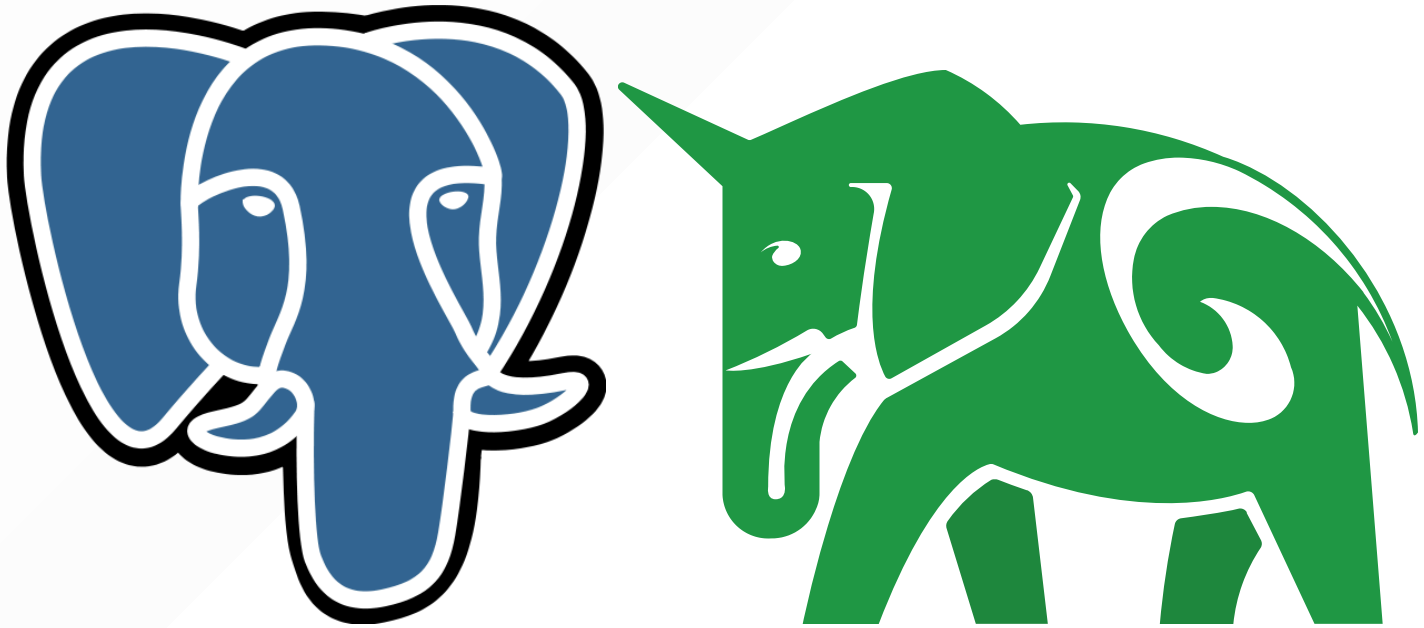
PostgreSQL en système distribué

Timothée Van Hove et Samuel Roland

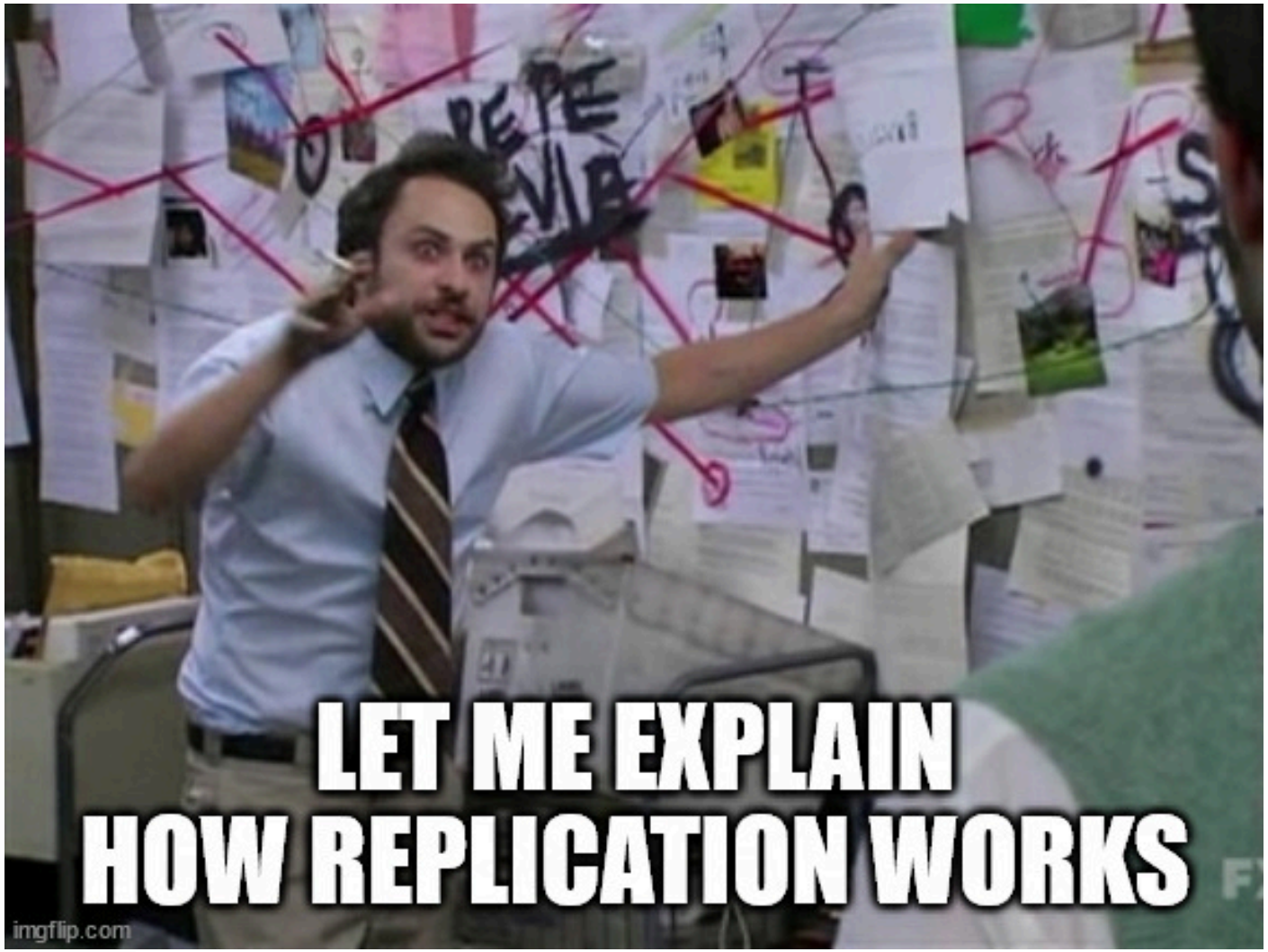


Comment faire de PostgreSQL un SGBD distribué?

Nous allons traiter 2 axes principaux de la distribution dans PostgreSQL :

- **Réplication** : Garantir la disponibilité des données en les répliquant entre plusieurs serveurs.
- **Partitionnement et Sharding** : Diviser les données pour améliorer la scalabilité et l'efficacité.

Réplication dans PGSQL

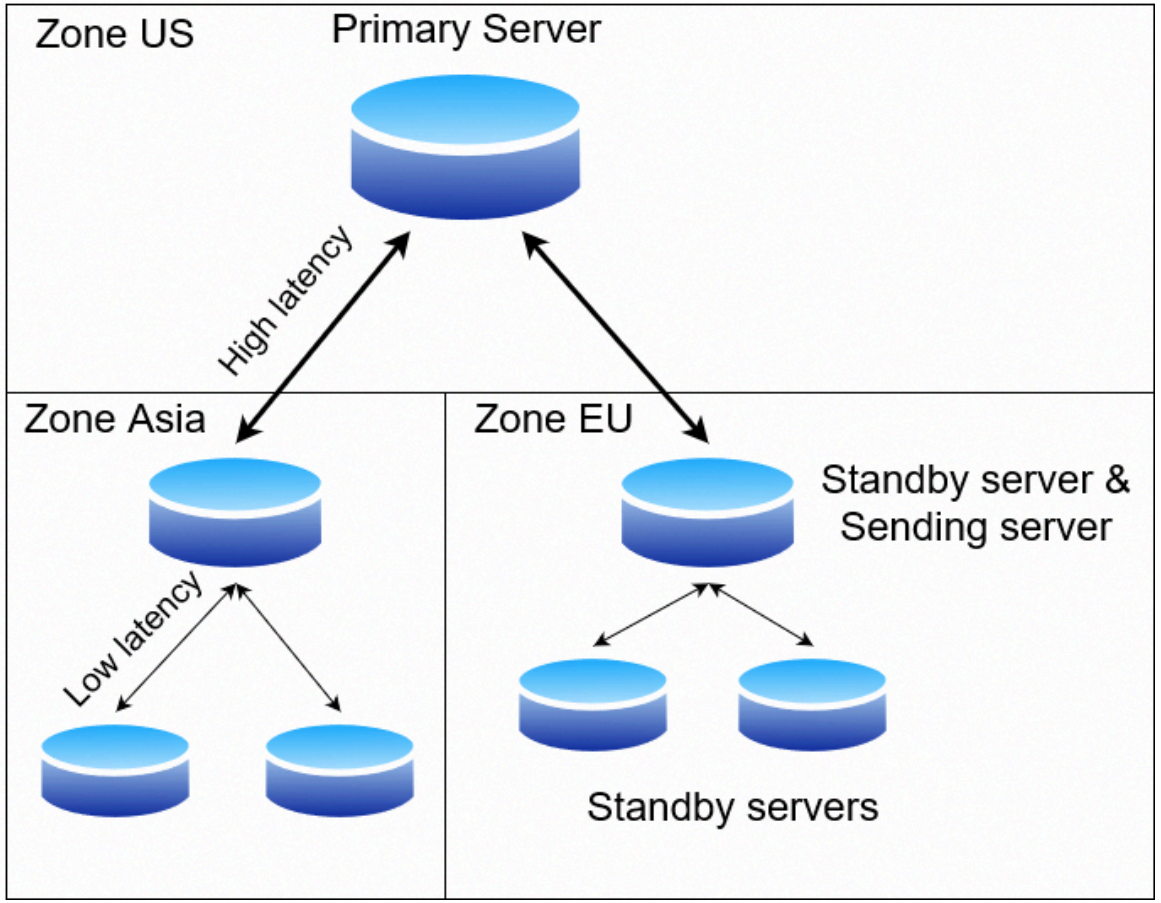


Qu’est-ce que la Streaming Replication ?

- Réplication la plus courante (leader unique)
- Utilise les journaux WAL pour synchroniser les followers (standby nodes) avec le leader (primary node).
- Les followers recoivent les journeaux WAL de manière quasi-continue

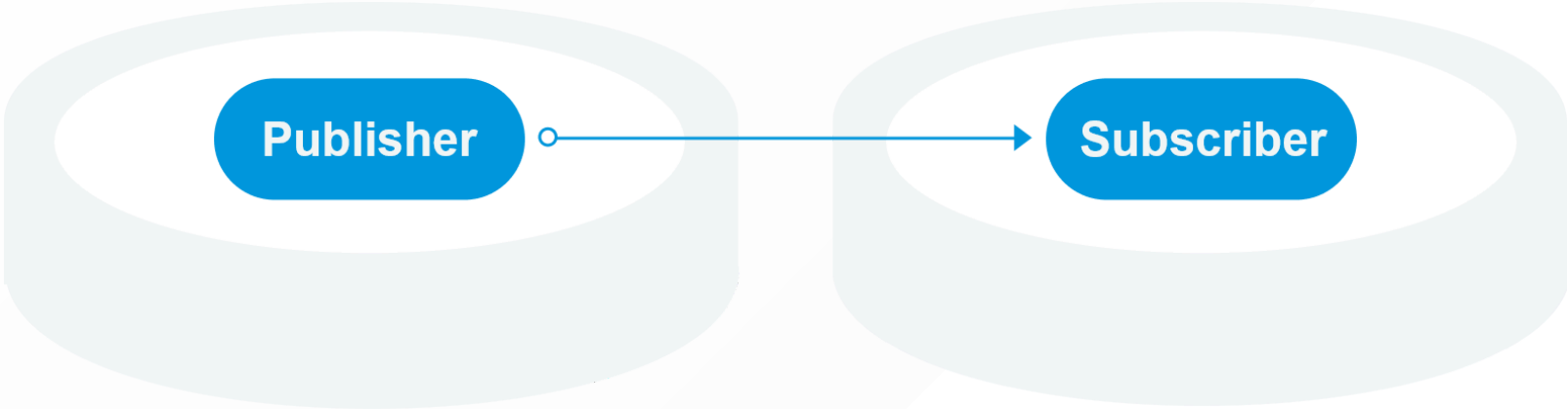
Streaming Replication - Modes

- **Mode Synchrone**
  - Le Primary Server (Leader) attend la confirmation des répliques avant de valider une transaction.
- **Mode Asynchrone**
  - Le Primary Server (Leader) n'attend pas de confirmation des répliques ; il envoie les WAL dès qu'ils sont générés.
- **Réplication en cascade**
  - Un Standby Server (Follower) peut avoir le charge d'envoyer les WAL à d'autres followers. On parle de "Sending server"
  - ça permet de réduire la charge du leader



Qu’est-ce que la Logical Replication ?

- Réplique les modifications au niveau des transactions (lignes/tables spécifiques).
- Fonctionne via un **publisher** (leader) et des **subscribers** (followers)
- Le Publisher transforme le WAL en opérations transactionnelles (UPDATE, INSERT, DELETE…)
- Les opérations sont envoyées aux subscribers, puis sont appliquées dans le même ordre transactionnel que sur le Publisher.
- Les schémas doivent être identiques ou compatibles entre Publisher et Subscriber.



Streaming vs Logical Replication

Streaming Replication

- Objectif : Maintenir une copie exacte de la base pour haute disponibilité et basculement.
- Avantages :
  - Simple à configurer.
  - Faible latence.
- Limites :
  - Réplique toute la base.
  - Pas de personnalisation ou de filtrage des données.

Logical Replication

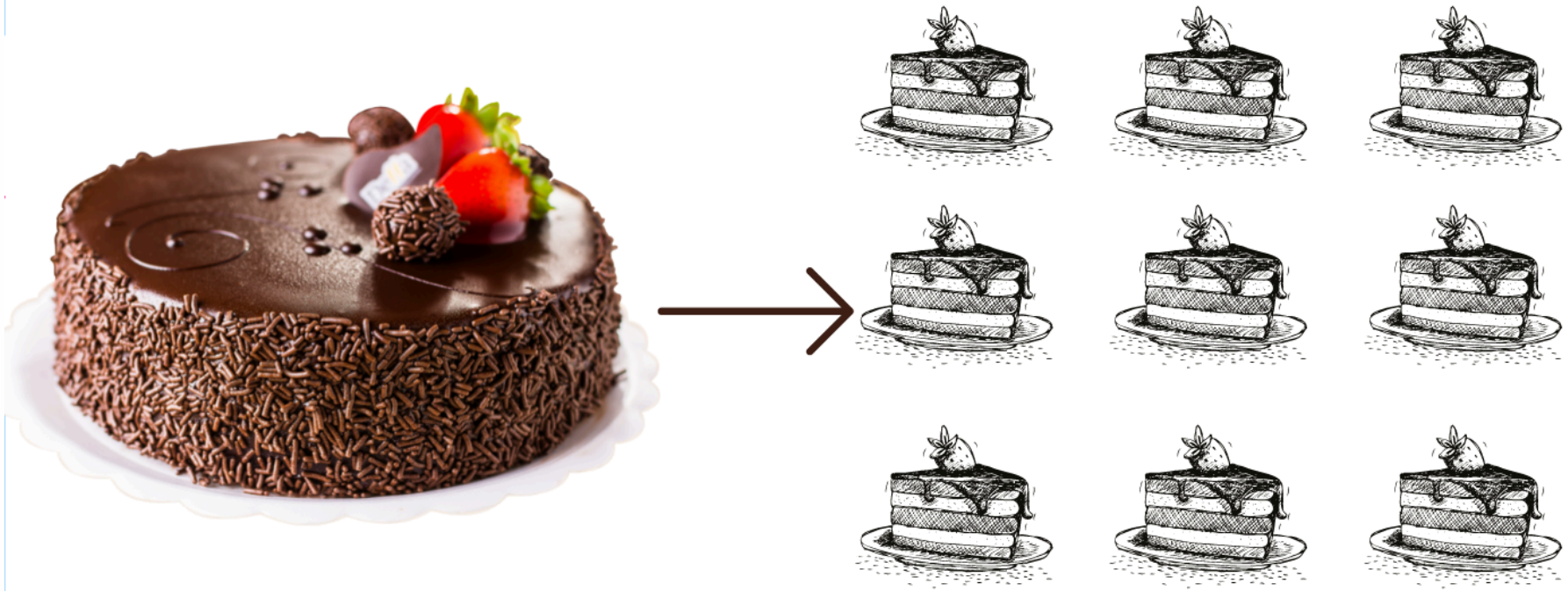
- Objectif : Partager des données spécifiques
- Avantages :
  - Flexible, permet de cibler des tables ou types de modifications.
  - Compatible entre versions ou plateformes.
- Limites :
  - Conflits possibles en cas d'écritures locales sur le Subscriber.
  - Schéma non répliqué automatiquement.

"Réplication multi-leader ? Bi-Directional Replication"

PGSQL ne supporte pas la réplication multi-leader nativement. BDR est une extension pour la réplication multi-leader basée sur le logical replication:

- Chaque nœud agit comme un leader capable d'accepter des écritures
- Les conflits d'écriture sont détectés lorsque plusieurs nœuds modifient les mêmes données.
- Utilise des résolveurs de conflits pour déterminer comment gérer ces situations.
  - Par défaut, BDR applique le résolveur `update_if_newer`, qui conserve la version de la ligne ayant le timestamp de commit le plus récent.
- **Avantages :**
  - Partage de la charge d'écriture entre plusieurs nœuds.
- **Limites :**
  - Complexité : gestion des conflits entre les nœuds.
  - Licence commerciale et "source disponible"

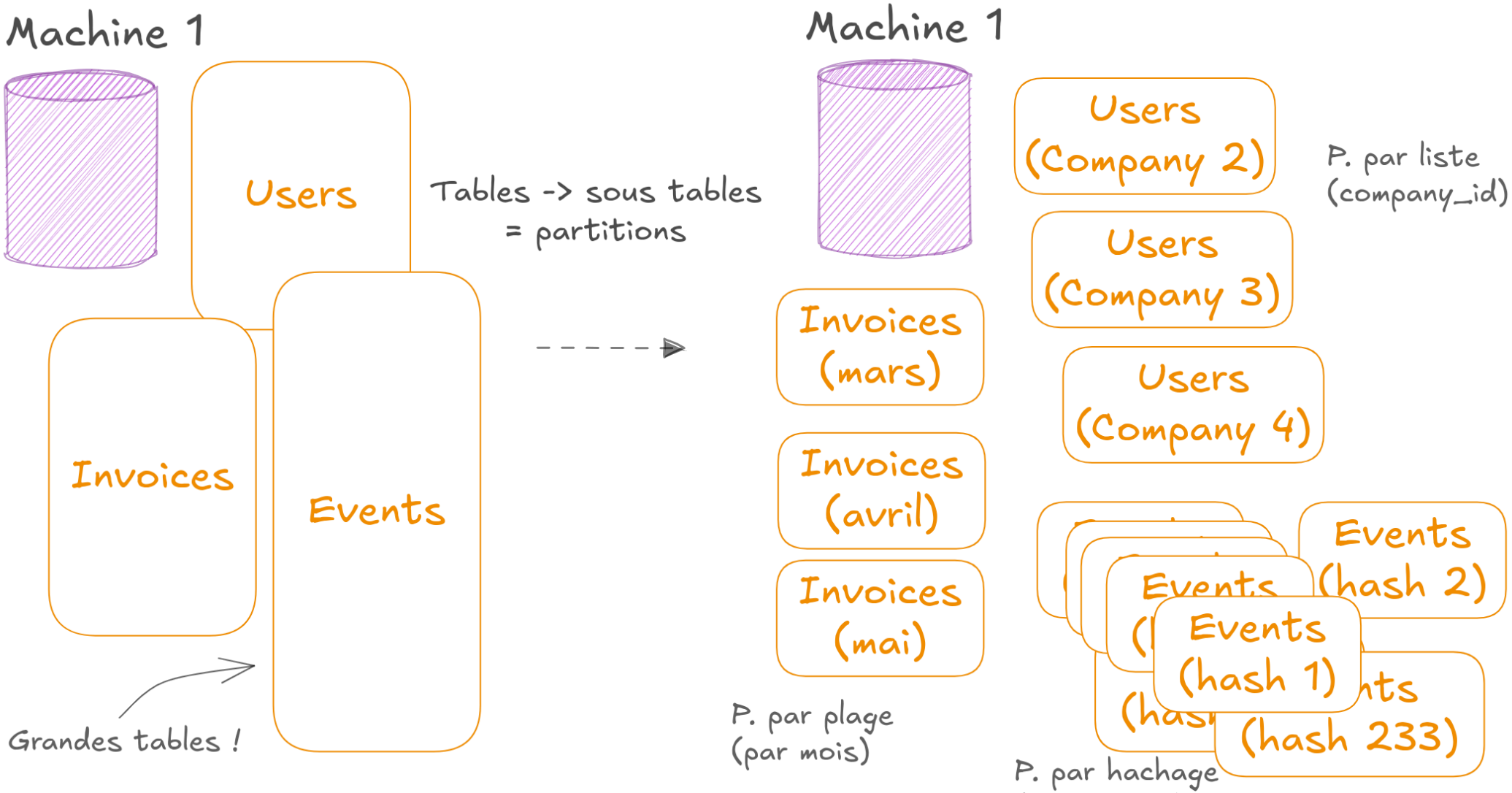
Partionnement et sharding



“ https://www.thegeekyminds.com/post/database-sharding-vs-partitioning ”

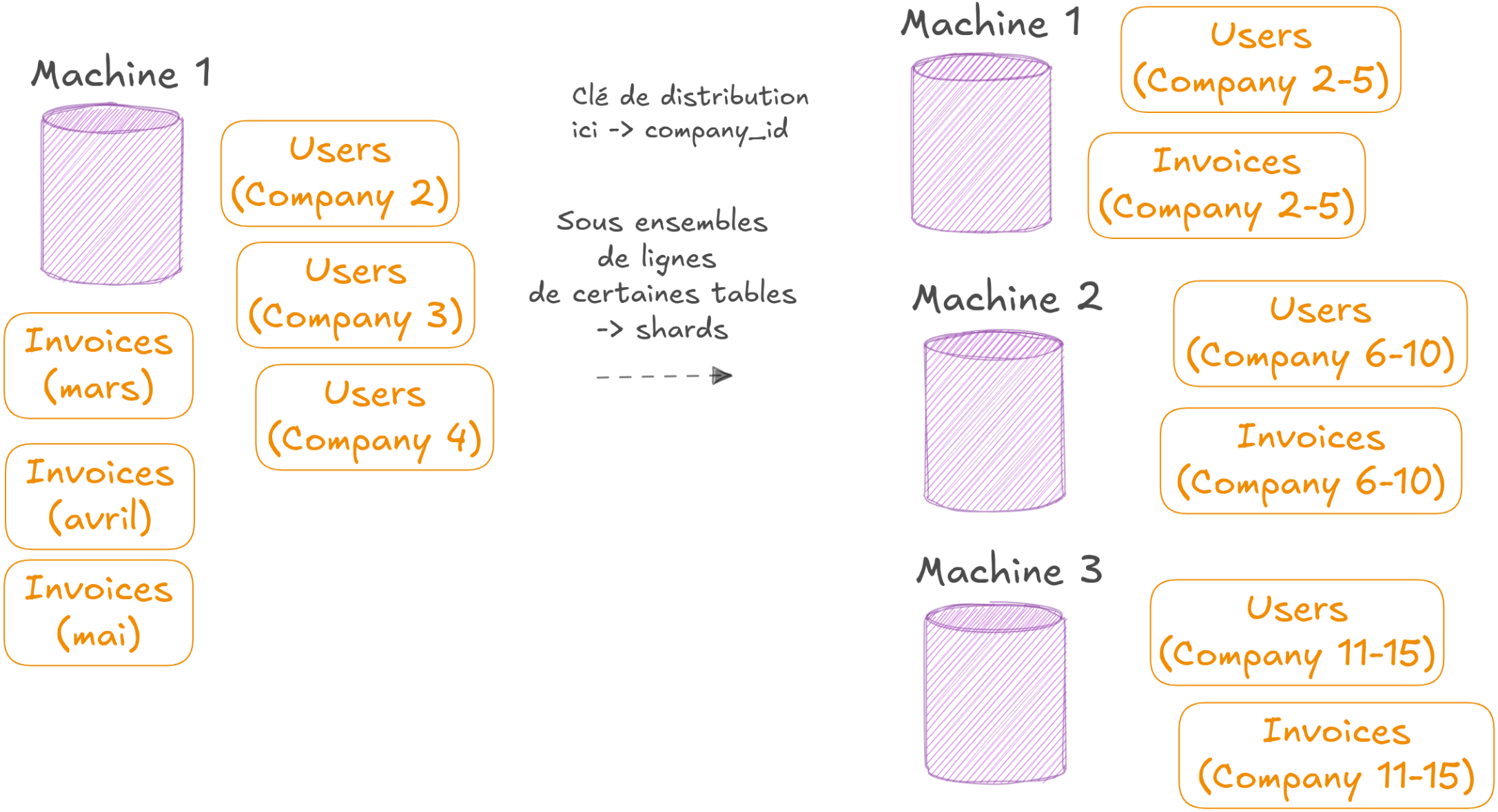
```sql
select username from users where company_id = 12; -- Entreprise 12
select price from invoices where date >= '2024-05-01' AND date < '2024-06-01' -- Mai 2024
select name from events where date = '2024-05-01' -- Jour spécifique
```

Partionnement natif PGSQL

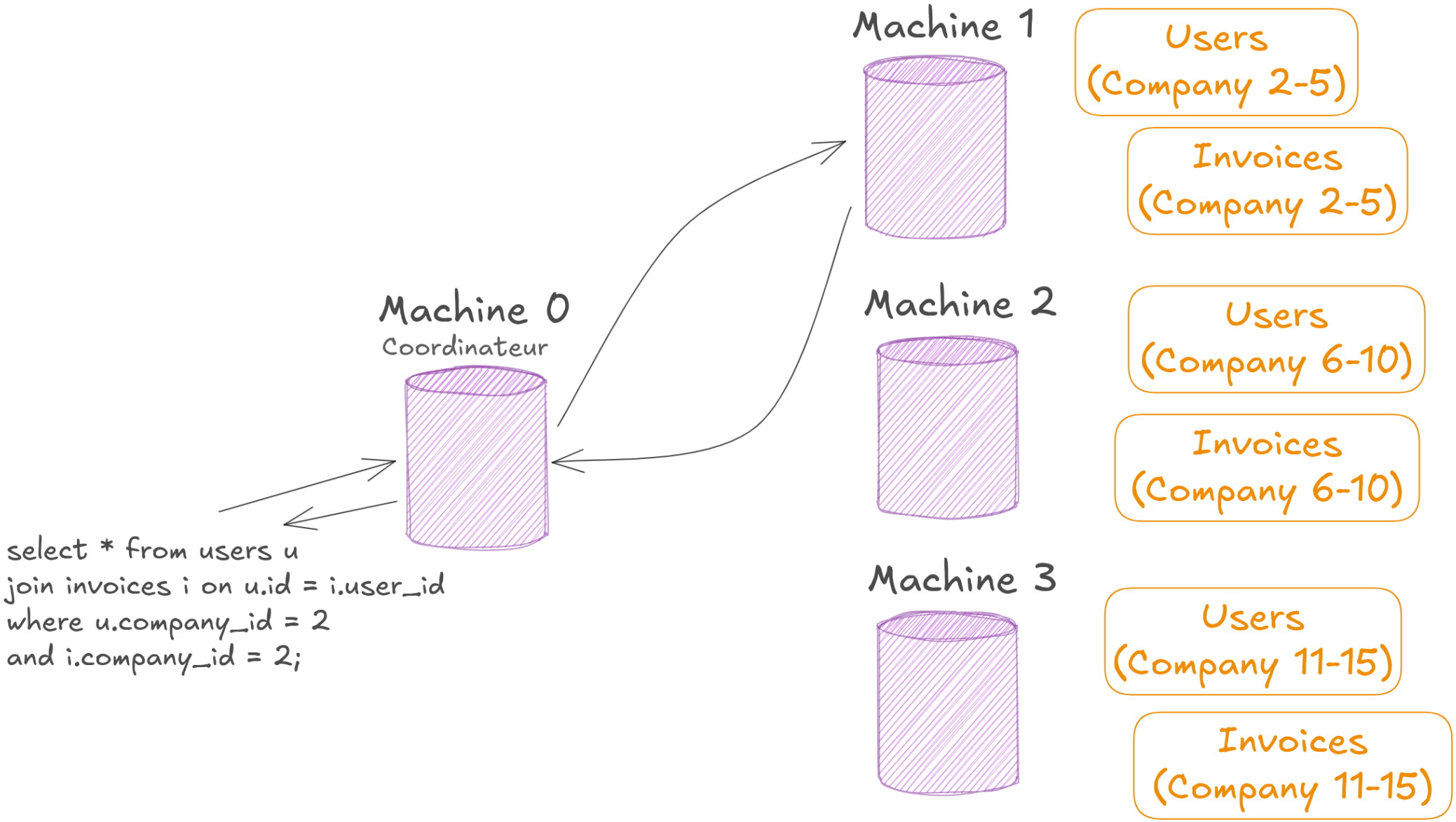


-> Amélioration des performances des requêtes
-> Suppression efficace des données par sous table

Sharding avec Citus



Citus - routage



Citus - parallélisation