

Résilience de la messagerie

Maintien en production, utilisation en 24/7

Thomas Cataldo & Laurent Coustet

Le dernier kilomètre

90/10

- Quand la messagerie s'arrête...
- Un atout : mise à jour in-place? (sans doubler l'infra)
- Une contrainte : mise à jour in-place (interruption de service)
- Le sujet : éviter les interruptions programmées
- Nombreuses optimisations pour améliorer les choses
 - Session persistentes
 - HotUpgrade (Montée en version Elasticsearch, Conversations)

Existant

Pour une utilisation 24/7

- Haute disponibilité
 - Via hyperviseur
 - peu de mise en place
 - coût (VMware) et contraintes infra
- Toujours indisponible lors des mises à jour
- Montée de version n'est pas toujours juste "log4shell", eg. passage de postgres 11 à postgres 14.

Objectifs

Dans un monde idéal

- Mise à jour doit être invisible pour l'utilisateur
- Indépendamment des changements à effectuer
- PRA efficace
 - PRA : Plan de reprise d'activité
 - RPO - Recovery Point Objective : Perte de Données Maximale Admissible
 - RTO - Recovery Time Objective : Durée Maximale d'Interruption Admissible
- On veut restaurer un backup **immédiatement** et **sans perte de données**

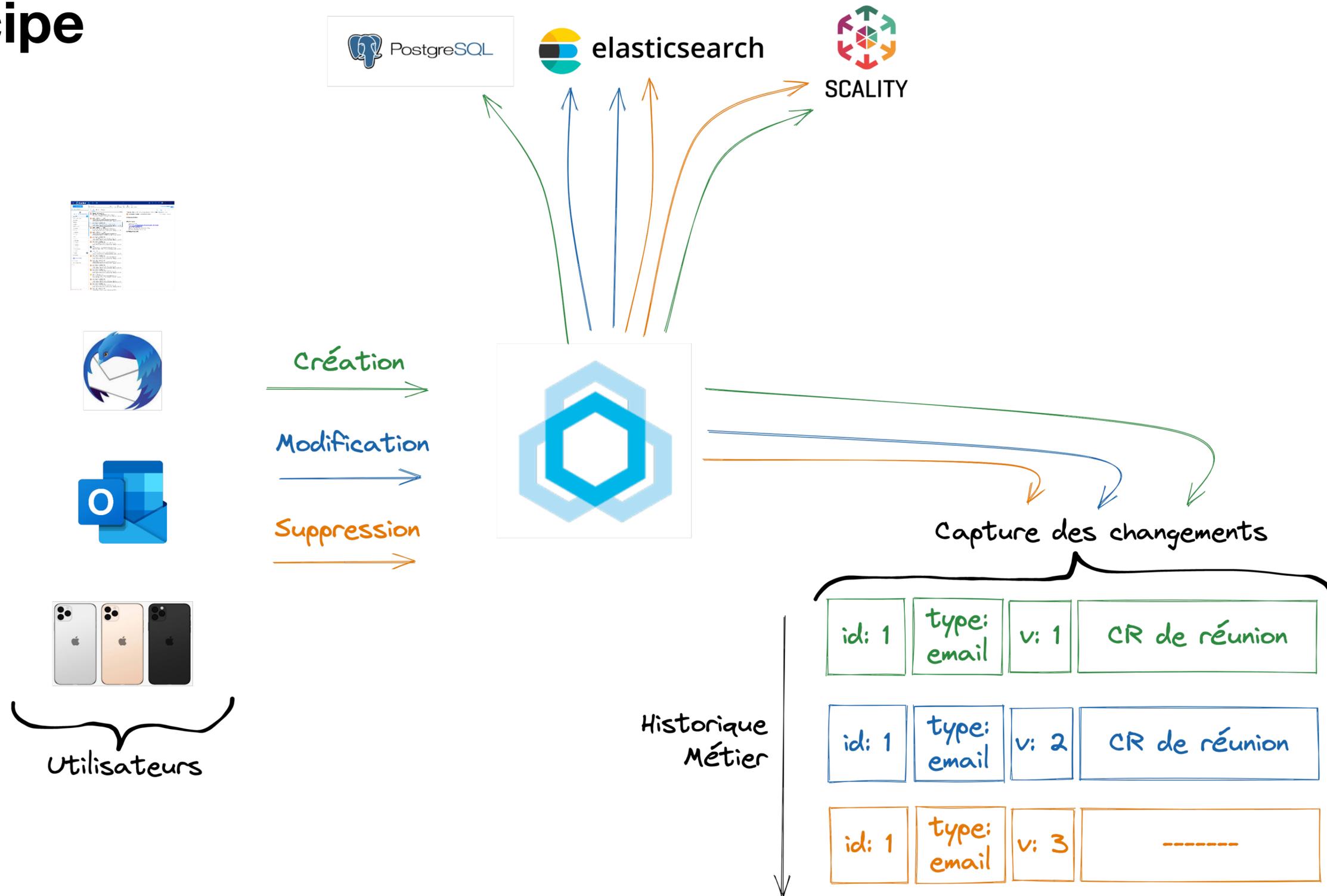
Objectifs

Comment on y arrive

- On renonce[?] au in-place (pour la maj à chaud)
- **Bascule transparente** (Proxy spécifique)
- Le **Stockage Objet**[?] pour les données non modifiables
 - Implémenté pour les mails et documents
 - Partagé entre instances
- Le reste (calendriers, contacts, annuaires, lu/non-lus, etc) : **Capture Des Changements**[?]
 - Savoir rejouer les changements
 - Sur version identique pour répondre à un incident
 - Sur version supérieure pour préparer une mise à jour

Capture des changements

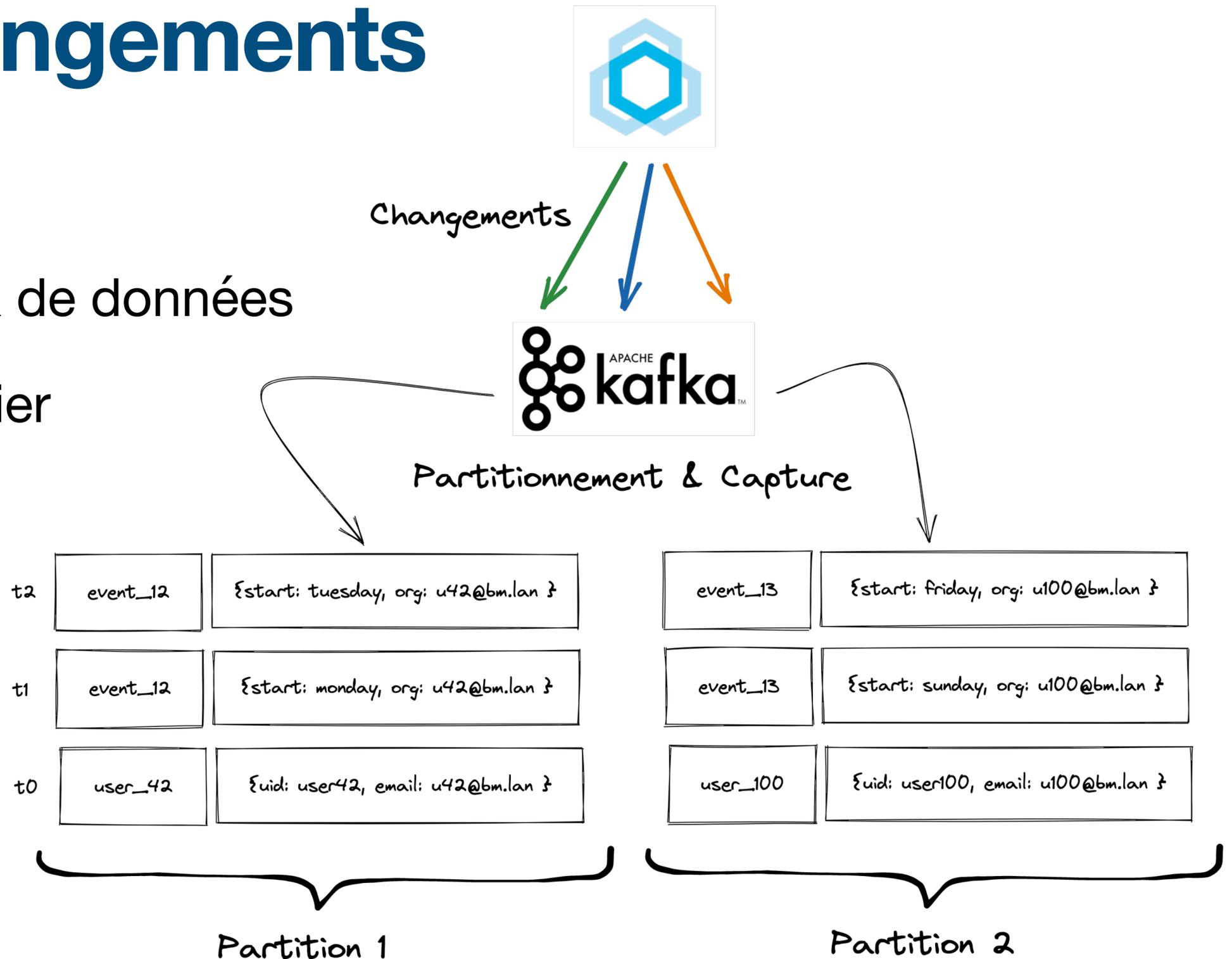
Le principe



Capture des changements

Apache Kafka 1/2

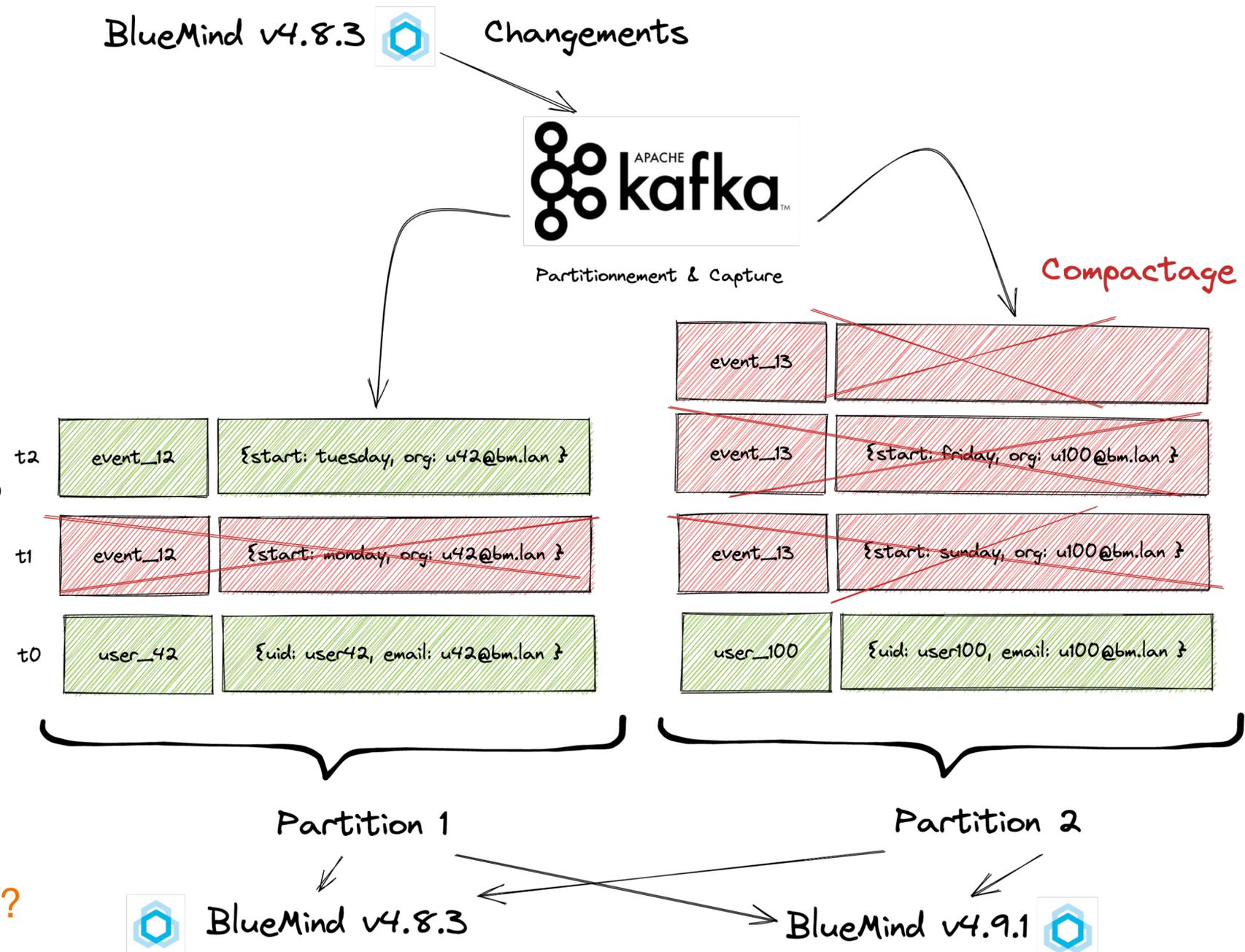
- Solution de gestion des flux de données
- Sauvegarde des objets métier
- Capacité de les rejouer



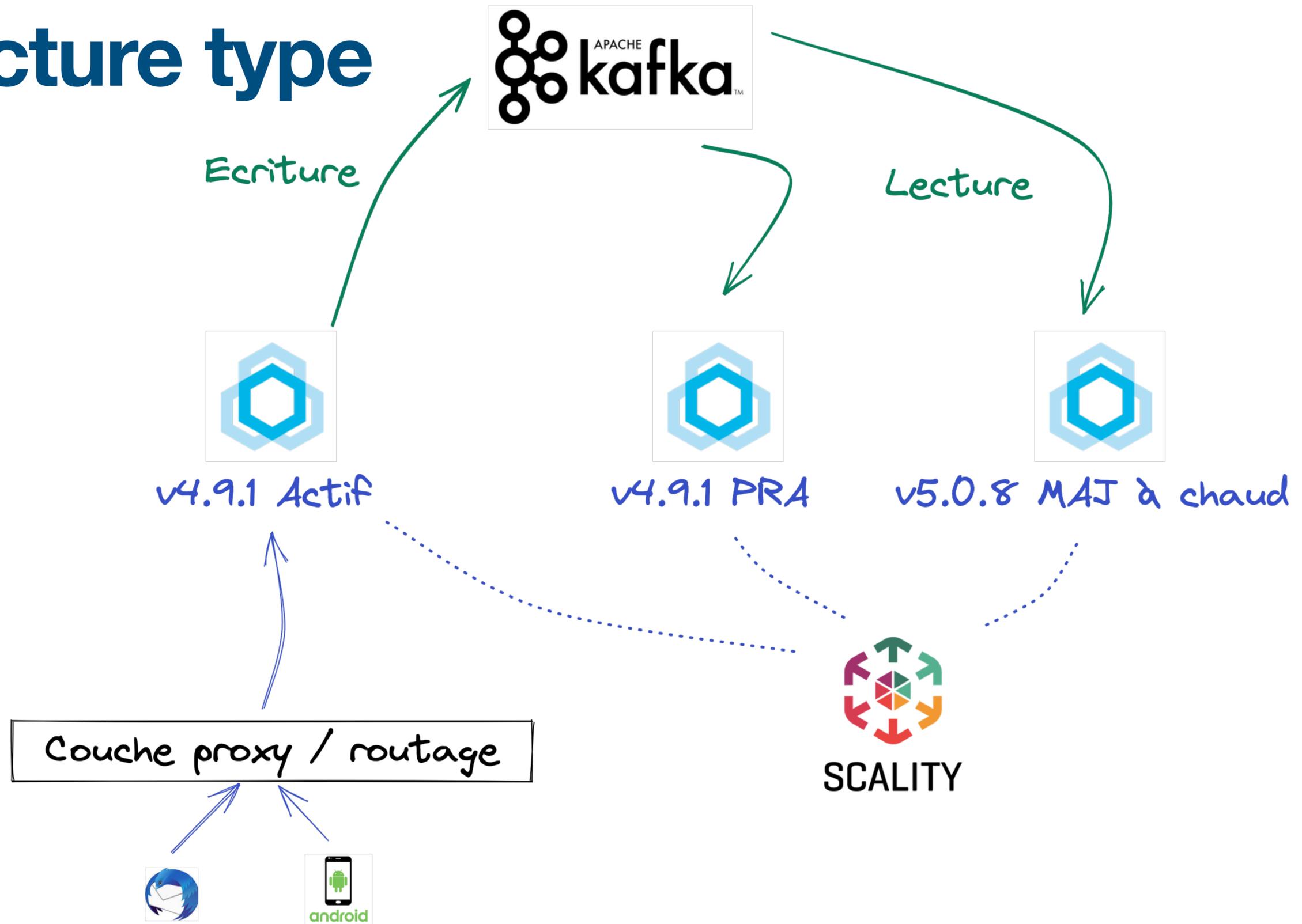
Capture des changements

Apache Kafka 2/2

- Compacté (et compressé)?
- Recréer à l'identique
 - identique pour l'utilisateur
 - identique pour les logiciels clients
- Temps réel
- Nouvelle procédure d'installation ou de mise à jour
- Données traitées par version X ou Y?



Architecture type



En résumé

Fin des interruptions programmées

- Pré-requis
 - infra Kafka et Stockage Objet
- Evolutions BlueMind
 - Nouvelle façon de mettre à jour
 - Enregistrement constant des changements
- Valeur ajoutée
 - Mise à jour, PRA et PCA sans interruption
 - Possibilité de cloner (ex: changement d'OS, préprod)