

Les Bases de données

« NoSQL »

Part 2

Présenté par

Marie Esther BOUA
Data Scientist

Fondamentaux du NoSQL



Fondamentaux du NoSql

1. Passage à l'échelle
2. Architecture
3. Sharding/Partitionnement
4. Réplication
5. Tolérance aux pannes



Fondamentaux du NoSql

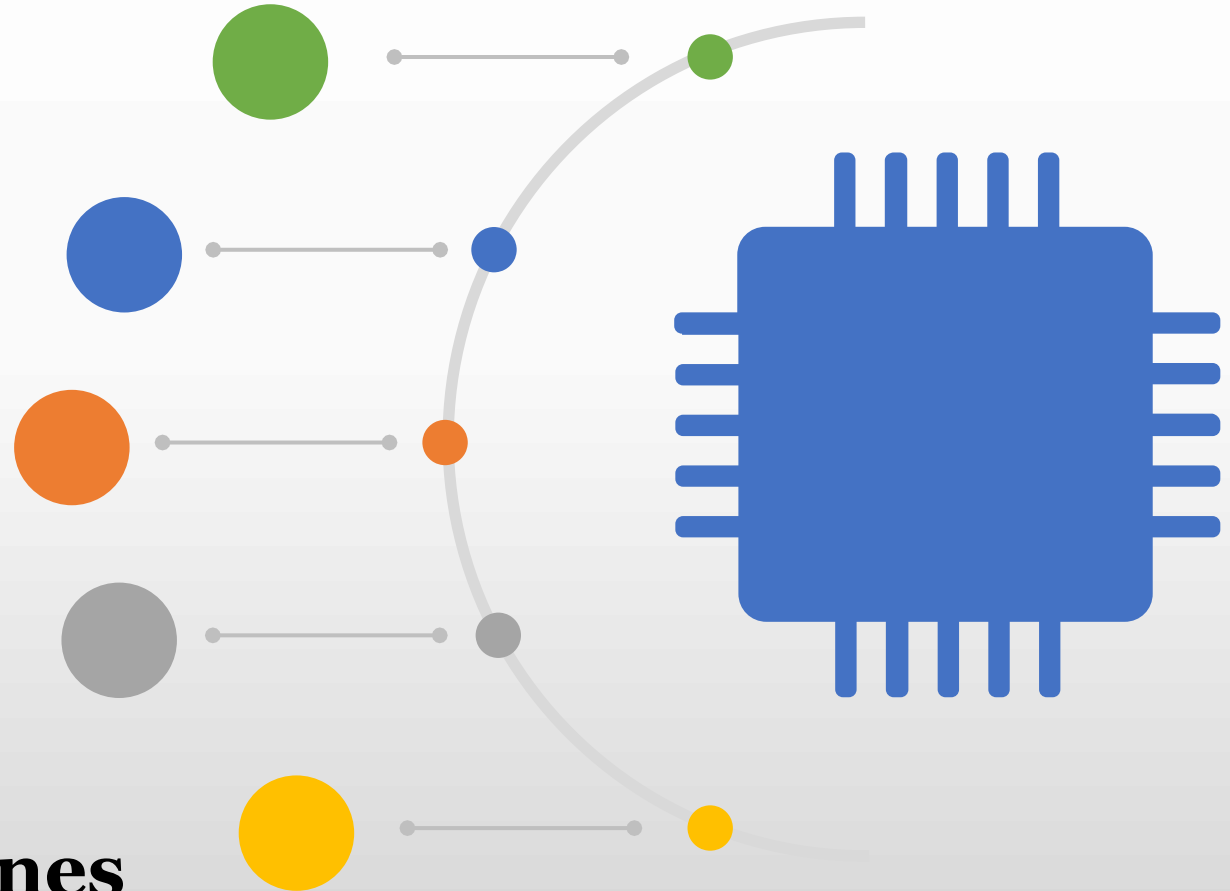
Passage à l'échelle

Architecture

Sharding/Partitionnement

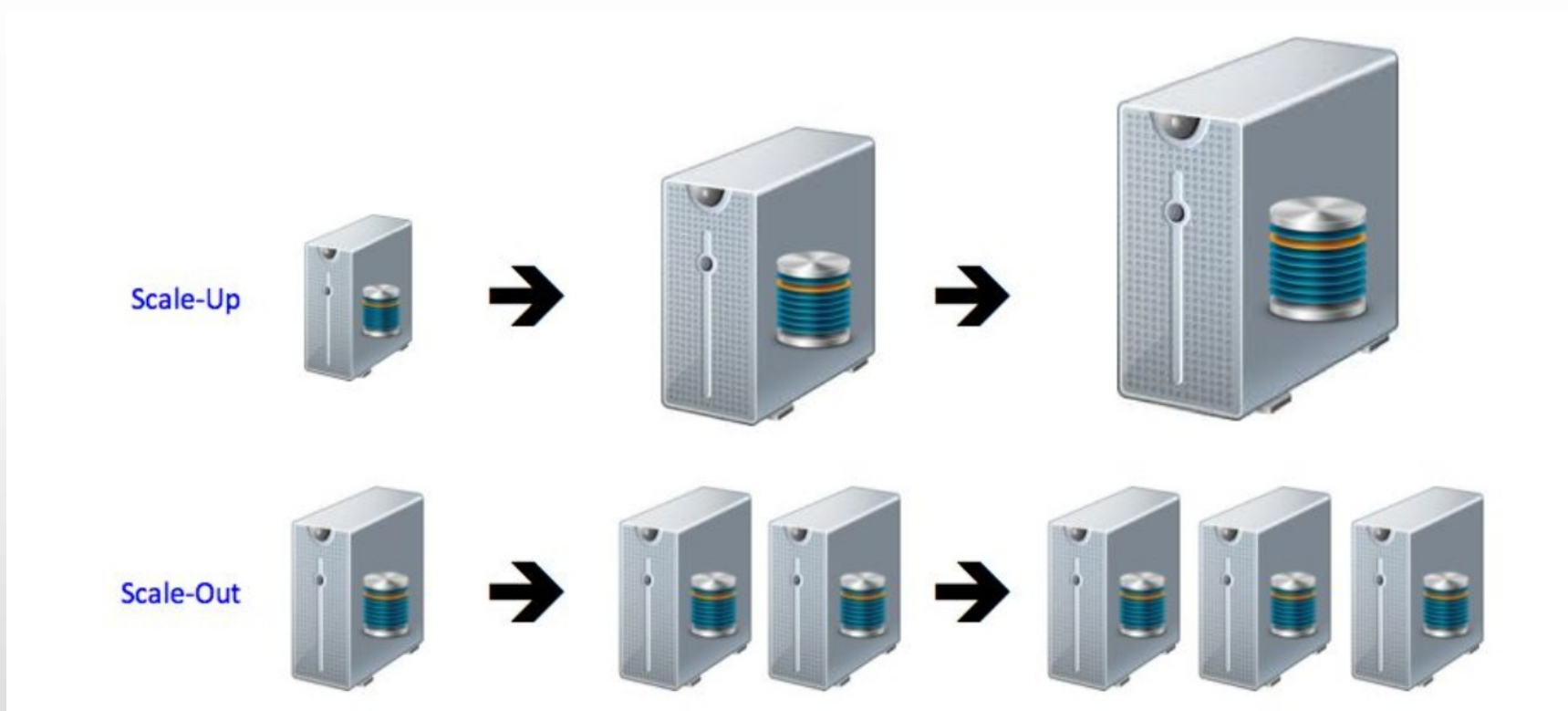
Réplication

Tolérance aux pannes



Passage à l'échelle

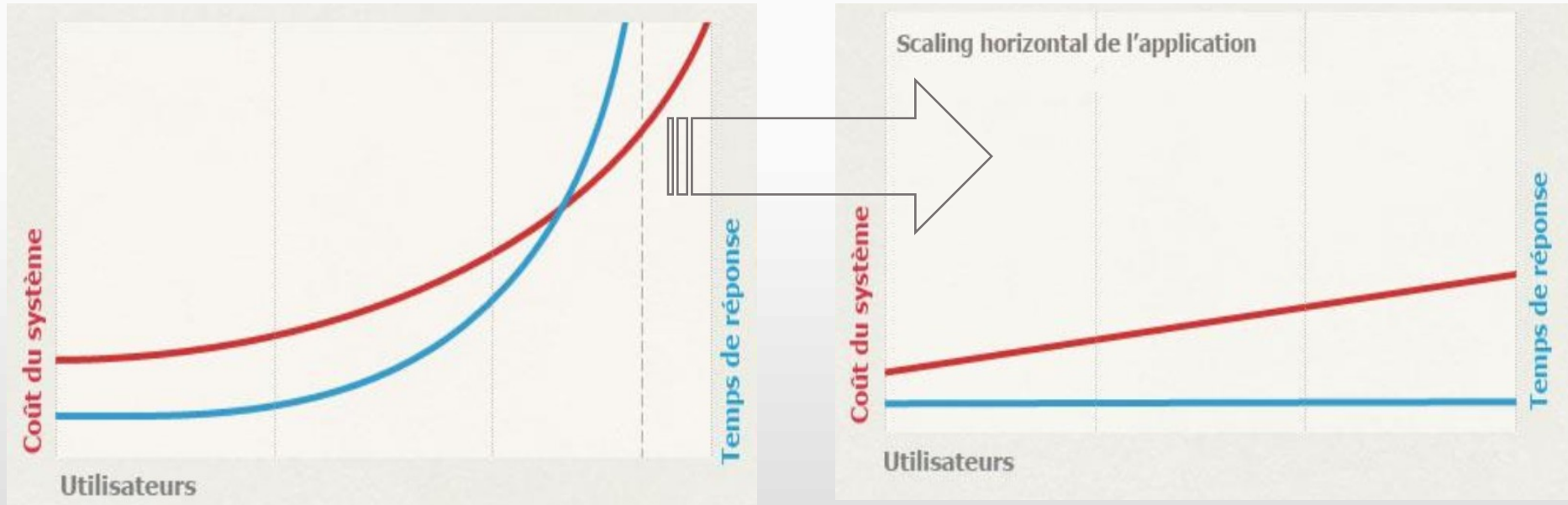
Augmentation des ressources sans dommage des systèmes
Horizontale / Verticale



- Taille
- Géographique
- Administrative

Passage à l'échelle

Assurer la scalabilité / Elasticité des systèmes



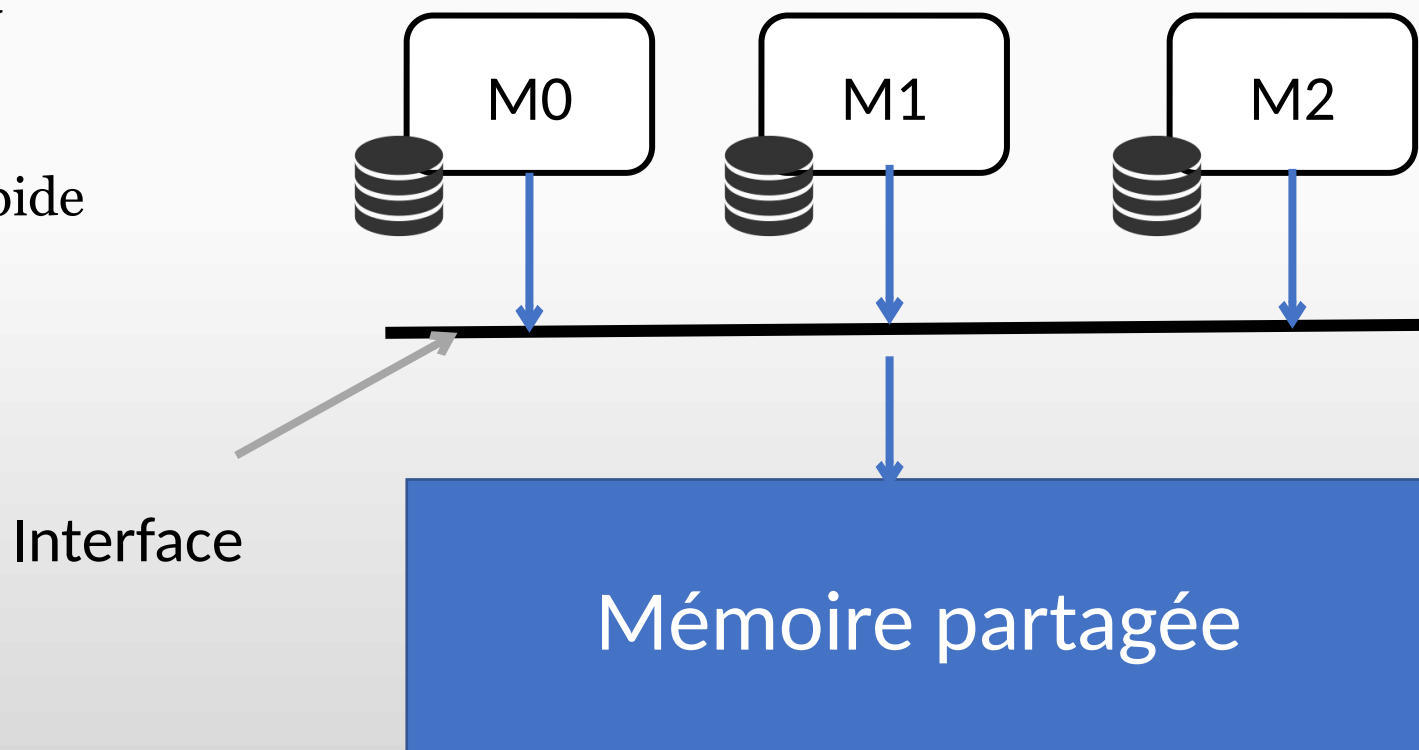
Scale out pour les OS propriétaire
Mauvaise qualité de scale up

Architecture BD NoSQL

Système/BD distribué ou repartitionné
4 familles de système

- **Shared Memory**

- traitement rapide

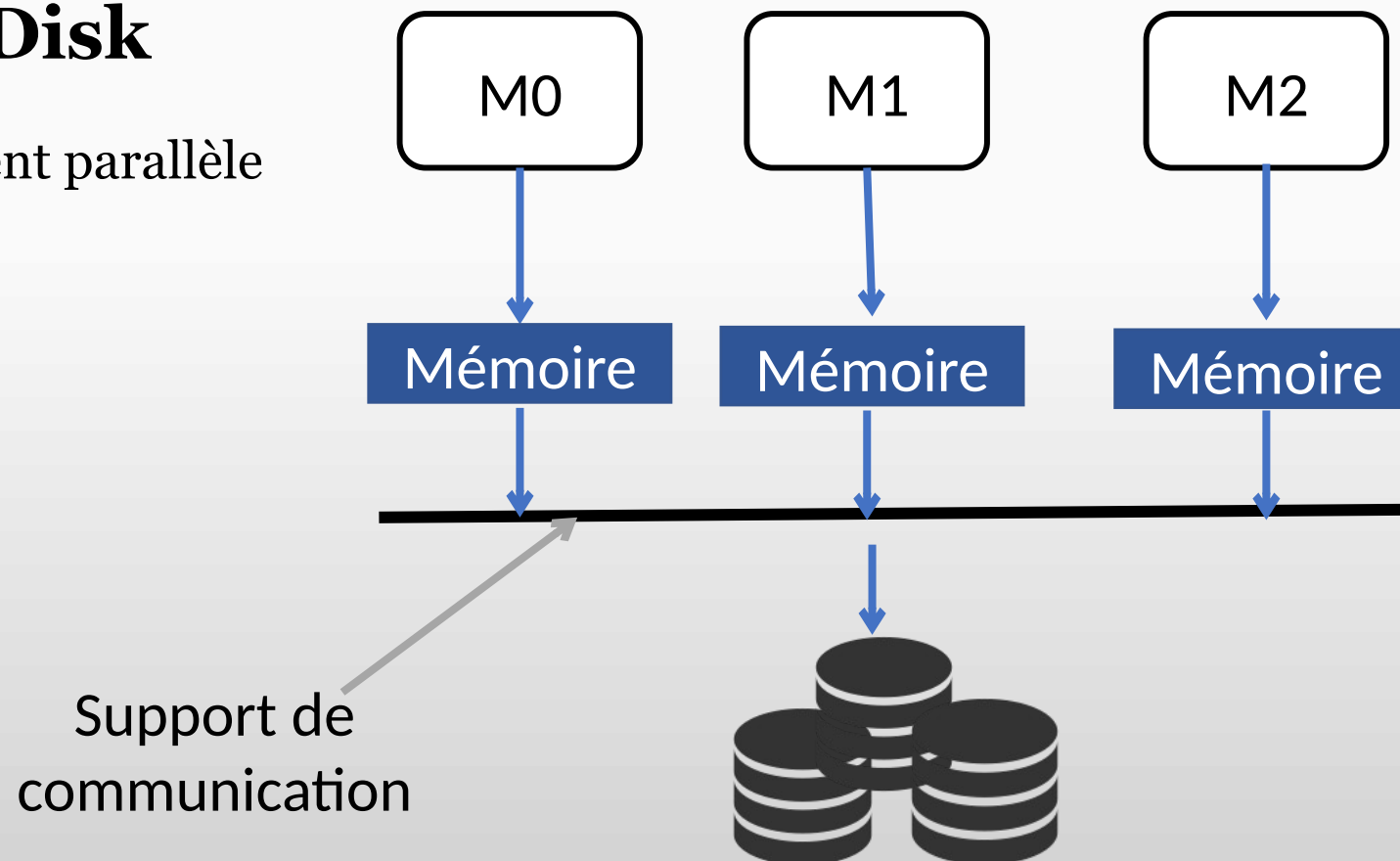


Architecture BD NoSQL

Systèmes, BD distribué ou repartitionné
4 familles d'architectures

- **Shared Disk**

- traitement parallèle

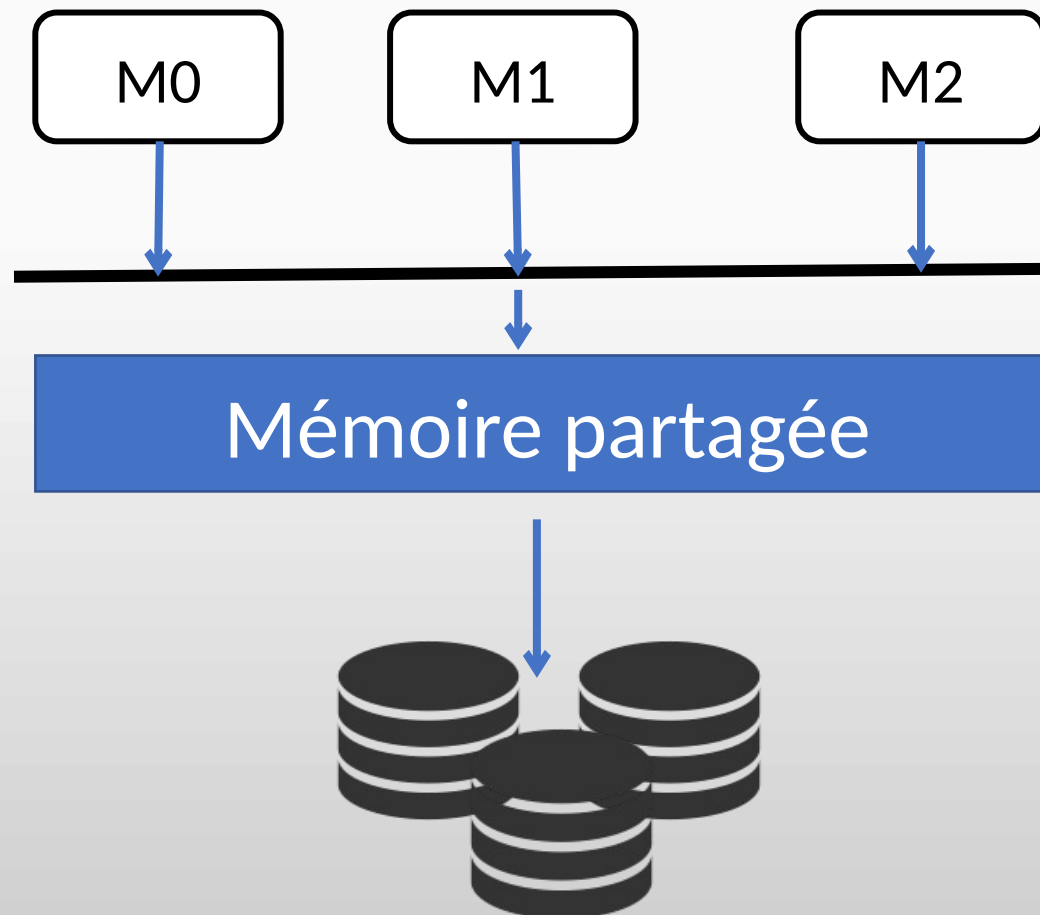


Architecture BD NoSQL

Systèmes, BD distribué ou repartitionné
4 familles d'architectures

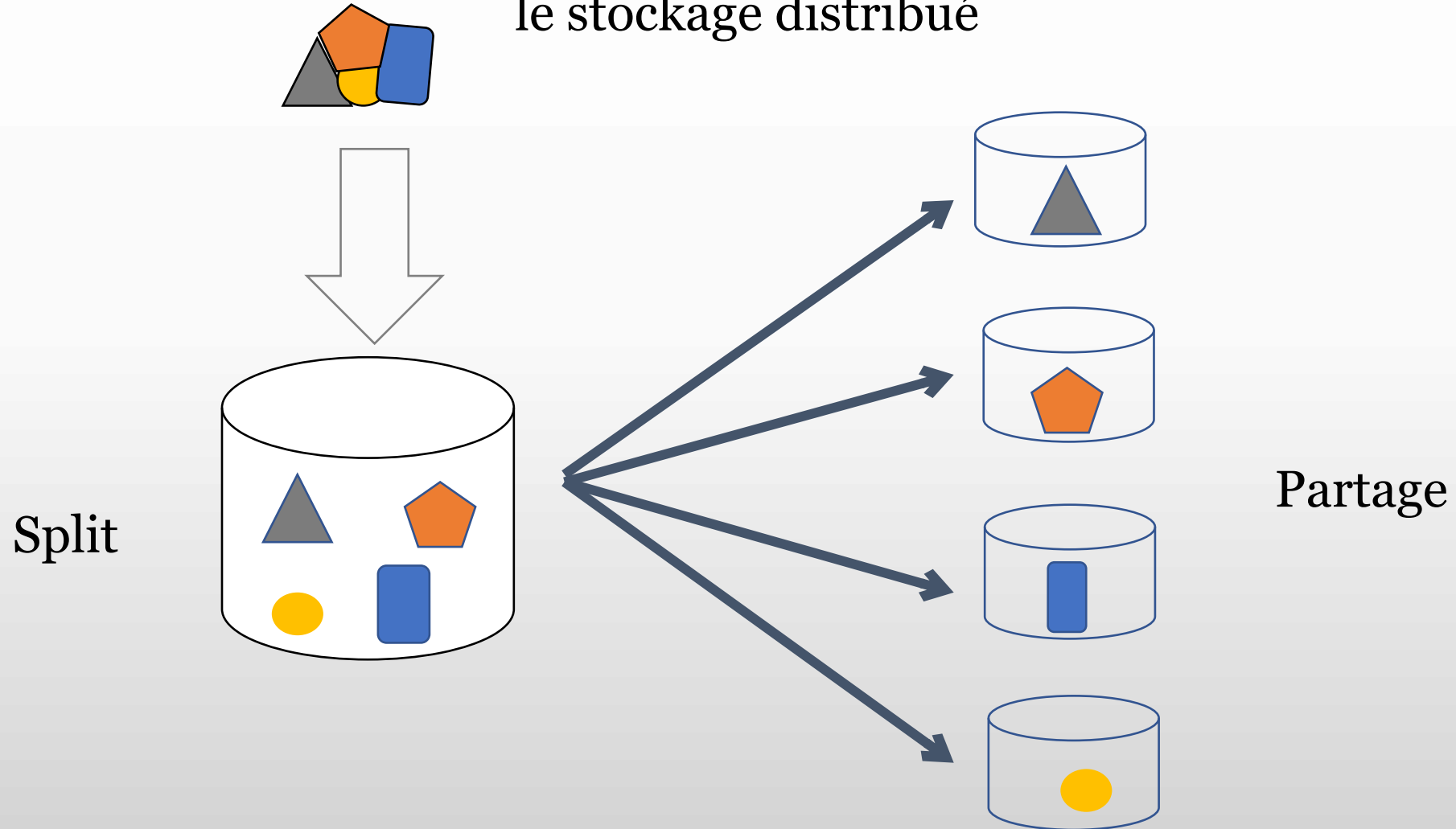
- **Shared Everything**

- Une horloge commune
- Ressources partagés
 - ✓ Processeur
 - ✓ GPU



Sharding

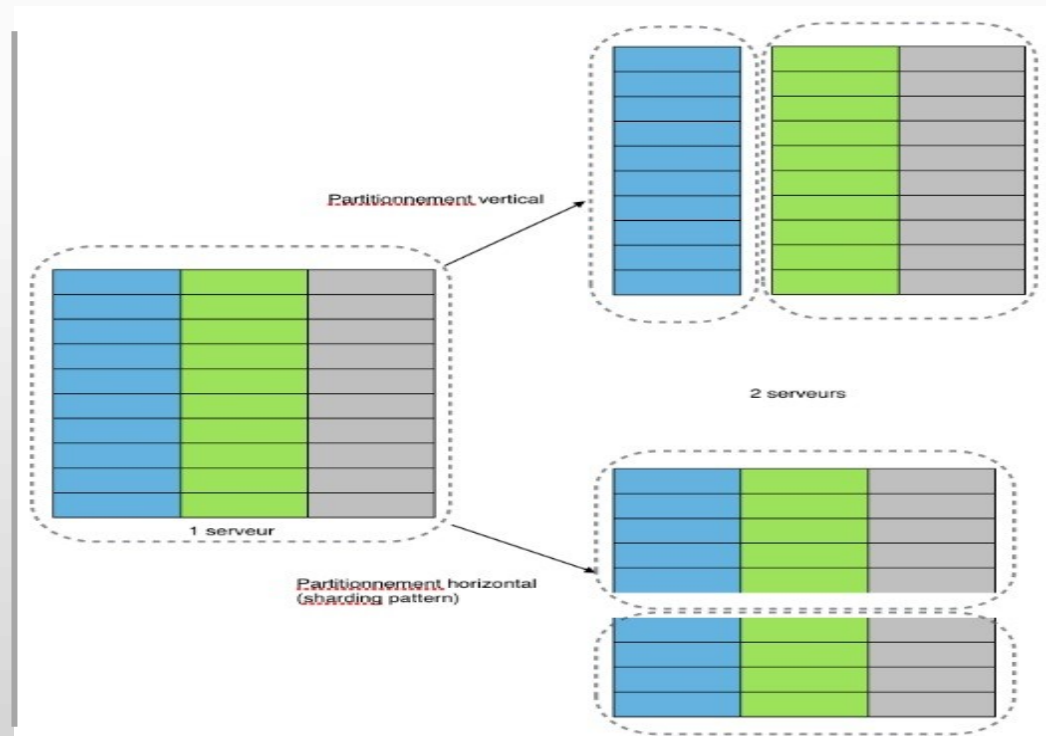
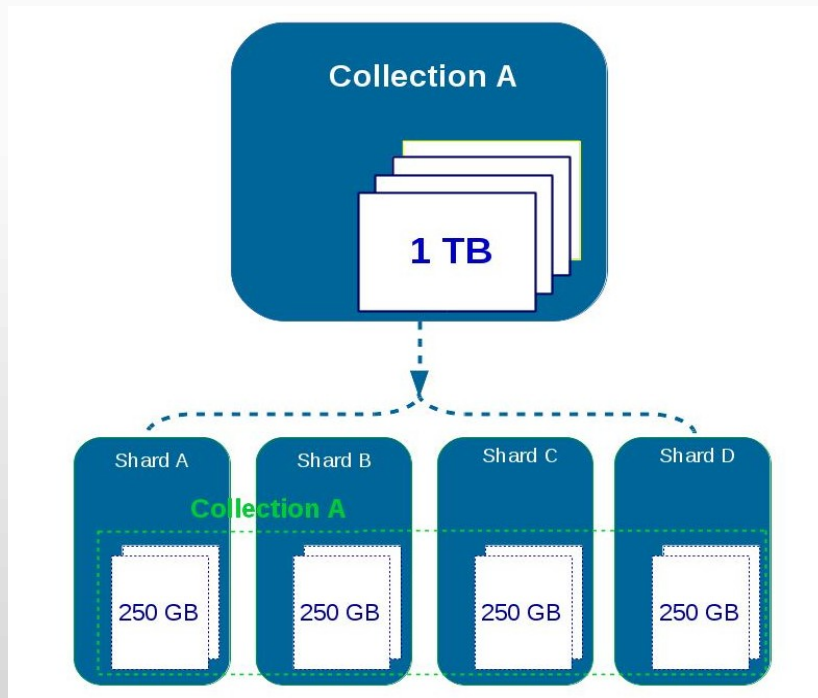
Le partitionnement des données
le stockage distribué



Sharding

Stratégie de Split

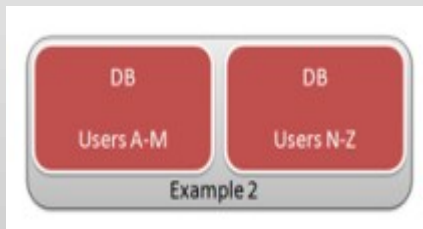
- Chunk size (into Block)
- Split Rows / columns



Sharding

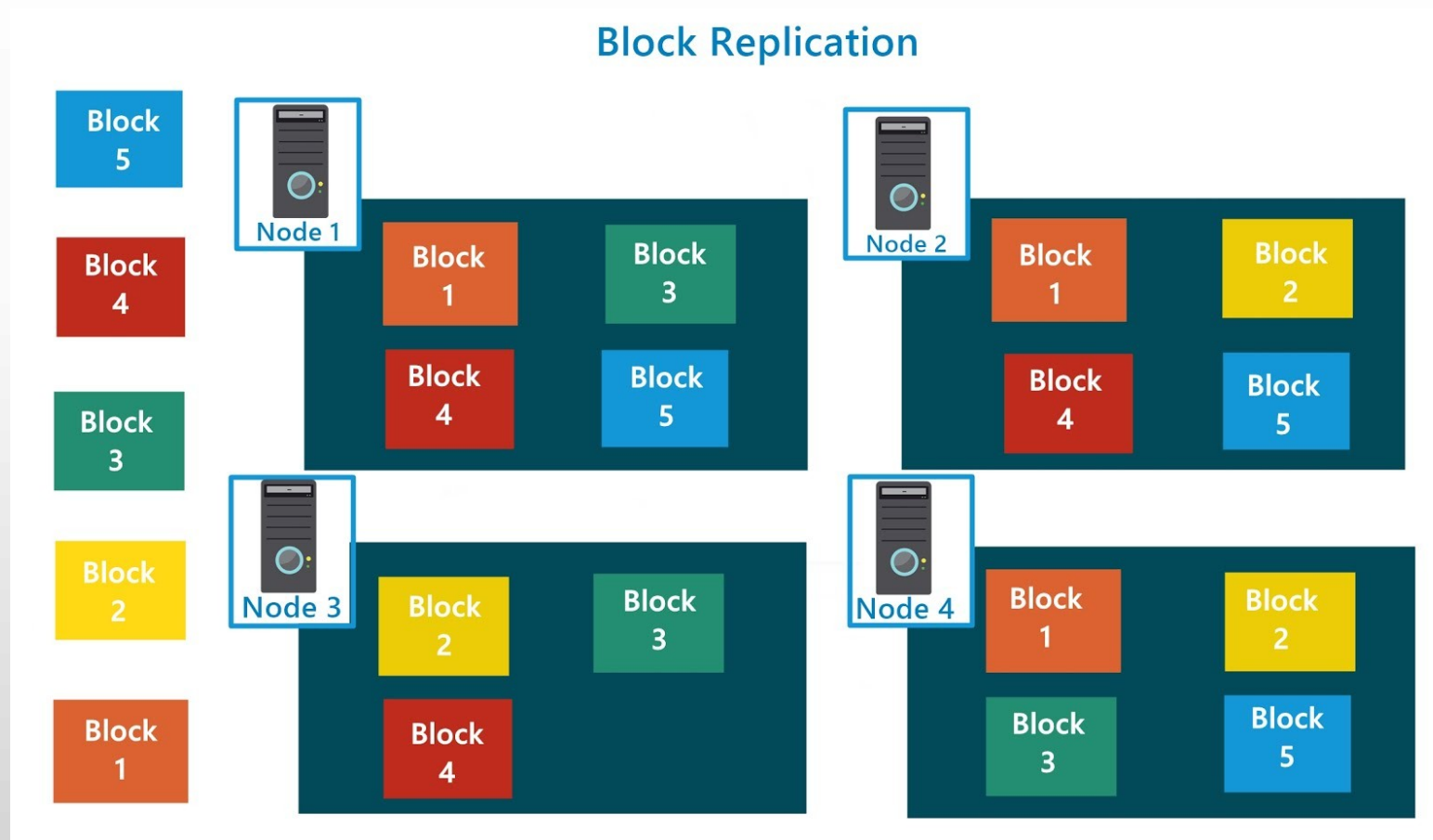
Pourquoi ?

- **Indexation pour accélérer les recherches**
 - Indexation clé – valeur
 - Hashage
- **Traitement distribué**
- **Favoriser l'élasticité**
- **Eviter la perte totale de données**



Réplication

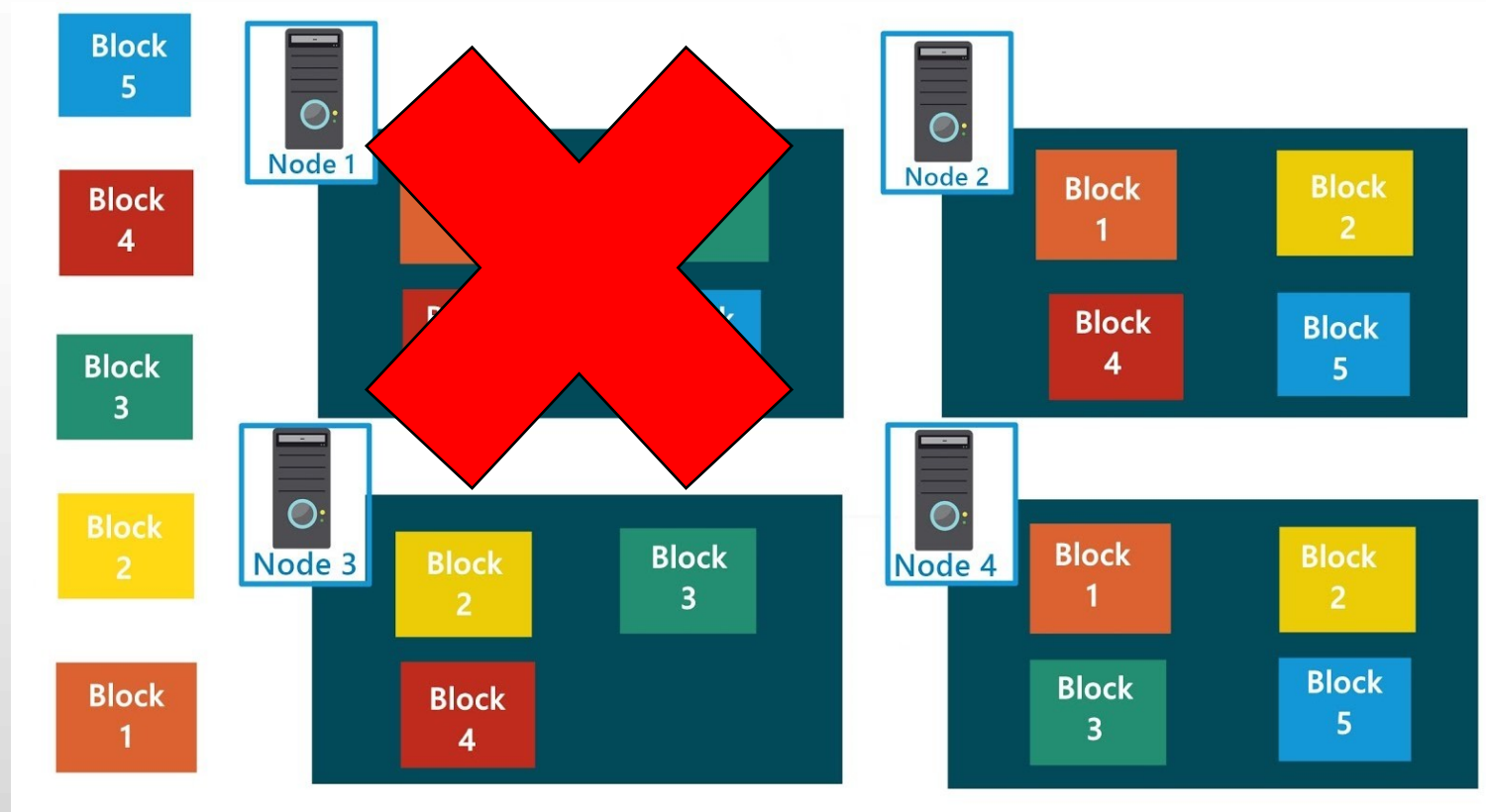
Copie de données ou partition de données
Temps de réponse / protège (Perte) données



3 copies par défaut sinon pris en charge par le système

Tolérance aux Pannes

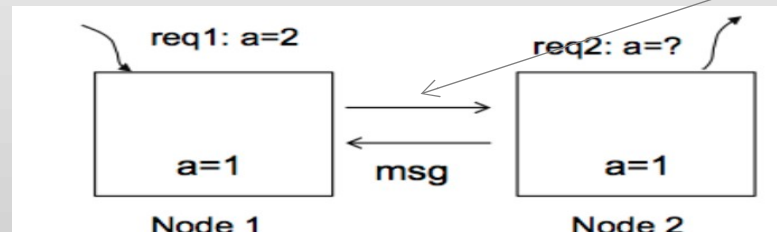
Machines hors services et continuité du Système
Continuité permanente des services du système



Propriétés et cohérence

Modèle de cohérence

- **Trois (3) niveaux de cohérences**
 - Sémantique : règle de gestion
 - Transactionnelle : mode de validation
 - Mutuelle : cohérences des copies
- **Diffusion des modifications et mise à jour**
 - Mode Asynchrone → Données obsolète
 - Mode synchrone → cohérence forte



Quelle
Stratégies ?

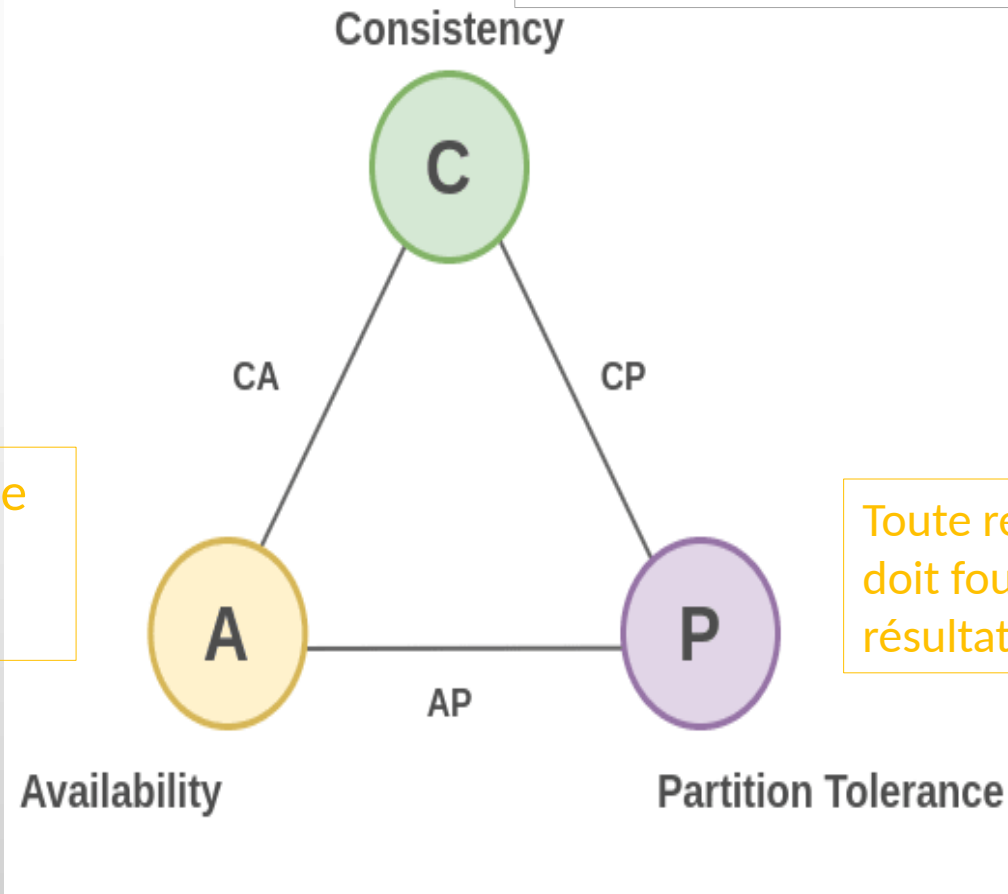
Théorème CAP

En 2000, Eric Brewer établit les fondements des bases de données :

A définir

Un seul état visible dans toutes les conditions

Disponible continue des données sauf système down

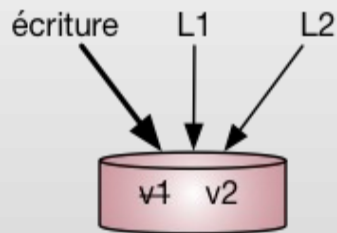


Toute requêtes doit fournir un résultat correct

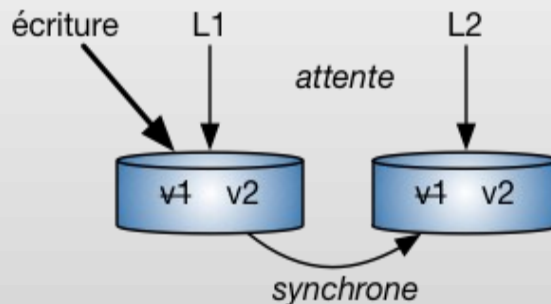
Théorème CAP

Dans toute base de données (distribuée), vous ne pouvez respecter au plus que 2 propriétés parmi la *cohérence*, la *disponibilité* et la *distribution*.

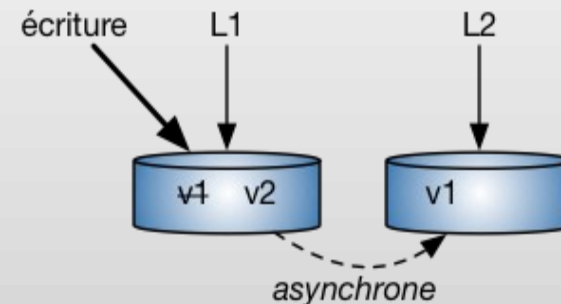
CA
Cohérence + Disponibilité



CP
Cohérence + Distribution



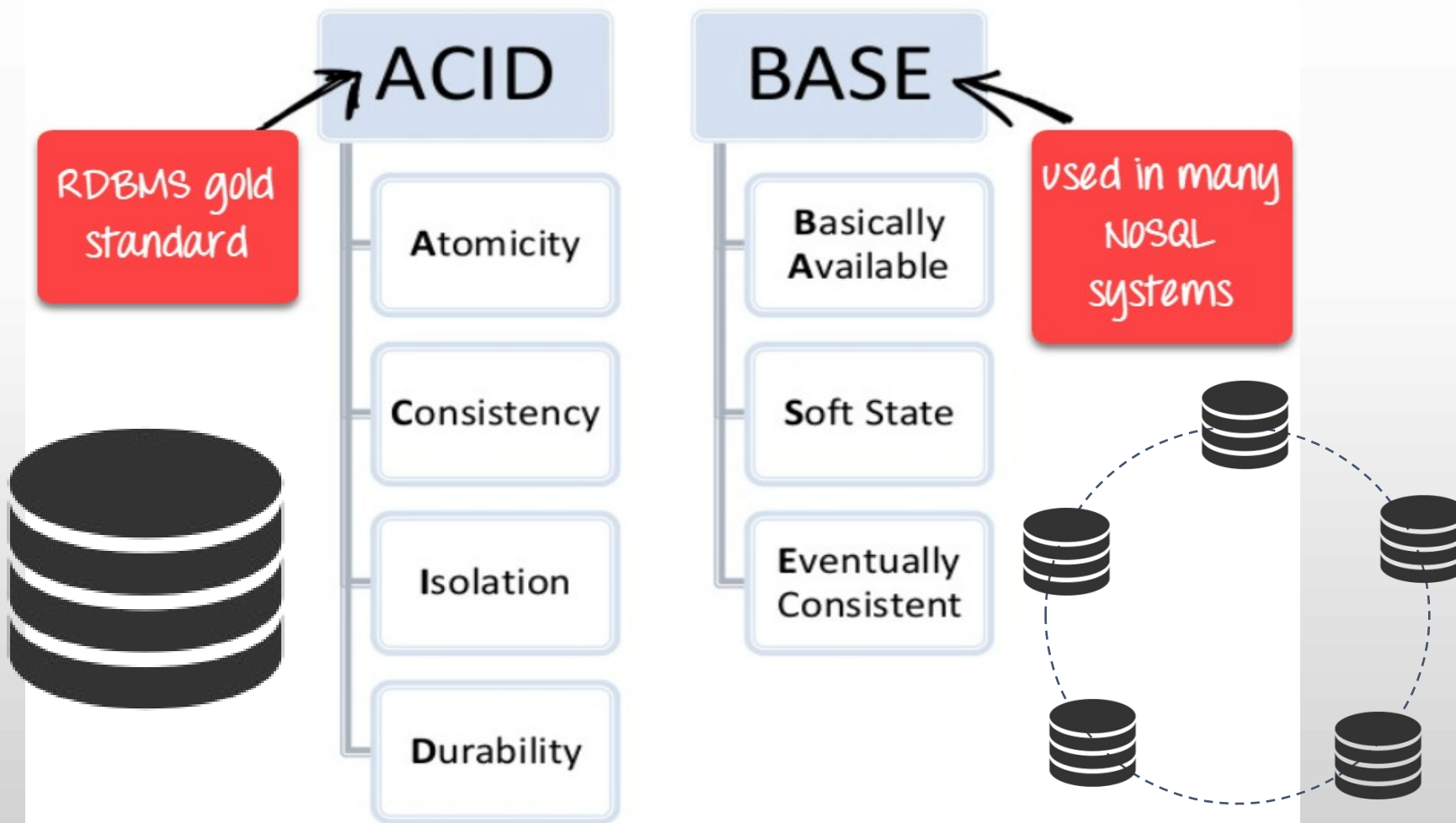
AP
Disponibilité + Distribution



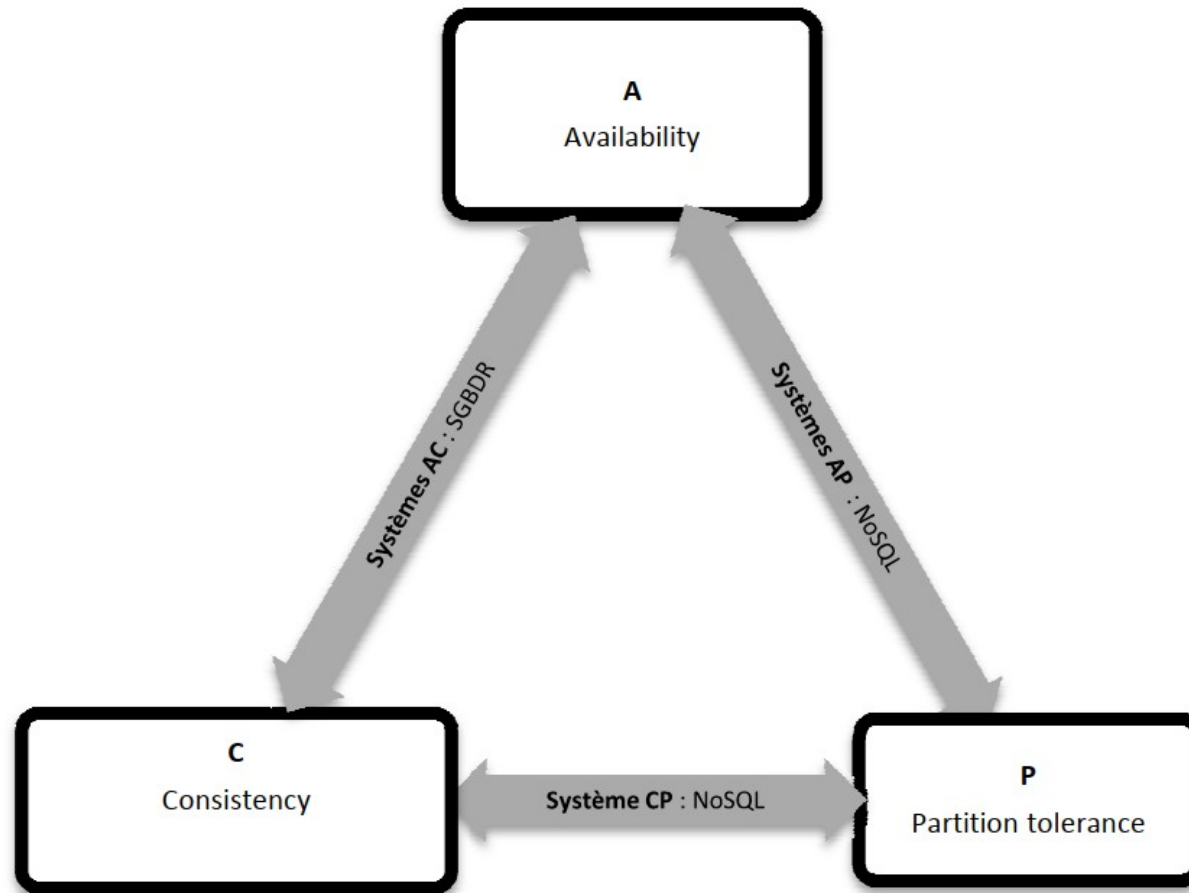


Propriétés BASE

Forte disponibilité / Cohérence relâchée
Rapide / Etat précaire



BASE dans le CAP

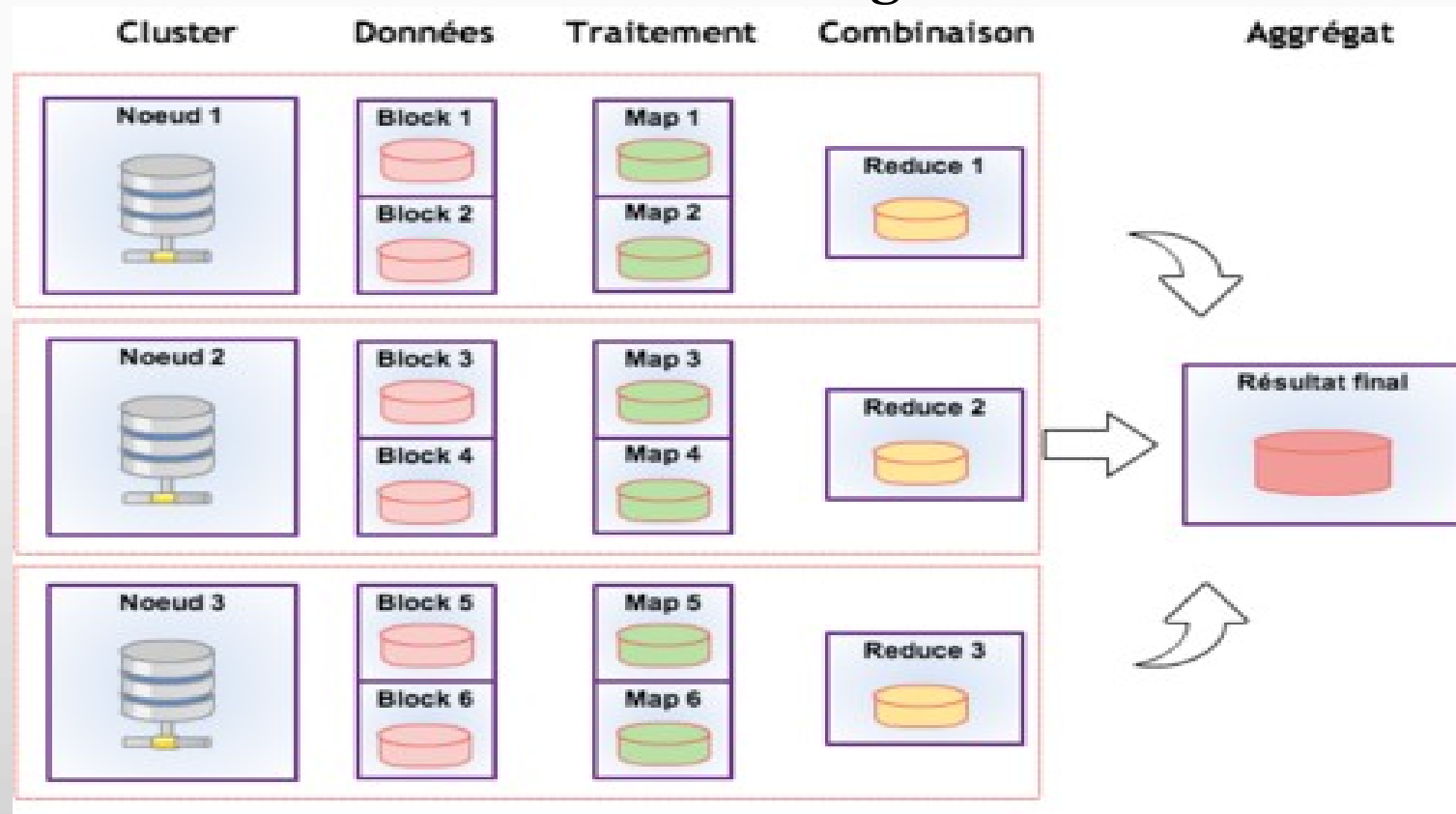




Traitement Distribué

Nouveau Paradigme / Calcul Parallèle

Map Reduce/ Rapidité / Shared
Nothing



Utilisation du NoSQL



CAS D'USAGE

- Gestion de larges pans de données non transactionnelles comme des logs.
- Système en temps réel ou la latence est vitale, comme des jeux par exemple.
- Représentation d'objets complexes en base de façon simple.
- Archivage continu et accessible en permanence



CAS D'USAGE

- Des systèmes embarqués qui ne souhaitent pas s'encombrer de l'overhead des bases traditionnelles
- Système de détection de fraudes qui compare les transactions en cours à des patterns connus
- Application web muni d'une base de données online/offline avec synchronisation à la connexion

MERCI