

Figure 1.Message Broker Diagram

The **MessageBroker** runs in a separate dockerized application. It contains six actors and two lists (Figure 1). On the top of the actors hierarchy, the message broker has the **ActorsSupervisor** (Figure 2) - the uppermost parent of all the other actors. The rest of the hierarchy: **MessageReceiving** actor, which is a queue that does exactly what its name says - receives messages from the producers. Whenever a message needs to be sent to a consumer, the **MessageReceiving** actor passes it to the **MessagesManager** actor, which will further send a message to the **MessagesConfirmer**, containing info about the message's ID and the consumer that have to receive it. Also, the initial message is passed to the **MsgSender** actor, which will send the actual message from the producer to the consumer. Whenever a consumer confirms the receipt (through a message stating the ID of the message it has accepted), the **ConfirmationReceiver** dedicated to that consumer will pass on the accept message to the **MessageConfirmer** deletes the corresponding record from its list of messages that wait for confirmation from its dedicated consumer. The **ConnectionActor** watches for new clients connecting to the broker and initiates / creates the actors **ConfirmationReceiver**, **MessagesConfirmer** and **MsgSender** (Figure 2) - one of each for each consumer. Also, each producer has a dedicated **MessageReceiving** actor. Only the **ConnectionActor** and the **MessageManager** are shared actors. Whenever a MessageBroker actors system is instantiated, the actors' supervisor is instantiated under the hood by default, and it takes care of the actors in the system.

An actor can kill itself in case it receives a poison pill or it malfunctions, after notifying its parent. If the parent actor - the one who created the child - can't handle the child's failure, it will also kill itself and the problem is passed on to the next parent, or even up to the supervisor.

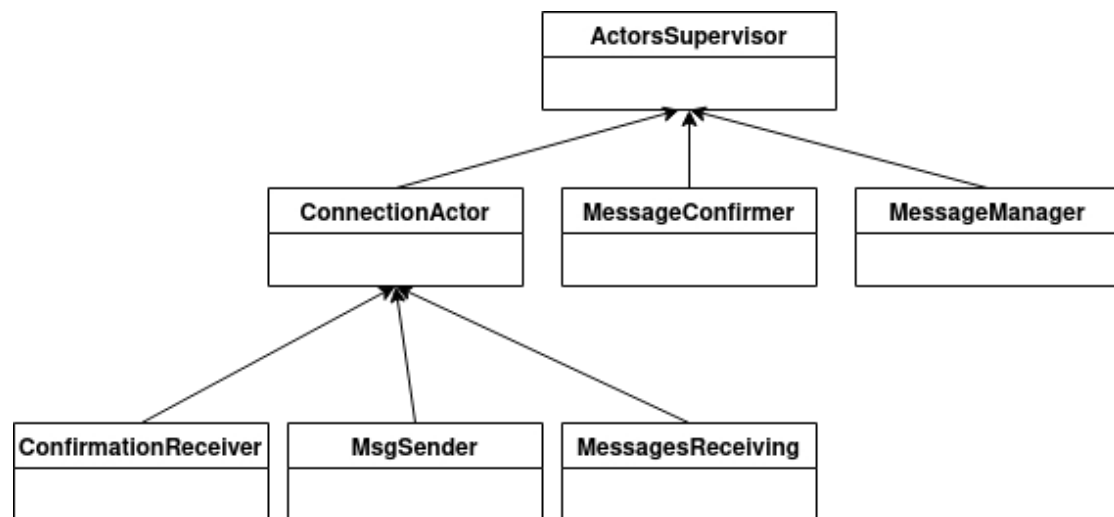


Figure 2.Actors Hierarchy Diagram