



**School of Computer Science & Engineering**

CZ3003 – Software System Analysis & Design

Project Name: CDA-911

Teammates:

Yiu Hong Sum U1621435G

Lim Yan Jun U1622311B

Chen Zhenni U1622603D

Bryan Lim Kian Hock U1620949L

Deon Tan Jun Wei U1621715F

**Table of Contents**

[1. System Thinking 2](#_Toc511168466)

[2. Architectural Thinking 3](#_Toc511168467)

[3. Software System Composition 4](#_Toc511168468)

[4. Initial Architecture Structure with Functions 5](#_Toc511168469)

[5. Initial architecture behaviour 8](#_Toc511168470)

[6. Imprinted Software Quality 8](#_Toc511168471)

[Integrity 8](#_Toc511168472)

[Flexibility 9](#_Toc511168473)

[Usability 9](#_Toc511168474)

[Correctness 9](#_Toc511168475)

[7. Analysis Modelling 10](#_Toc511168476)

# System Thinking

For the assignment, we belong to the 911 subsystem. To help us explain the requirements of the subsystem and how we should be designing such a subsystem, we had the assistance of an artist illustration. This is so that it will be useful to all the stakeholders as an ideation of the new software develops as it is being designed.

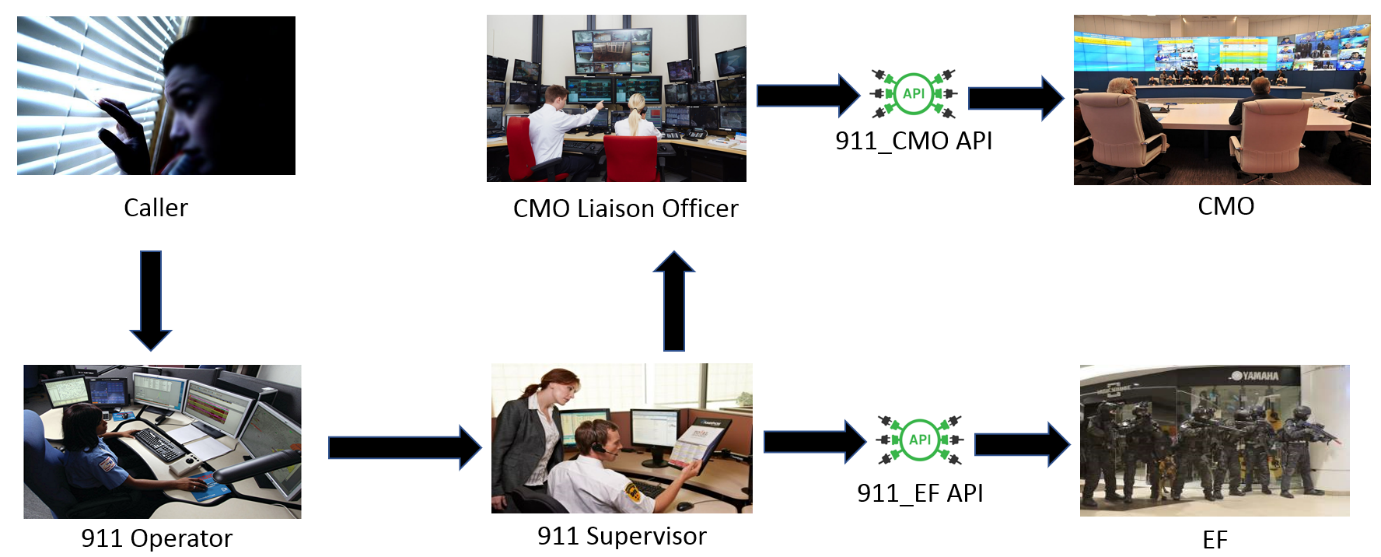


Figure 1. Artist Illustration for 911 subsystem

From the above image, it can be shown that the 911 subsystem consists of different parts such as people, software, hardware, etc. These people include 911 operators, a 911 supervisor in the classification team and a CMO liaison officer. The software may include a web graphical interface as well as a highly secured cloud database whereas the hardware may comprise of computers, monitors, telephones, access passes, etc.

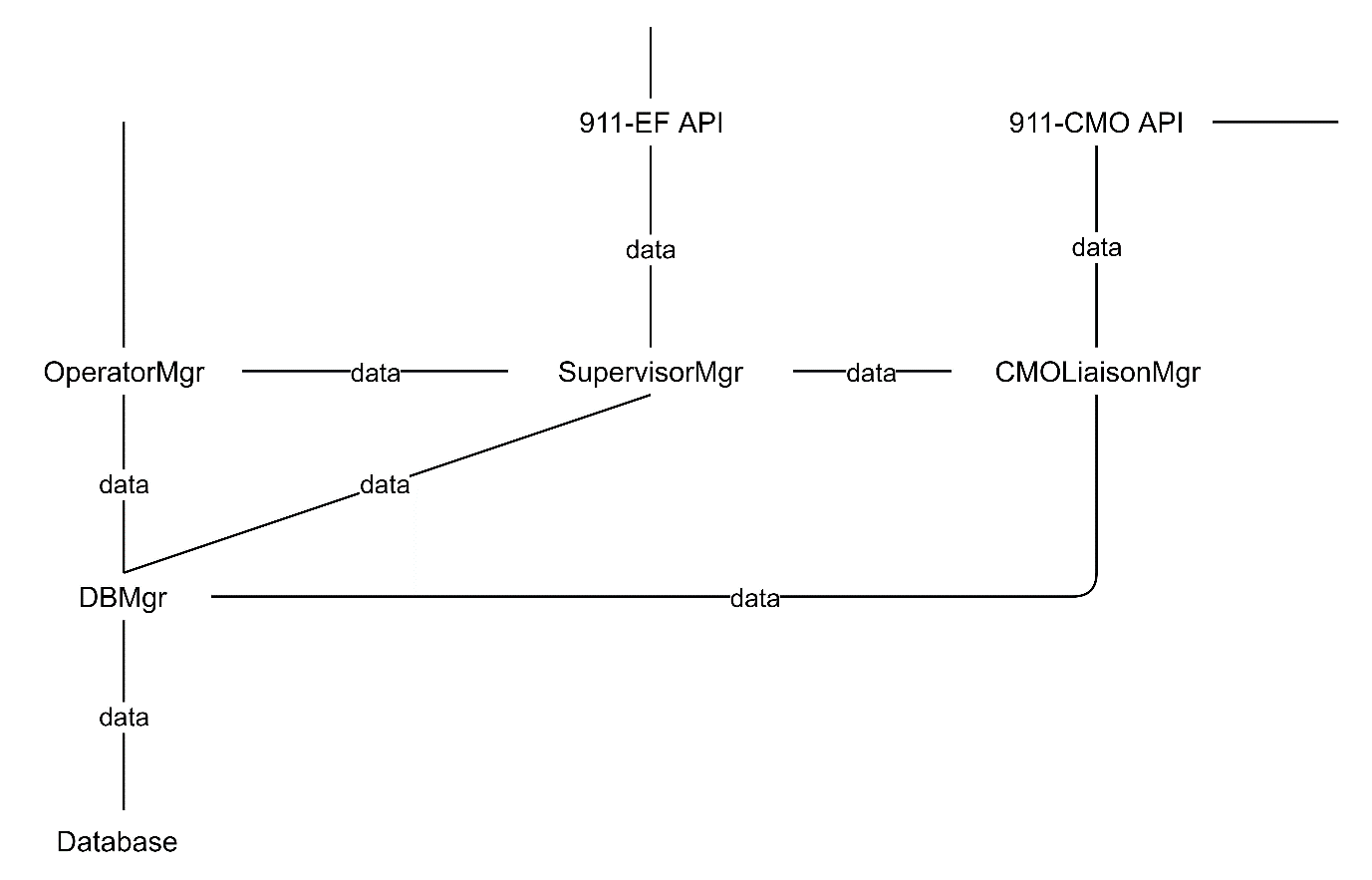
When there is an emergency call received from the caller to the 911 operator, the operator will record all important information pertaining to the call. Subsequently, the operator will store all the information into the database. The 911 supervisor will then classify the class of the incident by accessing the database. After which, should a suspected crisis be found, the CMO liaison officer will be informed to determine the authenticity of the crisis. In the event of a legitimate crisis, the incident report will be forwarded to the Crisis Management Office to handle the situation.

Thus, from the above illustration and after applying system thinking, we are able to design the 911 subsystem.

# Architectural Thinking

The opposite of system thinking, architectural thinking is a way of composing a meaningful solution to a problem given the abstract assemblages. After disassembling the possible subsystems for this 911 project using system thinking, the architectural thinking combines those subsystems into what would be a fully functioning 911 subsystem.

From the artist's illustration shown in Fig. 1, we need to determine whether data and/or control will be passed, to whom they will be passed to make the system work according the requirement. We could see from that figure that the 911 subsystem will include the operator manager, supervisor manager, database manager, database, Google Map API, 911-EF API to connect to EF Subsystem and an 911-CMO API to connect to the CMO system, which is separately build.



911 Subsystem will follow the following crisis classification in terms of impact of crisis area and how serious the crisis is. The classification will be from lowest to highest impact: Monkey, Wolf, Tiger, Demon, Dragon, God. The call handler will be manned by the operators, who will directly liaise with the caller and gather the data from the conversation to be put into the handler. If the data provided show signs of crisis, the supervisor manager will allow supervisors to receive the data from the call handler, analyse the data using current and past reports.

Should the crisis be Monkey level which indicates a prank call, Supervisor will send the data to Operator using Supervisor Manager.  If the crisis level is classified to be either Wolf or Tiger, Supervisor Manager will send the data to EF subsystem using 911-EF API. If crisis level is classified Demon and above, the Supervisor Manager will send the data to CMO Liaison Manager who will then send the data to CMO subsystem using API. The database handles all requests from the call handler and supervisor manager including requests to select, enter, or update data.

# Software System Composition

Software system is a system whose development effort is majorly software in accordance with the Software Development Life Cycle(SDLC) and functionality is commanded and controlled by one or more software applications (product, or developed subsystem), all working to a common mission. Furthermore, the system likely bears the name of the leading software application.  Therefore, in this assignment, the software system of our 911 subsystem can be defined — in EBNF language — as follows:

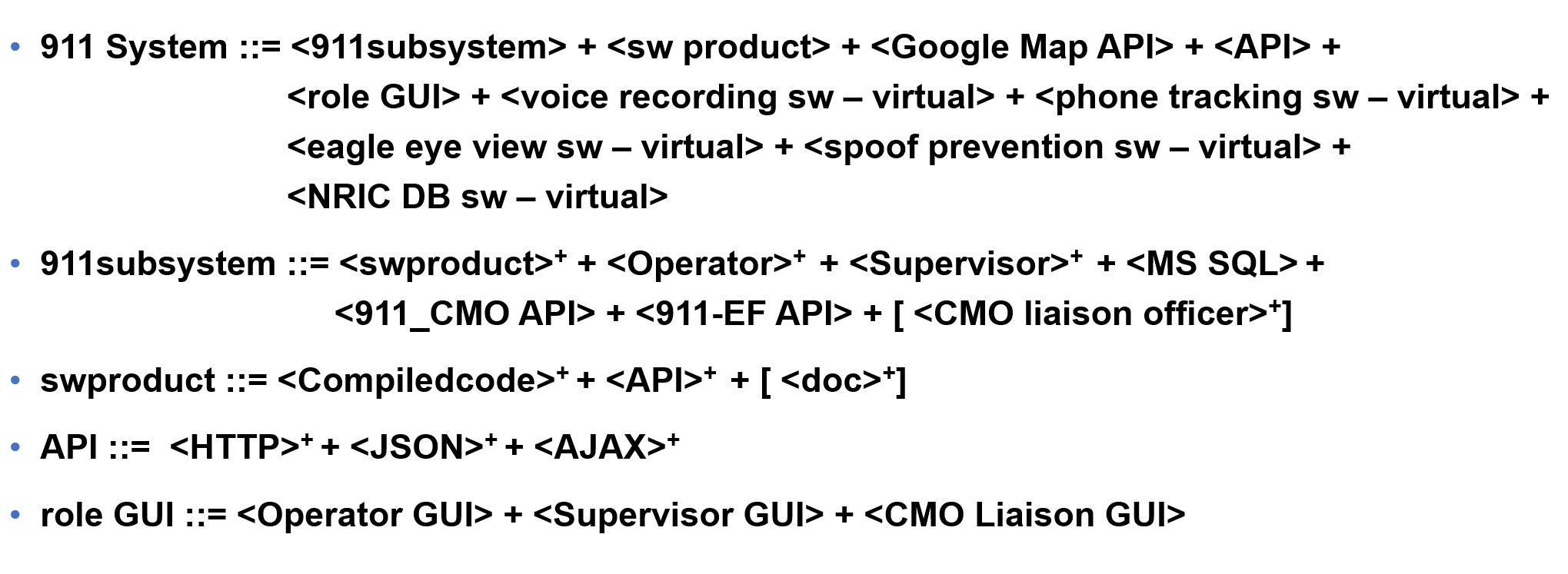


Figure 2. EBNF Statement for 911 subsystem

**Legend:**

::=     means “can be broken down into”

+means multiplicity from 1 to infinity

<X> means mandatory part

[]    means optional part

There are five EBNF statements in the above figure. The interpretation of the first statement is that 911 subsystem is composed of at least one part from each of the following:

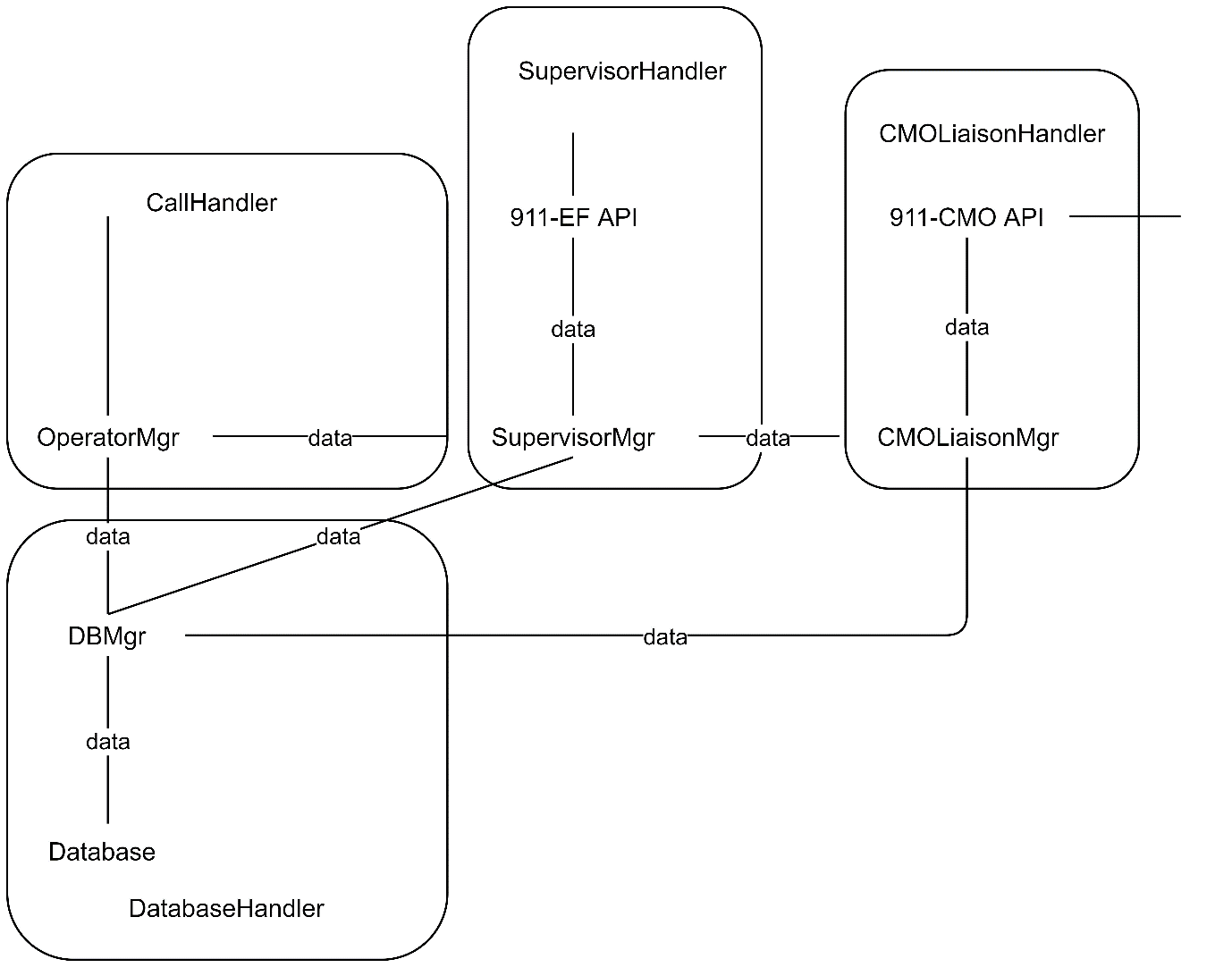
1. software product
2. 911 operator
3. 911 supervisor
4. MS SQL
5. 911-CMO API
6. 911-EF API
7. CMO Liaison Officer

The google map API is used to pinpoint the location of the called based on the address given. Upon classifying the crisis level, detailed report of the crisis will be sent to the EF or CMO accordingly through either 911-EF API or 911-CMO API. The above two API will be named appropriately as API since the data is being passed from 911 subsystem to others (i.e. we are the givers or providers of data).

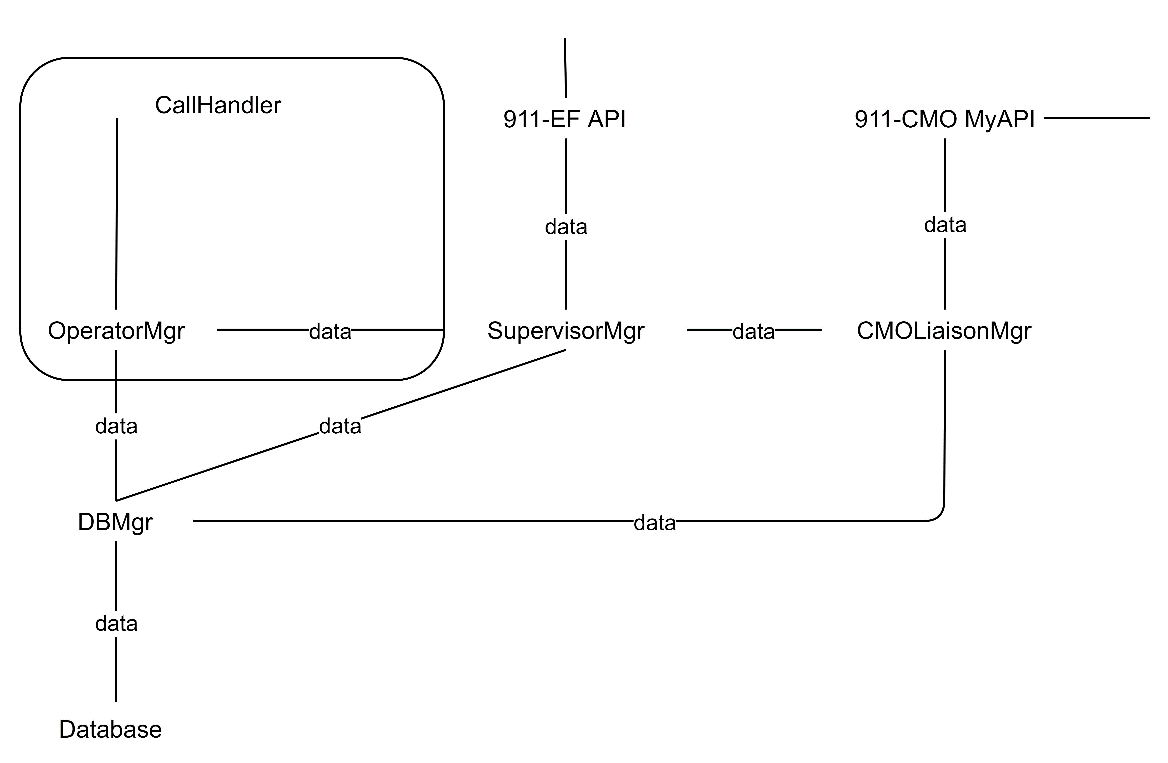
In the second statement, the software product is said to be broken down into the compiled code that make use of relevant APIs for implementing the 911 subsystem software and optional storage of documents or incident reports. Finally, as described by the last statement, APIs that will be used in our software consist of a HTTP API for implementing web graphical user interface and a JSON API for passing required data to the other subsystem in JSON format. AJAX API is used for communication. Therefore, EBNF statements are effective in examining composition of a software system.

# Initial Architecture Structure with Functions

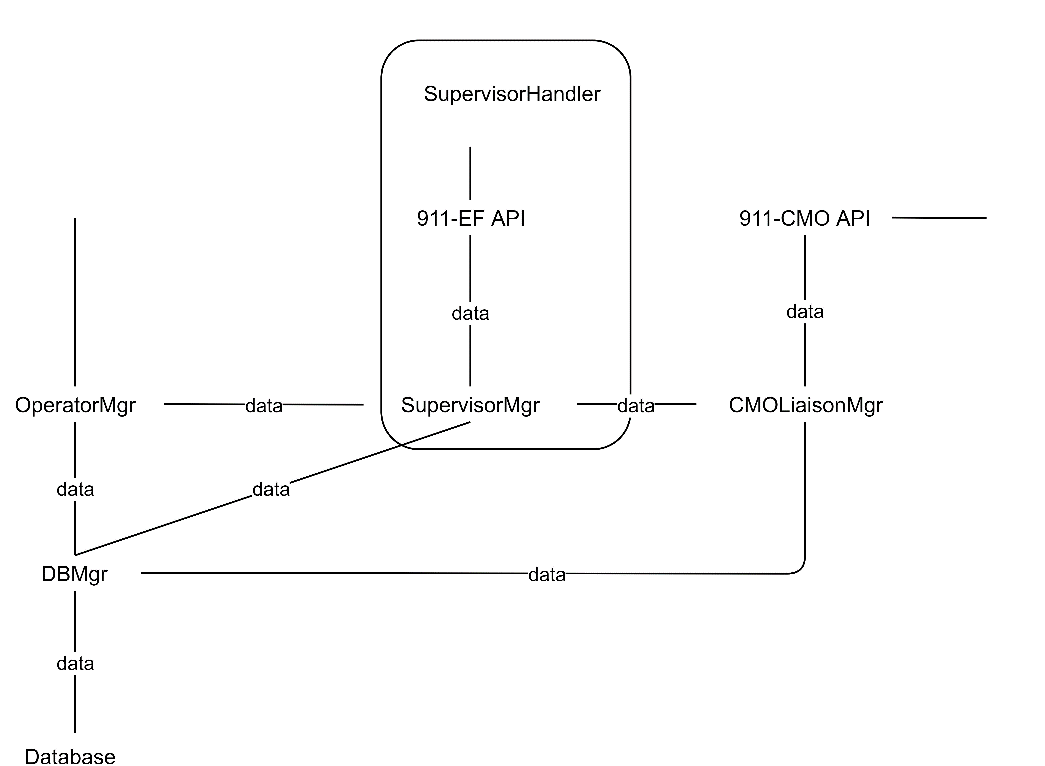
Initial architecture structure with functions is essential for a software designer to understand how each part of the system connects with each other. As a result, initial architecture structure with functions require more details than the usual initial architecture even though the diagrams are derived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



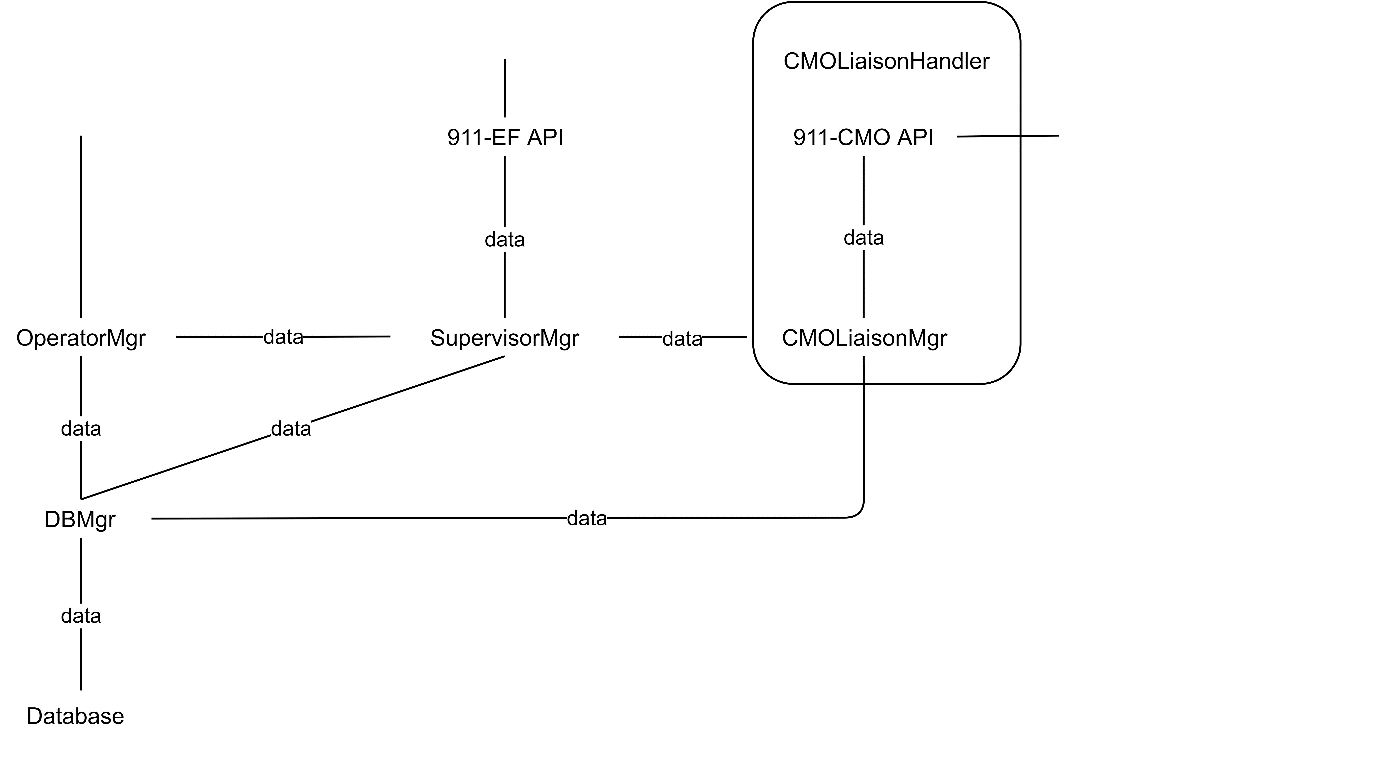
The architecture above contains all 4 components present in the system, namely CallHandler, SupervisorMgr, CMOLiaisonHandler and DatabaseHandler. However, as a user can only run one role at the same time, we need to separate the initial architecture based on the cluster of components as below:



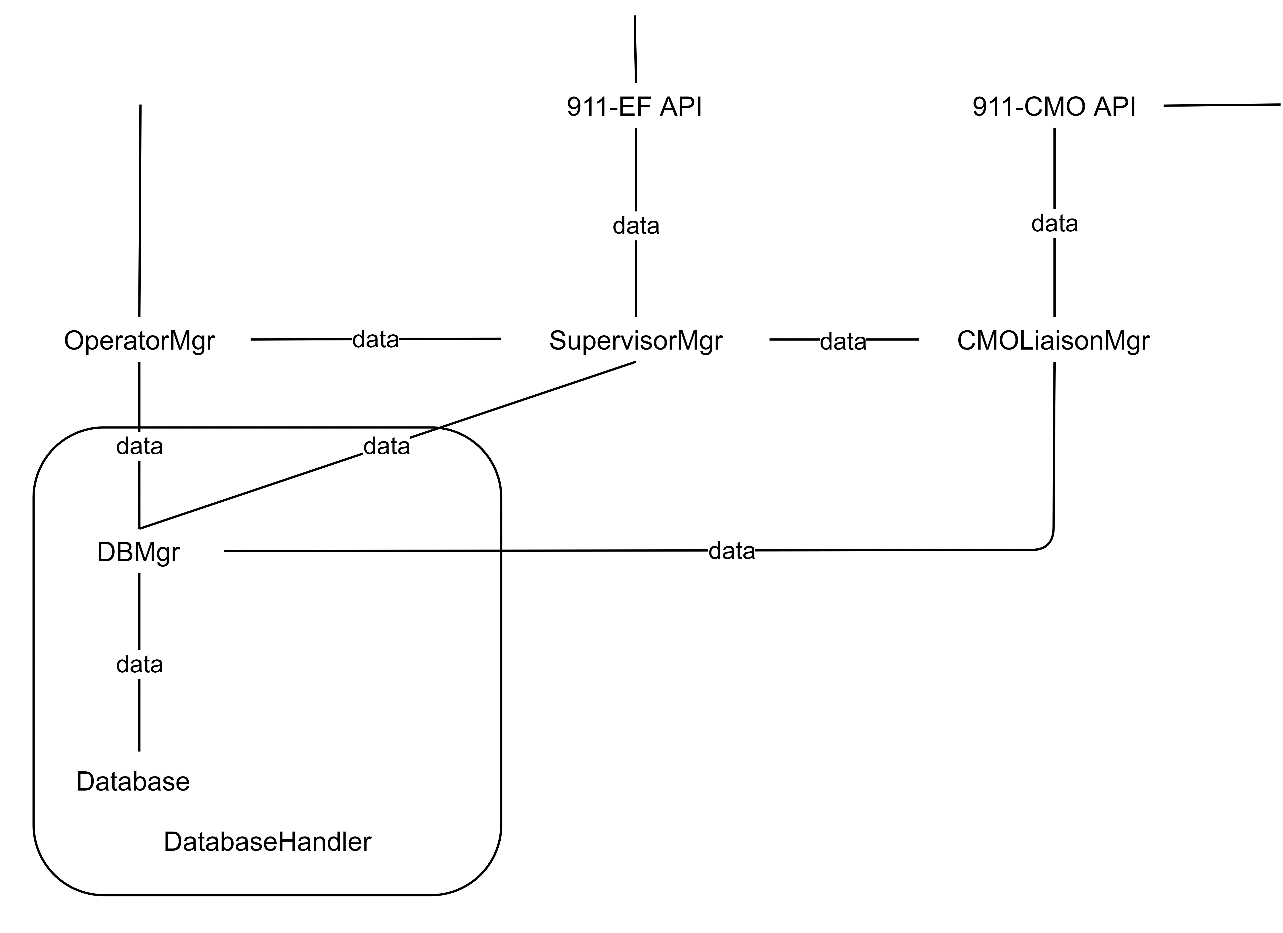
The example above shows how the components in CallHandler relate with others and within itself. The calls are handled by specially trained operators who then entered the call details into the operator manager. During the call, Operator will determine the location of crisis based on the details provided 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 API will be initiated to send data to EF.



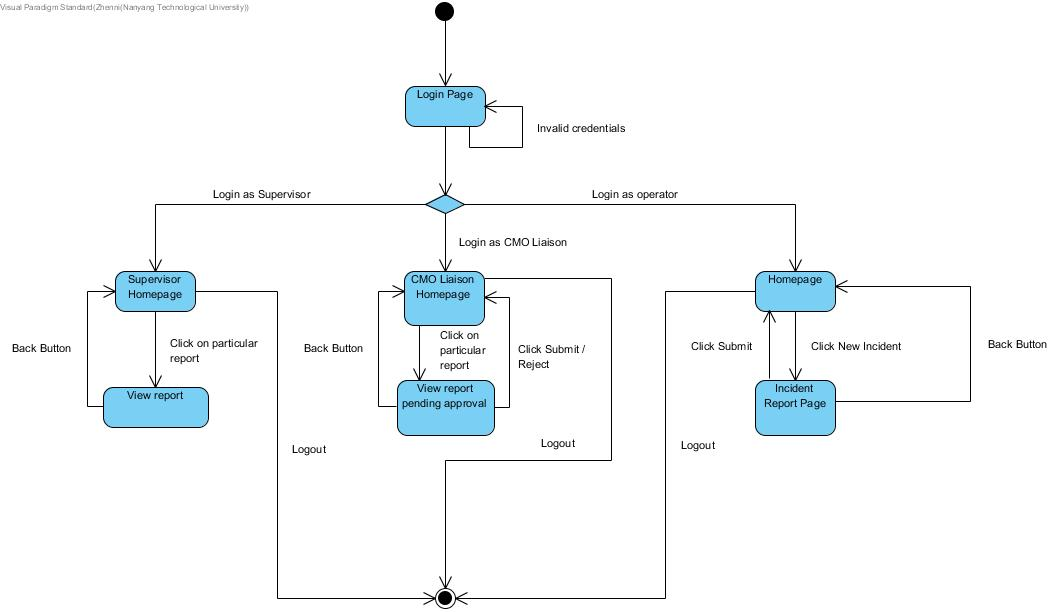
After the CMO liaison Manager receives the crisis data, it then analyses the report. If the reported crisis is indeed Demon level and above, 911-CMO API will be initiated to send the data through to CMO, else CMO Liaison Manager will send data back to Supervisor Manager



The system must be able to withstand a large number of 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.

# Initial architecture behaviour

For initial architecture structure with behaviour, we will use dialogue map to illustrate. Dialogue map is a specific form of state machine diagram which is used to show discrete behaviour of a part of designed system through finite state transition and focus more on user interface. The dialogue map for 911 subsystem is shown as below:



# Imprinted Software Quality

Quality is the measurable degree to which the software satisfies requirements, conforms to specifications, and meets vendor’s expectations. The 911 subsystem has implanted following McCall quality attributes to strive for a better performance of the subsystem:

## Integrity

LogIn component is to ensure that control is only passed after authentication. This is to prevent unauthorized personnel from accessing the data in 911 system. The following diagram shows the structural implementation of this quality attribute:

https://lh3.googleusercontent.com/I3-JySUVj1WXYcSeUr7bgDQNnpeaGrFrLp8xoN5kW2Y7TKzPFZABtYAjQ0nsqUbvMfvymBmc_w2f8AlWEdCTG6d-7o8e1Qdh7Ggk1sattul9eLNISXMtht4tEnhFBgEh3foM3rYC

As seen from the diagram, operators need to authenticate themselves before being able to enter the system. They need to enter their own userID and password for log in purposes, and the information will be then passed to LogIn component for verification. Only upon successful authentication will the control be passed to the next process, in this case, CreateRequest.

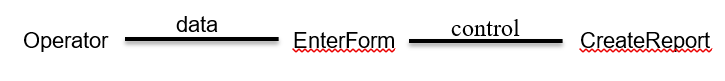
## Flexibility

The system should be able to handle exceptions risen from the system. For example, in the 911 subsystem, if a user fails to log in for a few times, some exception handling process will be triggered. The following diagram illustrates our idea:

flexibility.png

In the case of several consecutive times of log-in failure performed by the user, the system will go to AuthenticationFailHandling component to work on this exception.

## Usability

Upon successful login, the operator, once answered the call, would be required to key in the details in the form before generating the report. The following diagram illustrates our idea:  


Upon creating the report, the system will go back to the main page and wait for another incident call, before creating another report.

## Correctness

The main objective of our system is to allow our operators to be able to create a report based on details given by the caller. So, our functionality allows the operator to log in, upon successful logged in, then would the operator be able to create report. However, if the operator’s authentication failed, then AuthenticationFailHandling will handle the exception, as per the specification of the 911 system.

# Analysis Modelling

Structured Systems Analysis & Design Method (SSADM) is a hefty set of structured analysis and structured design techniques and graphical tools for identifying and transforming business requirements into software design specifications. For this portion, our group will be making use of the Data Flow Diagram (DFD) to process mapping of system.

DFD is a behaviour diagram which shows (from the process perspective) inputs, outputs, and data storage. It does not indicate timing between design objects inside a process, nor sequence of distinct processes. Fig 7 below shows our DFD.

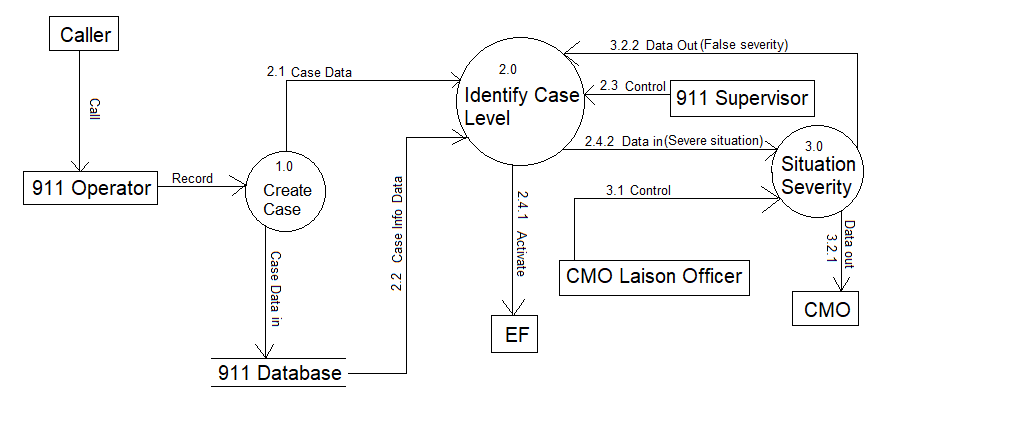


Fig 7: Data Flow Diagram

Referring to the data diagram, the caller first calls 911 operator which the operator create the data of incident report to the our create case process. In this process, the 911 operator creates a case based on this information, this data is passed into our 911 Database and at the same time it would pass to Identify Case Level process. Once Identify Case Level received the case, it will notify the 911 supervisor in-charge of classifying the case.

Upon classifying the case, firstly if the case is monkey (false incident), it would be pass back to the operator. Else if the case is Wolf or Tiger, the supervisor would activate (2.4.1) the EF. If the case determines to be a Demon, Dragon or God it would notify CMO Liaison officer via Situation Severity process through Data in (2.4.2). If CMO Liaison officer determines if the situation is true, the Data would be pass to CMO on Data out (3.2.1). Else if the situation determine to be non-severe, the data would be returned to 911 supervisor on data out(3.2.2) and it would then activate EF via Identify Case level Process.