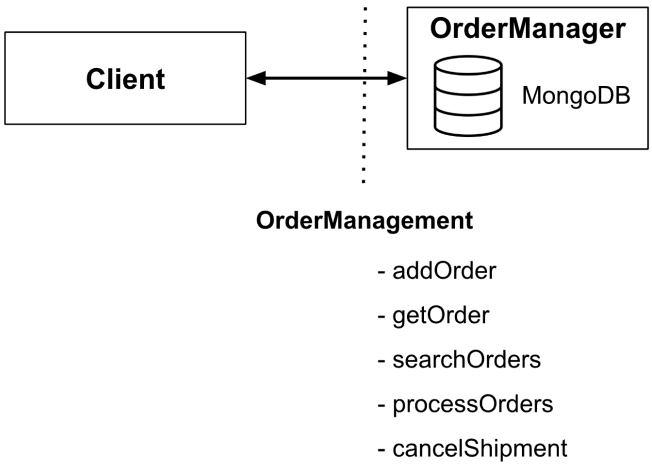
Università degli Studi di Napoli Federico II
Advanced Computer Programming

Il candidato realizzi un'applicazione Python basata su gRPC e MongoDB per la gestione di ordini e spedizioni. Il sistema è composto da 2 entità, come illustrato in figura:



Un ordine è descritto da un ***id*** (stringa), una lista di ***items*** (stringa), una ***descrizione*** dell'ordine (stringa), il ***prezzo*** totale (float), e la ***destinazione*** (stringa). Una spedizione è descritta invece da un ***id*** (stringa), da uno ***stato*** (stringa, può essere *PENDING* o *PROCESSED*), e da una lista di ordini.

1. **OrderManager** offre i servizi specificati dall'interfaccia OrderManagement, la quale è descritta dal file `OrderManagement.proto` fornito, che comprende i metodi:
   - **rpc addOrder(Order) returns (StringMessage):** permette l'aggiunta di un ordine presso il Manager. Il Manager riceve un ordine (*Order*) e genera un *ID univoco* per quell'ordine. Tale ID viene inserito nel campo *id* dell'ordine, il quale è salvato come document in una collection MongoDB denominata *orders*. La funzione restituisce come risposta (*StringMessage*) l'*id* generato per l'ordine.
   - **rpc getOrder(StringMessage) returns (Order):** permette di recuperare un ordine dato il suo *id*. Il Manager riceve un *id* (*String Message*) che sfrutta per ricercare l'ordine all'interno della collection MongoDB *orders*. Se l'ordine esiste, la risposta (*Order*) conterrà l'ordine trovato, altrimenti viene ritornato un ordine vuoto.
   - **rpc searchOrders(StringMessage) returns (stream Order):** permette di cercare un prodotto tra gli ordini gestiti dal Manager. Il Manager riceve il nome di un *item* (*StringMessage*) e cerca un'occorrenza di tale item all'interno della collection MongoDB *orders*. La funzione ritorna uno stream di ordini (*Order*) che contengono il nome dell'*item* ricevuto.
   - **rpc processOrders(stream Order) returns (stream CombinedShipment):** permette di processare degli ordini preparando le spedizioni per destinazione. Il Manager riceve uno stream di ordini (*stream Order*), salva gli ordini nella collection MongoDB *orders* e genera una spedizione per ogni destinazione differente. Ogni spedizione è caratterizzata da un *id* (da generare) e dalla lista degli ordini con la stessa destinazione. La funzione salva ogni spedizione in una collection MondoDB denominata *shipments*, e ritorna uno stream di spedizioni (*stream CombinedShipment*).
   - **rpc cancelShipment(StringMessage) returns (StringMessage)**: permette di cancellare una spedizione. Il Manager riceve l'*id* (*StringMessage*) della spedizione da cancellare, e procede alla cancellazione della spedizione dalla collection MongoDB *shipments*. La funzione restituisce il messaggio *DELETED* (*StringMessage*) per cofermare l'operazione.
2. **Client**: genera 5 ordini, e ne richiede l'aggiunta al Manager attraverso *addOrder*. Per ogni invocazione di *addOrder*, il client stampa a video l'id ottenuto, e verifica l'aggiunta dell'ordine attraverso la *getOrder*, stampando a video i dati dell'ordine.

Successivamente, il client richiede la ricerca di un particolare item aggiunto ad uno degli ordini precedentemente inviati al Manager, stampando la lista di ordini che contiene quell'item.
Il Client crea poi 4 ordini, due dei quali hanno la stessa destinazione, e richiede al Manager la creazione delle relative spedizioni attraverso la *processOrders*, stampando a video le informazioni sulle spedizioni.
Infine, il Client richiede la cancellazione di una delle spedizioni attraverso la *cancelShipment*, stampando a video la risposta ottenuta.