

TP7 Event-driven Micro-Service Pattern : CQRS & Event Sourcing

Soit une application micro-Service (voir **figure 1**) qui est constituée de deux micro-services *Ms-Library* et *Ms-Editeur*. On souhaite ajouter une fonctionnalité qui affiche pour chaque Chambre son statut → *Chambre-Status*(*idCh*, *idH*, *nbReservation*).

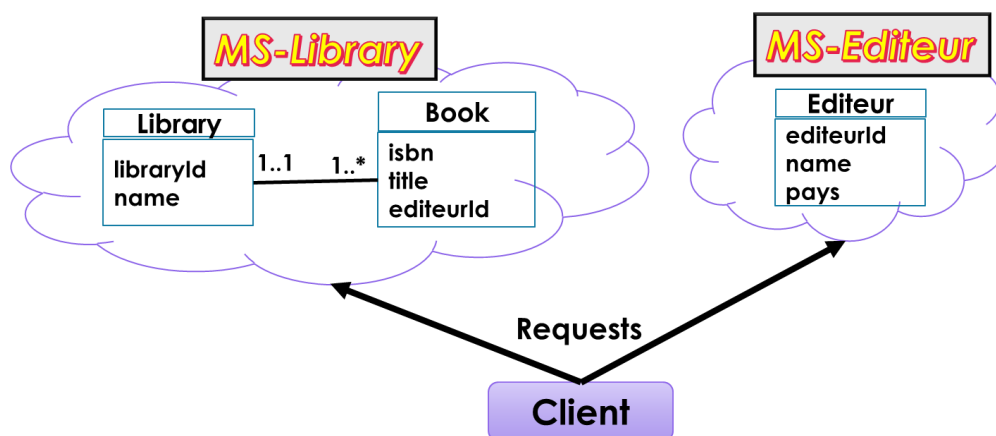


Figure1. Application Microservice sans CQRS

1. Core-Api :

- Commads** : LibraryCreationCommand, AddBookCommand, RemoveBookCommand, EditeurCreationCommand
- Events** : LibraryCreatedEvent, BookAddedEvent, BookRemovedEvent, EditeurCreatedEvent
- DTO** : LibraryDTO, BookDTO, EditeurDTO

2. Lab1 : Separate Command and Query Services (Voir Figure 2)

- Ms-Command-Library**: Ce micro service gère un **Aggregate Root** *LibraryAggregate* (without state storage).
 - LibraryAggregate*: **Not Null** [LibraryId, name, isbn], **Unique** [For each Library the isbn must be unique].
- Ms-Query-Library**: Ce microservice gère la partie **Query** en mette à jour deux entités : **Library** et **Book** en se basant sur les évènements envoyés par la partie **Command** : **Ms-Command-Library**.
 - Entity, Repository, Projection
 - BDD: jdbc:mysql://localhost:3306/Query-Bdd

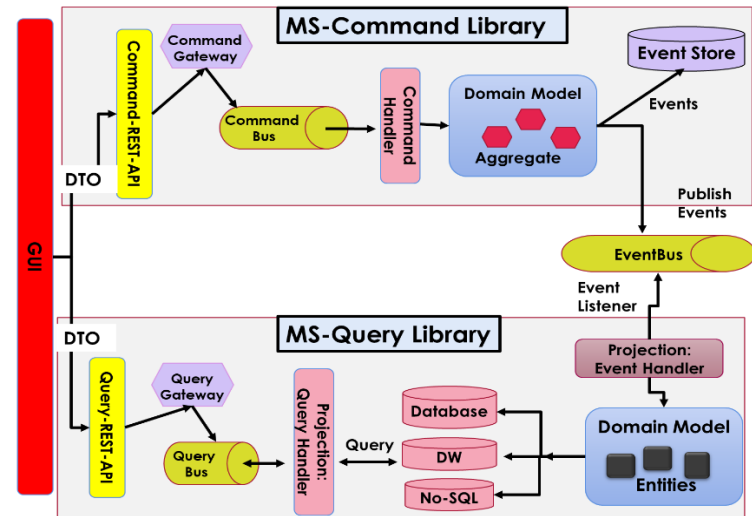
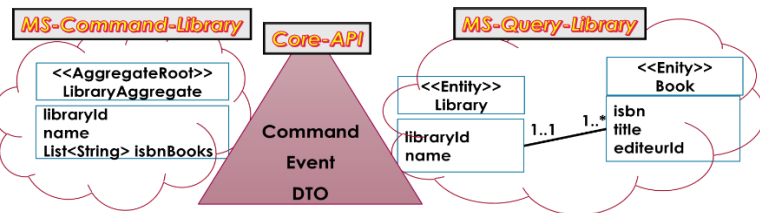


Figure2. Lab1: Separate Command & Query Services

3. Lab2 : Materialize join & State stored aggregate (Voir Figure 3)

- Ms-Command-Editeur:** Ce micro service gère un *state Stored Aggregate* (i.e., Entity/Aggregate) *EditeurAggregate*
 - State Stored Aggregate, Repository, Projection et Command Controller.*
 - BDD: `jdbc:mysql://localhost:3306/Editeur-bdd`
- Ms-Command-Library-Book:** Ce micro service gère deux Aggregates: un **Aggregate Root Library**, et un **Aggregate Member Book**.
 - Library & Book: Not Null [LibraryId, name, isbn, Title], Unique [For each Library the isbn must be unique].*
- Ms-Query-Library:** Ce micro service gère la **materialized-Join EditeurJoinBook** en mettant à jour cette dernière en fonction des événements publiés (*EditeurCreatedEvent*, *BookAddedEvent* et *BookRemoveEvent*).
 - Entity, Repository, Projection*
 - BDD: `jdbc:mysql://localhost:3306/Join-Query-Bdd`

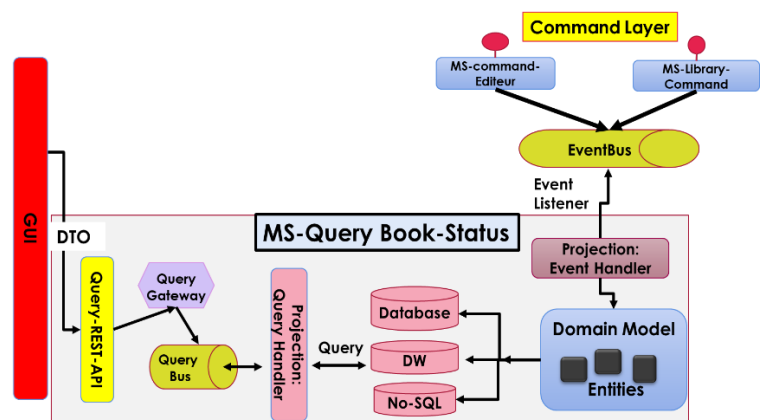
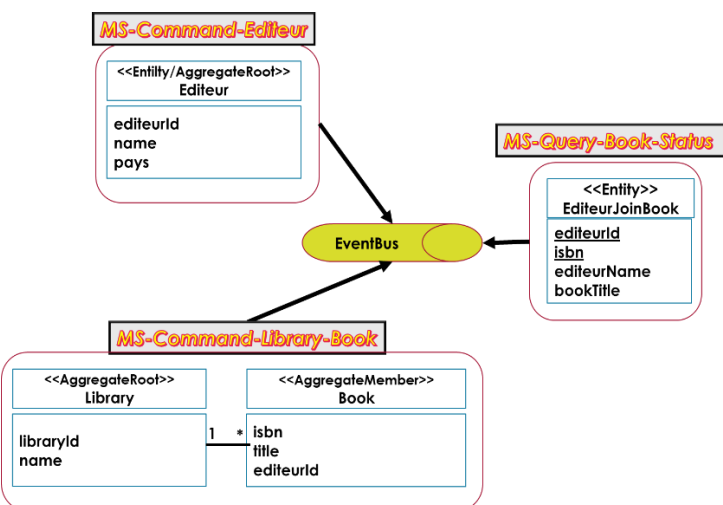


Figure3. Lab2: Materialize join & State stored aggregate

4. Lab3 : State stored & Materialized Join with MongoDB (Voir Figure 4)

- a. **Ms-Command-Library-Book-Stored:** Ce micro service gère deux state-stored Aggregates: un Entity/Aggregate-Root *Library*, et un Entity/Aggregate Member *Book*.
- Library & Book: Not Null [LibraryId, name, isbn, Title], Unique [For each Library the isbn must be unique].*
 - State Stored Aggregate, Repository, Projection et Command Controller.*
 - BDD: jdbc:mysql://localhost:3306/tp-cqrs-stored*
- b. **Ms-Query-Join-Library-Book:** Ce micro service gère la **materialized-Join** *JoinTable* en mettant à jour cette dernière en fonction des événements publiés (*LibraryCreatedEvent*, *BookAddedEvent* et *BookRemoveEvent*).
- MongoDb Document, MongoRepository, Projection*
 - BDD: mongodb://localhost:27017 & database=join-library-book*

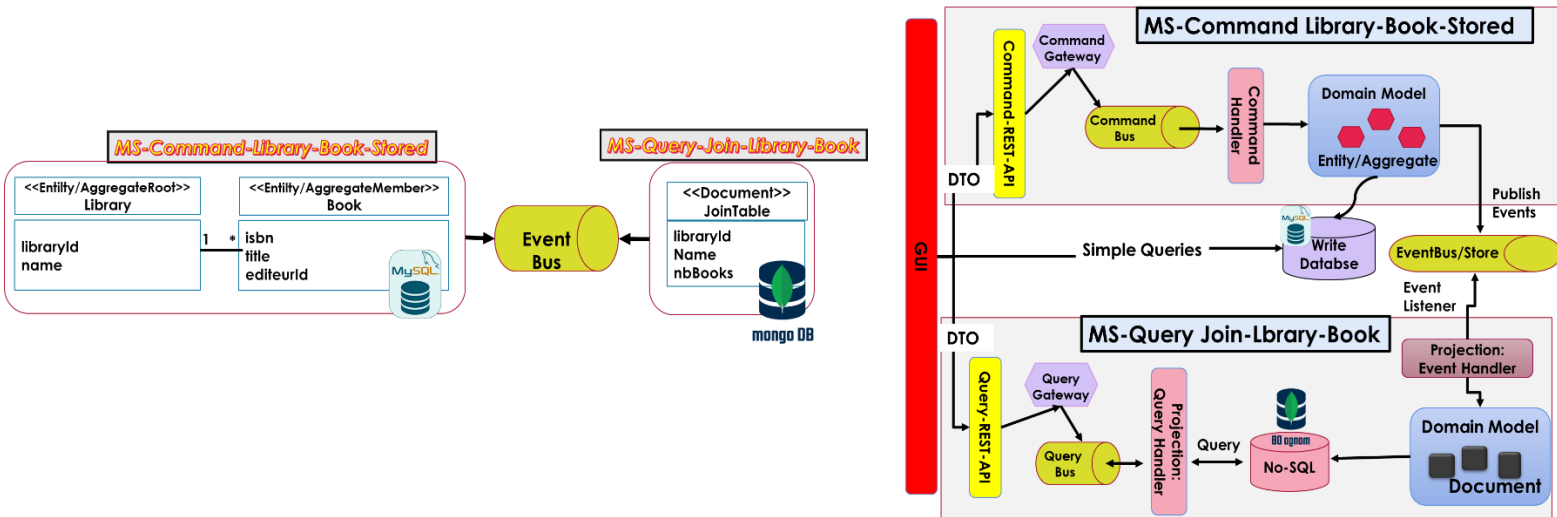


Figure4. Lab3: State stored & Materialized Join with MongoDB & Materialized Join with MongoDB