# 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:

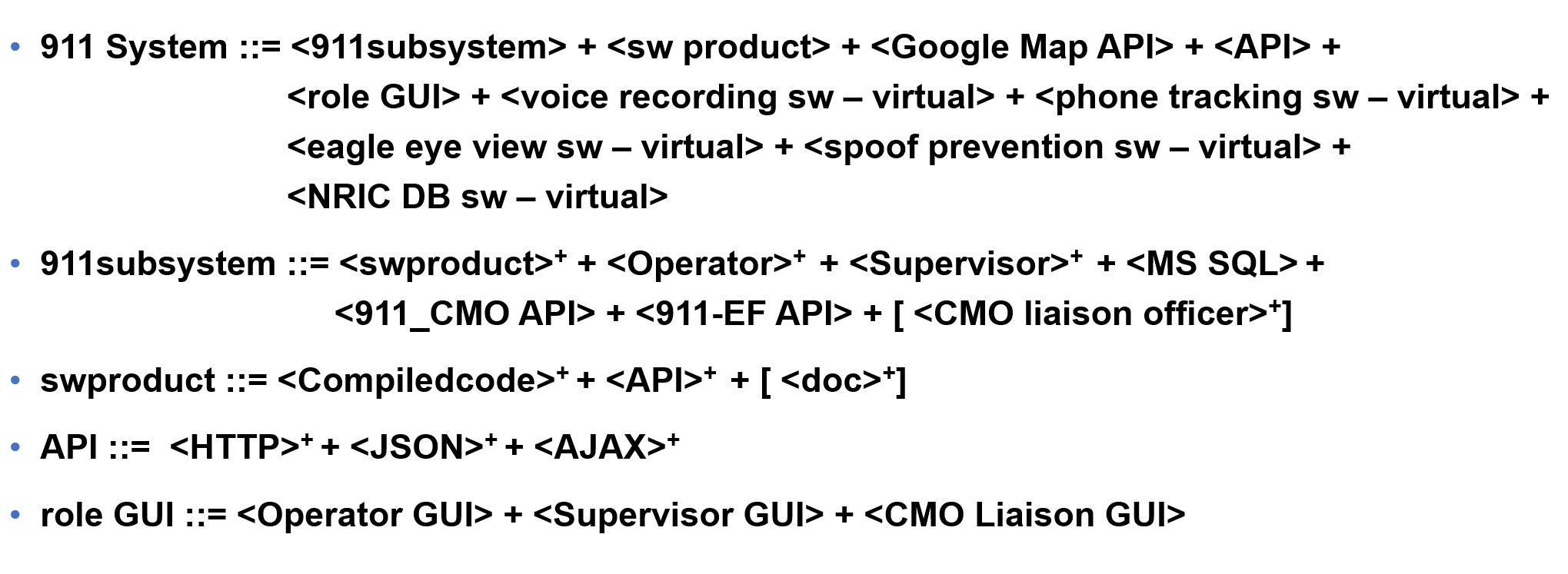


Figure 2. EBNF Statement for 911 subsystem

**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.