POLITECNICO
MILANO 1863

SafeStreets

Design Document

Davide Cocco - 944122
Marco Gasperini - 944922

A.Y. 2019/2020 - Prof. Di Nitto Elisabetta

Contents

1 INTRODUCTION

1.1 Purpose

This `Design Document (DD)` for the SafeStreets software will provide a functional description of the system by describing its architecture. It will eventually by used by the development team as a blueprint to guide the engineering of the application.

1.2 Scope

The main objective of the S2B will be assisting (thus not substituting) authorities and officers in handling traffic violations though a crowd-sourced platform in which civilians can participate. A mobile application will allow users, both civilians and officers, to report violations through the use of the camera and the GPS, sending the data to authorities who will process such reports. Law enforcement will be aided by a data mining system which will produce relevant data about the registered violations, and the S2B will be able to automatically compile a ticket and send it to the municipality's VLA as soon as an officer personally convalidates a violation.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- `Violation | Offence:` we will strictly refer to any kind of static traffic violation, especially parking violations.
- `Authority:` a law enforcement authority which is manages traffic violations, it could be the local police, the traffic wardens etc. We will refer with this term also to the personnel which operates the web application in the authorities' headquarters.
- `User:` refers to the users of the mobile application, that is officers and civilians.

1.3.2 Acronyms

- `S2B:` software to be.
- `HFVZ:` high violation frequency zone.
- `GPS:` Global Positioning System.

- `GDPR`: General Data Protection Regulation.
- `DW`: data warehouse.
- `VLA`: Vehicle Licensing Authority.
- `RACS`: Reliable Array of Cloned Services.
- `RAPS`: Reliable Array of Partitioned Services.
- `GUI`: Graphical User Interface.
- `DMZ`: Demilitarized zone.
- `ADS`: Application and Data server of the authorities.

1.3.3 Abbreviations

- [Gn]: n-goal.
- [Rn]: n-requirment.
- [Dn]: n-domain assumption
- App: application.

1.4 Revision History

1.5 Reference Documents

1.6 Document Structure

- **Introduction**: includes the purpose and scope of the document along with some relevant definitions and acronyms used throughout the document.
- **Architectural design**: this section is focused on the main components used for this system and the relationship between them, providing information about their deployment and how they operate. it also focuses on the architectural styles and the design patterns adopted for designing the system.
- **User interface design**: this section provides an overview on how the User Interface will look like, but it will be omitted due to the presence of this chapter in the RASD.

- **Requirements traceability**: in this sections the requirements highlighted in the RASD document will be associated with the design elements explained in this document.
- **Implementation, integration and test plan**: in this section we identify the order in which we plan to implement the subcomponents of the system and the order in which we plan to integrate and test them.

2 ARCHITECTURAL DESIGN

2.1 High level overview

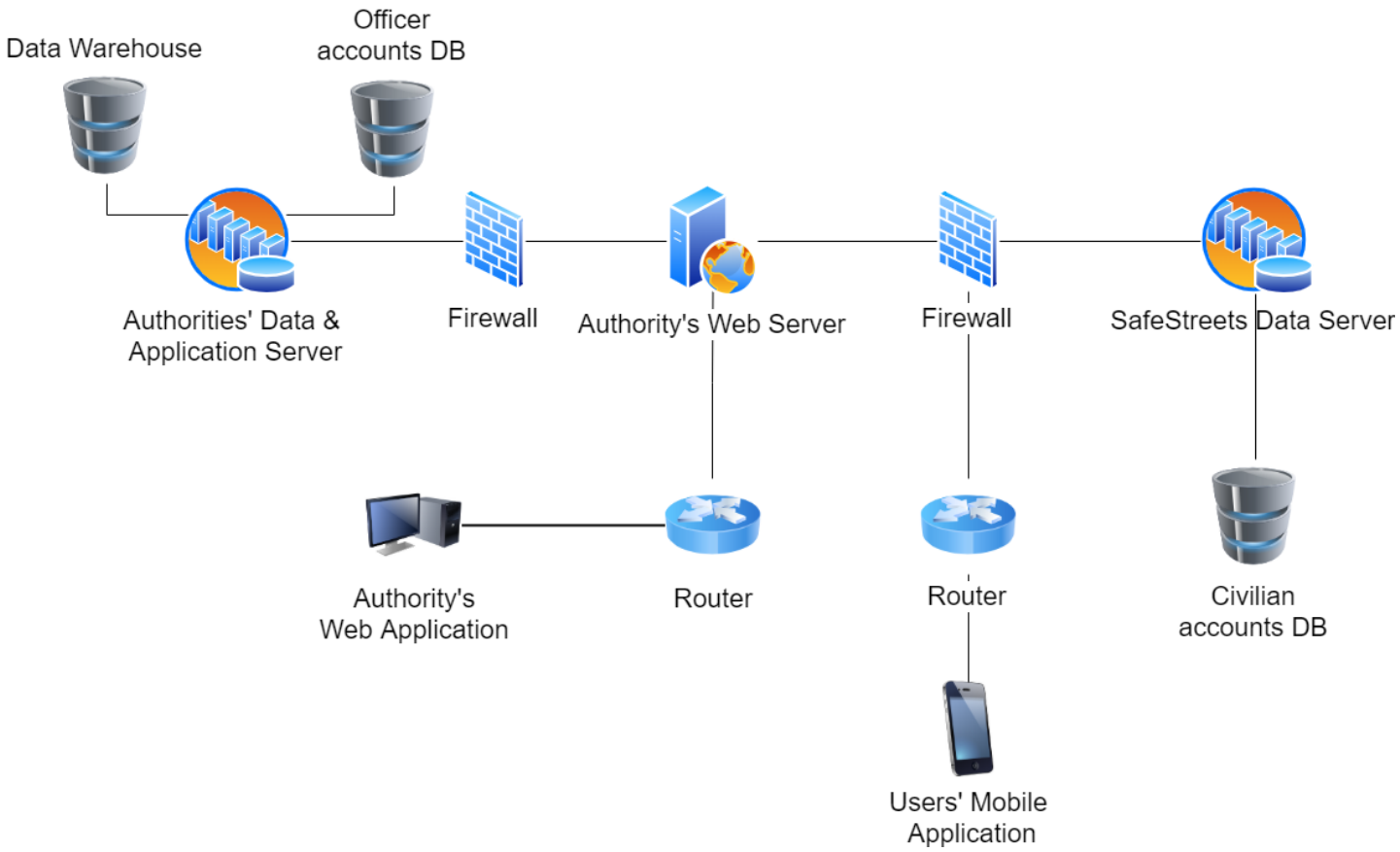

Figure 1: Figure 1: High Level Structure with a single Authority and a single User

The three logical layers of the S2B can be seen in different ways depending on the actor's client, as different actors can access different services. From the **Authority's perspective** we can see the layers distributed in a **two-tier architecture**:

- **Presentation layer:** the GUI at the authority's web application is the tool thanks to which the authority's personnel can add officers, receive reports from the users through HTTP calls thanks to a REST architecture embedded in the web server, and validate or invalidate them. It can thus interact with the Application and Data server (we

will call it ADS from now on) to which it sends validated reports and receives updates, the users' clients, and rarely with the SafeStreets' server to which it sends ban requests. The web server is isolated from the outside by firewalls for increased security and to protect the ADS even if the web server is compromised.

- **Application and Data layer:** the ADS manages application functionalities such as ticket emission and update spreading, and also data functionalities such as managing the officers' accounts in an ad-hoc database and enriching a standalone data warehouse with reports to be processed. This server's location can either be in a central authority location for reduced latency, a municipality etc. and can be replicated through a RAPS architecture, (e.g. every authority can have its own server) to be able to handle multiple requests in settings like big cities.

From the **Civilian's perspective** we can see the layers distributed in a **three-tier architecture**:

- **Presentation layer:** the GUI at the mobile application, which can be used to file reports and send them to the Authority's web server.

- **Application layer:** the hypermedia at the mobile application moves from a state to another thanks to the updates received by the authorities' web servers (which are themselves sent to the authorities by their application server), that are periodically spread to the clients in the network. This is done to avoid having the clients know the application and data servers' access points. This way the mobile application can only contact one of the authorities' web servers and all the report processing and updating procedures are black boxed from the users perspective.

- **Data layer:** the data server at the SafeStreets headquarters, which can be replicated through a RACS architecture to be simply scalable, manages civilian accesses and bans.

From the **Officer's perspective** we have a similar perspective as the civilian's, but officers' accounts are managed by the authorities' Application and Data server, thus we can see the layers distributed in a **two-tier architecture**:

- **Presentation layer:** the GUI at the mobile application which possesses the same features as the civilians' one but also includes the abilities to check a report and sign the automated ticket by inserting the

officer's secret PIN, and also receives slightly different updates which include sensitive data to be made unreadable by civilians.

- **Application and Data layer:** just as the civilians' perspective but in this case accounts are managed in the same location where the application functionalities resiede (as officers account aren't located in the DB at SafeStreets headquarters) and thus the application and data layers are merged. Officers, just as civilians, only communicate directly with one of the city's authorities' web server which then handles interacting with the Application and Data server to provide the functionalities.

It is important to denote the fact that since every Authority owns a web server, calls received from the mobile units can be handled by any of the authorities for load balancing. The only logic which will be implemented at the authorities' client will be the supervised algorithm to recognize plates: this way we can reduce latency, because since the overhead to effectively send a correct plate frame will be pretty heavy, we remove the step to contact the application server and instead execute this function directly at the authorities' location. Anything else will be handled as notifications forwarded to the ADS to limit the computational power used at the authorities' locations.

2.2 Component view

The following diagrams illustrate the system components and the interfaces through which they interact to fullfil their functionalities. The diagram is focused on the application tier, so the remaining tiers (the presentation tier and the data tier) are shown as black-box. First we have done a distinction between Client side and Server side:

- The Client side is composed by two components, *Web Application* and *Mobile Application*. The first is referred to the Authorities client, latter to the User client (Civilian and Officer).

- The Server side is composed of three main components: *Authorities Web Services* manages the authorities client, through the *AuthoritiesWebServices* interface server-side; the other two services are the *Civilian Mobile Services* and the *Officer Mobile Services* that manage the *Mobile Application* from two distinct interfaces.

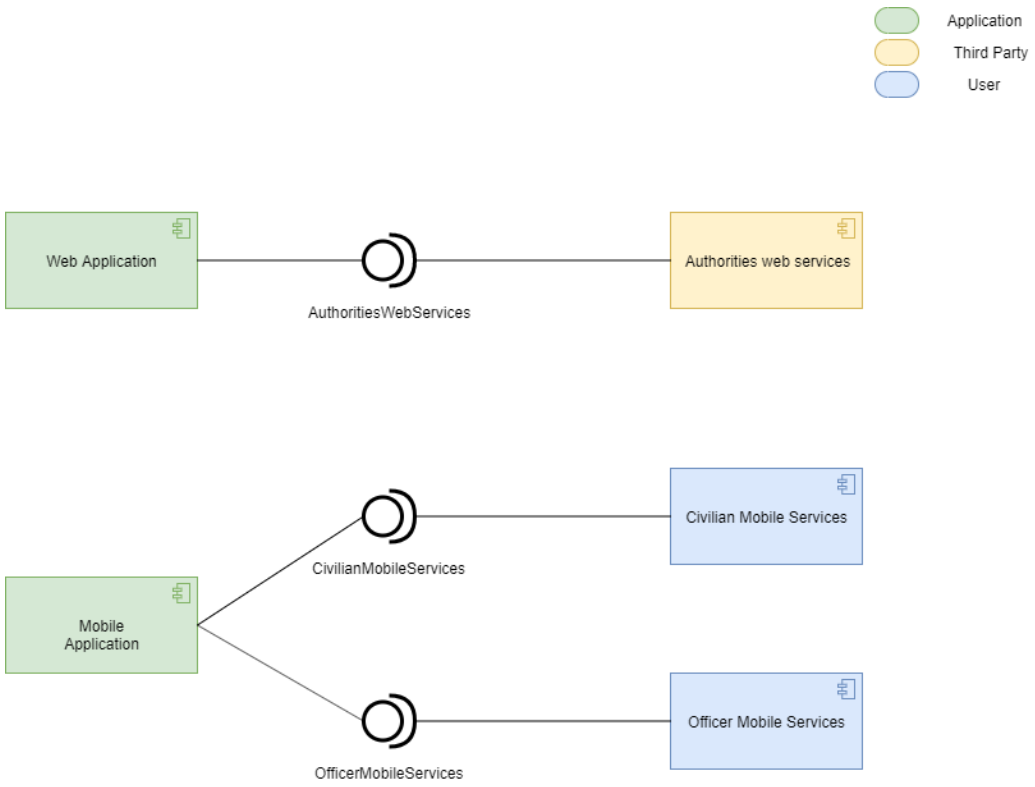

Figure 2: Figure 2: Component diagram

- Davide Cocco

Day	Subject	Hours
Total		22.5

- Marco Gasperini

Day	Subject	Hours
Total		19.5

7 References