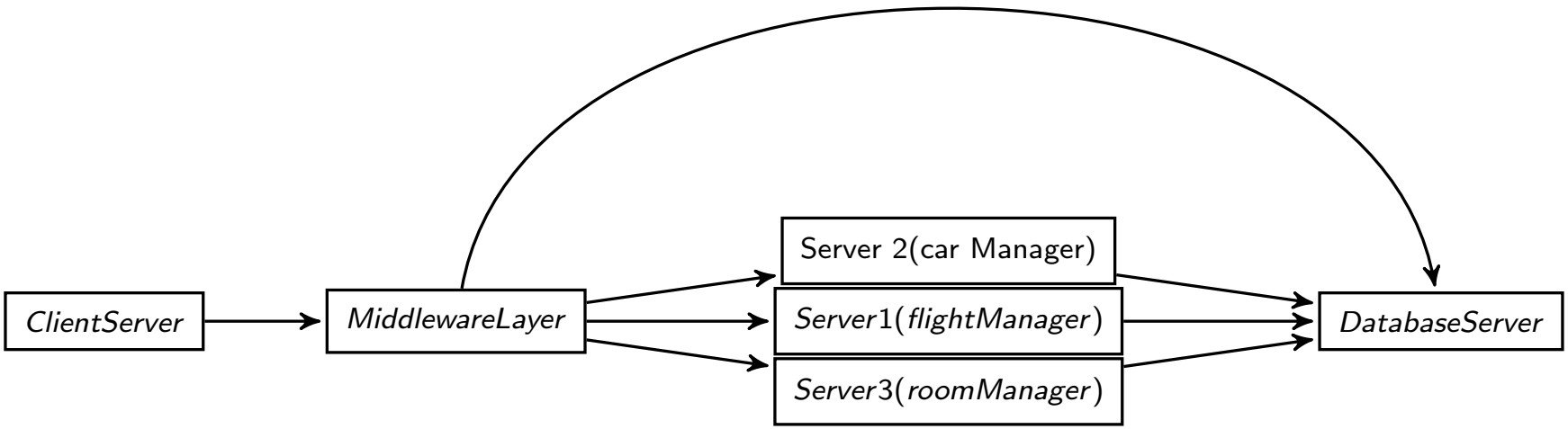
Webservices Architecture

The architecture of the webservices distributed system is as follows



(1)

Sockets Implementration:

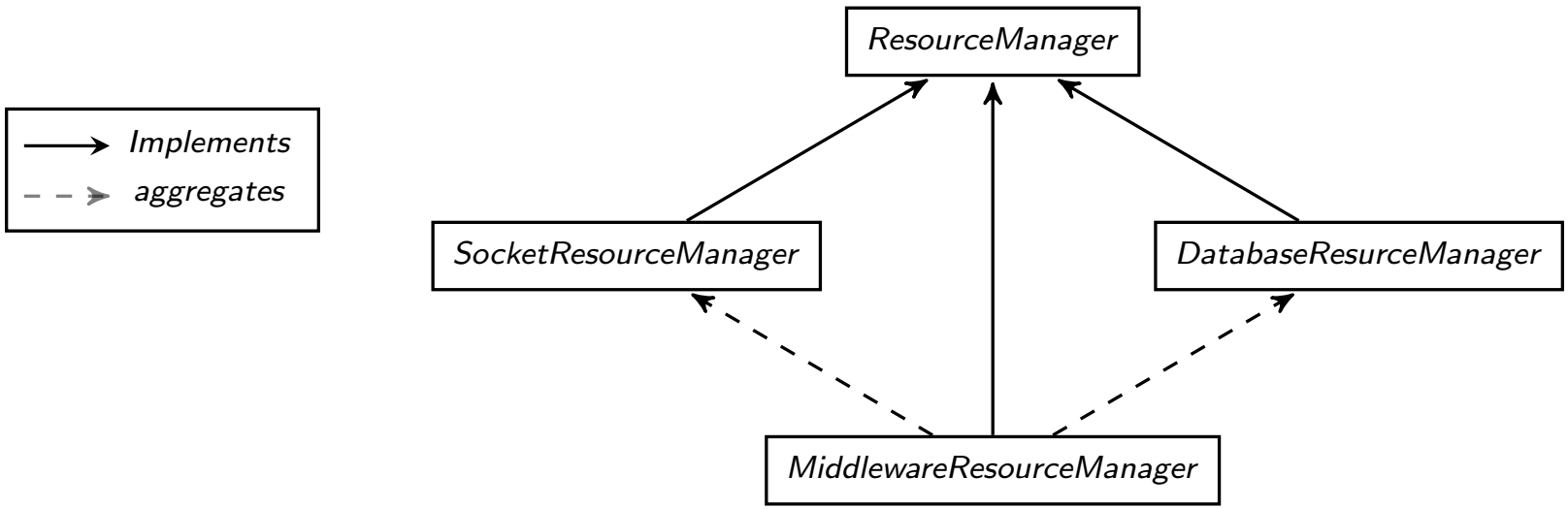
The overaching architecture of the websockets code is identical to the structure shown in (1). As such the description will be focused on the implementations of webservices on top of websockets.
We begin by creating an executor service and oppening a websocket. The server then listens on the socket of all incomming connections, once a connection is opened it is accpted and packaged into a request context which contains a RequestManager and the socket connection. The request manager implements runnable and is passed to the executor service which assigns the connection a thread and invokes Request Manager's run. The executor service guarantees that connections are run asynchronously and act as a threadpool.

Sockets Implementation:

The RequestManager's run fetches the socket from the request context, upon receiving an input stream we use Java builtin serialization to pass our own request and response objects, this allows us to gracefully handle exceptions in a network transparent manner. The Resquest Handler of the recipient then uses Java's build in reflection toolset to identify which method should be invoked providing us with a method of remote method invocation through sockets. As per the webservices code database access is handled through DBCP connection pools.

Sockets Implementation

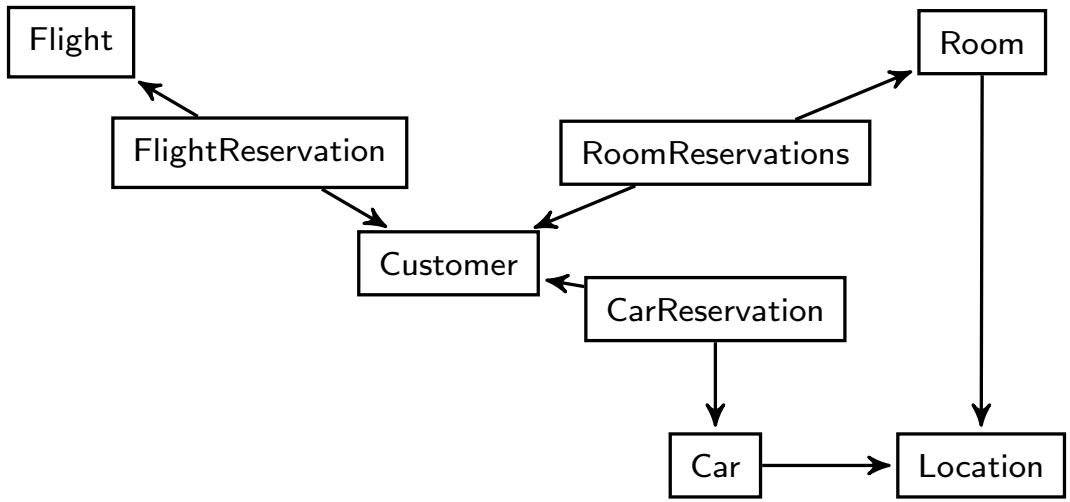
The resource managers thus have the following set of relations.



(2)

Shared Design

Shared design between the webservices and Sockets implementations is restricted to the POSTGRESQL database which is identical in both cases and has the following schema.



(3)