# 

**S4 Smart Surveillance System   
Technical manual   
[Version 1.0]**

Contents

[Application / Network Components 5](#_Toc175147065)

[Data Flow 7](#_Toc175147066)

[Application API 8](#_Toc175147067)

[Server Administration 9](#_Toc175147068)

[Socket Server 9](#_Toc175147069)

[MySQL Server Administration 9](#_Toc175147070)

[Apache Server Administration 9](#_Toc175147071)

[Socket Server Logs 9](#_Toc175147072)

[Application Server Logs 9](#_Toc175147073)

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version*** | |  |  |
| *Version* | *Date* | *Author* | *Description* |
| **V1.0** | 20-Aug-2024 | Shakil Momin | Initial version |
|  |  |  |  |

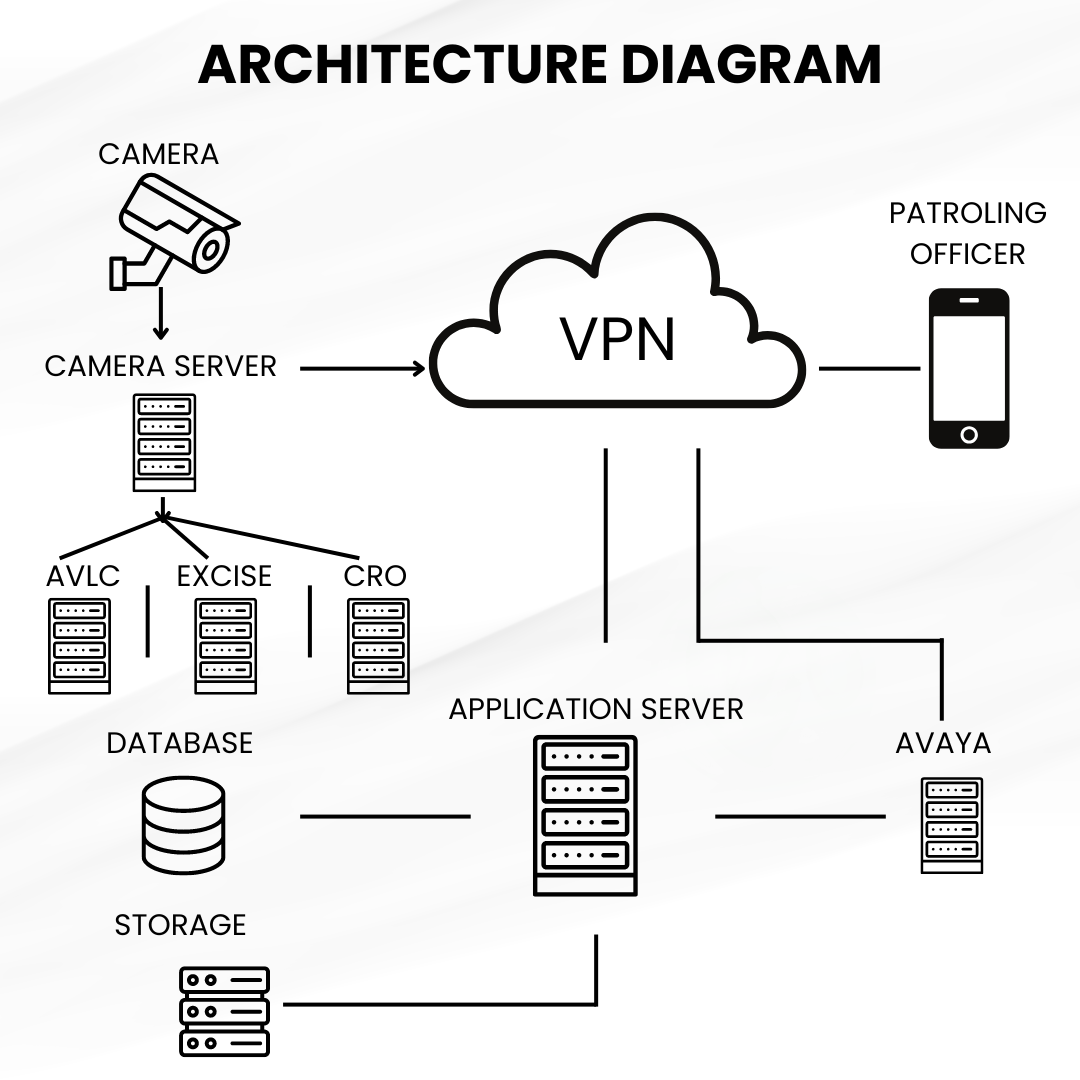
Distribution List

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version*** |  |  |  |
| *Name* | *Position* | *Reviewer* | *Approver* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Confidentiality Notice:**

***This document, along with any attached files, is provided in confidence to the recipients listed above and is intended solely for evaluation purposes. Disclosure or sharing of this information with outside parties is prohibited without express written consent. If you are not the intended recipient, please refrain from using any part of this document and destroy it immediately, notifying the original sender of the action taken.***

**Following diagram is showing flow of data in application to generate violation alert for patrolling officers.**



# Application / Network Components

* **Camera**
  + The cameras are installed at the toll plaza to capture images of vehicle and rider.
  + These cameras are connected to the **Camera Server**, where the captured images are processed.
* **Camera Server**
  + The camera server is responsible for processing the images captured by the cameras. It performs Optical Character Recognition (OCR) to extract license plate numbers from the images.
  + The server is connected to multiple external API (AVLC, Excise, CRO) to verify the details of vehicle and rider.
* **External API**
  + **AVLC (Anti-Vehicle Lifting Cell)** stores records of stolen, snatched, wanted, and burnt vehicles.
  + **Excise** provide registration details, including vehicle color and manufacturer information.
  + **CRO (Criminal Record Office)** Holds records related to criminal.
* **VPN (Virtual Private Network)**
  + Securely connects the Camera Server with the Application Server and patrolling officers' devices, ensuring that data transmission is encrypted and protected from unauthorized access.
  + Patrolling officers access the system through this VPN to receive real-time alerts and updates.
* **Application Server**
  + The core component that manages the application logic, processes queries from the Camera Server, and handles user interactions from both the patrolling officers and the admin interface.

**Functions:**

* + - Processes data received from external API.
    - Generates alerts for detected violations.
    - Manages user roles and permissions for patrolling officers and administrators.
    - Integrates with AVAYA for alert management and communication.
* **Database**
  + Centralized storage for all processed data, including license plate records, alerts, and acknowledgment logs.
  + Handles queries and data retrieval for both real-time operations and historical data analysis.
* **Storage Server**
  + Provides additional storage capacity for large volumes of data, particularly images and logs generated by the system.
* **AVAYA System:**
  + An integrated communication system that may be used for handling alerts, notifications, and direct communication between patrolling officers and the control center.
  + Connected to the Application Server for managing alert acknowledgment and other communication functions.

# Data Flow

**Step 1: Vehicle and Rider Photo Capture**

The system initiates surveillance by capturing images of vehicles and riders as they pass through the camera installed at toll plaza.   
**High-Resolution Cameras are** installed to capture photos of license plates and vehicle.

**Step 2: Photo Processing**

Captured photos are OCR for plate number, vehicle make, color, mobile use and seatbelt unfasted.

Using AVLC, Excise and CRO API, application will generate violation alerts for on duty patrolling office for respected location.

**Step 3: Violation Notification**

Camera application will notify violations to S4 API. S4 API will store and process violation and notify on web socket to respected on duty patrolling officer.

Patrolling officer can submit action details with photo evidence.

# Application API

Developer can load postmen collection for API documentation.

Below are postman collection and environment setting file location in API git repository.

* tollplaza-api/docs/tollplaza-postman-environment.json
* tollplaza-api/docs/tollplaza-postman-collection.json

|  |
| --- |
| Refer Full documentation: https://learning.postman.com/docs/introduction/overview/ |

# Server Administration

## Socket Server

Socket server process has been manage using PM2.

Using below command you can run socket process.

* pm2 start 'node /var/www/html/tollplaza-socketserver/index.js > /var/www/html/tollplaza-socketserver/output.log' --name 'Toll Plaza Socket Server'

Using below command you can list all process manage using PM2.

* pm2 list

Using below command you can restart pm2 process.

* pm2 restart <<PROCESS\_ID>>

|  |
| --- |
| Refer Full documentation of PM2: https://pm2.keymetrics.io/ |

## MySQL Server Administration

MySQL Service Status

* sudo service mysql status

MySQL Service Status

* sudo service mysql status

|  |
| --- |
| Refer Full documentation of MySQL: https://dev.mysql.com/doc/ |

## Apache Server Administration

Apache Service Status

* sudo service apache2 status

|  |
| --- |
| Refer Full documentation of Apache: https://ubuntu.com/server/docs/how-to-install-apache/ |

## 

## Socket Server Logs

Socket Server log in below file.

* /var/www/html/tollplaza-socketserver/output.log

## Application Server Logs

Application Server date wise log will find in below folder.

* /var/www/html/tollplaza-api/storage/logs/