

Evolution du stockage

Cyrus & bluemind v5

Thomas Cataldo & Laurent Coustet

Evolutions d'architecture dans BM

Un flux constant de changements (transparentes)

Version	PostgreSQL	ElasticSearch	Java	Cyrus	Autre
BM 1.0	8.4 - 9.1	Nope	6	2.4.1	
BM 3.0	8.4 - 9.1	2.3	7	2.4.8	+IPS
BM 3.5	9.6.22	6.4	8	2.4.13	+TICK
BM 4	11.2	7.x	8	3.0.8	-IPS +Réplication
BM 5	15.x	8.x	17 !		

Cyrus

Rôle dans BlueMind

- Serveur IMAP & POP3 (protocoles standards d'accès aux courriers électroniques)
- Serveur de stockage des emails
- Multiples autres copies des métadonnées pour répondre aux demandes
- Référence pour les mails dans bluemind
 - Quand on synchronize, cyrus a raison.

Accès aux mails

IMAP en minorité



EAS
(Smartphones)

MAPI
(Outlook)

Mail App
(nouveau webmail)

Roundcube
(Ancien webmail)

Thunderbird

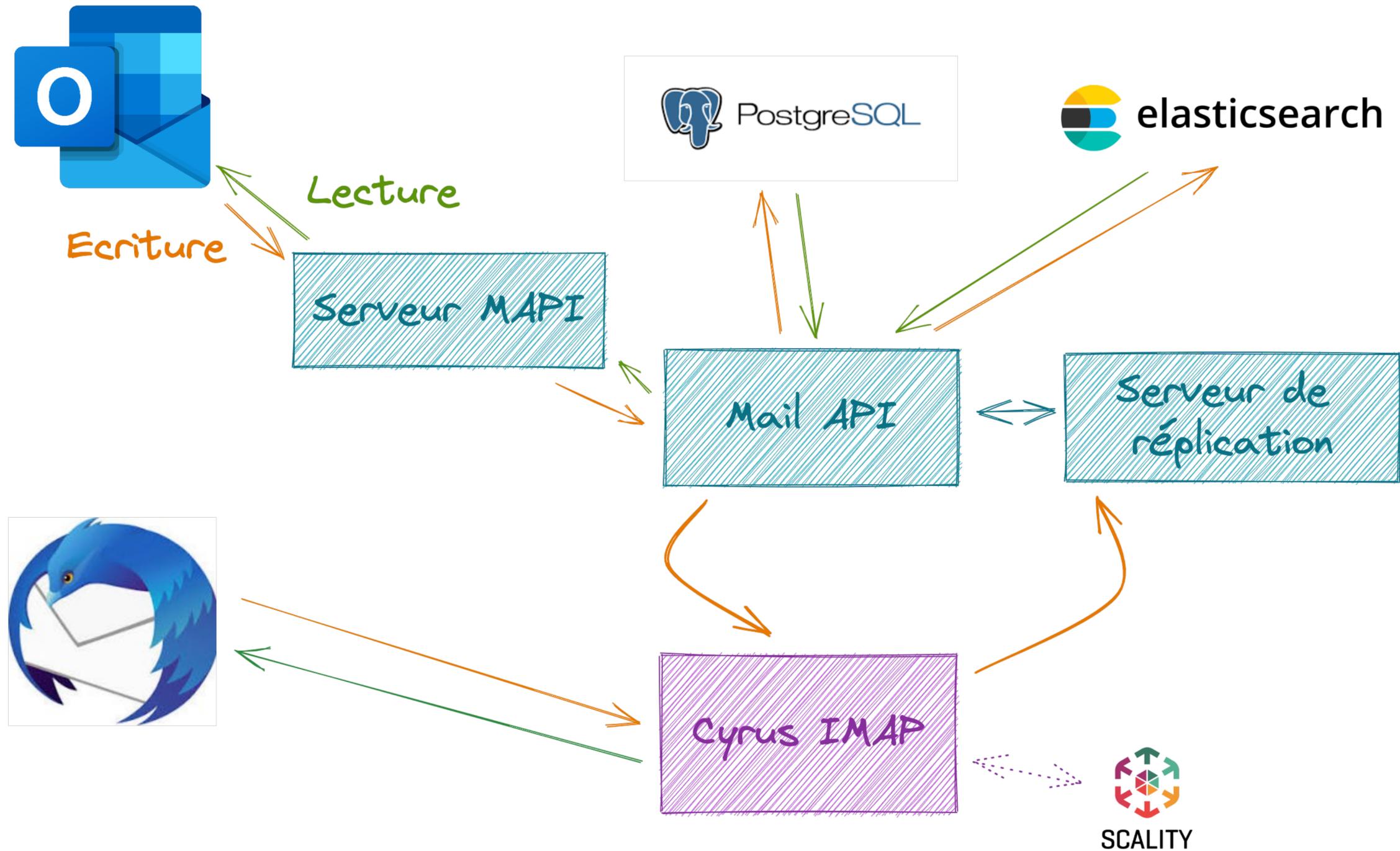
MAIL Api
(HTTP - REST)

Cyrus
(IMAP)

Conception 1994 - 2001
Stockage objet non officiel
Modèle 1 connexion = 1 process

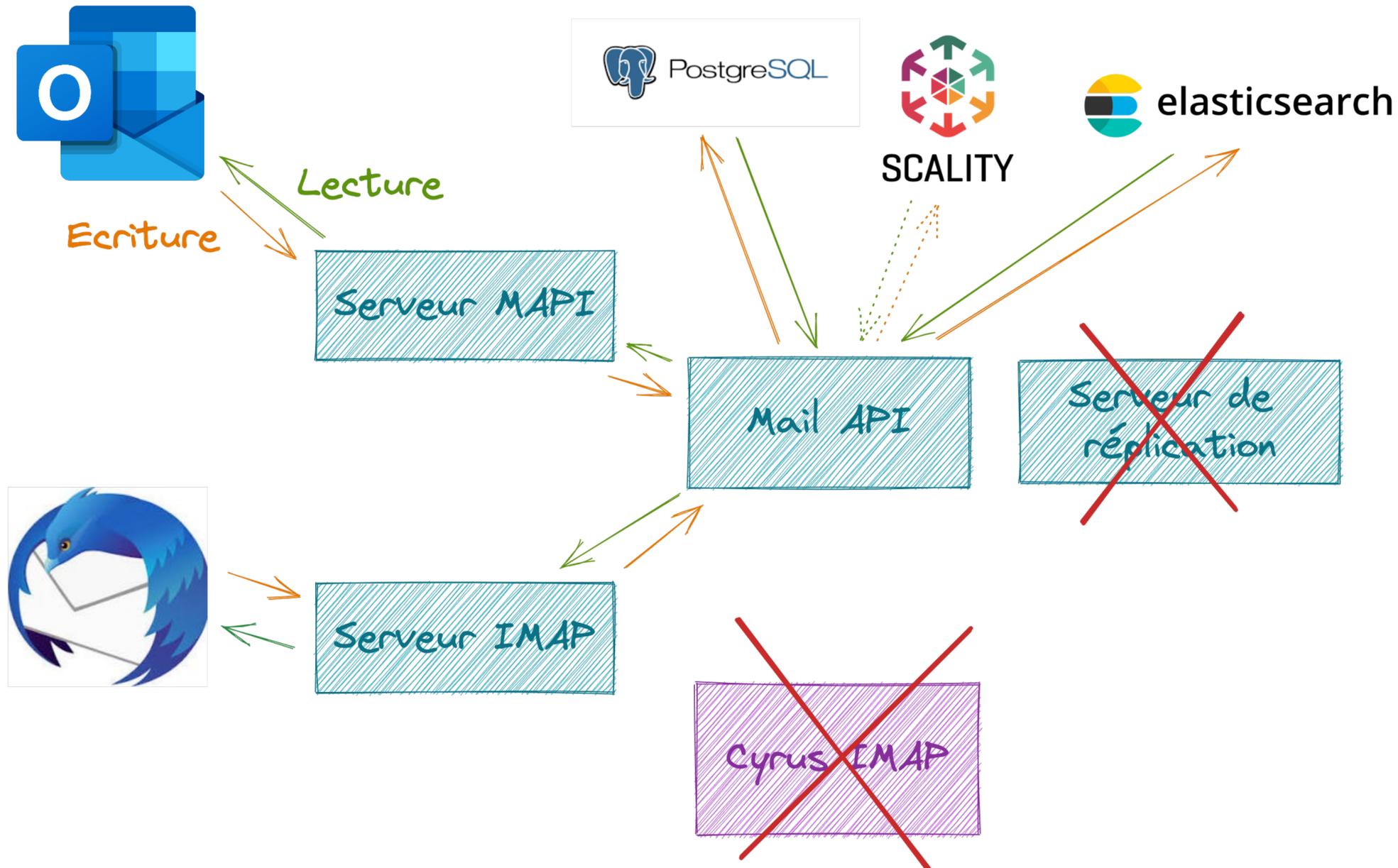
Accès aux mails

Interactions avec Cyrus



Evolutions

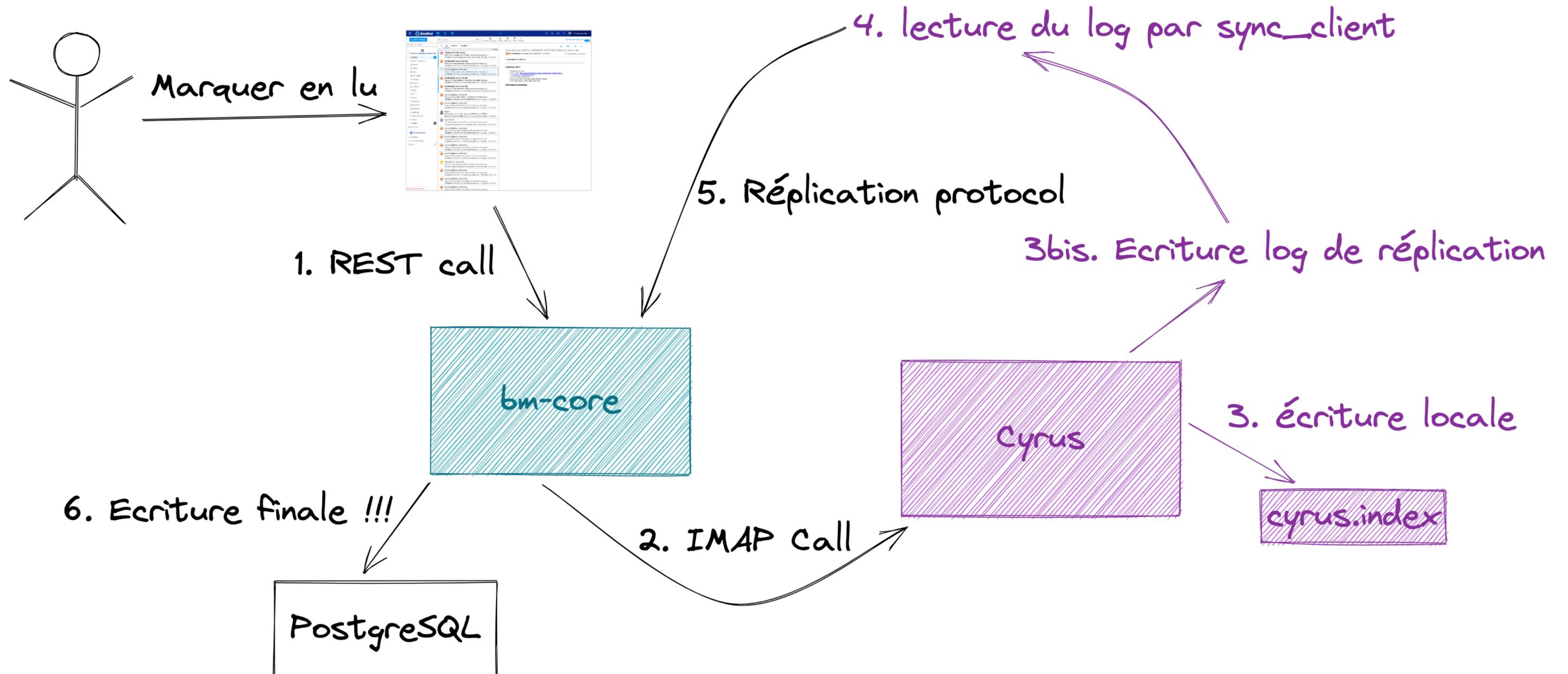
Une complexité nécessaire ?



- - asynchronisme fragile : on prends la main
- + perf !!
- + maitrise du format
- + proche du stockage
- + compression?

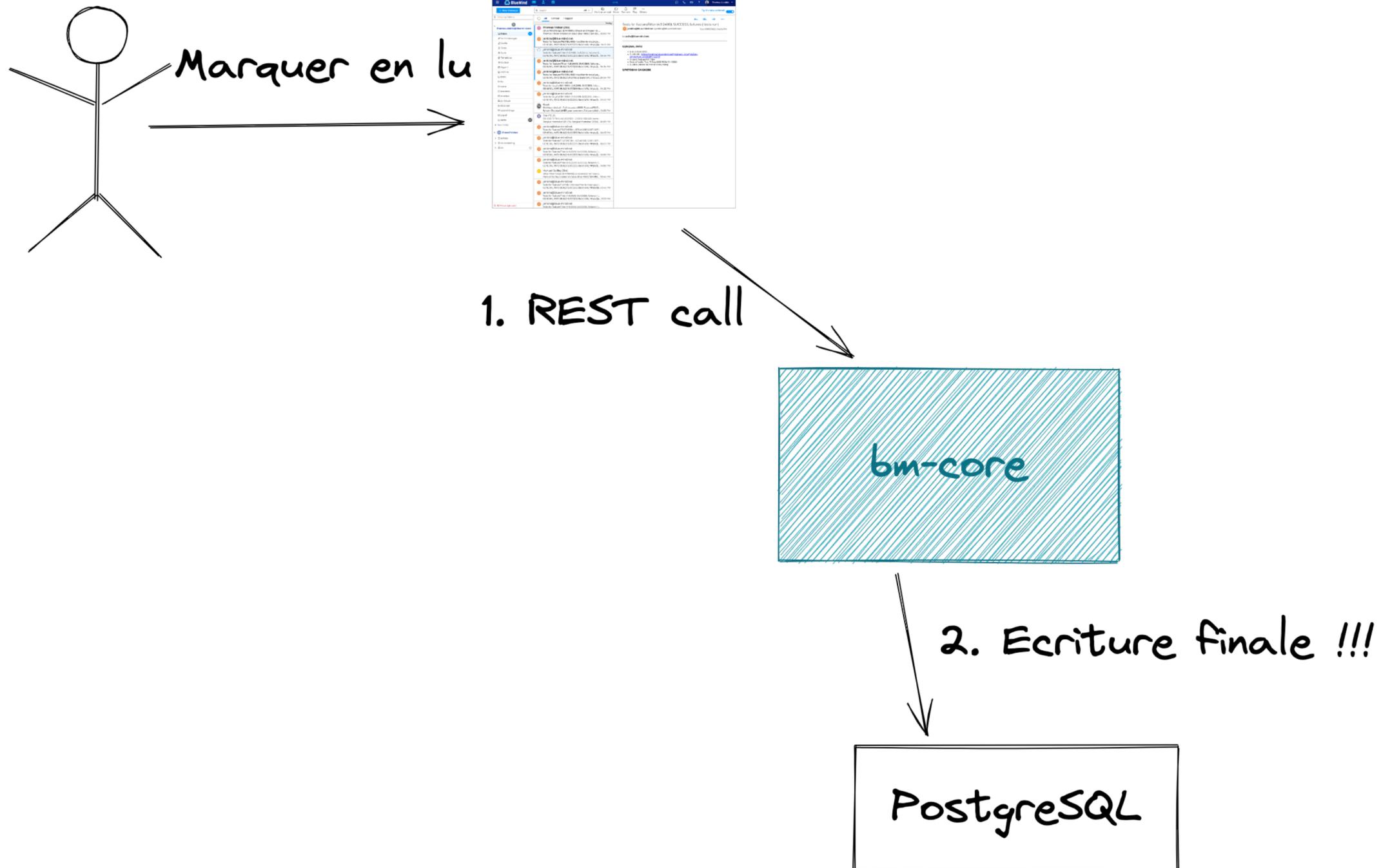
Performances

Marquer un mail en lu - v4



Performances

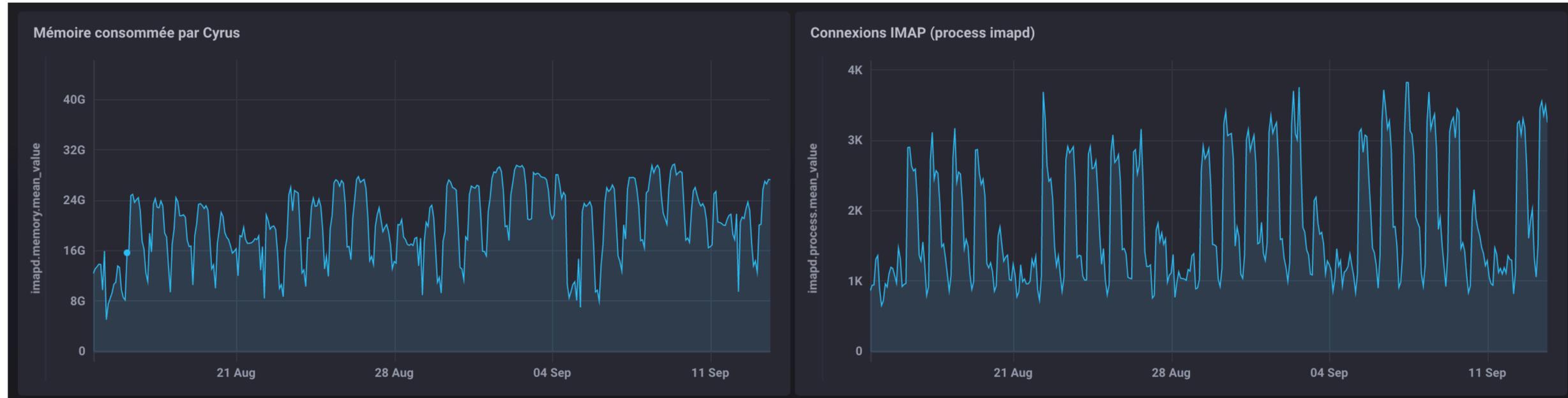
Marquer un mail en lu - v5



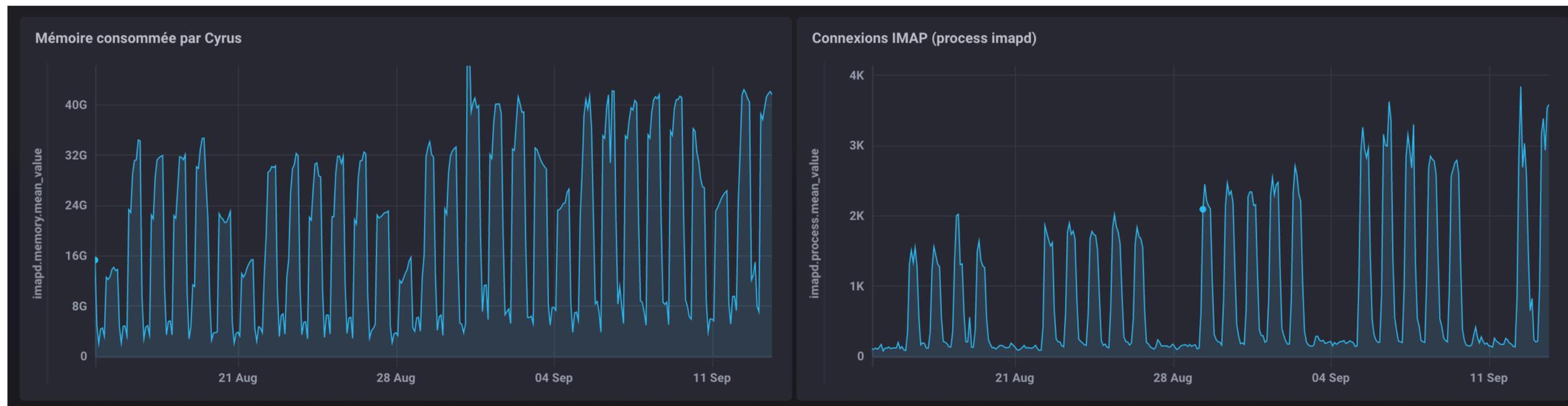
Evolutions

Performances ? (sur-consommation de Cyrus)

4.6.8



4.7.5



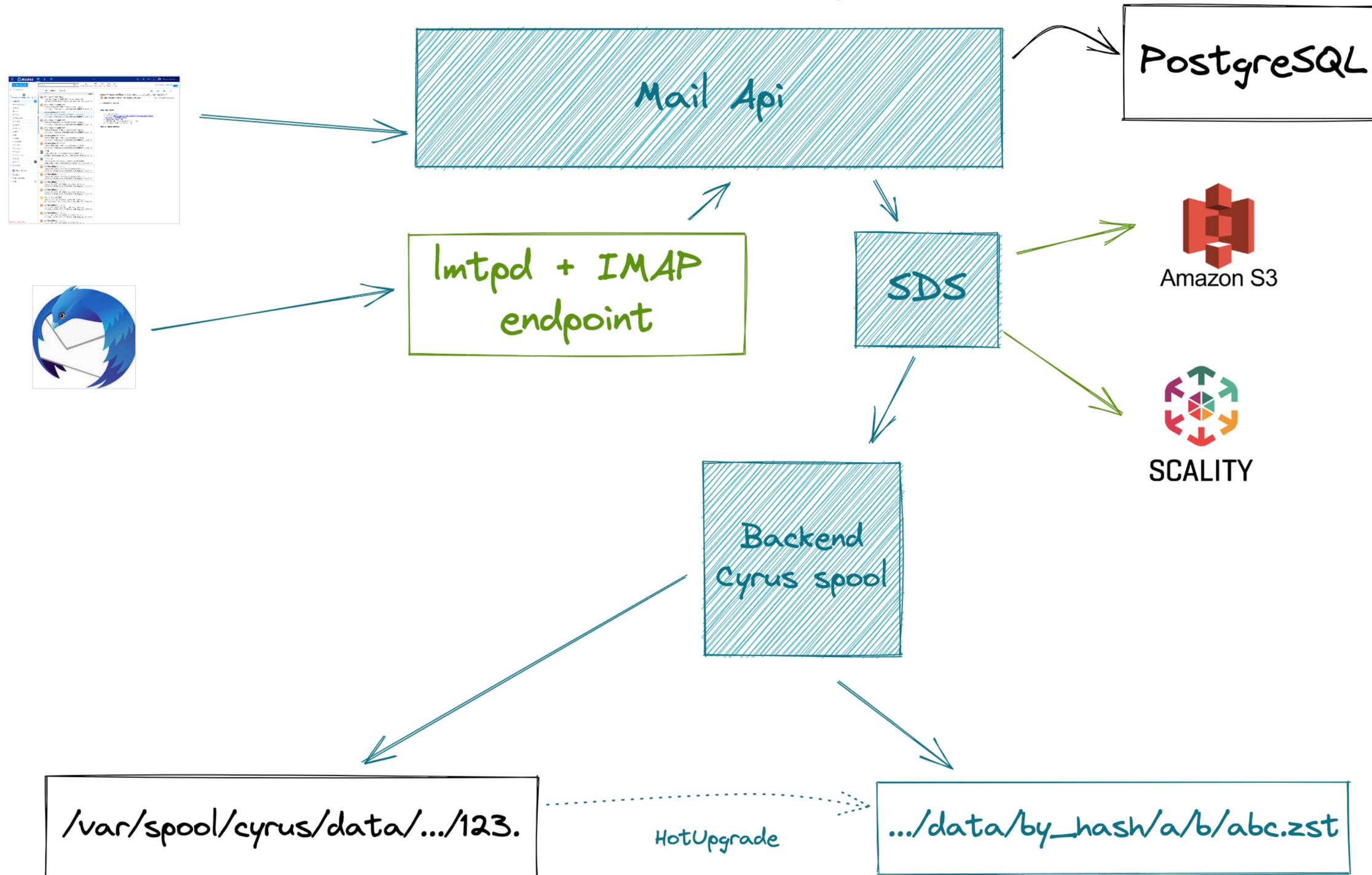
Evolution

Eliminer le verrou des partages

- Cas d'une infrastructure multi-backend
- 2 possibilités théoriques avec cyrus
 - proxy nginx + placement intelligent en fonction des partages (bm)
 - cyrus-murder qui fait des synchros - fusions de listes de dossiers
- mail-api de bluemind est indépendante du placement des boîtes
- Fin des limitations sur les partages lecture écriture !

Migration

Transparente - In place - Sans stockage additionnel



Résilience et MAJ sans interruption

Mieux, moins bien ?



20To >= 3jours à 1Gb/s



Référence au Stockage Objet



Quelques Go à écrire*

En résumé

Les limites de cyrus

- Architecture (très) dépassée
 - 1 connexion par processus
- Données de référence en local
 - cyrus.index, cyrus.header, mailbox_NN.conversations, annotations.db, mailboxes.db
 - Intégration avec stockage objet “primitive” (traitements additionnels)
- Mail immuable, format impossible à faire évoluer
- Partages très contraignants

En résumé

Les avantages BlueMind v5 sans Cyrus

- Performances de fonctionnement nettement améliorées
- Consommation de mémoire (et stockage) diminuée
- Partages globaux sans limitations de backend
- Pensé pour le stockage object
- PRA / PCA & mise à jour à chaud possible