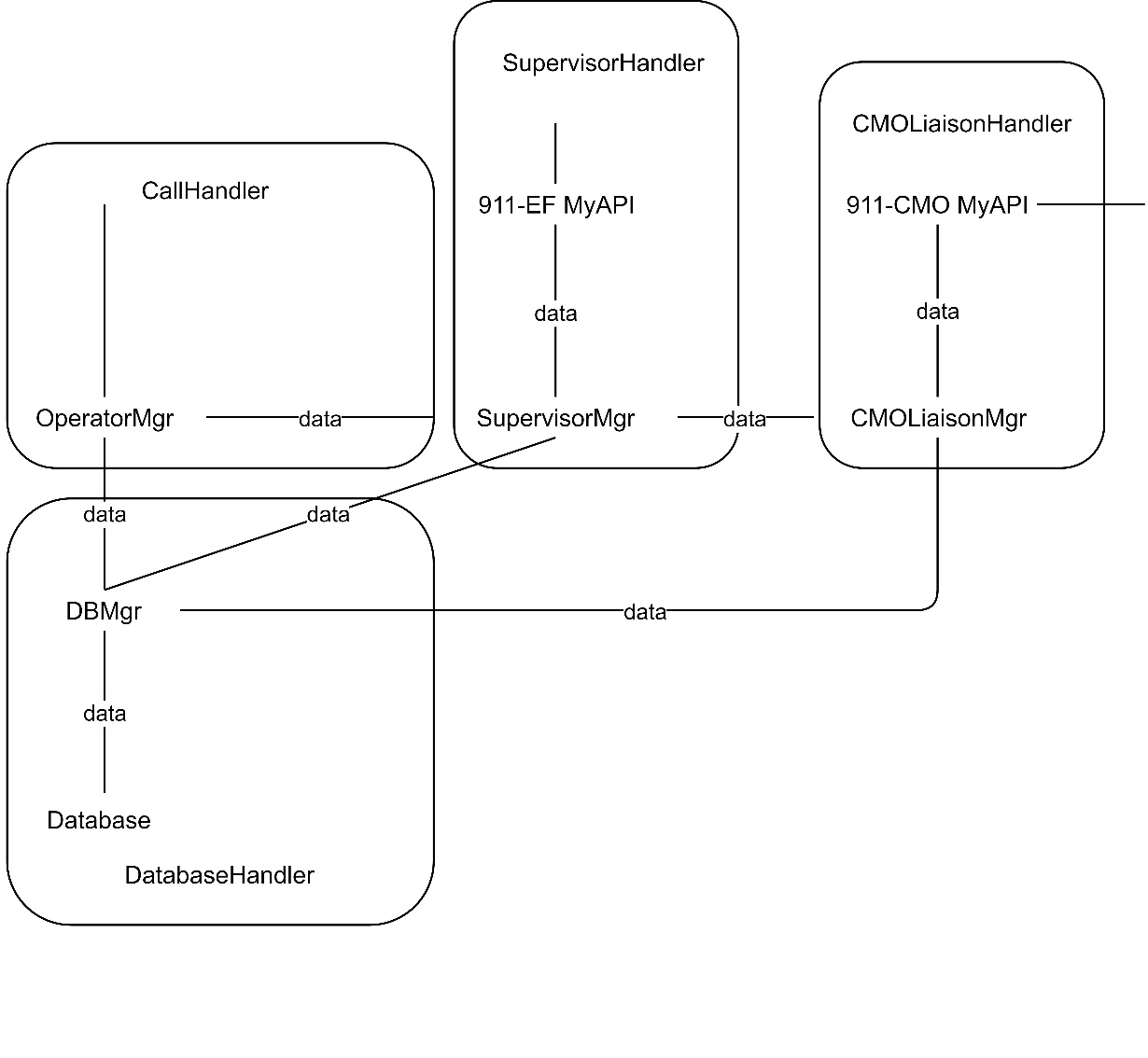
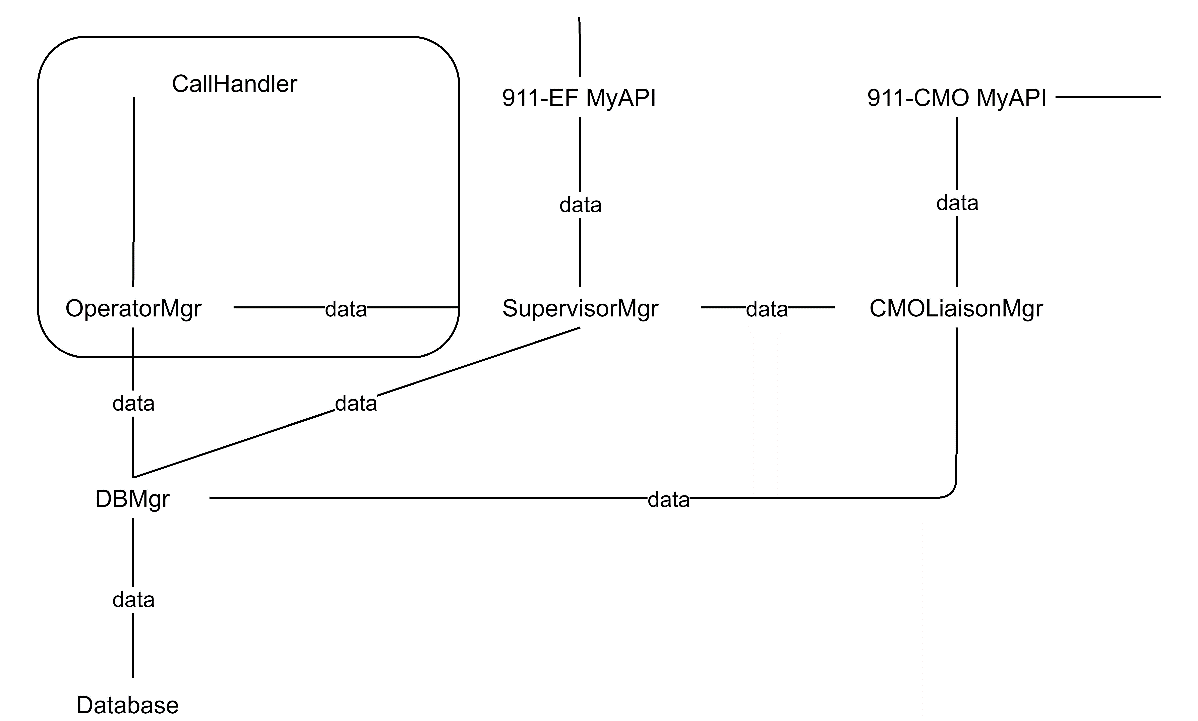
# Initial Architecture Structure with Functions

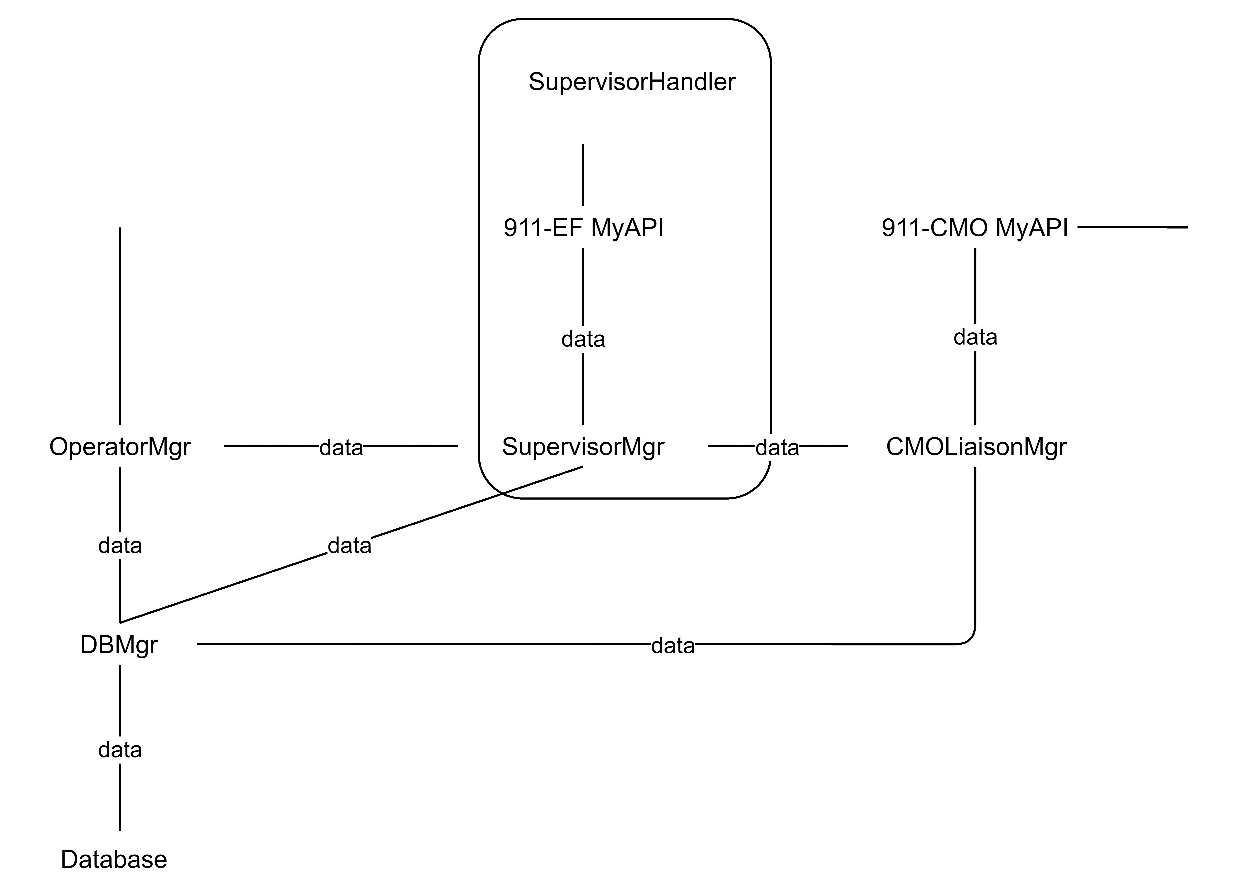
Initial architecture structure with functions is essential for a software designer to understand how each part of the system connects with each other. As a result, initial architecture structure with functions require more details than the usual initial architecture even though the diagrams are derived from it.



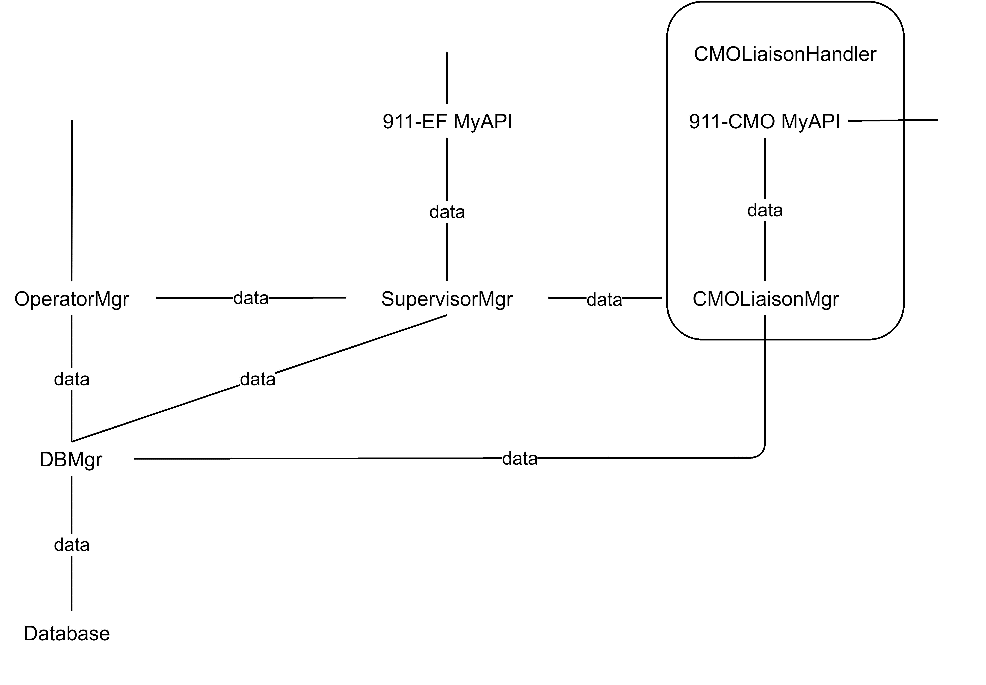
The architecture above contains all 4 components present in the system, namely CallHandler, SupervisorMgr, CMOLiaisonHandler and DatabaseHandler. However, as a user can only run one role at the same time, we need to separate the initial architecture based on the cluster of components as below:



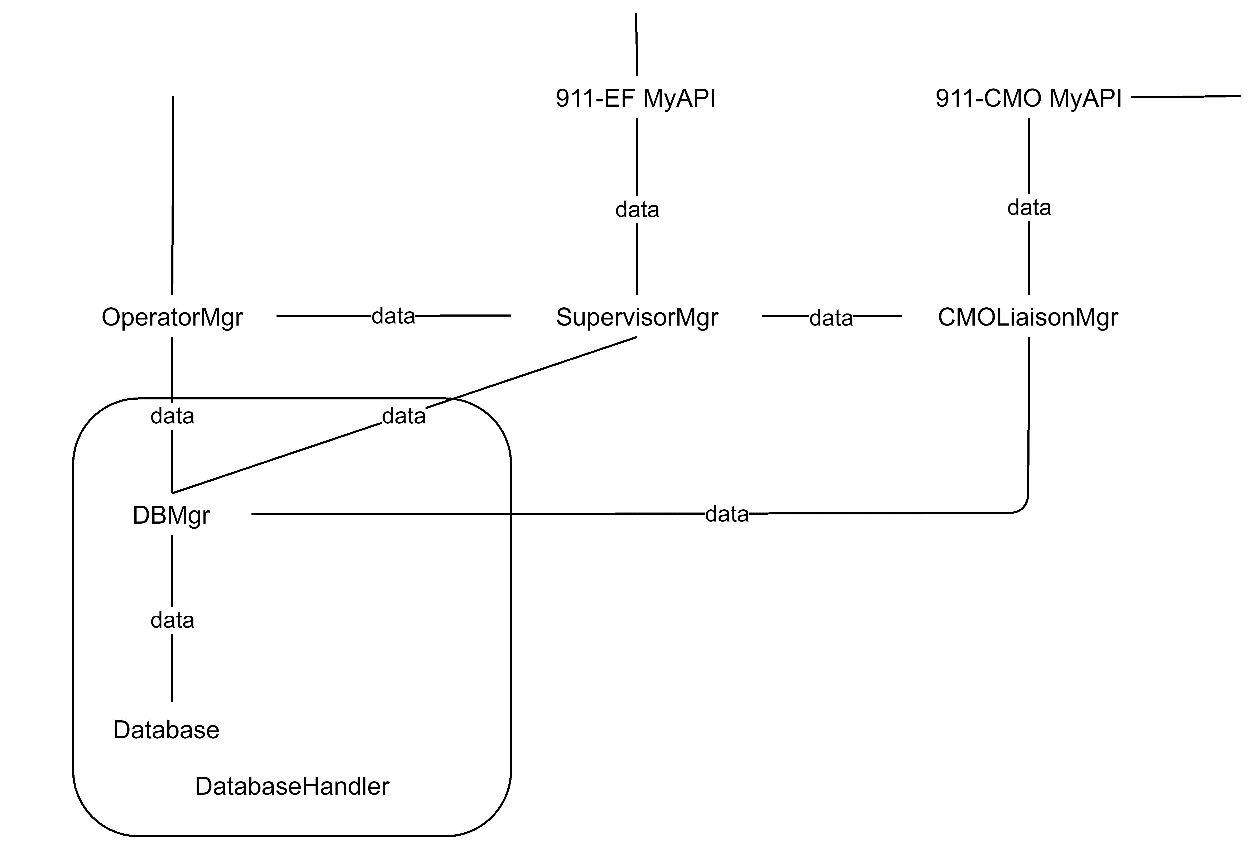
The example above shows how the components in CallHandler relate with others and within itself. The calls are handled by specially trained operators who then entered the call details into the operator manager. During the call, Operator will determine the location of crisis based on the details provided by the caller using Google Map API. Data gathered by the operator will be entered to the database manager and if the operator suspects a crisis the data and request to assess will be sent to the supervisor.



After the supervisor receives the crisis data and the request to assess the report, it then analyses the report using data from the caller and past data retrieved through the database manager. After the supervisor makes the outcome for the assessment, the classifier will send the updated outcome to the database manager for recording. If the reported crisis is Demon level and above, Supervisor Manager will be initiated to send the data through to CMO Liaison Manager else, 911-EF MyAPI will be initiated to send data to EF.



After the CMO liaison Manager receives the crisis data, it then analyses the report. If the reported crisis is indeed Demon level and above, 911-CMO MyAPI will be initiated to send the data through to CMO, else CMO Liaison Manager will send data back to Supervisor Manager



The system must be able to withstand many calls, and therefore also able to cope with the large number of requests to the database. As a result, the database manager controls the data flow between the operator or supervisor and the database, preventing conflicts or overload.