SafeStreets

Software Engineering II Project

Politecnico di Milano

December 19, 2019

Authors:
Matteo Pacciani
Francesco Piro

Professor:
Matteo Giovanni Rossi

Outline

RASD

- **Goals**
- **Boundaries of the system**
- **Meaningful Use-Cases**
- **Relevant Requirements**
- **Relevant Assumptions**
- **Alloy Model**

DD

- **Component View**
- **Components Interfaces**
- **Runtime View**
- **Architectural Styles and Patterns**
- **Deployment View**
- **Implementation Integration and Test Plan**

RASD

Goals

G1 Users should be able to notify authorities when traffic violations occur, in particular parking violations.

G2 Users and authorities should be able to mine the information stored by SafeStreets, with different levels of visibility.

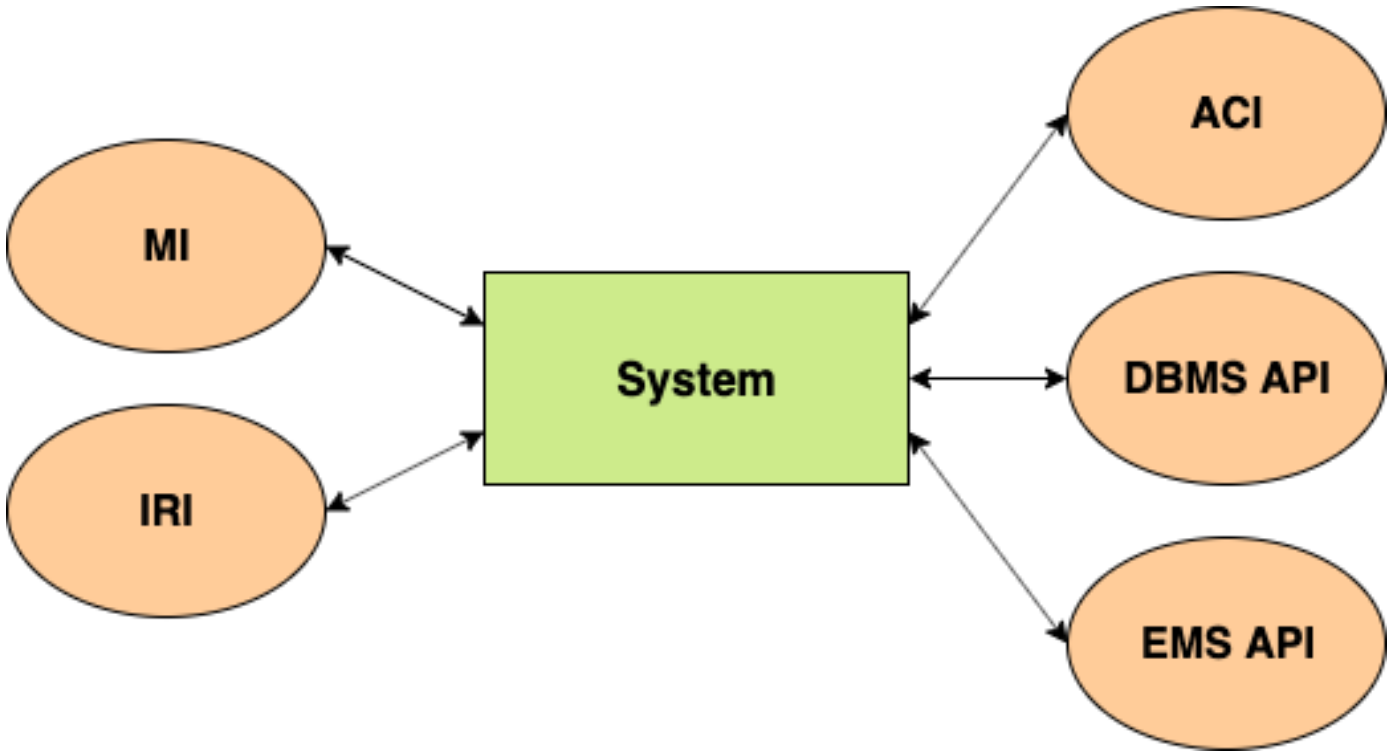
- **G2A** Users and authorities should be able to know where the highest number of violations occur.
- **G2B** Users and authorities should be able to know what types of vehicle make the most violations.
- **G2C** Authorities should be able to consult every violation report sent by users.

G3 Users and authorities should be able to know which streets are safe and which ones are not.

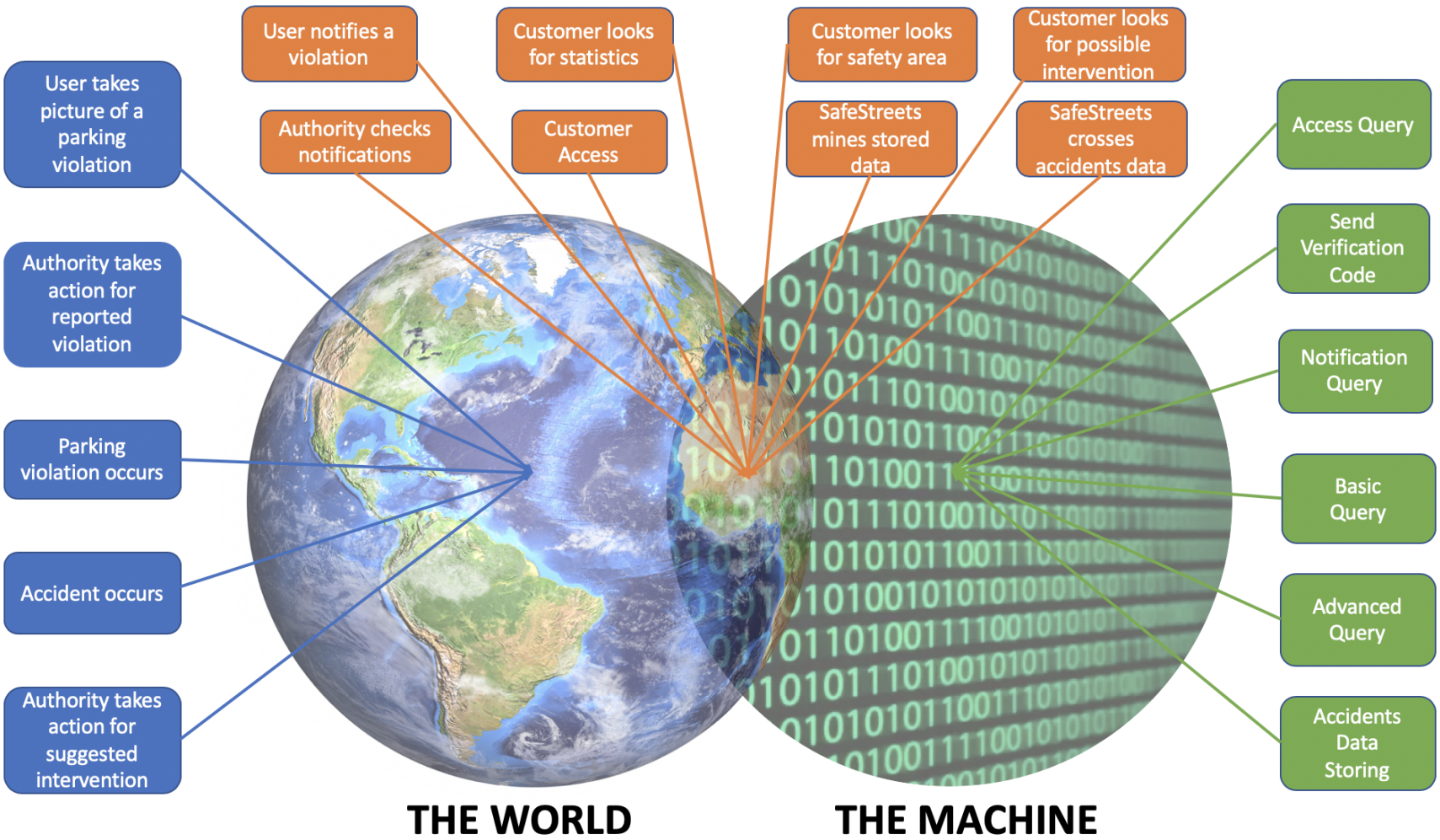
G4 Users and authorities should be able to know the possible interventions that could be done in a city.

System Interfaces

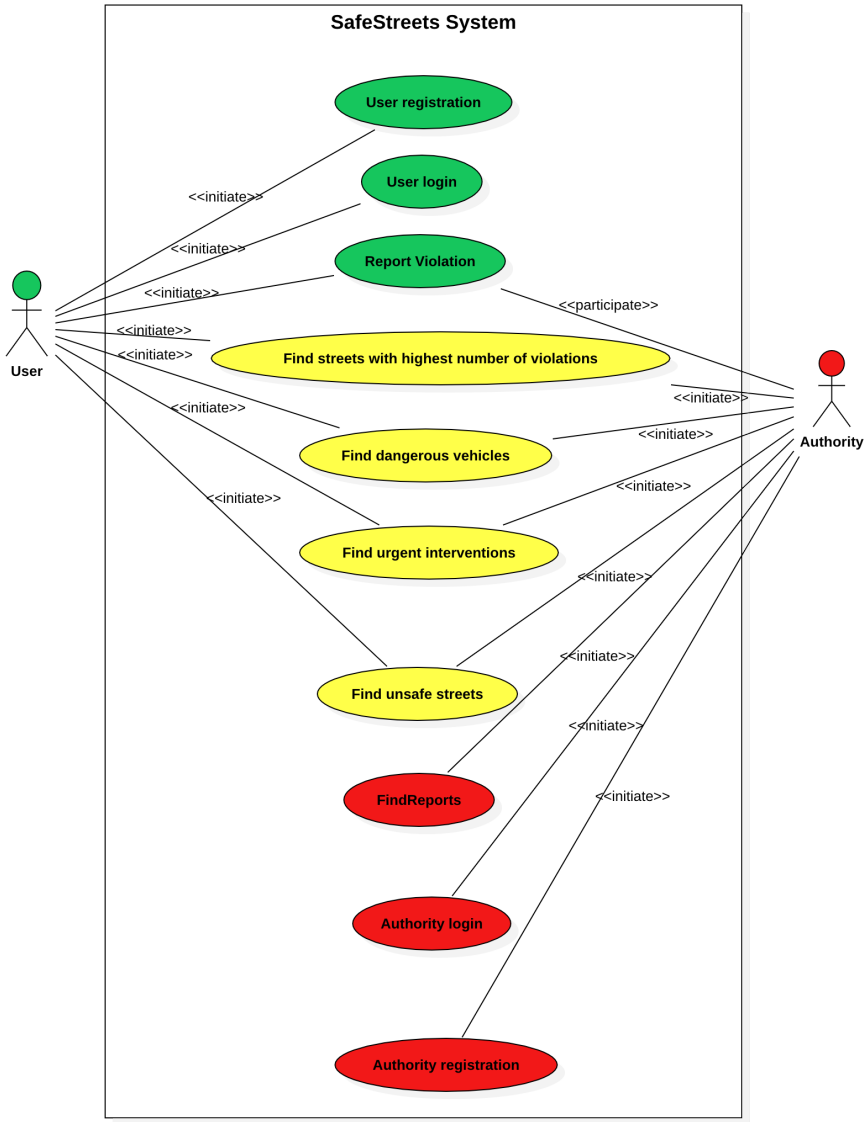
We want now to define the boundaries of our system first with the external interfaces it has to interact with, second with the *World And Machine* that helps to highlight the phenomena.

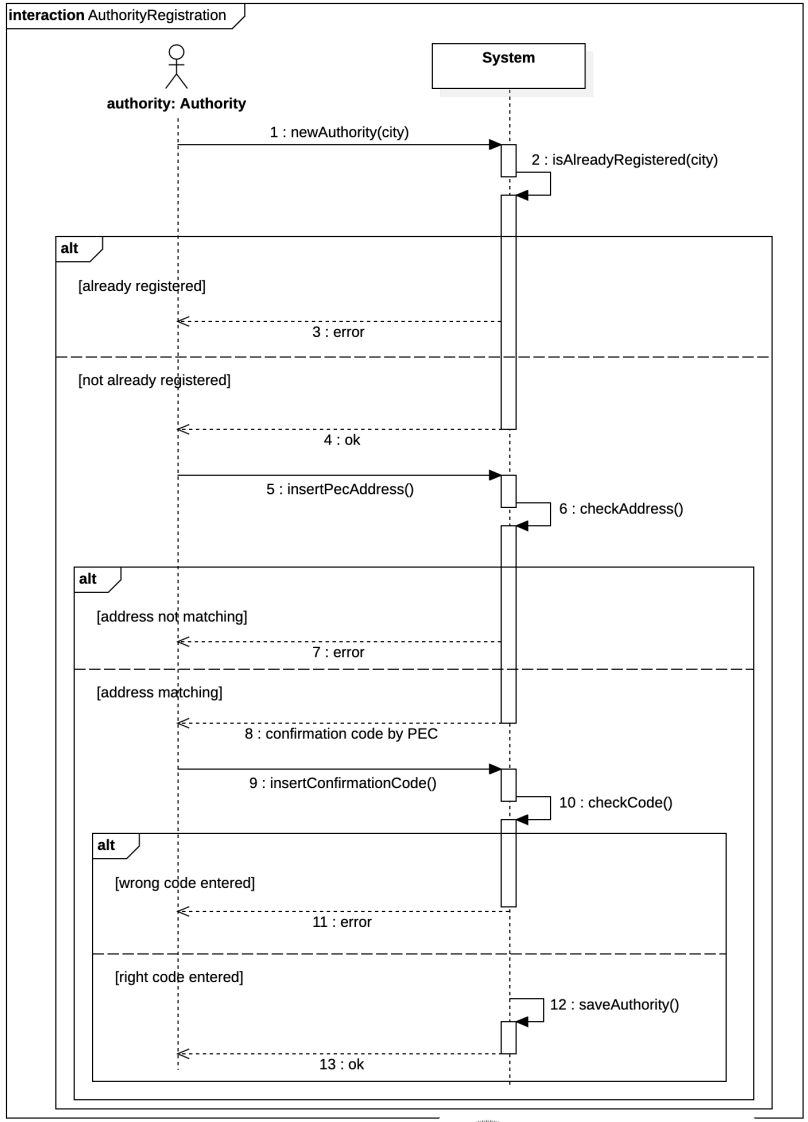


The World and the Machine

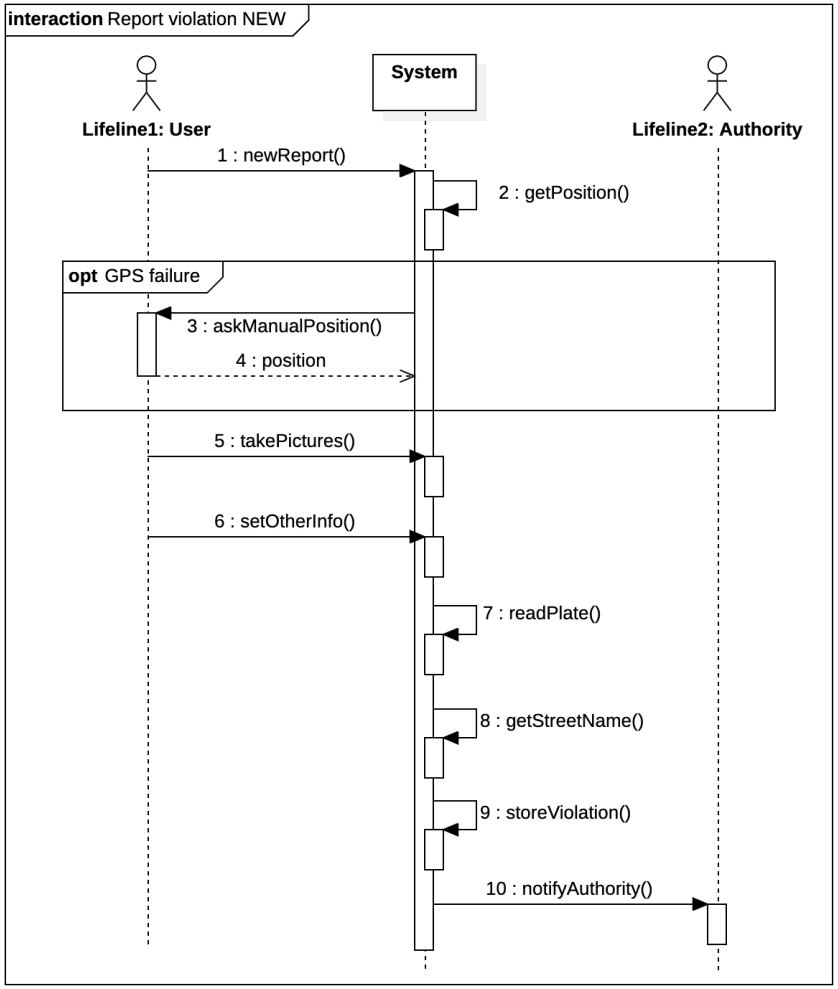


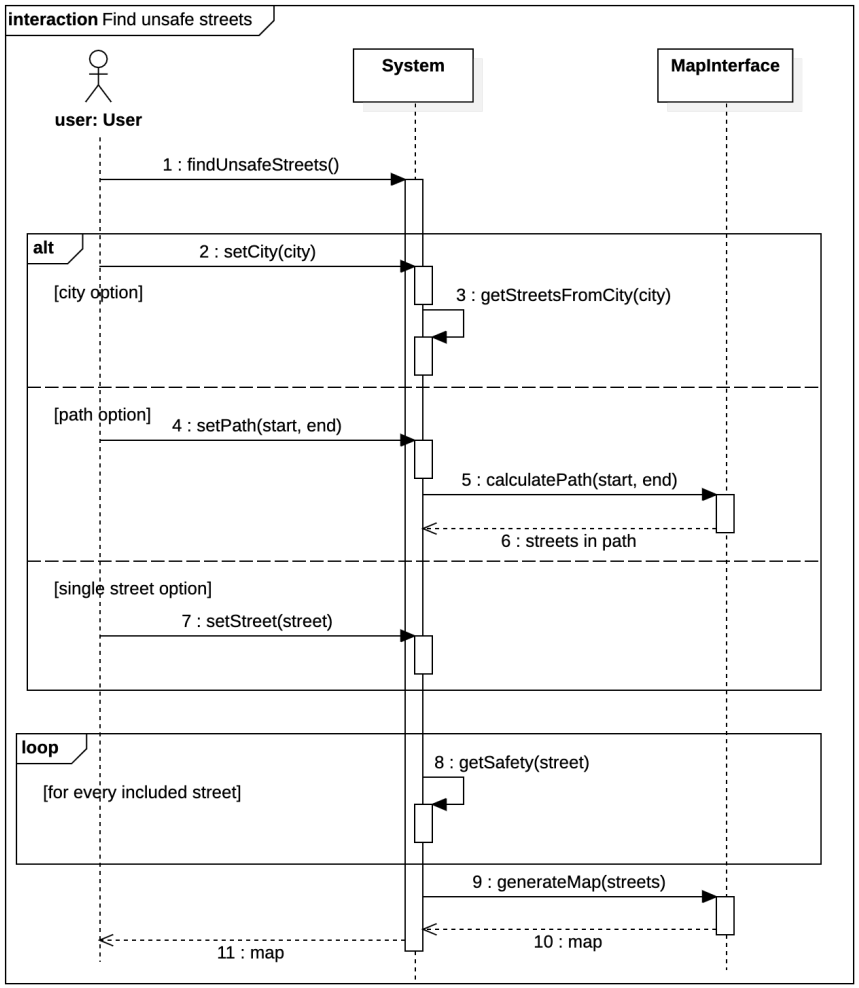
Use Case and Authority Registration





Report Violation and Unsafe Streets





R8 The system must allow users to take pictures of the violation they want to report

R17 The system must be able to show authorities all the violation reports sent by users

R18 The system must be able to mine the stored violation reports

R31 The system must be able to determine whether a street is safe or not

R35 The system must be able to determine the most urgent interventions in a street.

Relevant Assumptions

DA2 Users always insert correct data while reporting violations to SafeStreets

DA7 The GPS module of the devices on which SafeStreets runs always works correctly and has an accuracy of 2 meters

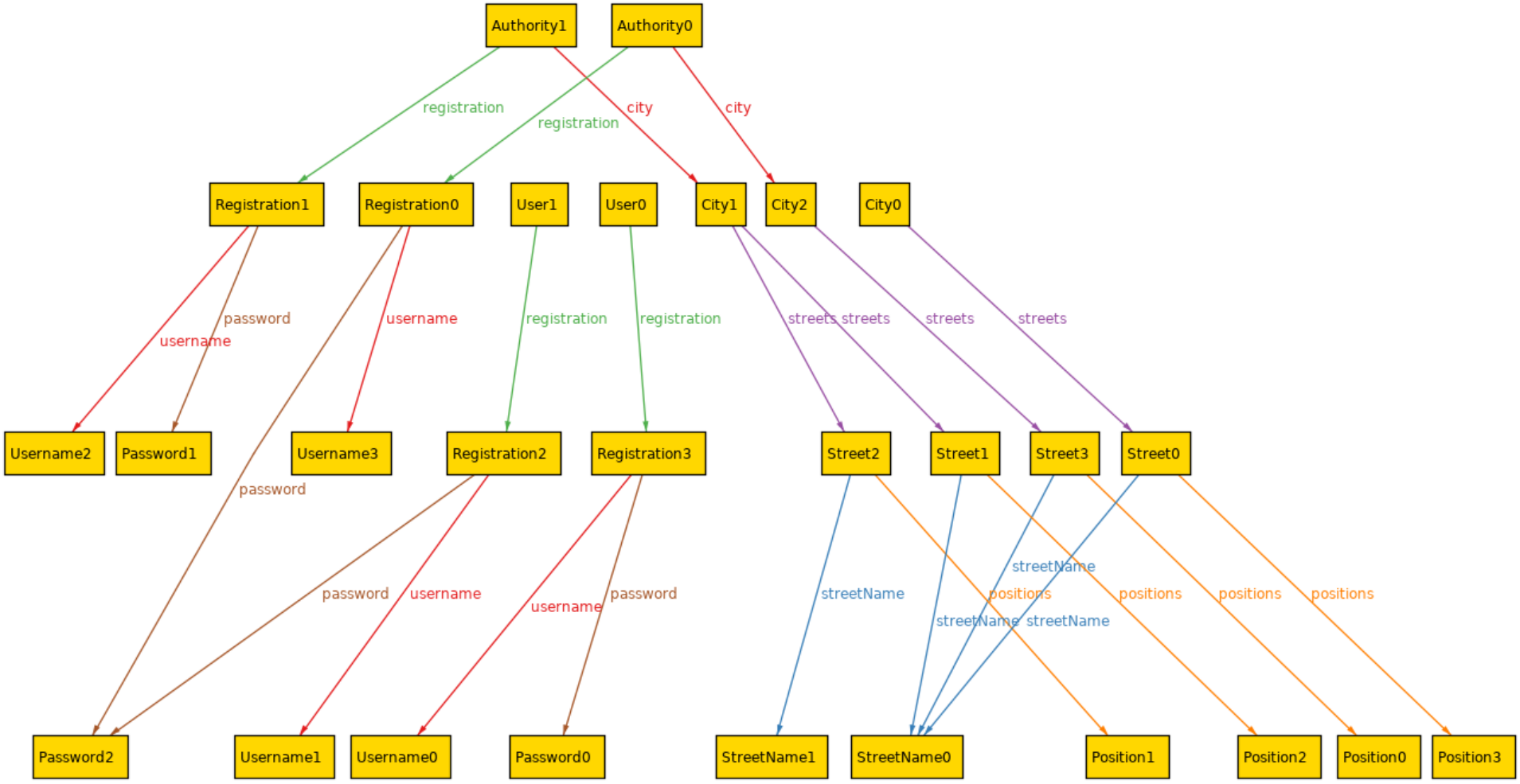
DA8 The camera module of the devices on which SafeStreets runs always works correctly

DA10 The system is assumed to work in Italy

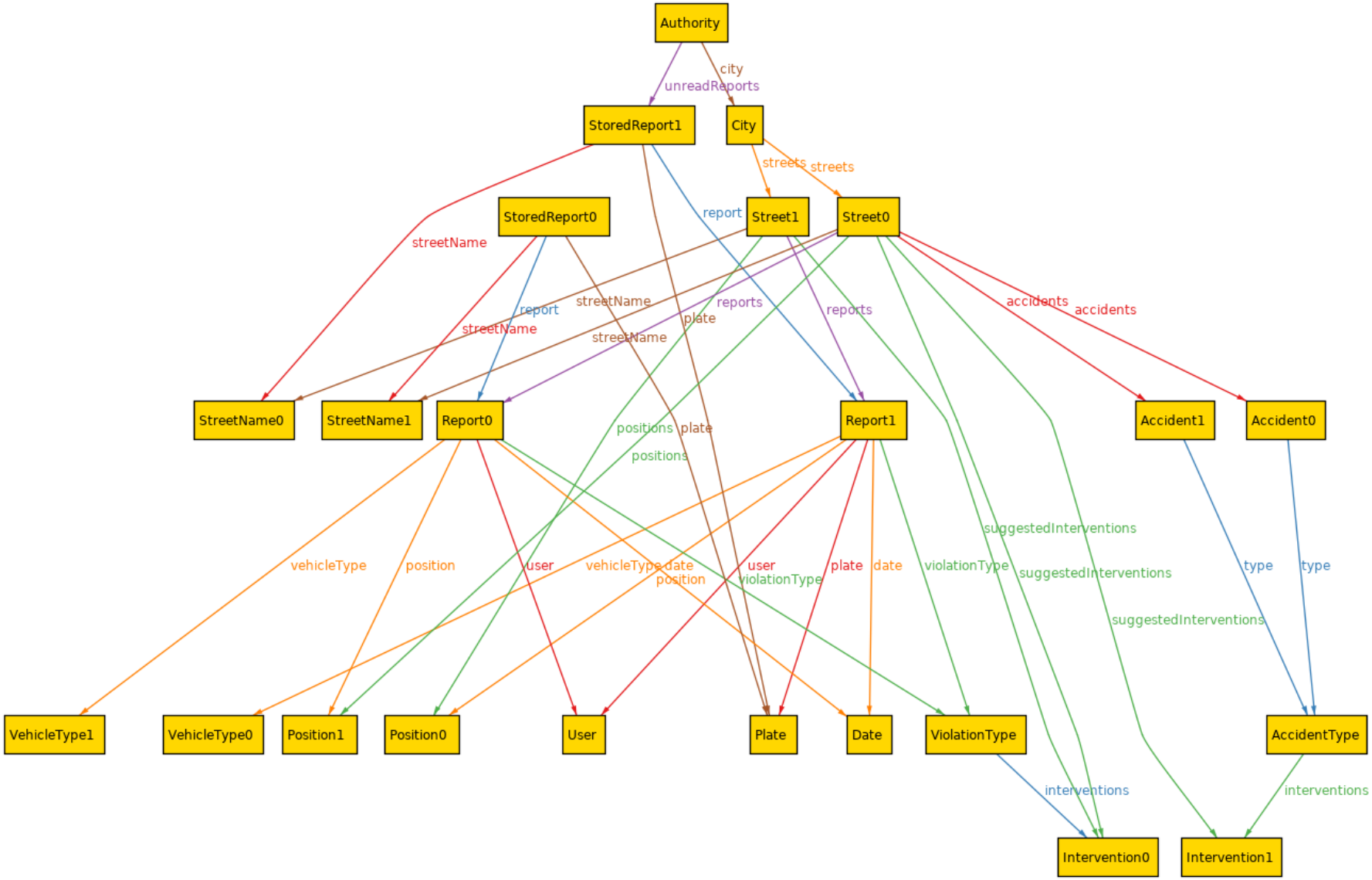
DA13 No one physically and maliciously replaces license plates

DA17 Accidents data provided by municipalities are always correct

First World


Authority1
Authority0
registration
registration
city
city
Registration1
Registration0
User1
User0
City1
City2
City0
password
username
username
registration
registration
streets
streets
streets
streets
Username2
Password1
Username3
Registration2
Registration3
Street2
Street1
Street3
Street0
password
password
username
username
password
streetName
streetName
streetName
streetName
positions
positions
positions
positions
Password2
Username1
Username0
Password0
StreetName1
StreetName0
Position1
Position2
Position0
Position3

Second World

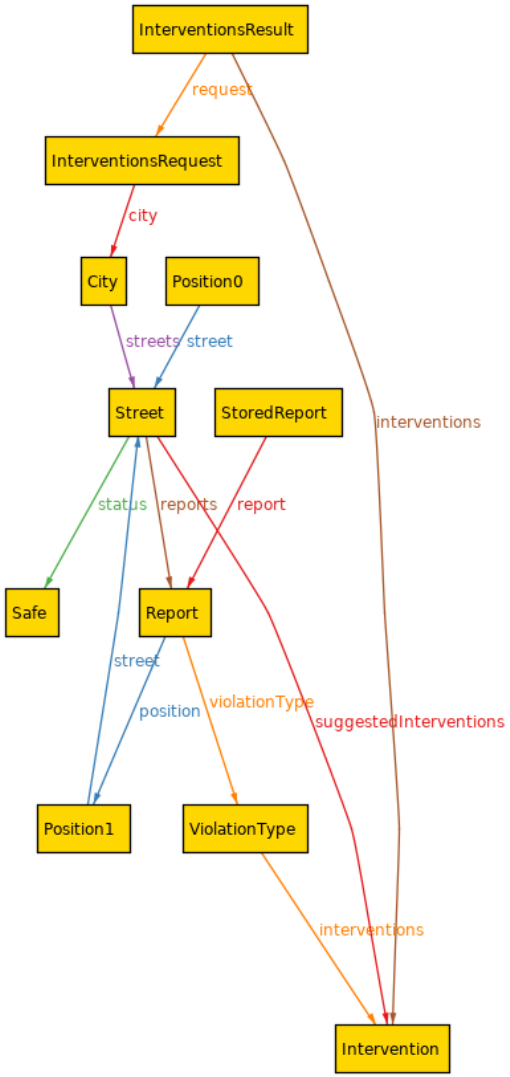


Facts

```
147 //an intervention is suggested for a street iff the number
        of reports in
148 //that street that contain violation types linked to that
        intervention
149 //is greater than 0 OR the number of accidents in that
        street linked to
150 //that intervention is greater than 0
151 fact InterventionSuggestedThreshold{
152   all s: Street, i:Intervention| (i in s.
        suggestedInterventions iff
153     (#getReportsFromInterventionAndStreet[i, s] > 0 or
154     #getAccidentsFromInterventionAndStreet[i, s] > 0))
155 }
156
157 //gets all the reports for that street, having as violation
        type one linked to the
158 //given intervention
159 fun getReportsFromInterventionAndStreet[i : Intervention, s
        : Street] : set Report{
160   {
161     r : Report | i in r.violationType.interventions and r in
        s.reports
162   }
163 }
```

Facts

```
177 --a street is considered unsafe iff
178 --the number of total accidents in that street is greater
        than 1 OR
179 --if the number of total violations in that street is
        greater than 1
180
181 fact UnsafeStreetThreshold{
182   all s: Street | s.status = Unsafe iff (
183     #s.reports > 1 or
184     #s.accidents > 1
185   )
186 }
```

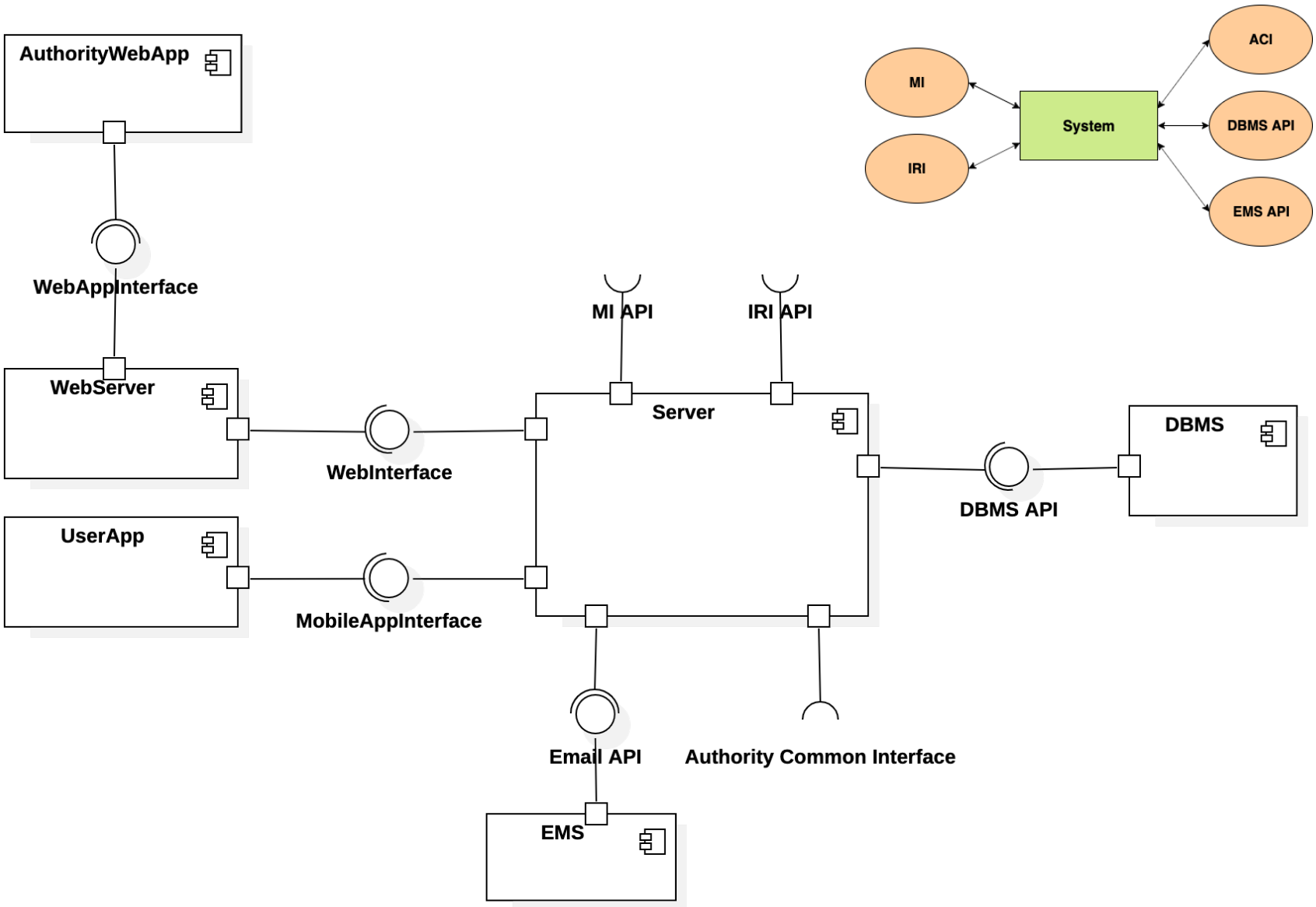



```
261 //this is the rule to apply the filter that comes with the
        InterventionsRequest
262 //only those interventions that satisfy the following
        conditions will be included
263 //in the result
264 fact InterventionsResultRule{
265   all i:Intervention, result:InterventionsResult | i in
        result.interventions iff
266   (
267     some s:Street | s in result.request.city.streets and
268       i in s.suggestedInterventions
269   )
270 }
```

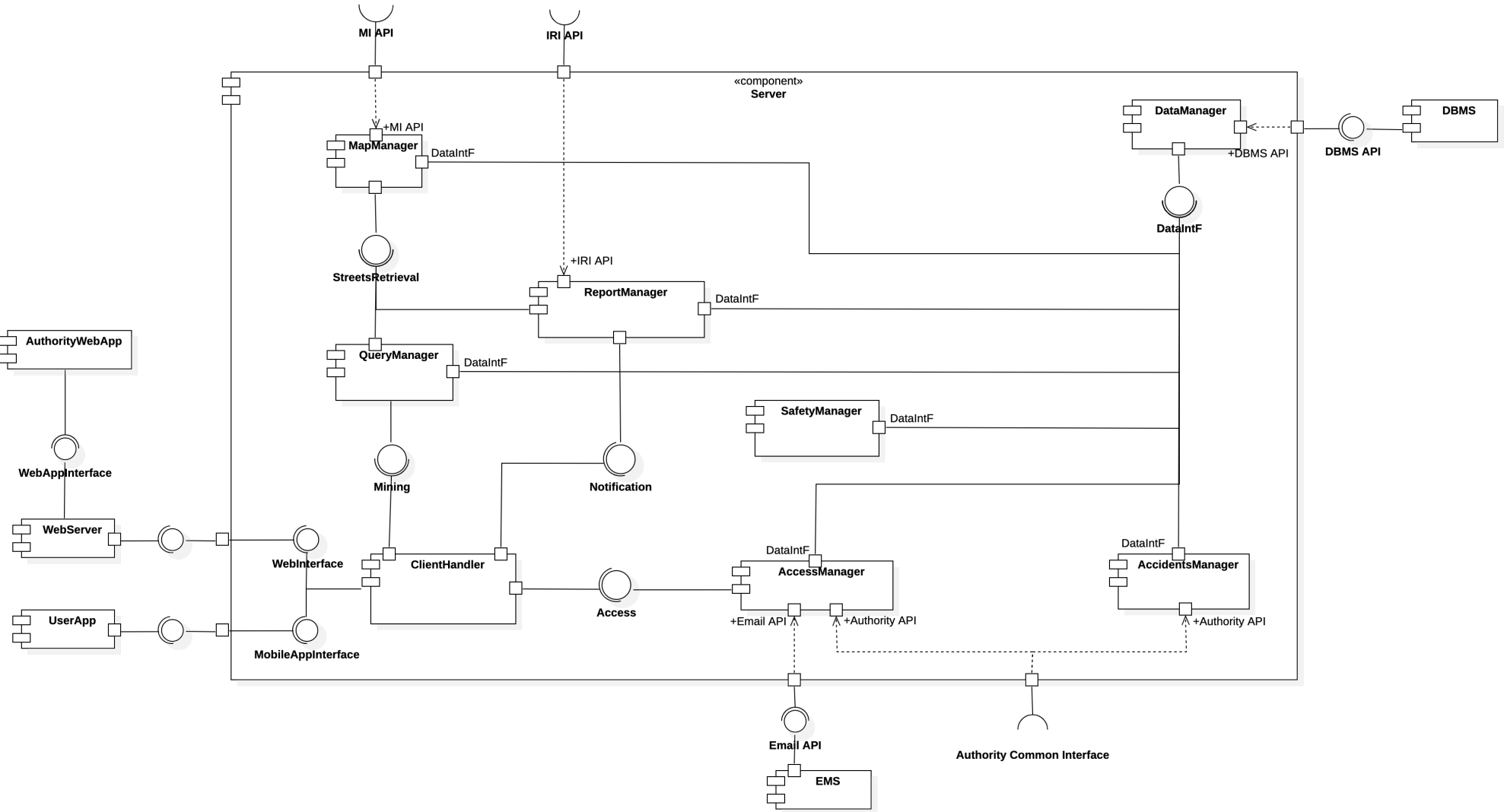
```
239 //this is the rule to apply the filter that comes with the
        ReportsRequest
240 //only those reports that satisfy these conditions will be
        included in the result
241 fact ReportsResultRule{
242   all sr:StoredReport, result:ReportsResult | sr in result.
        storedReports iff
243   (
244     sr.report.violationType in result.request.violationTypes
       and
245     sr.report.date in result.request.dates and
246     sr.report.time in result.request.time and
247     sr.report.vehicleType in result.request.vehicleTypes and
248     (result.request.street = none implies sr.report.position
      .street.city in
249       result.request.authority.city else sr.report.position.
      street in
250           result.request.street
251     )
252   )
253 }
```

DD

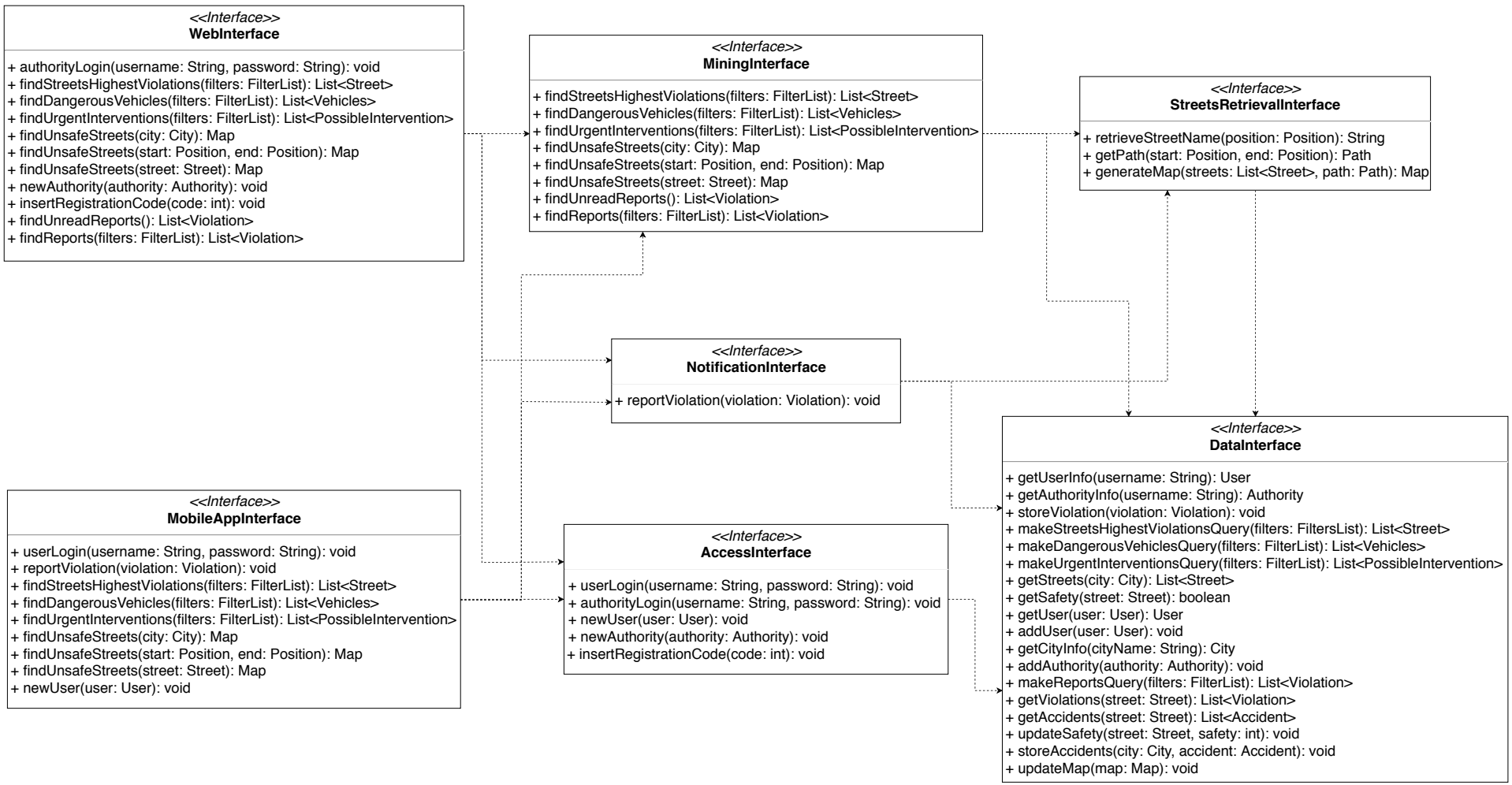
High-Level View



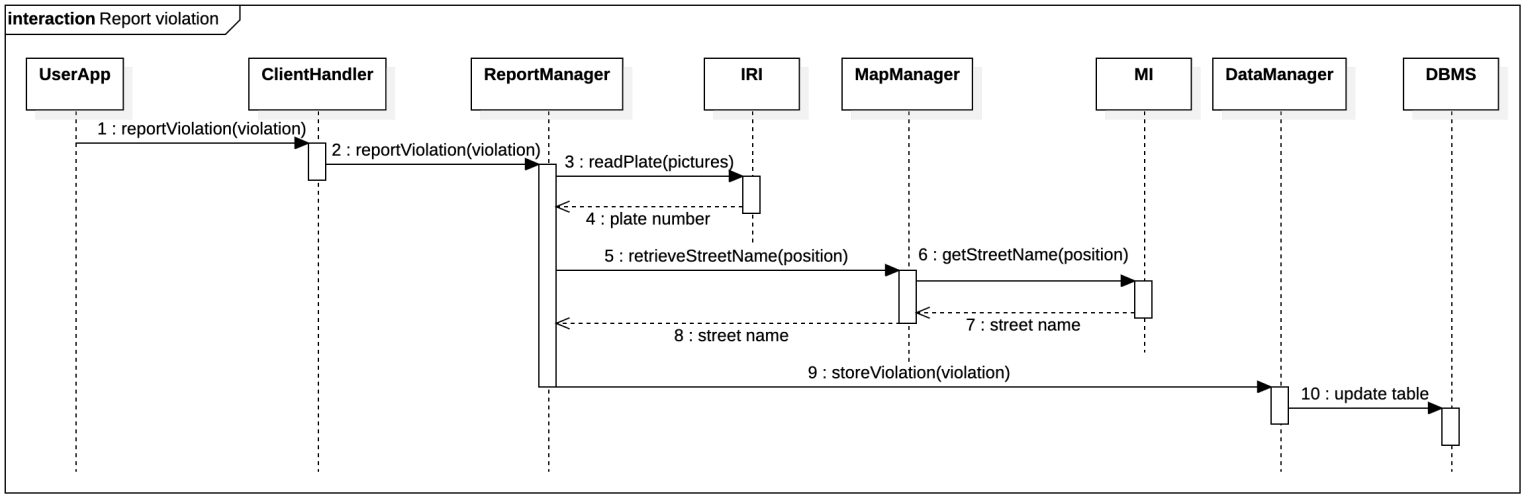
Server Component

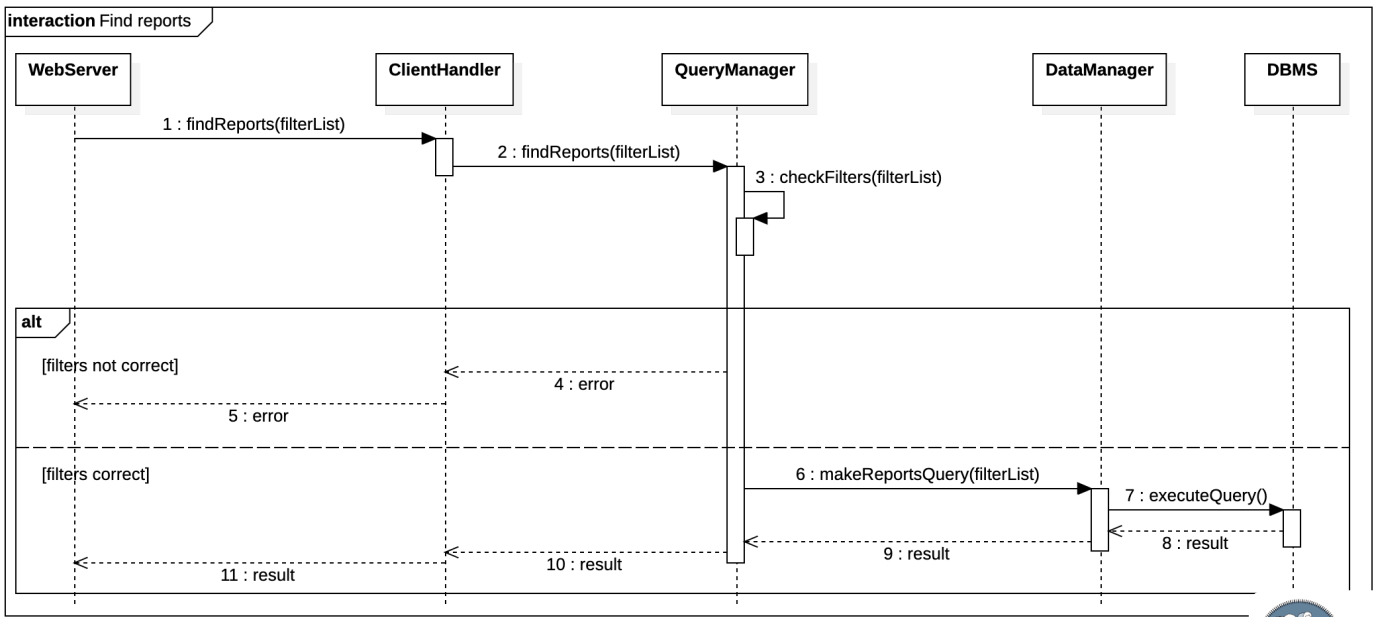


Component Interfaces

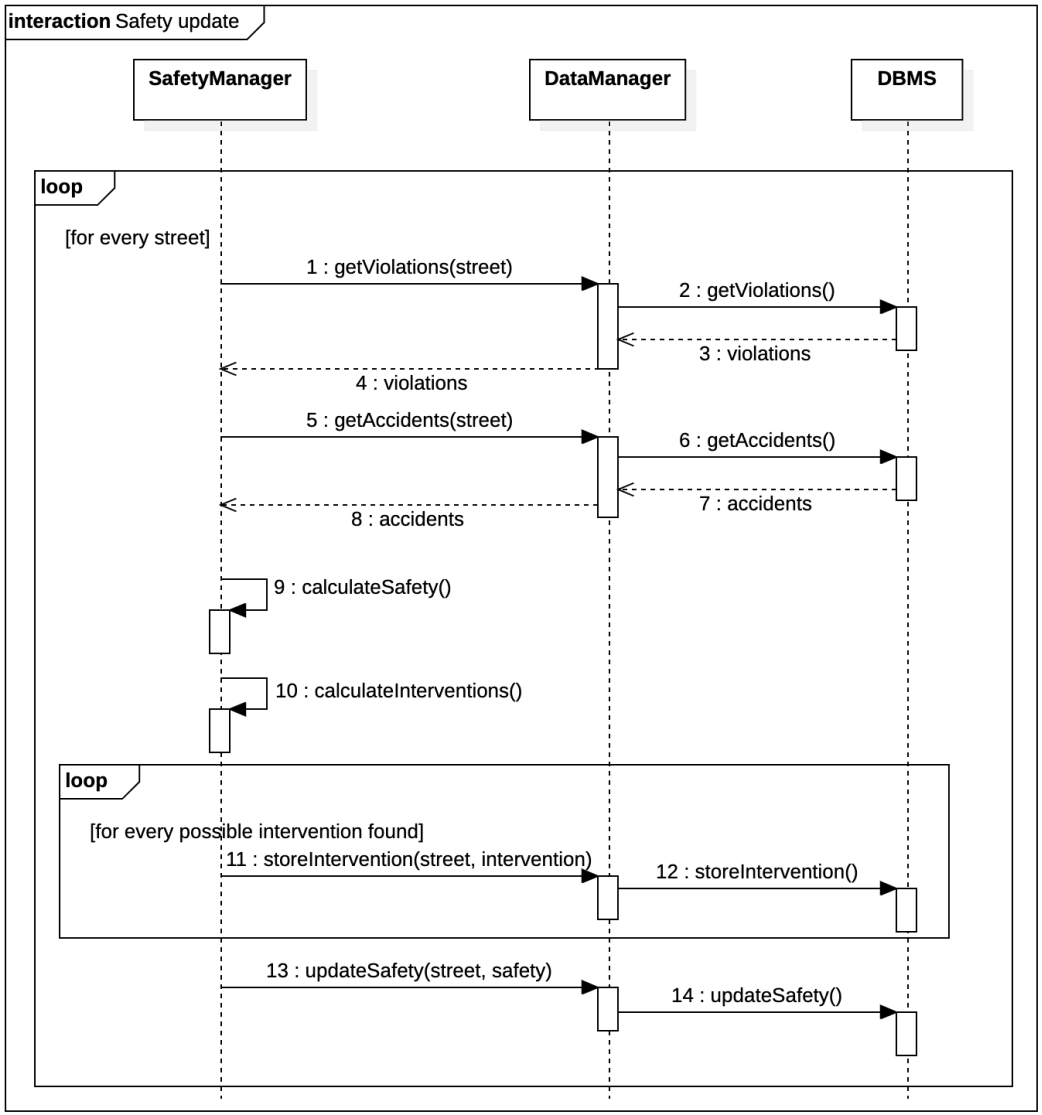


Report Violation and Find Reports



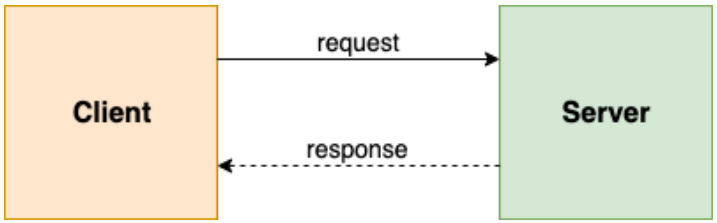


Update Safety

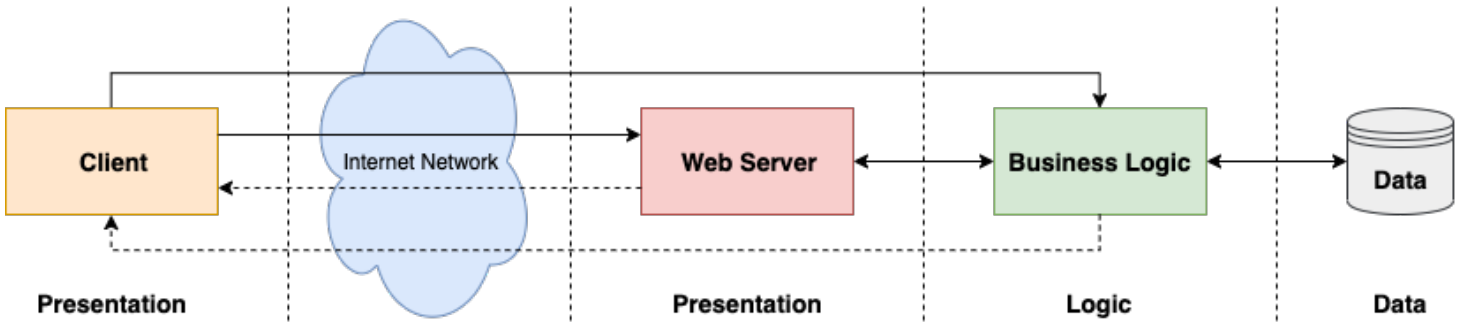


Client-Server in Four-tier Architecture

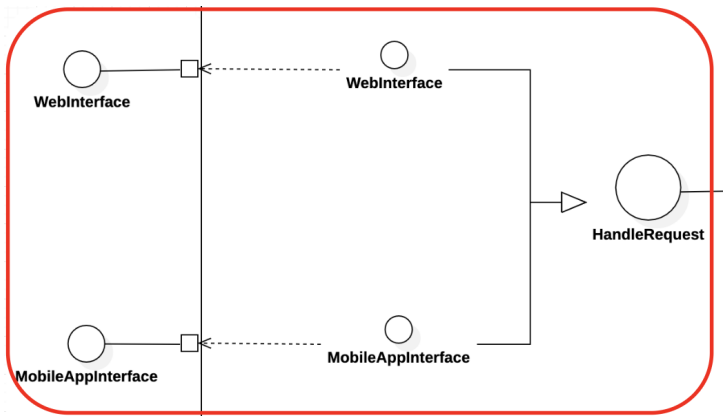
The system is a **crowed-sourced** application based on the client-server paradigm where the *users* represent the crowd on which the functionalities of SafeStreets are funded



The layers of the system are deployed to a *four-tier architecture*

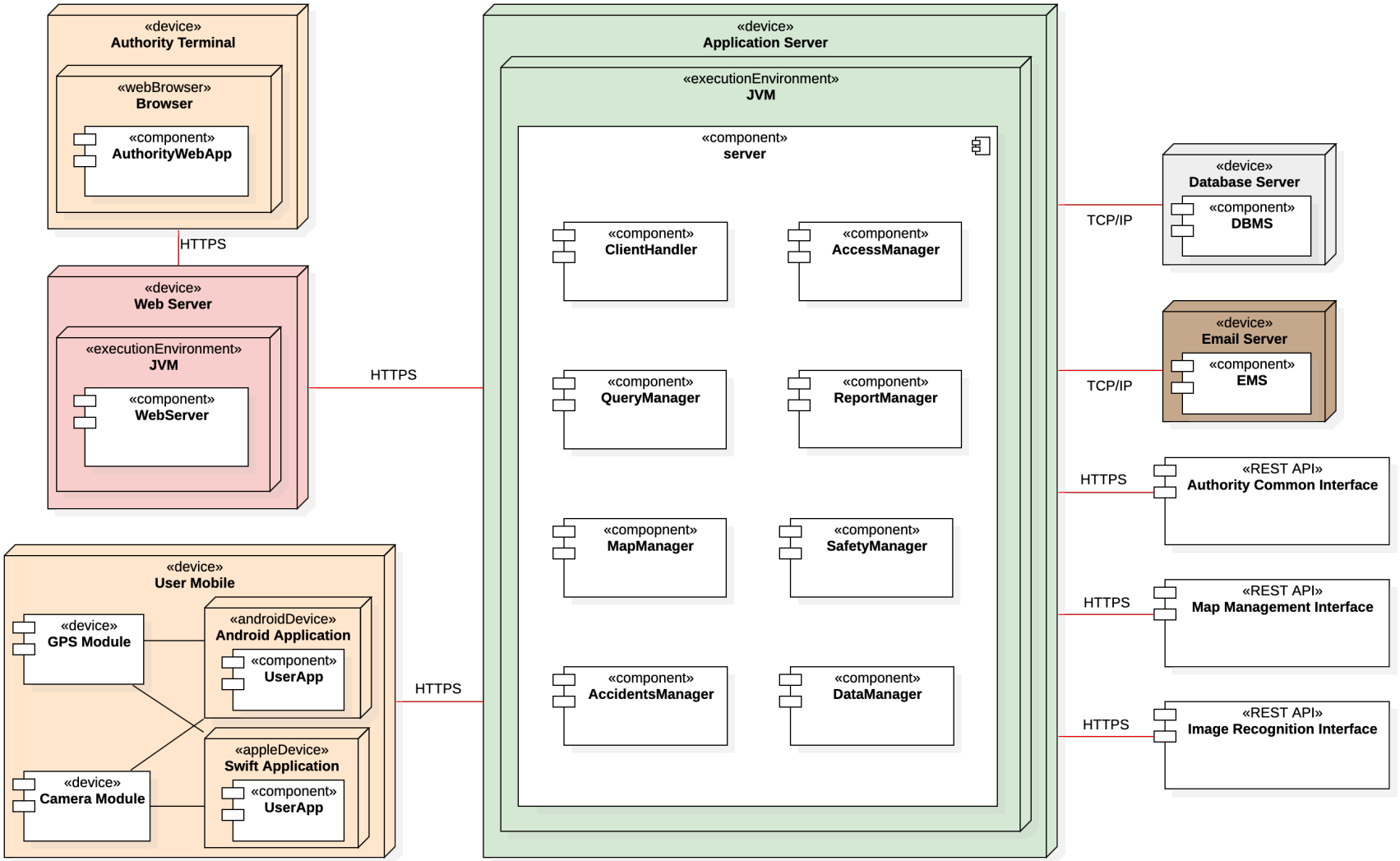


SafeStreets provides its services through **REST** thanks to an interface that is extended twice to define the communication with the **mobile app** for the users and the **web app** for the authorities



Moreover basic communication with the TCP/IP protocol is also used in particular for the external components that allows the system to manage its data: the *DBMS* and the *EMS*

Deployment View



The strategy on which the *Implementation, Integration and Test Plan* is developed, and thus the use relation hierarchy diagram, is based on the following 4 features:

1. **Bottom-Up and Top-Down approach**

The strategy on which the *Implementation, Integration and Test Plan* is developed, and thus the use relation hierarchy diagram, is based on the following 4 features:

1. **Bottom-Up and Top-Down approach**
2. **Core Functionality**

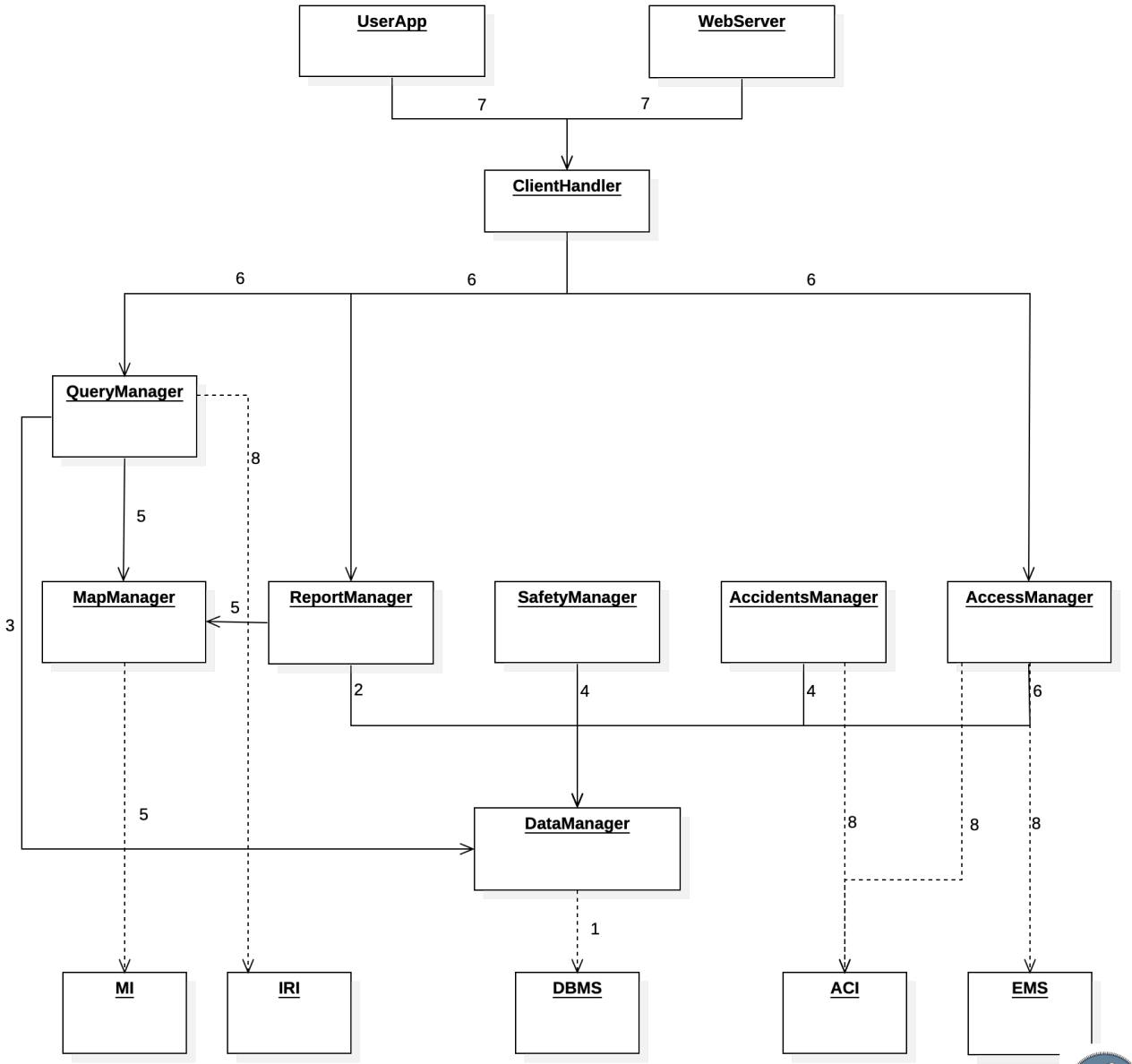
The strategy on which the *Implementation, Integration and Test Plan* is developed, and thus the use relation hierarchy diagram, is based on the following 4 features:

1. **Bottom-Up and Top-Down approach**
2. **Core Functionality**
3. **Decoupling of Components**

The strategy on which the *Implementation, Integration and Test Plan* is developed, and thus the use relation hierarchy diagram, is based on the following 4 features:

1. **Bottom-Up and Top-Down approach**
2. **Core Functionality**
3. **Decoupling of Components**
4. **Reliability of Eternal Systems**

Use Relation Hierarchy Diagram



Thanks for your Attention