

SafeStreets

Software Engineering II Project



POLITECNICO DI MILANO

December 17, 2019

Authors:

Matteo PACCIANI
Francesco PIRO

Professor:

Matteo Giovanni ROSSI

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

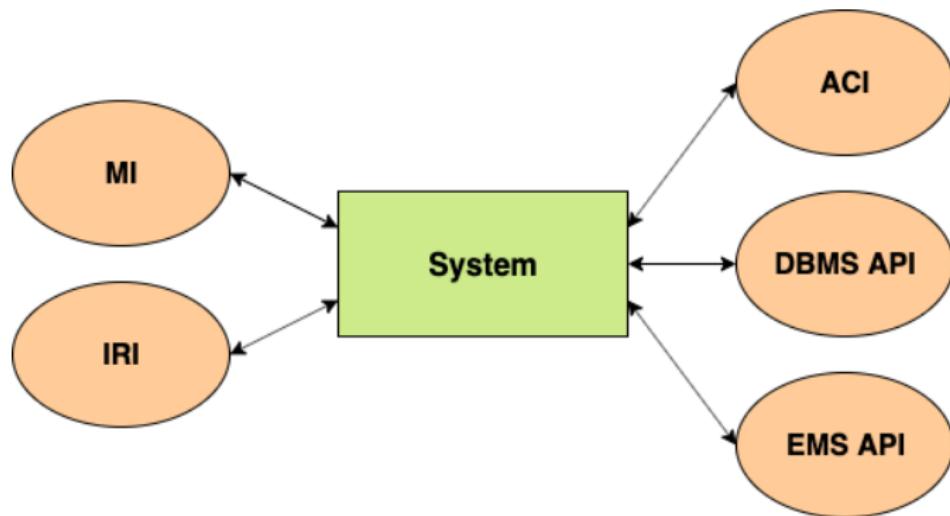


RASD

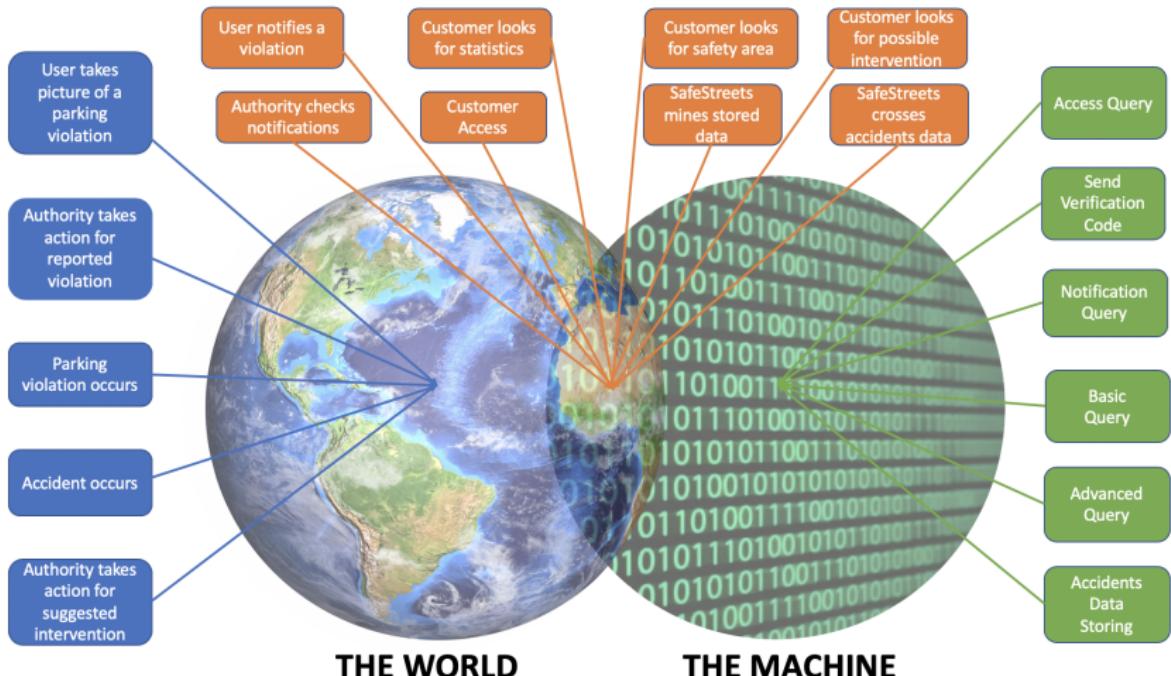
- G1** Users should be able to notify authorities when traffic violations occur, in particular parking violations.
- G2** Users and authorities should be able to mine the information stored by SafeStreets, with different levels of visibility.
 - G2A** Users and authorities should be able to know where 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.



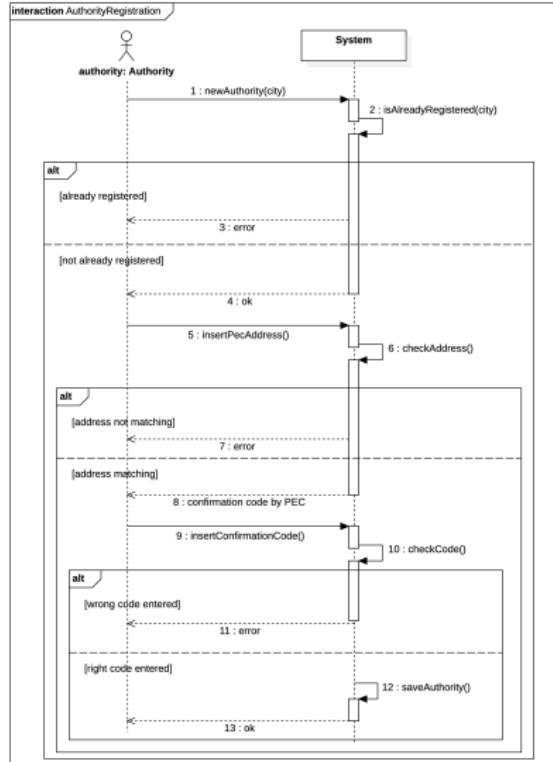
We want now to define the boundaries of our system first with the external interfaces it has to interact with, second with the *World And Machine* that helps to highlight the phenomena.



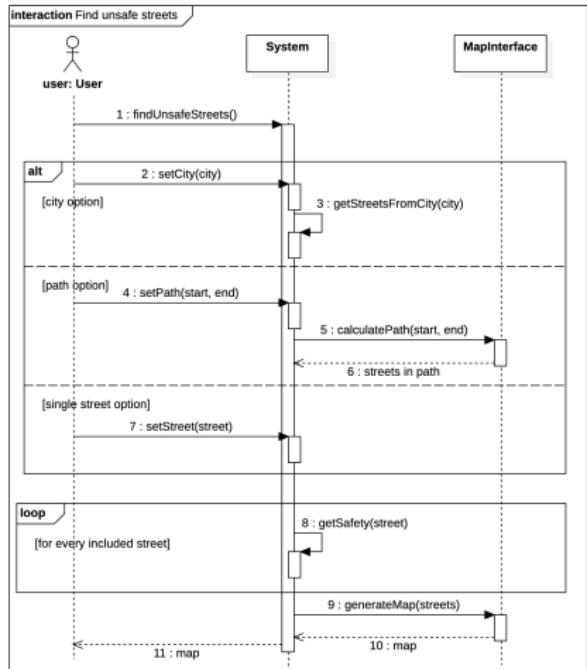
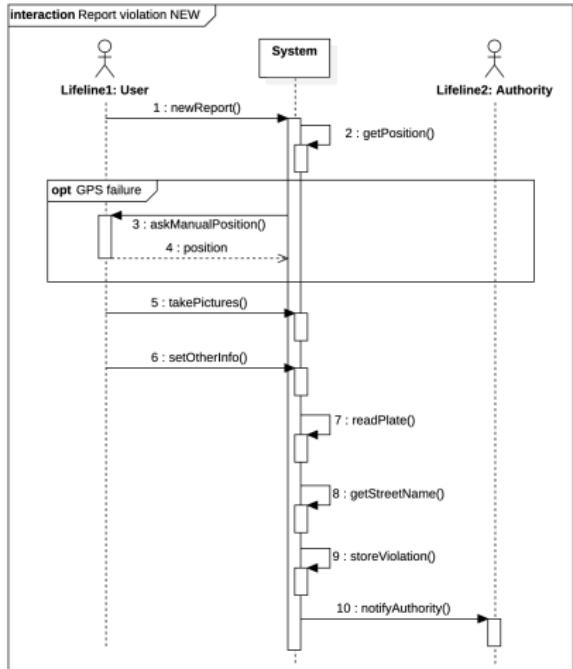
The World and the Machine



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



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



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



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



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.



DA2 Users always insert correct data while reporting violations to SafeStreets



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



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



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



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



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

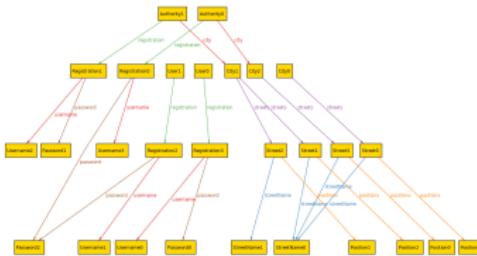
DA8 The camera module of the devices on which SafeStreets runs always works correctly

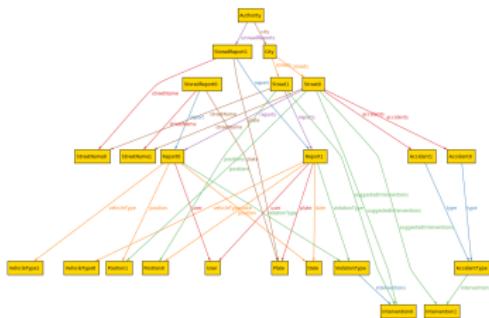
DA10 The system is assumed to work in Italy

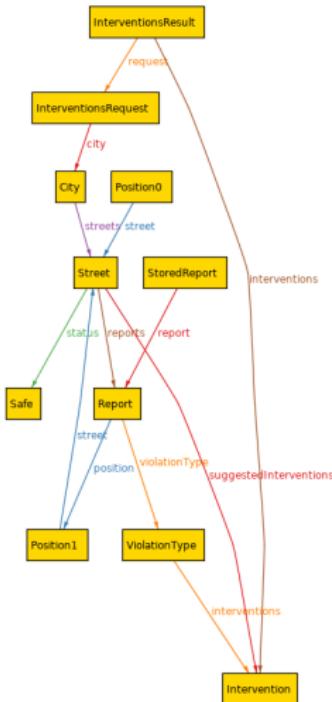
DA13 No one physically and maliciously replaces license plates

DA17 Accidents data provided by municipalities are always correct







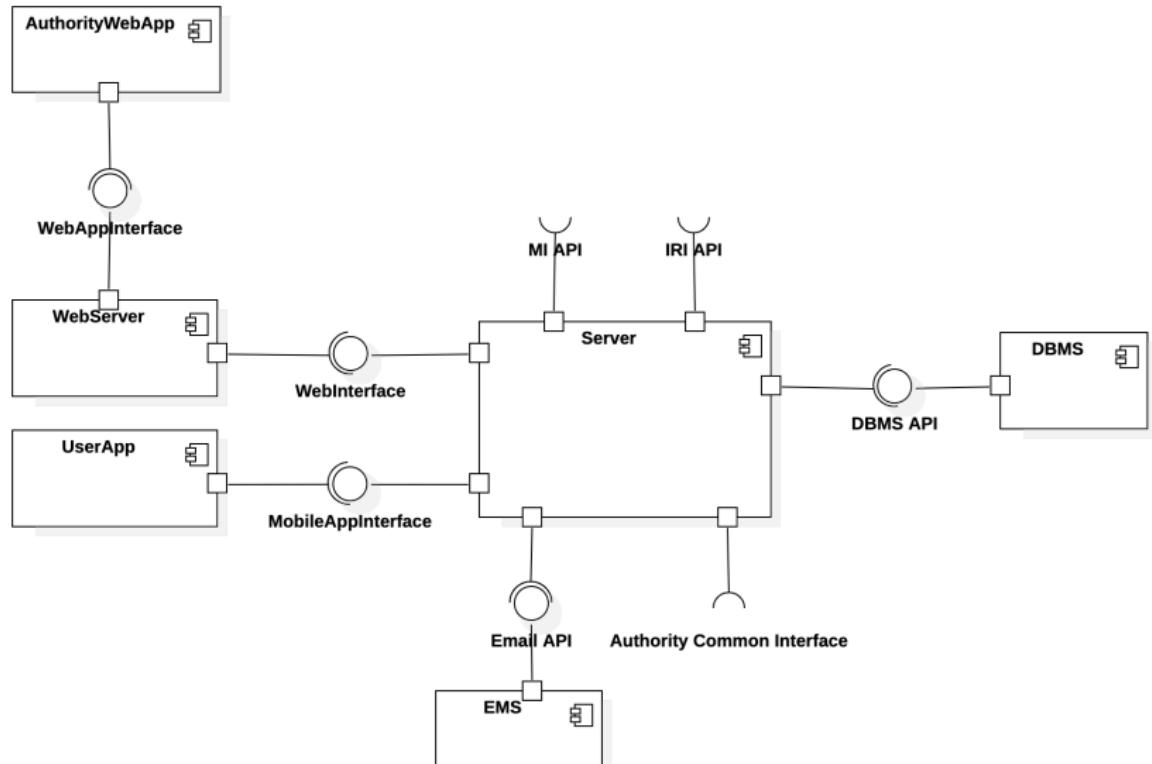


```

261 //this is the rule to apply the filter that comes with the
262 //InterventionsRequest
263 //only those interventions that satisfy the following
264 //conditions will be included
265 //in the result
266 fact InterventionsResultRule{
267   all i:Intervention, result:InterventionsResult | i in
268     result.interventions iff
269   (
270     some s:Street | s in result.request.city.streets and
271     i in s.suggestedInterventions
272   )
273 }
  
```

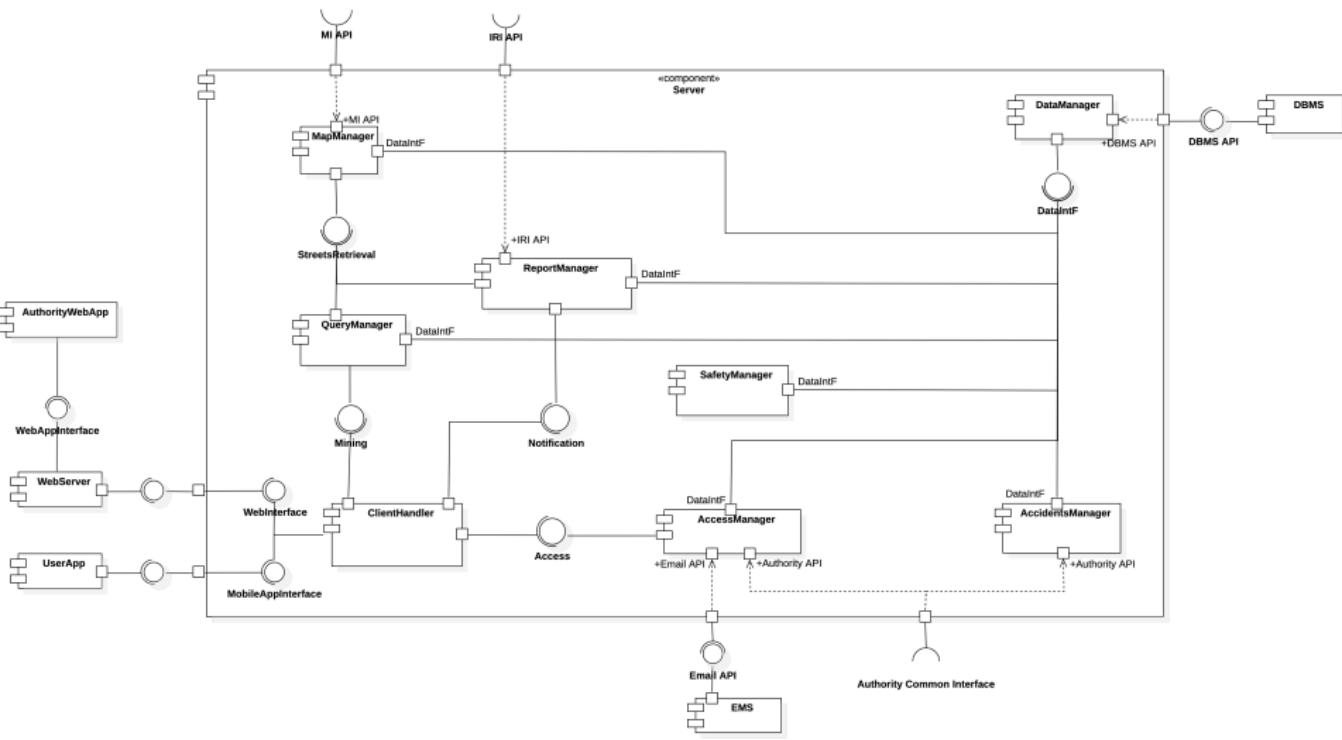


DD

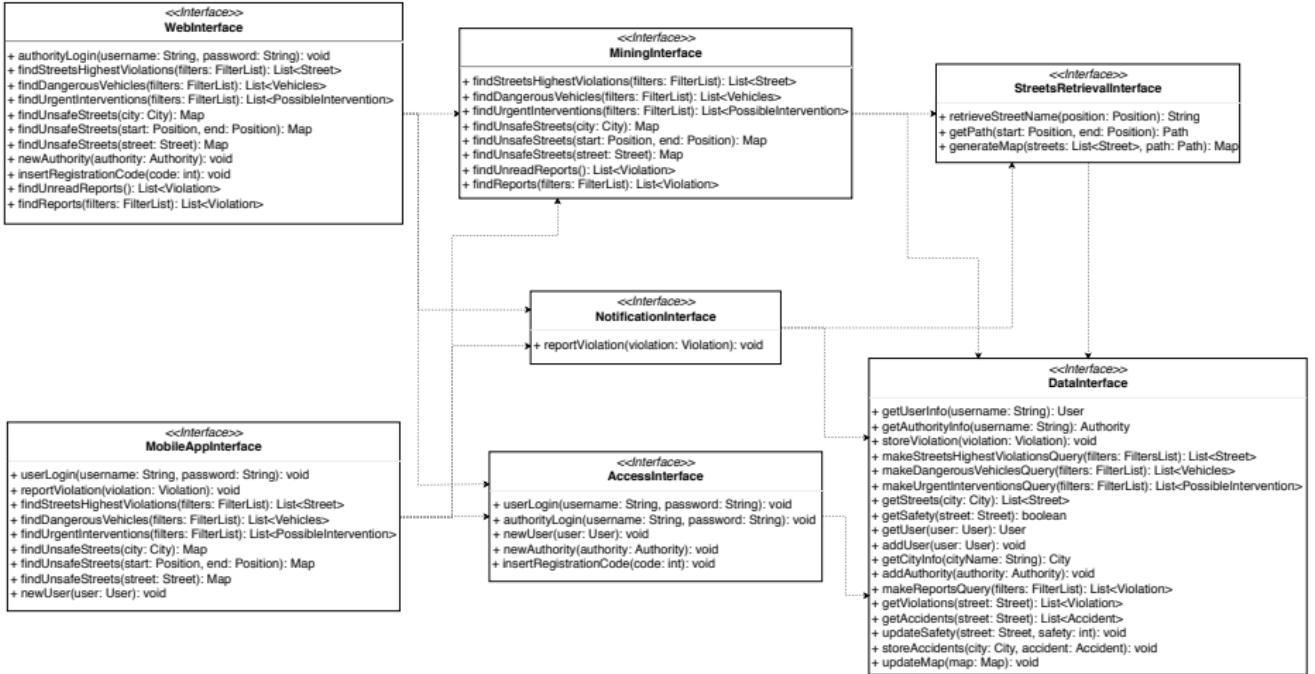


Server Component

16

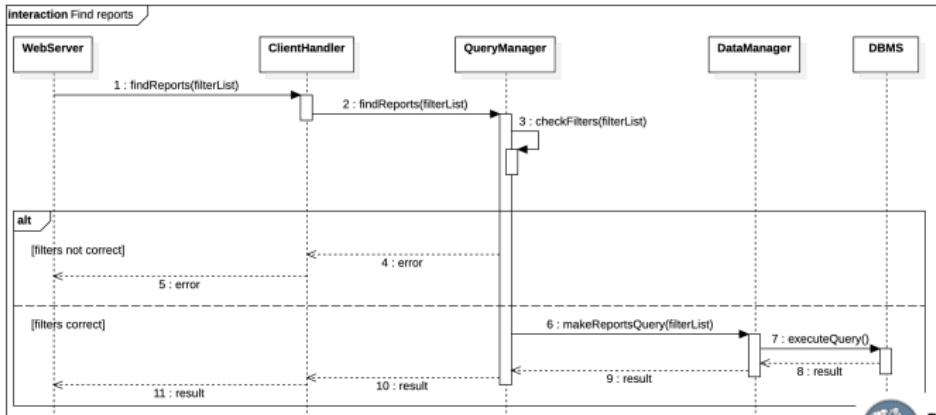
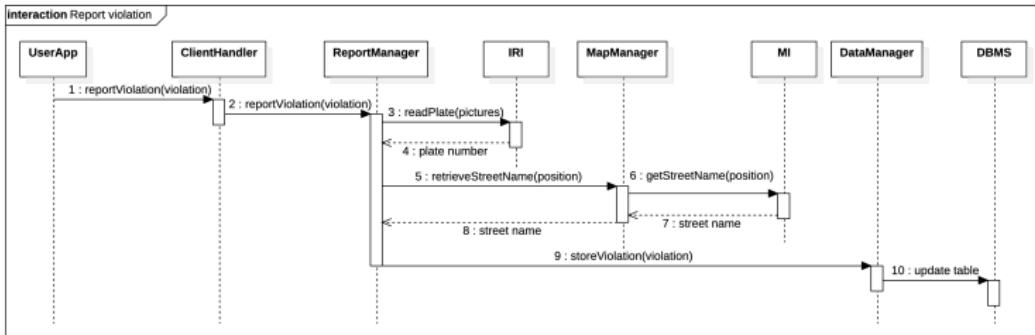


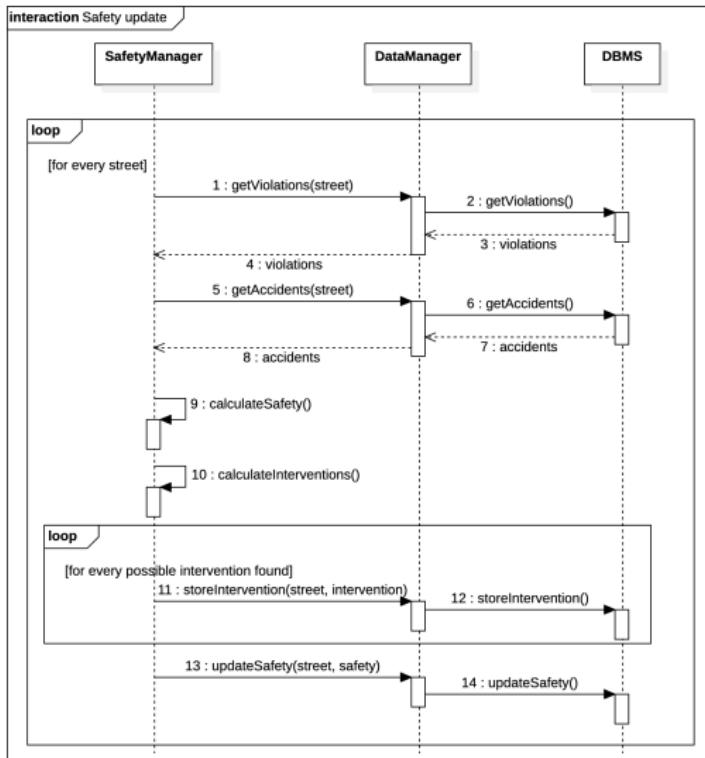
Component Interfaces



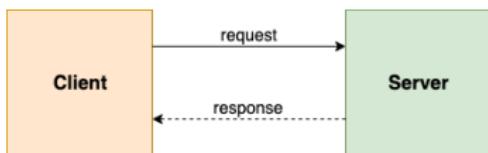
Report Violation and Find Reports

18

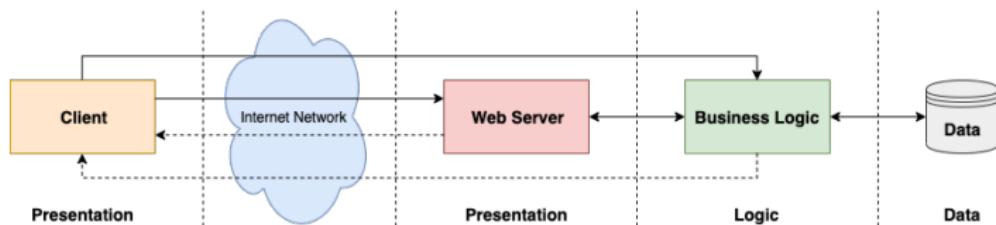




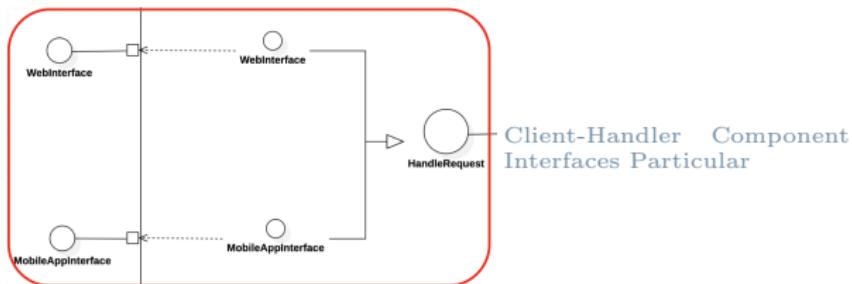
The system is a **crowd-sourced** application based on the client-server paradigm where the *users* represent the crowd on which the functionalities of SafeStreets are funded.



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.

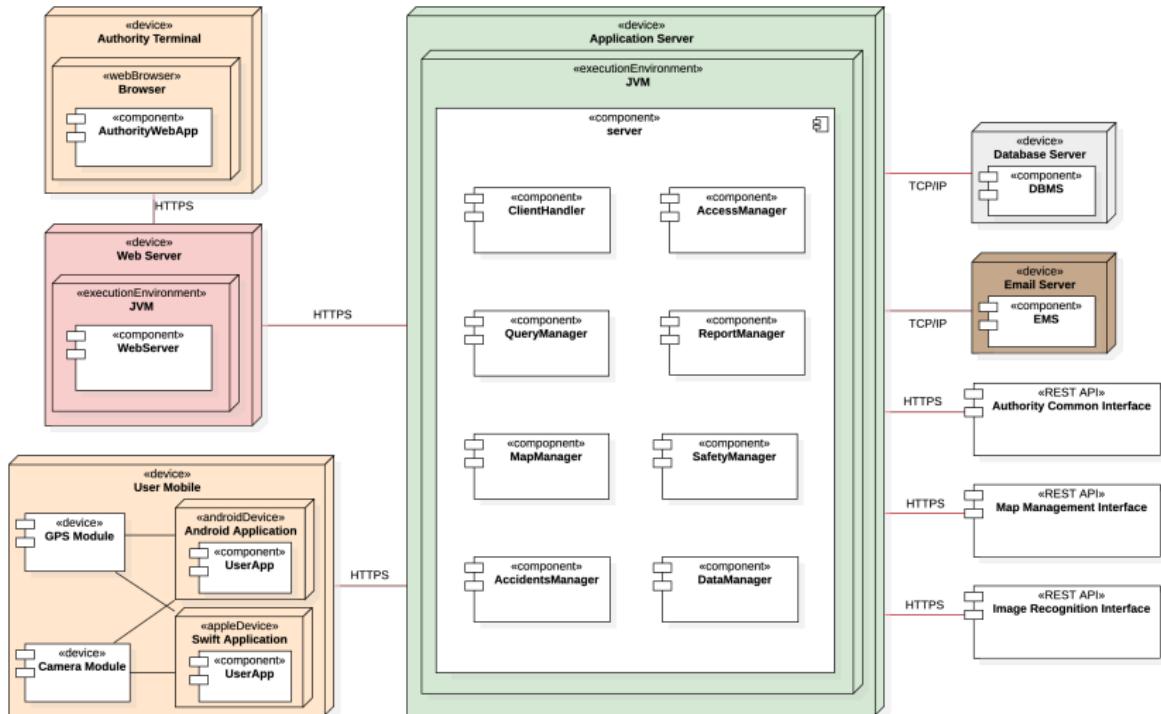


Moreover basic communication with the TCP/IP protocol is also used in particular for the external components that allows the system to manage its data: the *DBMS* and the *EMS*.



Deployment View

22



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

24

