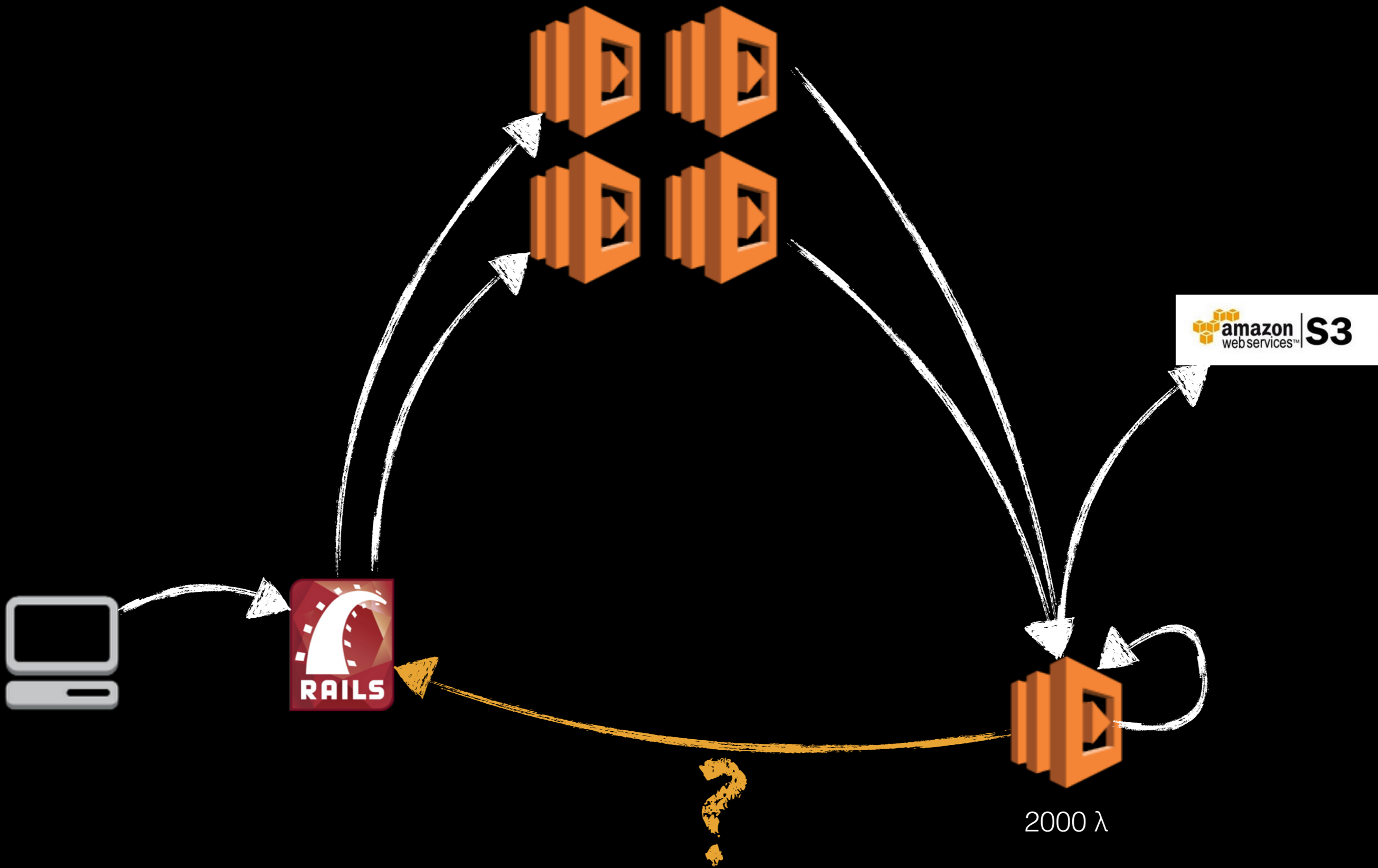
Latence

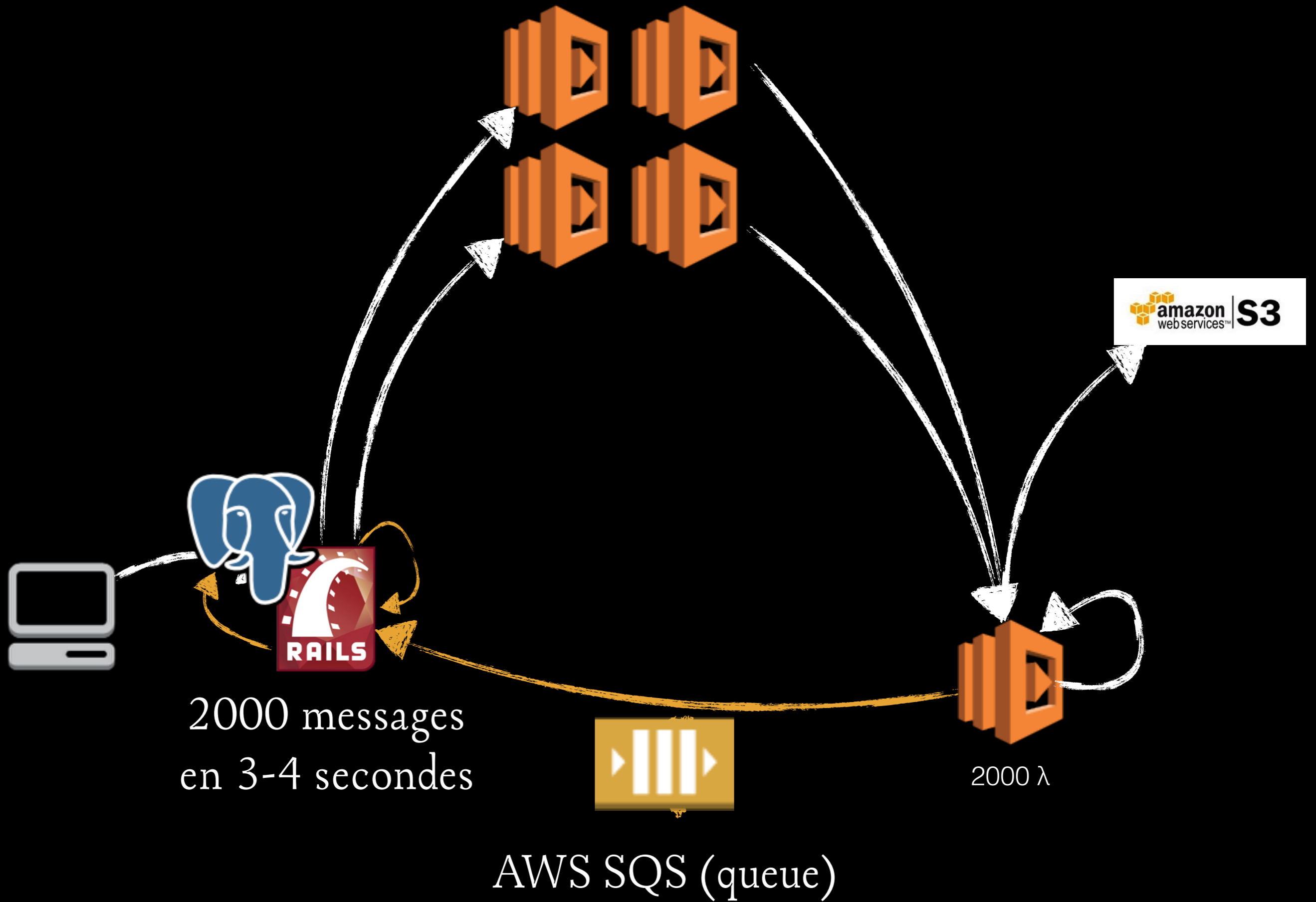
Appel des lambda non //

Lambda utilisées

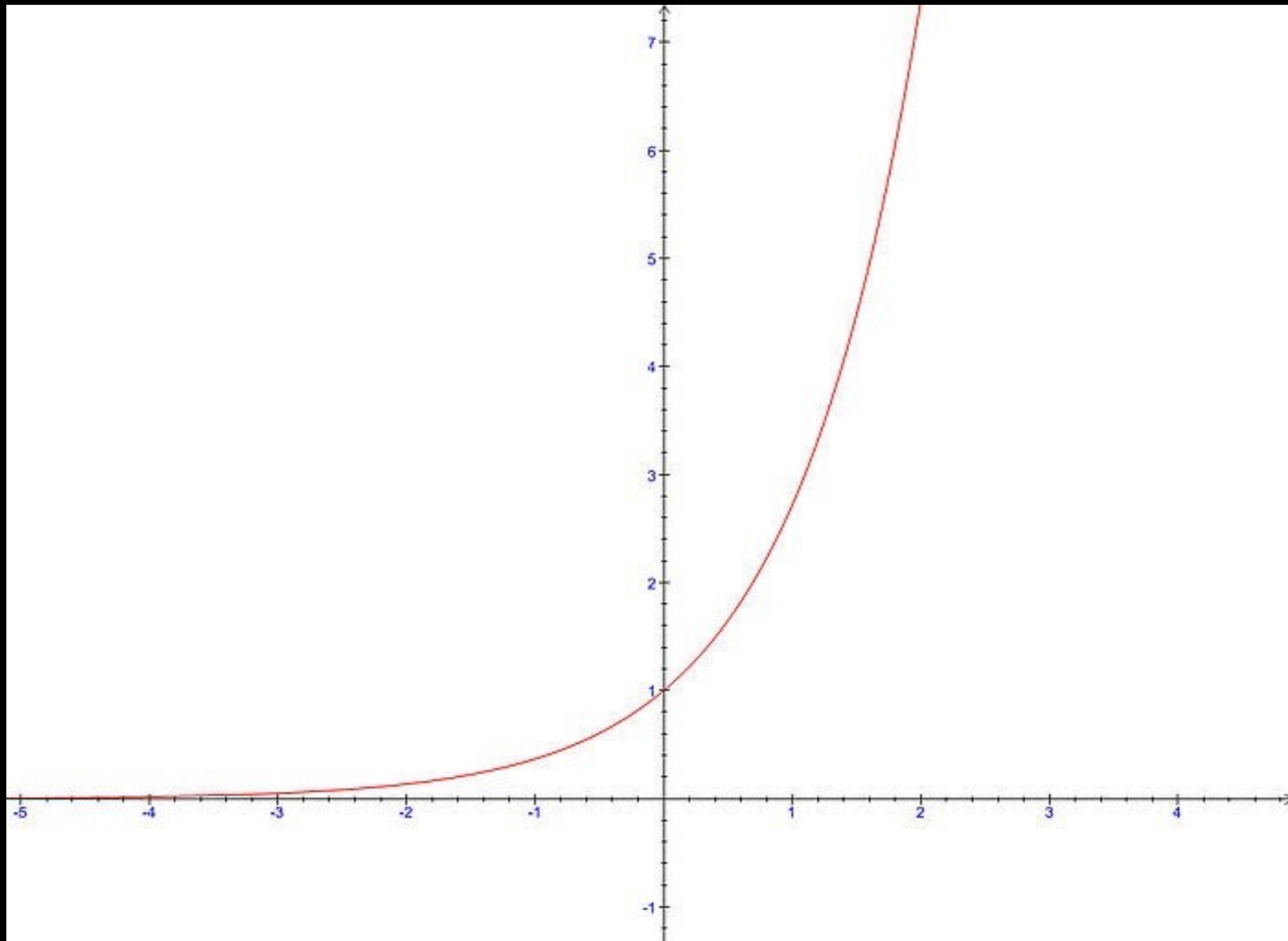
Chapitre 4

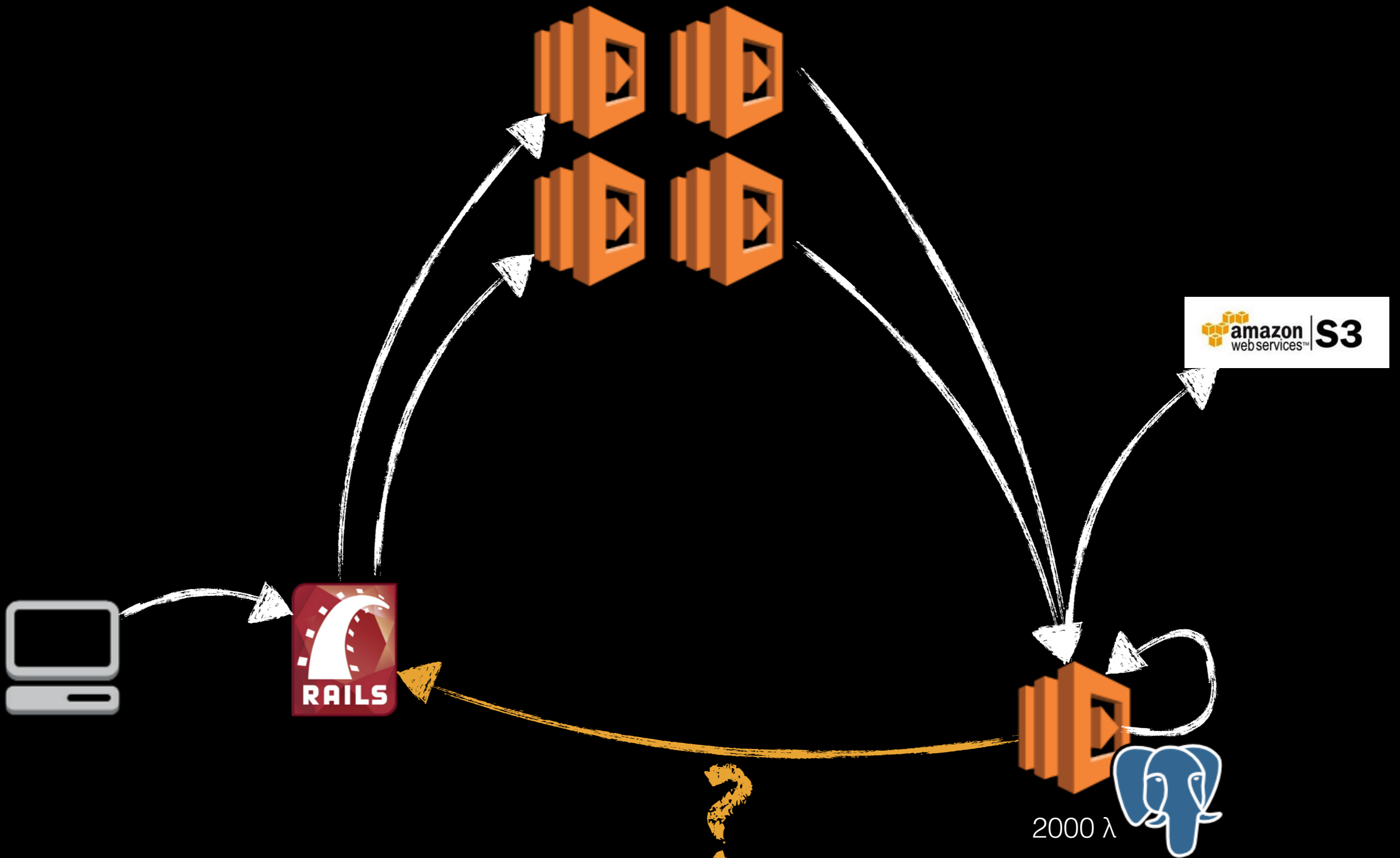
Chérie, ça va couper !



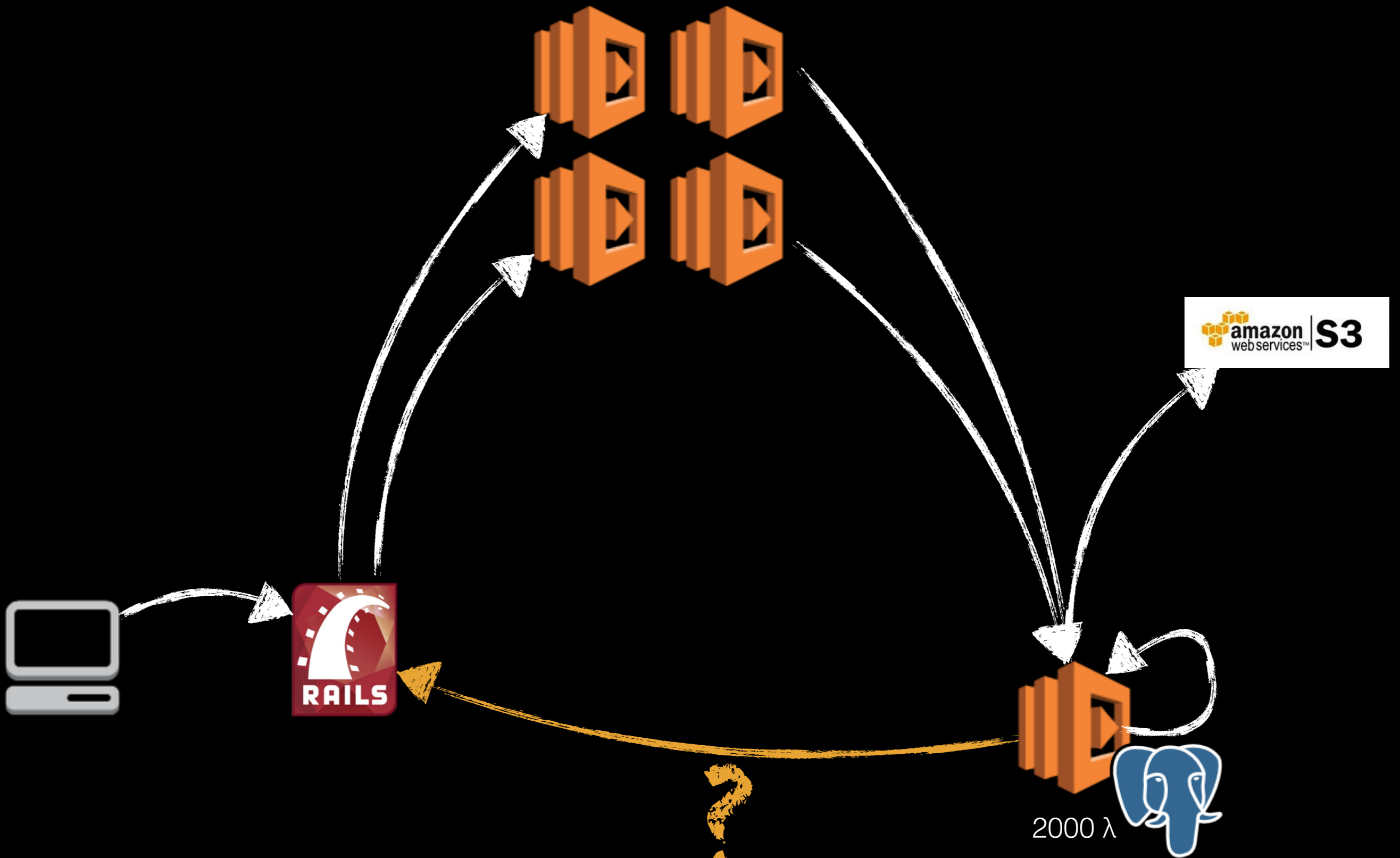


CPU Load

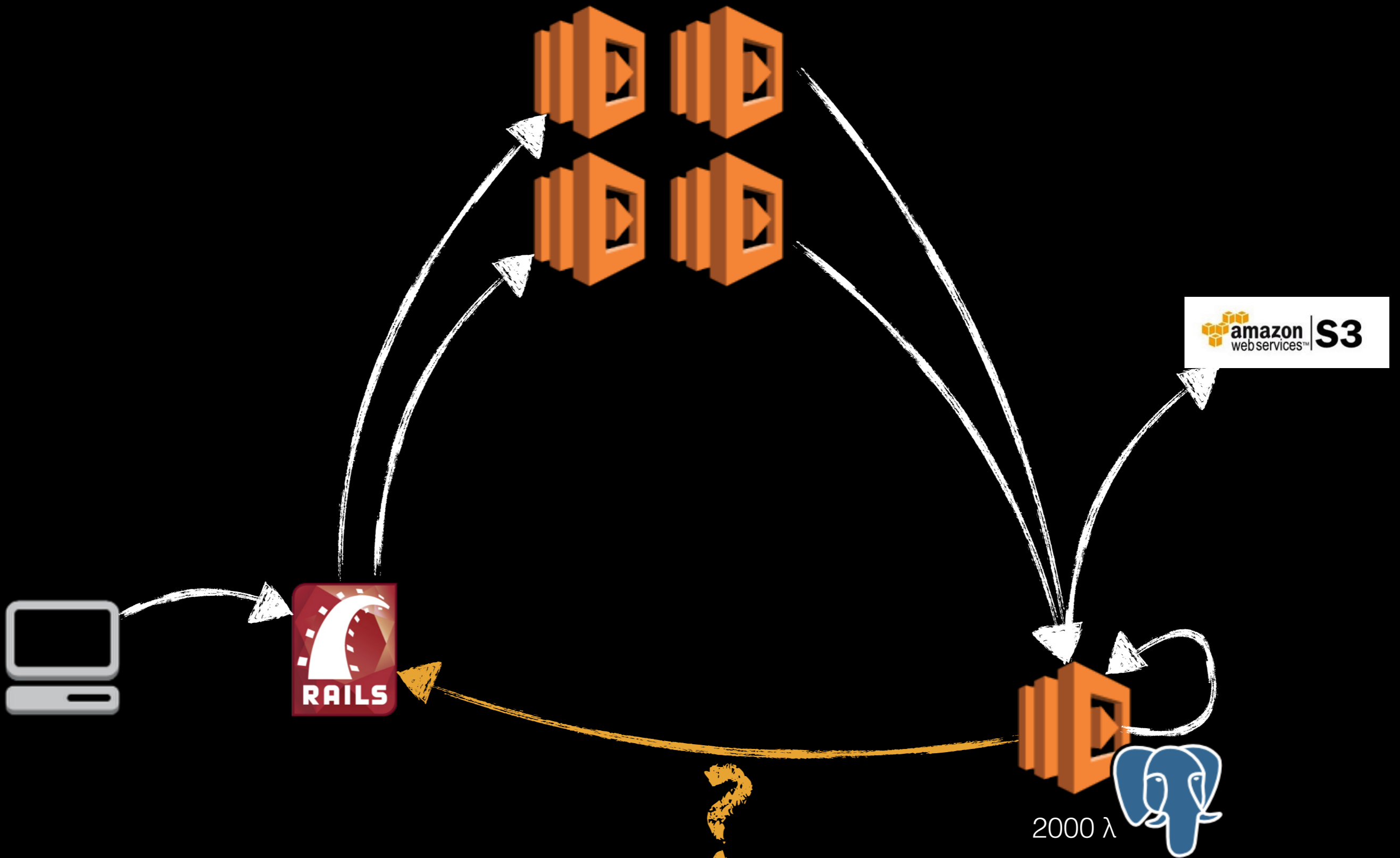




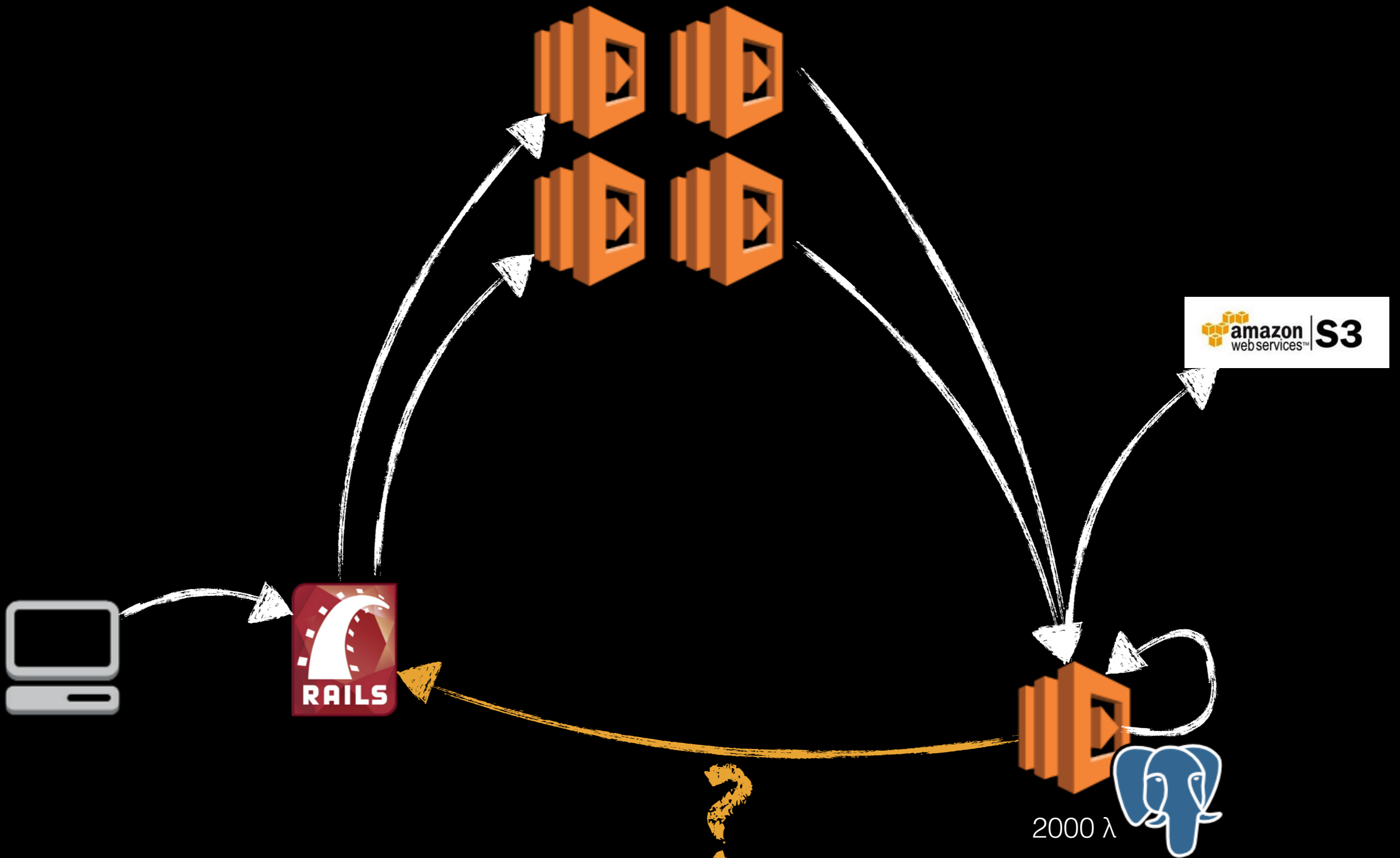
$\lambda + S3 + RDS$ (postgresql) + SQS (queue de notification)



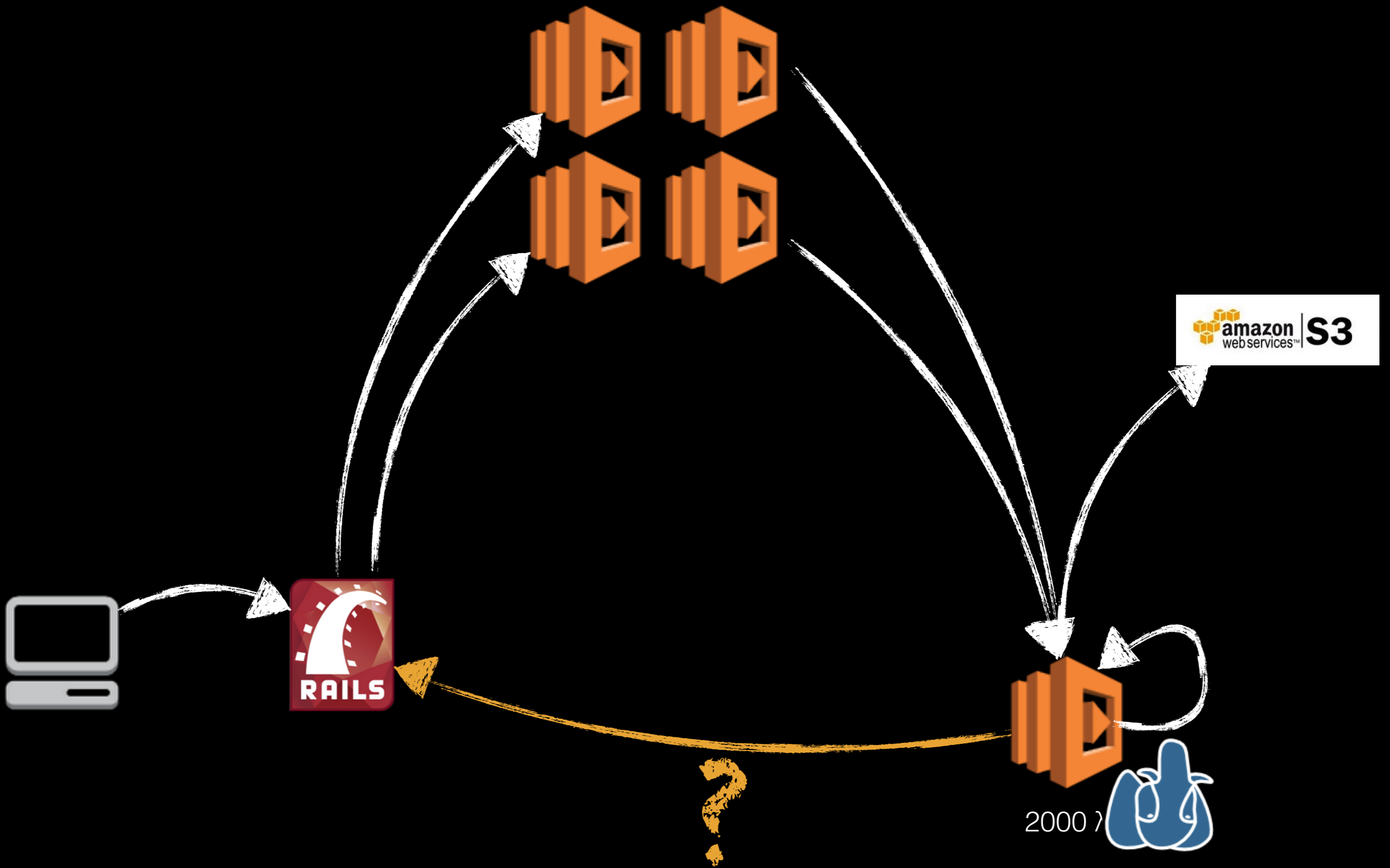
λ + S3 + RDS (postgresql) + SQS (queue de notification)
⇒ conf réseau dans les lambda (VPC, Gateway)



$\lambda + S3 + RDS$ (postgresql) + SQS (queue de notification)
 \Rightarrow perf :-(
?



$\lambda + S3 + RDS$ (postgresql) + SQS (queue de notification)
⇒ découper les lambda... en fonctions !



2000 updates + lock en 3-4 s

*« When deploying containers + microservices,
there is an unstated truth: you are developing
a distributed system »*

— Bryan Cantrill, CTO of Joyent — Hashiconf

« When deploying container **serverless** microservices,
there is an unstated truth: you are developing
a distributed system »

— Bryan Cantrill, CTO of Joyent — Hashiconf



**KEEP
CALM
AND
THINK
SIMPLE**

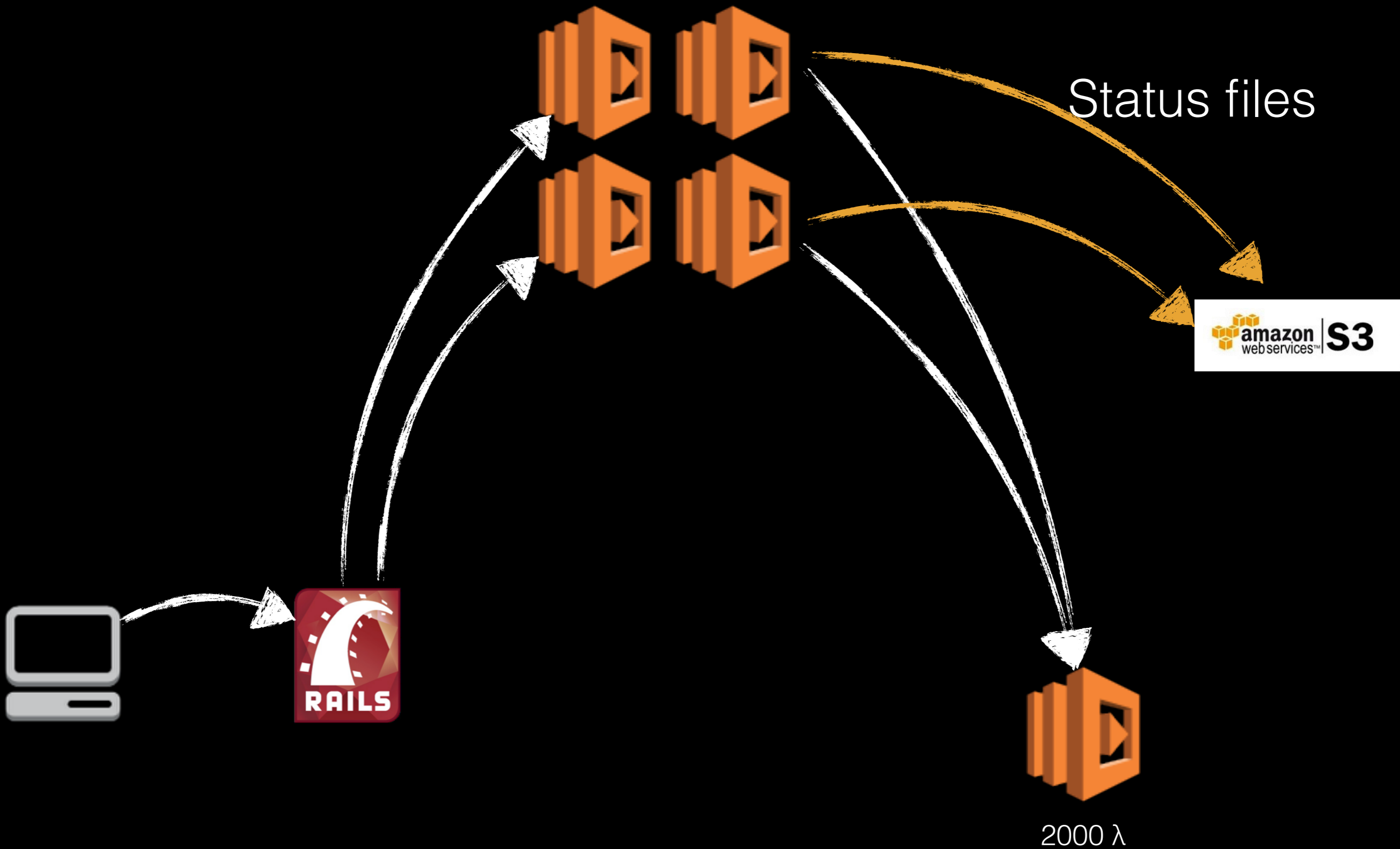
AWS S3

DB clé \Rightarrow valeur



2000 λ



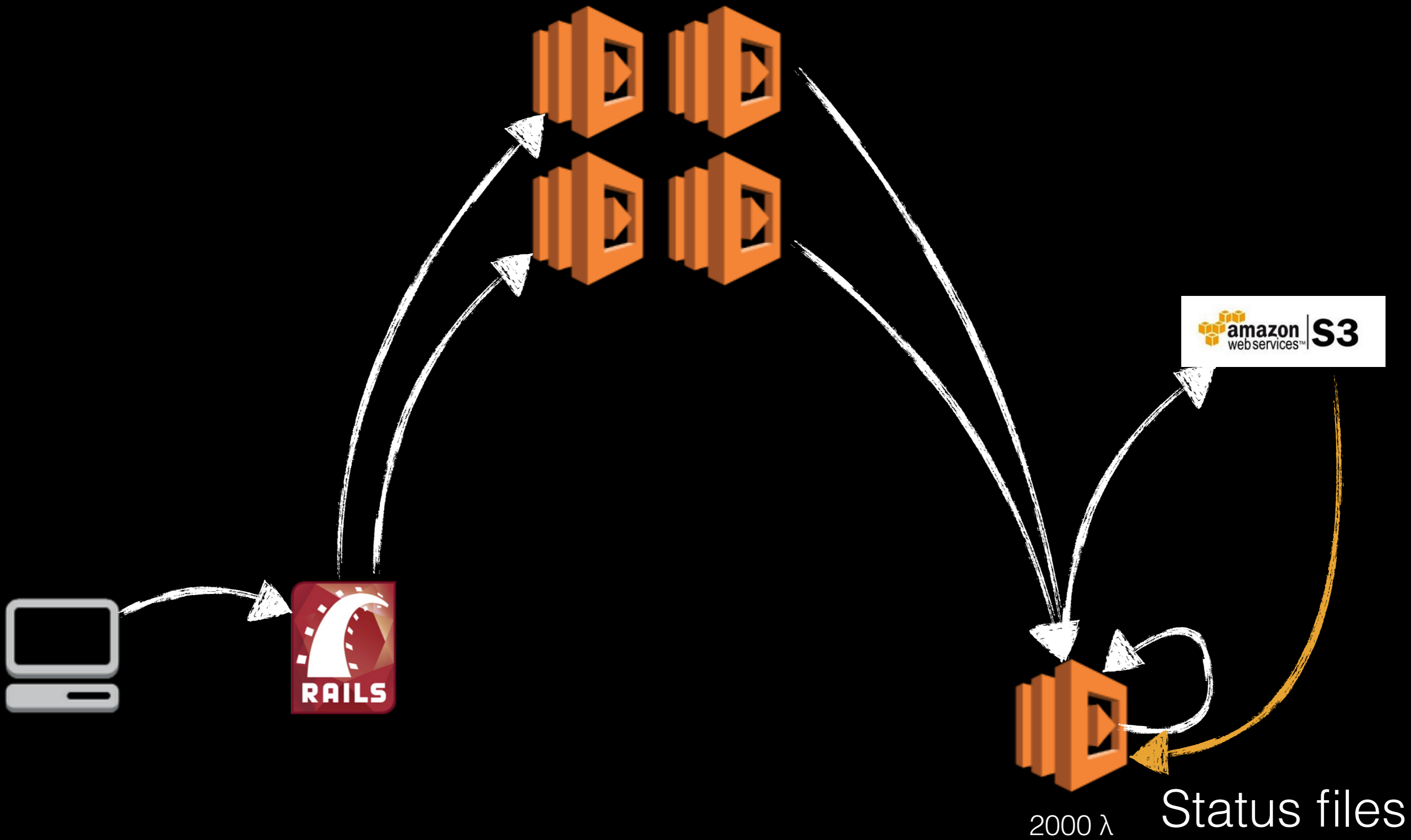


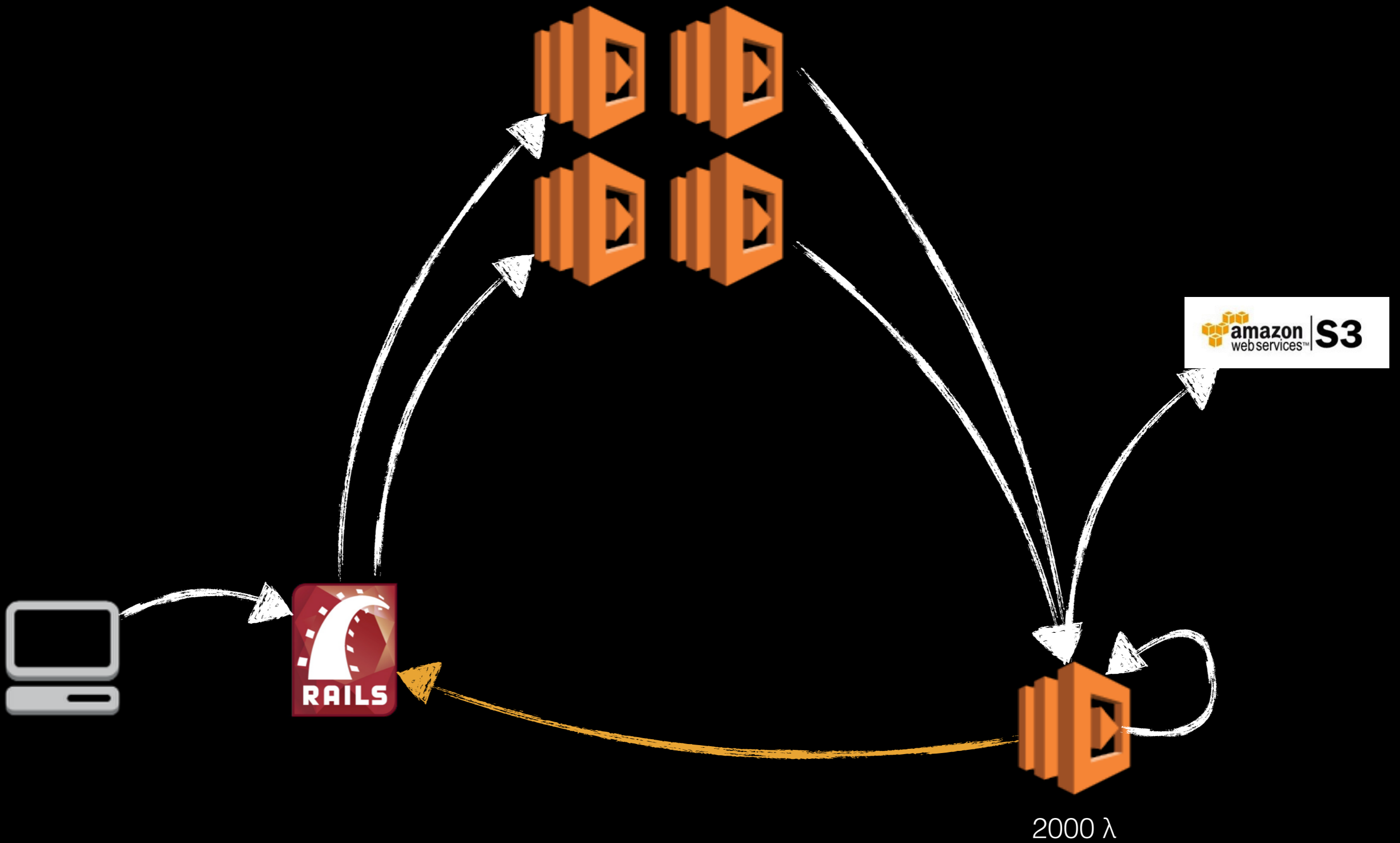
Status files



2000 λ







Quelques updates (< 10) + lock en 3-4 s



YOUHOU !

Conclusion

Alors, ça donne quoi ?

C'est viable !

En production avec les 2000 λ

With great power comes great responsibility.
- Ben Parker



Restez simple

Restez simple

Une λ = une fonction

Une λ peut appeler une λ

Isolation des problématiques réseau, droits, etc.



think

BIG

2000 λ c'est loin d'un serveur sur heroku...

I SEE λ



EVERYWHERE

Merci !

Questions ?