

Denis GERMAIN

@zwindler / @zwindler_rflx / d.germain@lectra.com



**Besoin de métriques Prometheus
à long terme ?**

Thanos fera des Marvels !

LECTRA®



~# whoami

Denis GERMAIN 

Ingénieur Cloud chez **LĒCTRA**.

Auteur principal sur blog.zwindler.fr

#geek #SF #semiMarathon



@zwindler

@zwindler_rflx (blog)

LECTRA®

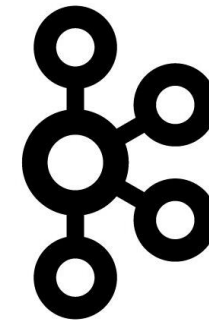
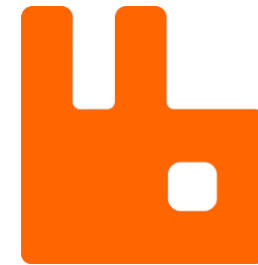
Leader mondial des solutions technologiques intégrées pour les entreprises utilisatrices de cuir ou textile



Et moi, je fais quoi chez LECTRA® ?



There is no cloud
it's just someone else's computer



LECTRA recrute !!

Je cherche un·e futur·e collègue dans l'équipe

- Cloud Engineer **#Ops #Cloud #K8s #InfraAsCode**
- Viens m'en parler après le talk

Et plein d'autres encore !!

- GOTO www.lectra.com/fr/carrieres

Denis GERMAIN

@zwindler / @zwindler_rflx / d.germain@lectra.com

Previously in Prometheus



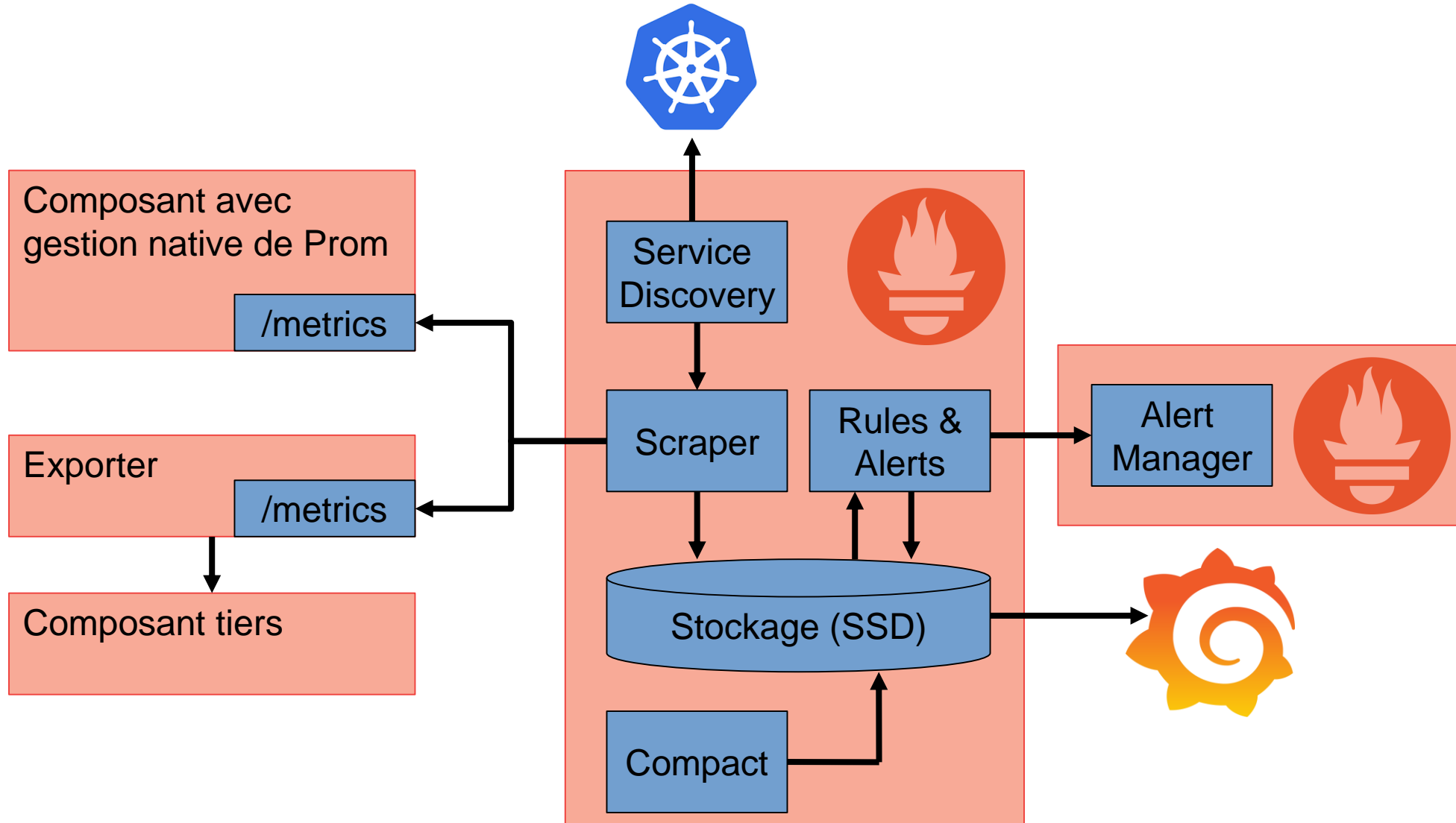
Prometheus

Outil open source

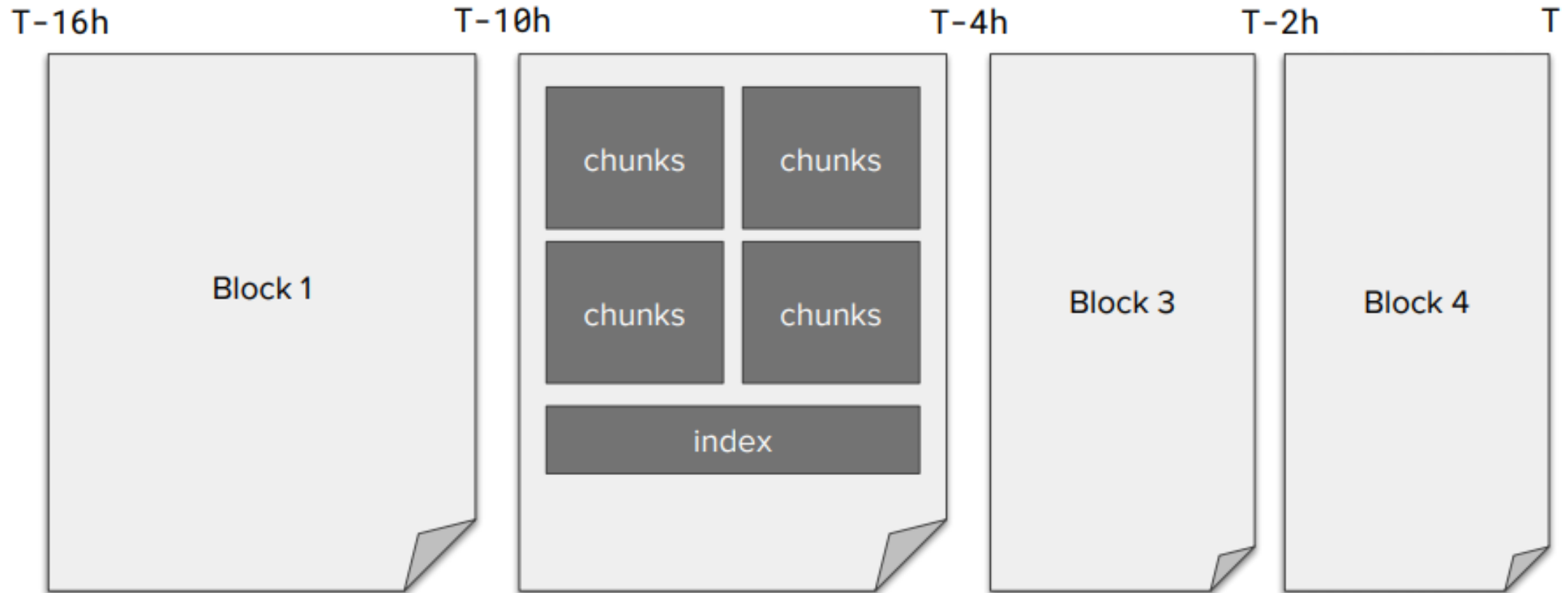
- Développé par SoundCloud en 2012
- Intégré à la CNCF en mai 2016
- Supervision via collecte de métrique
- Langage (PromQL) permettant « requêter » sur ces métriques
- Système d'alertes



L'architecture de Prometheus



Le stockage des métriques dans Prometheus



Performances (1/3)

Logiciel très performant, capable de

- stocker des millions d'échantillons
- requêter sur de nombreux échantillons/timeseries

Prometheus à l'échelle (pour de vrai) chez Digital Ocean

- 2 millions d'échantillons/s
 - 200 millions de timeseries
- ⇒ ~200 serveurs Prometheus



Performances (2/3)

Dépendent grandement de la quantité de métriques stockées

Best practices Prometheus

- Métriques stockées sur des **disques locaux** et de **type SSD**
- Par défaut, 15 jours de rétention max
- Possible d'étendre cette limite
 - mais « Prometheus n'est pas fait pour ça »

Performances (3/3)

Workaround 1a \Rightarrow grossir les disques tant qu'on peut

- couteux
- pas extensible à l'infini
- au delà de 2-3 mois, ça commence à devenir compliqué

Workaround 1b \Rightarrow déporter vers du stockage distant

- pas supporté par Prometheus
- Devs farouchement opposés

<https://prometheus.io/docs/prometheus/latest/storage/>

Failure domains

On n'a pas tous les mêmes besoins que Digital Ocean (ouf!)

Pour autant, vous allez sûrement devoir gérer plusieurs



Dans les bonnes pratiques

- 1 serveur minimum par « Failure Domain »
- 1 Failure Domain = 1 datacenter (voire 1 cluster)

<https://www.robustperception.io/scaling-and-federating-prometheus>

Problème n°2 : Répartition manuelle

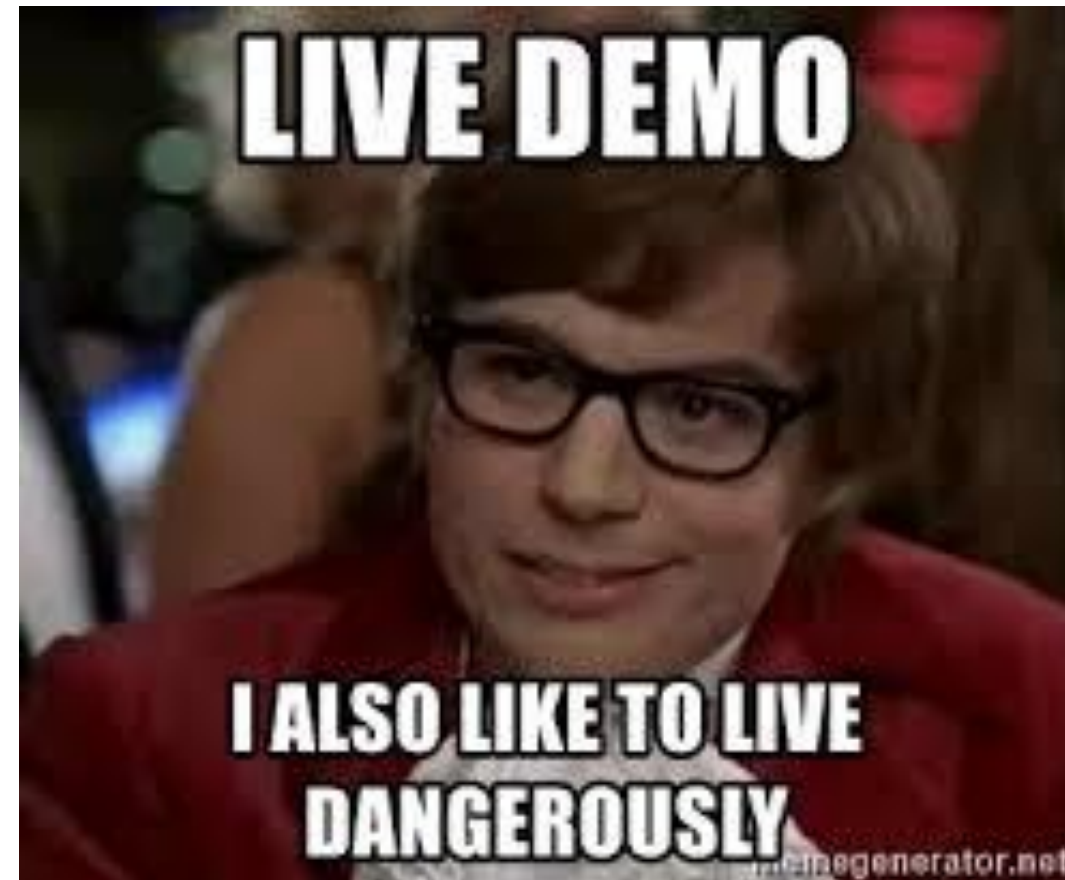
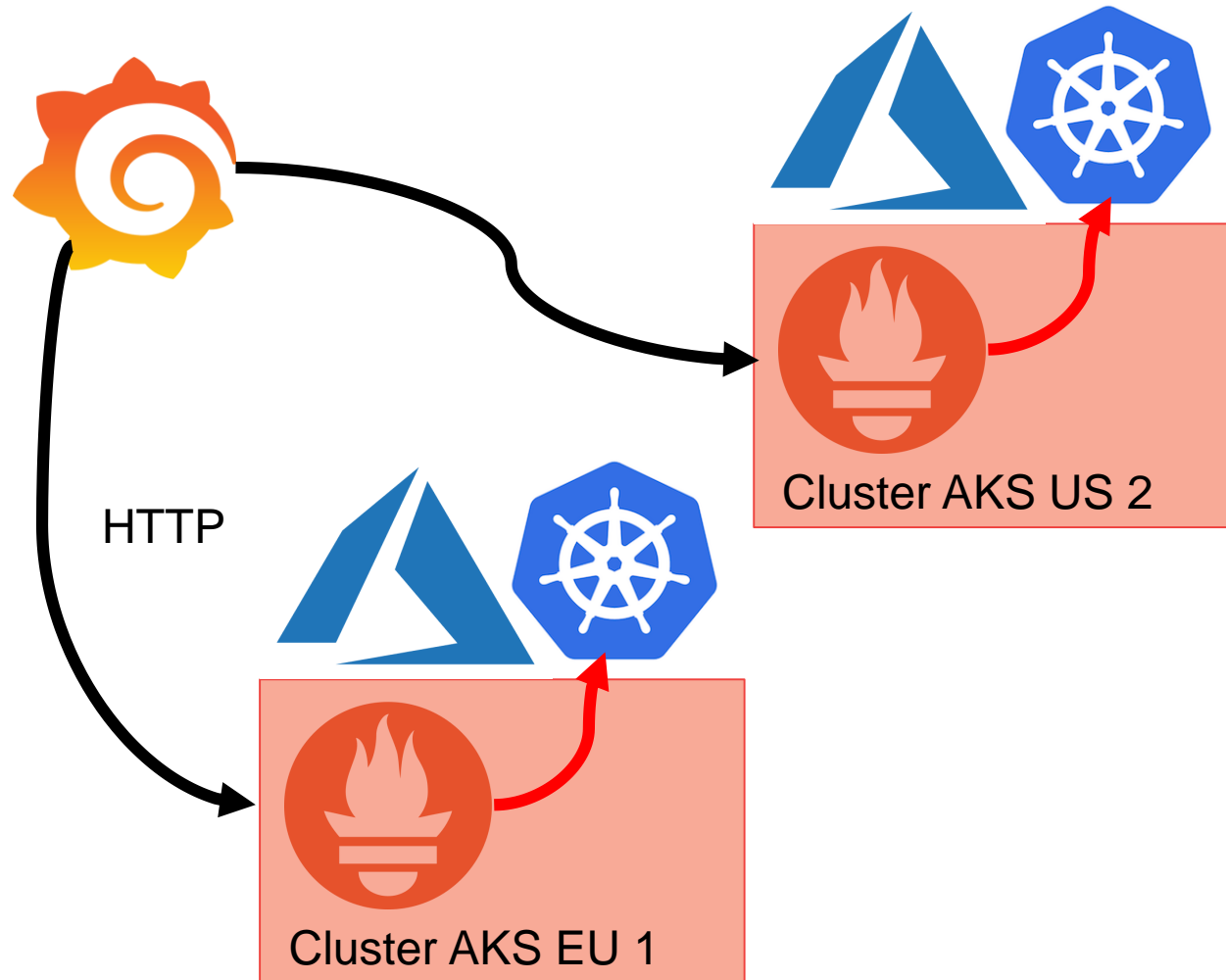
Plusieurs serveurs  ⇒ plusieurs sources de données

Point « Captain Obvious » :

- Il est **déconseillé** de stocker des métriques que vous souhaitez corréler sur des serveurs Prometheus **différents**



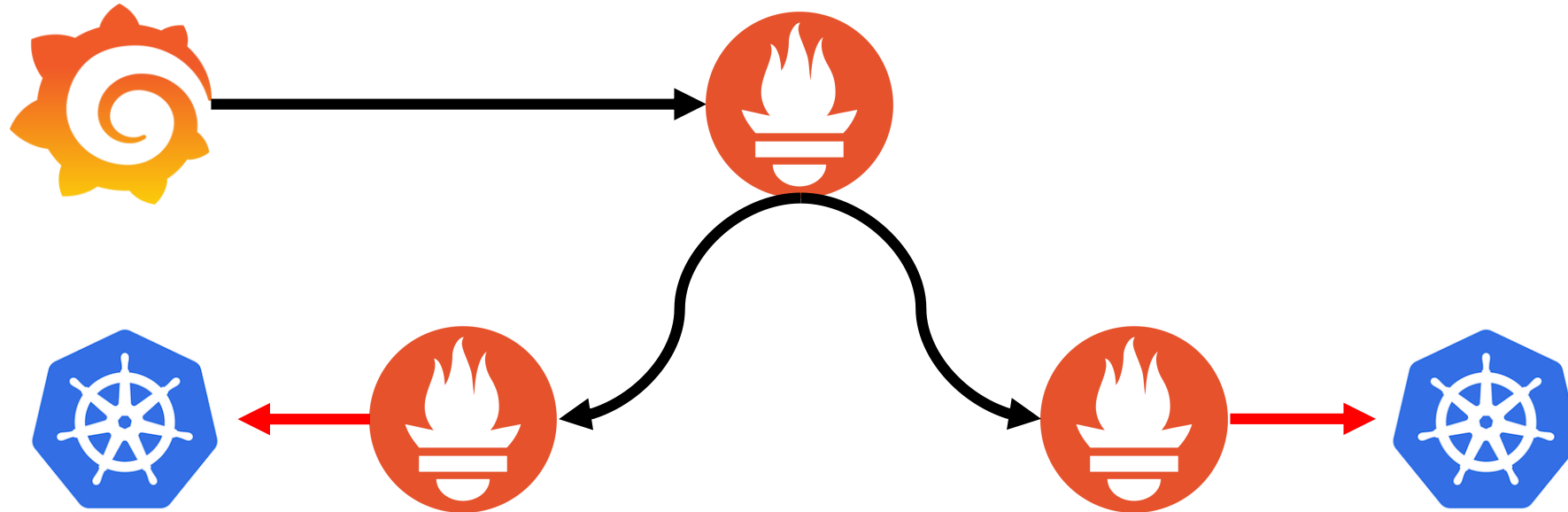
Workaround 2a : Agréger tout ça dans Grafana



Workaround 2b : la Fédération

Réponse des Devs Prometheus : Fédération

- Prometheus « racine » collecte les données des « feuilles »



- Attention à la charge sur le Prometheus « racine » !

Problème n°3 : SPOF !

Un seul serveur Prometheus \Leftrightarrow on a un SPOF

Réponse des Devs Prometheus : « easy, on double tout »

- 2 serveurs identiques par « failure domain »
 - ⇒ toutes les cibles sont scrappées 2 fois (CPU ++)
 - ⇒ 2 sources de données « théoriquement identiques »

Workaround 3a : Loadbalancer

Postulat : les données sont identiques

Si on met un loadbalancer L7 devant les

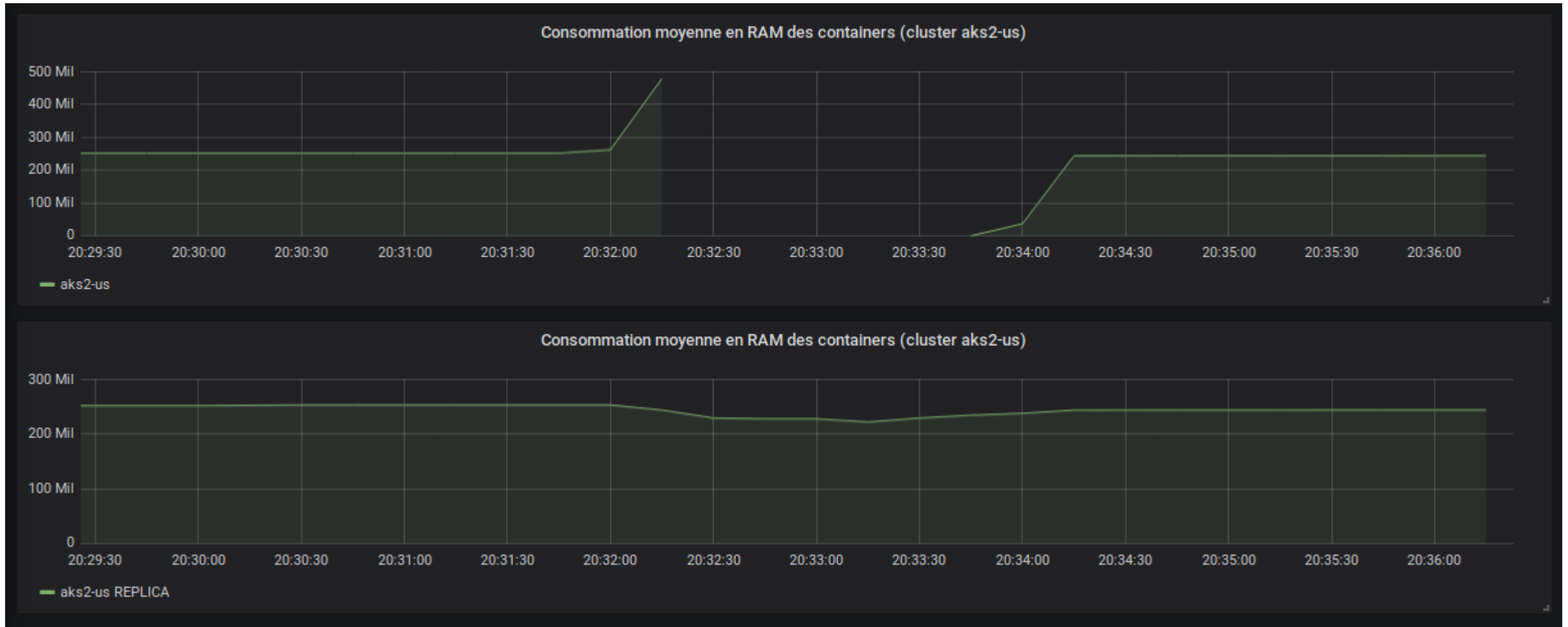


- HA en cas de panne
- Distribue la charge PromQL

FBI : Fausse Bonne Idée !

- IRL \Rightarrow Les données ne sont pas identiques
- (en cas de panne par exemple, on aura un « trou »)

Workaround 3a : Loadbalancer



Workarond 3b : Tout dans Grafana BIS

On double toutes les sources de données dans Grafana !



On a déjà autant de graphes que de sources...

On en est plus à ça près !



Denis GERMAIN

@zwindler / @zwindler_rflx / d.germain@lectra.com

Thanos vous veut du bien



Thanos

Outil open source

- Développé par Improbable depuis nov. 2017
- Intégré à la CNCF en août 2019 *

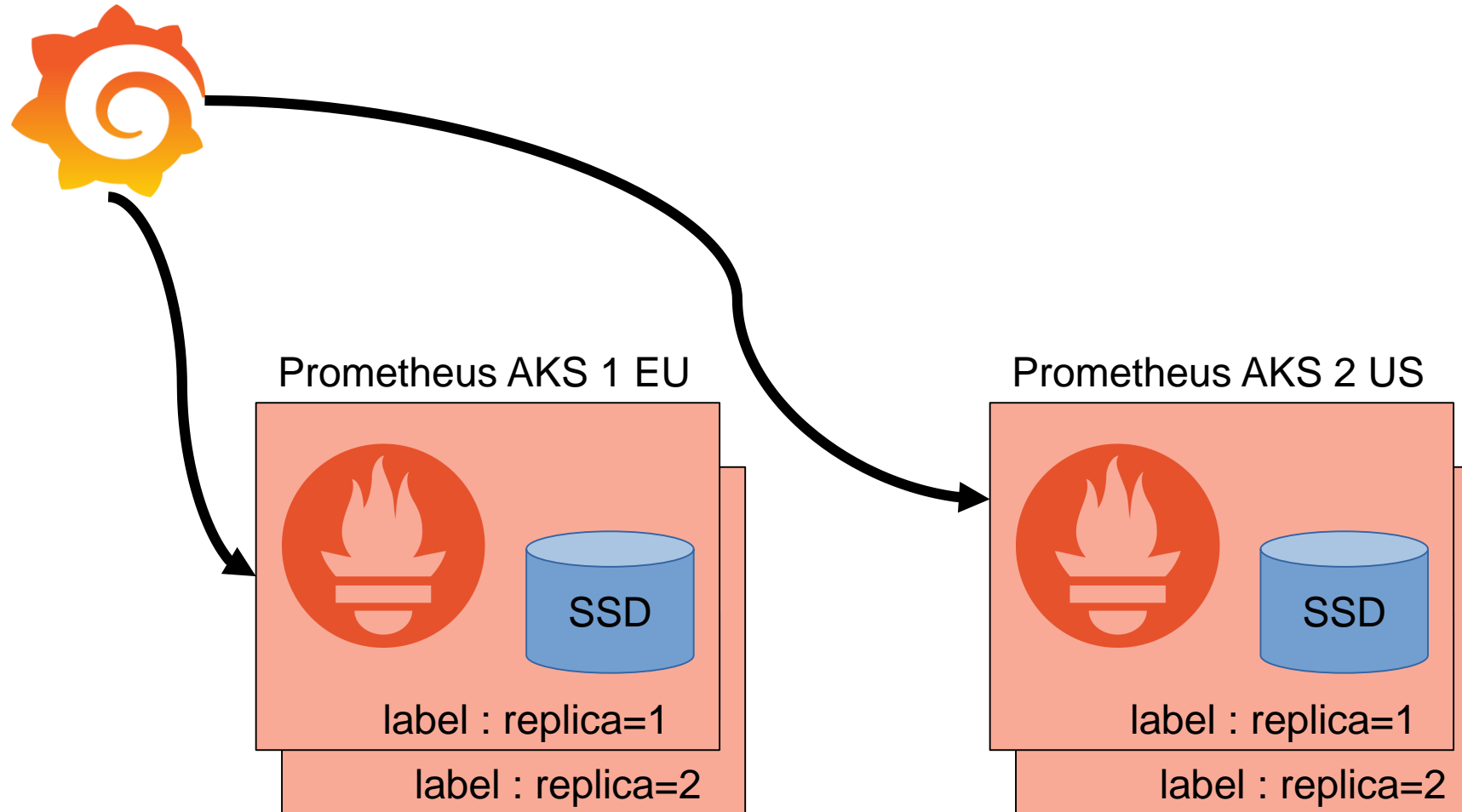
« Prometheus at scale »

- 100 % compatible avec Prometheus + écosystème
- Rétention « infinie » (externalisation S3)
- Corrélation de plusieurs Prometheus + gestion des replicas
- Meilleure compaction
- Downsampling

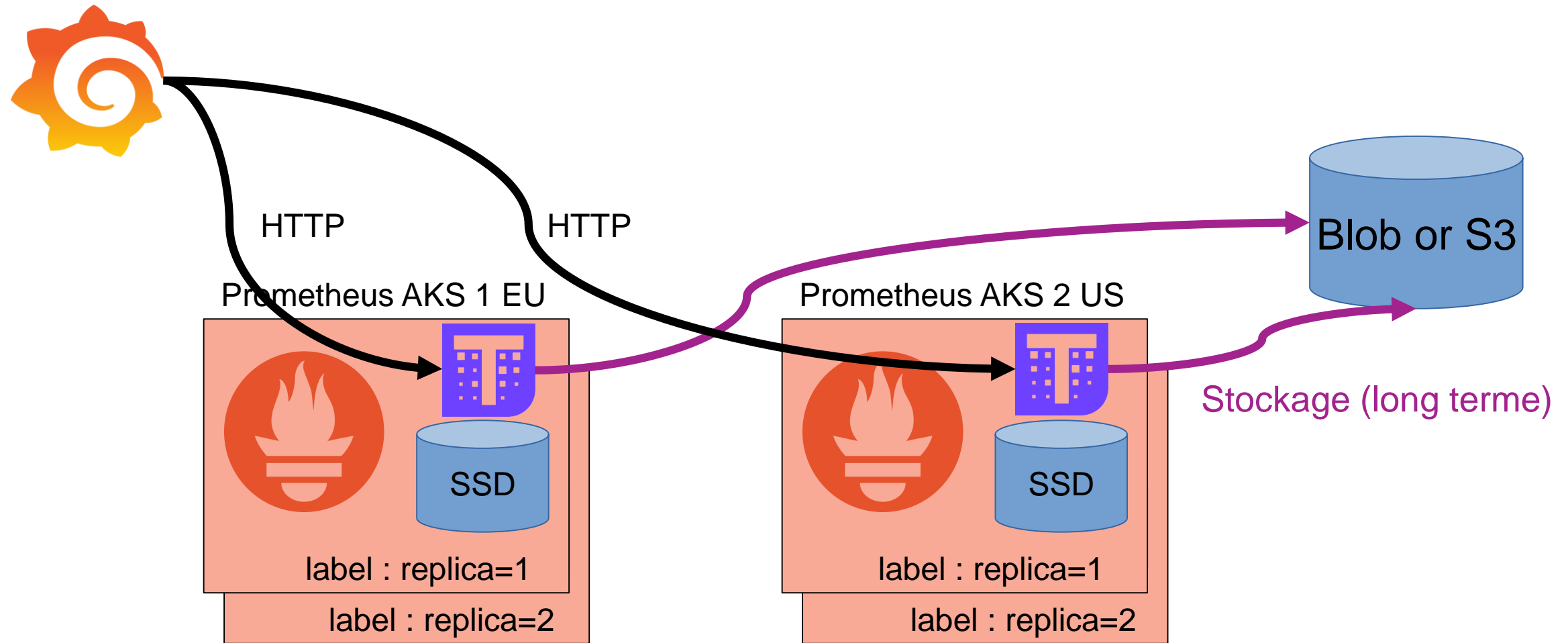


* <https://improbable.io/blog/improbable-donates-thanos-to-cloud-native-computing-foundation>

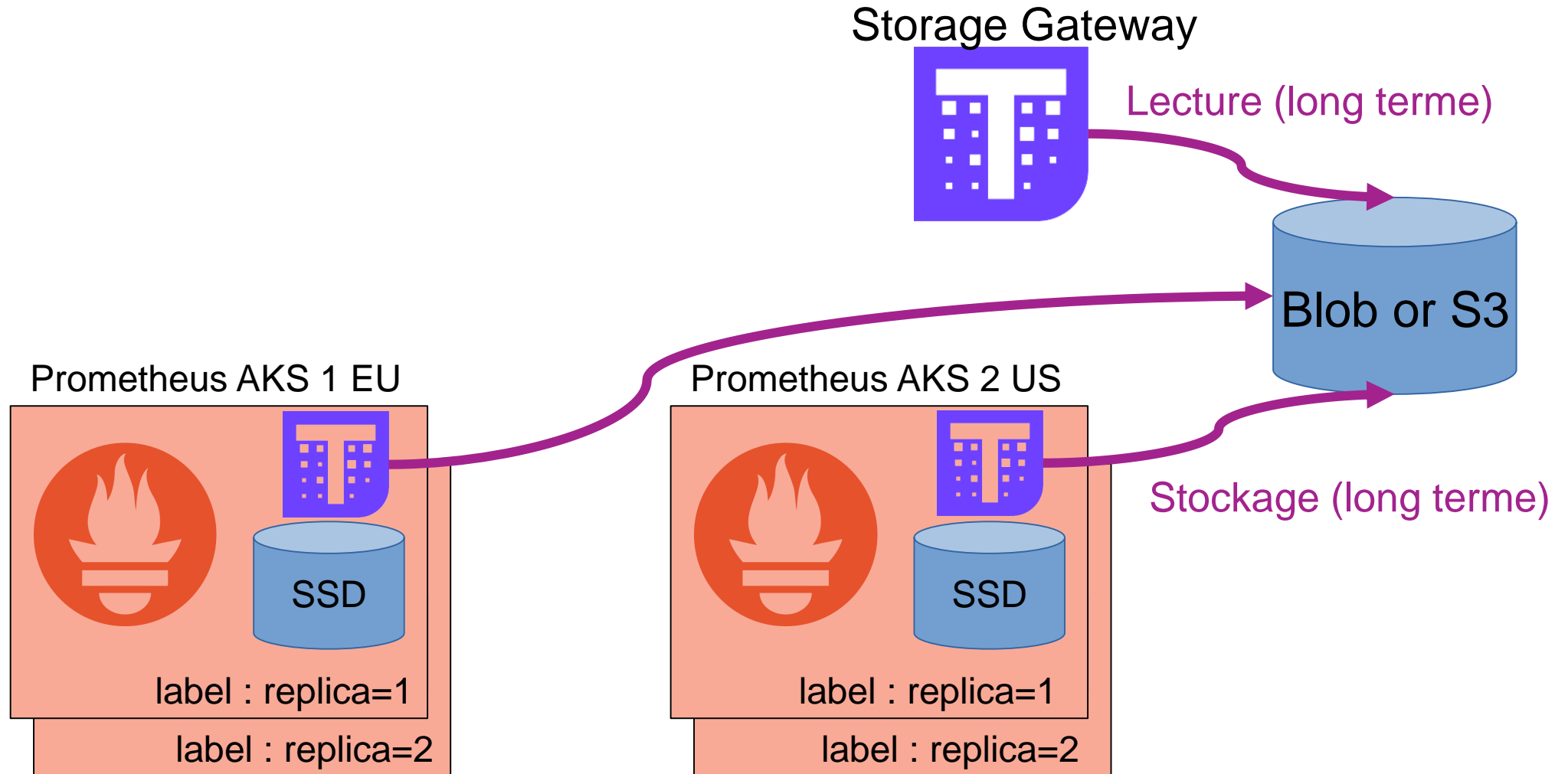
Principe de Thanos



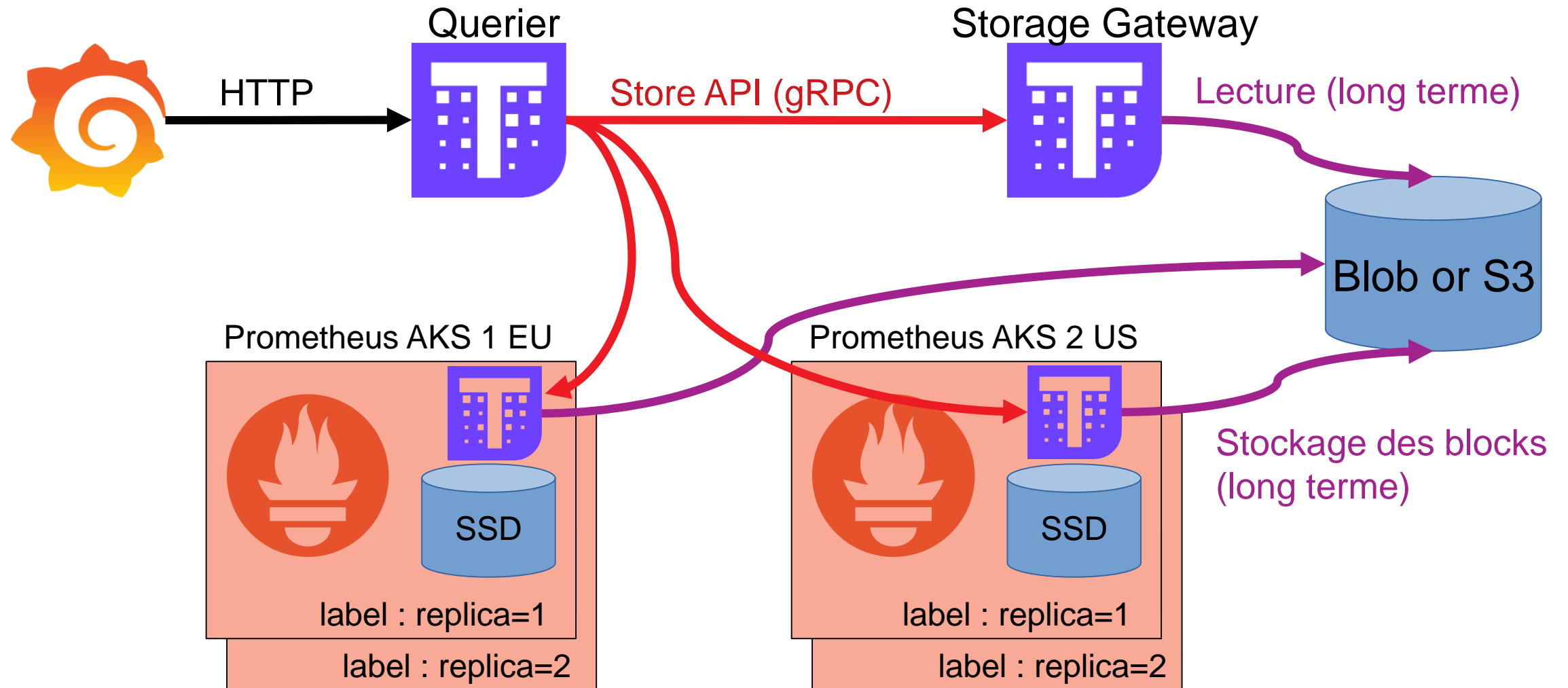
Principe de Thanos



Principe de Thanos



Principe de Thanos



Thanos (en vrai)



Thanos = « Prometheus at scale » ?

100 % compatible avec Prometheus + écosystème ✓

Rétention « infinie » (externalisation S3) ✓

Corrélation de plusieurs Prometheus

- Gestion centralisées de plusieurs DC ✓

- Gestion des replicas ✓

- Meilleure compaction ✓

- Downsampling ✓



Denis GERMAIN

@zwindler / @zwindler_rflx / d.germain@lectra.com

C'est déjà fini ?

On aurait pu en parler

- Global compactor
- Downsampling
- Thanos Ruler (alerting centralisé multi sources)
- Mettez à jour Prometheus en 2.13.0 (et Thanos en 0.8.1)
 - apporte de grandes améliorations CPU/RAM/latence pour Prometheus **et** Thanos

<https://prometheus.io/blog/2019/10/10/remote-read-meets-streaming/>

Denis GERMAIN

@zwindler / @zwindler_rflx / d.germain@lectra.com



Denis GERMAIN

@zwindler / @zwindler_rflx / d.germain@lectra.com

Des questions ?

