



INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
HAUTS-DE-FRANCE

Projet Final

Desticourt Matt - Choquet Mathias
Bouchendhomme Alexis - Delhoute Clara

FISA 4 INFO

Organisation de la Base de Données.....	2
Architecture de notre projet.....	3

Organisation de la Base de Données

Initialement, nous avons envisagé de réaliser une fragmentation horizontale (sharding) basée sur l'ID des groupes. Cette approche aurait impliqué la création d'un serveur dédié pour chaque groupe, gérant à la fois le matériel et les utilisateurs associés. Dans chaque document, une référence à un **idGroupe** aurait permis d'assurer un sharding par groupe. Cependant, ce système nécessitait la mise en place d'un processus automatisé capable de déployer et configurer dynamiquement un nouveau conteneur Docker à chaque création de groupe. Par manque de temps pour implémenter une solution de ce type, nous avons opté pour une autre approche de fragmentation.

Nous avons choisi de réaliser une fragmentation horizontale basée sur le **code postal** des données. La distribution des données est la suivante :

- Le **premier serveur** stocke toutes les données associées à des codes postaux inférieurs ou égaux à 50 000.
- Le **deuxième serveur** stocke toutes les données associées à des codes postaux strictement supérieurs à 50 000.

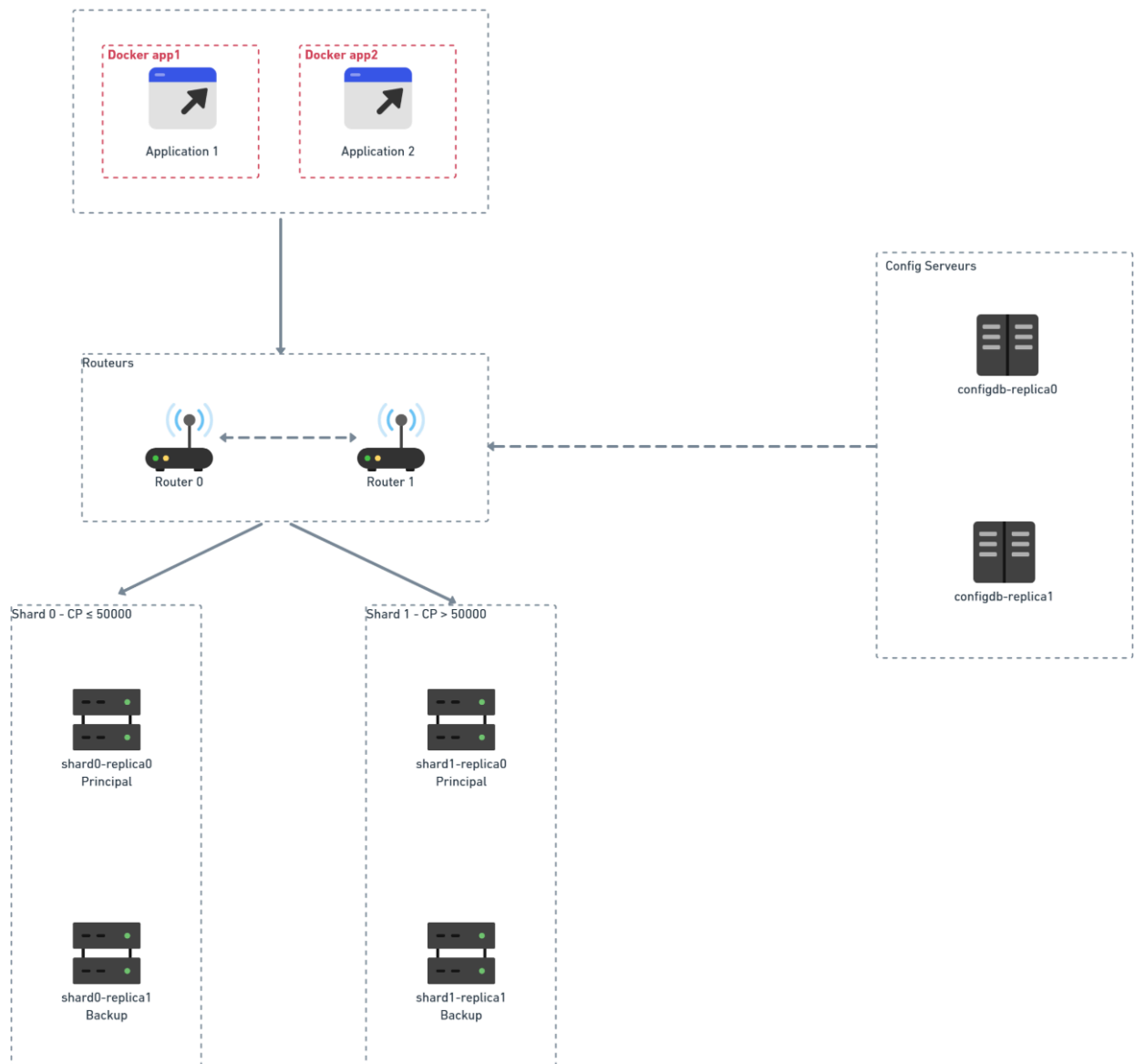
De cette manière, nos prédicats respectent les règles de minimalité et de complétude.

Chaque **shard** dispose d'un **replica** qui réplique l'intégralité des données du shard principal. Cela garantit une haute disponibilité et une redondance des données en cas de panne.

Pour assurer l'interopérabilité et l'accès uniforme aux données, nous avons configuré **deux routeurs MongoDB** (mongos). Ces routeurs servent d'interfaces entre les clients et les shards, en dirigeant les requêtes vers le shard approprié en fonction de la clé de fragmentation (le code postal). Cette configuration permet de maintenir la disponibilité et la performance du système.

Enfin, la base de données repose sur **deux serveurs de configuration** (config servers) qui stockent les métadonnées nécessaires à la gestion des shards et des routes. Ces serveurs garantissent la cohérence et l'intégrité des données réparties entre les différents shards.

Architecture de notre projet



Composition de nos tables

membres	
_id	
nom	
prenom	
adresse	
ville	
codePostal	
email	
typeMembre	CLIENT ou ACTIF
password	
confirmPassword	
_class	org.example.classes.Membre

commandes	
_id	
idMembre	
idVendeur	
date	
liste	
prixTotal	
_class	org.example.classes.Commande

materiels	
_id	
numeroSerie	
modele	
type	
marque	
prix	
_class	org.example.classes.Materiel

groupes	
_id	
numero	
nom	
ville	
codePostal	
_class	org.example.classes.Groupe