LG Software Architectures Training Program

**<ALPR Project SW Architecture Document>**

**Document Information**

|  |  |
| --- | --- |
| Issuing authority | Team ONE |
| Status of document | In Progress |

**Revision History**

|  |  |  |
| --- | --- | --- |
| Version | Date | Comment |
| 0.1 | 2022-06-15 | - Initial Draft |
| 0.2 | 2022-06-16 | - Append: Functional Requirements, Quality Attributes, Design Constraints |
| 0.3 – 0.7 | 2022-06-20 | - Revise/Update: Functional Requirements, Add more Quality Attributes items  - Append: Architectural Approach and Analysis for each QA |
| 0.8 | 2022-06-21 | - Revise: Quality Attributes table, tactics, risks, tradeoffs, QA scenarios  - Append: QA table with Priority Decision(Utility Tree) |
| 0.9 | 2022-06-22 | - Append: System Context Diagram for overall SW Architecture |
| 1.0 | 2022-06-23 | - Remove: System Context Diagram  - Update : Overall SW Architecture for module views |
| 1.1 | 2022-06-29 | - Append: Add QA#7, QA#7 QA Table.  - Revise: QA Table Priority Decision (Utility Tree), SW Overall Architecture. |
| 1.2 | 2022-07-06 | - Append: Context Diagram, View(Module), Views(Runtime(C&C), Allocation)  - Update: Planned Test Result update, Test Plan, Risk  - Append: ADR(Architecture Decision Record) |
| 1.3 | 2027-07-18 | - Update: Overall ALPR Runtime View  - Append: Component references for each runtime views. |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of Contents**

[1 Project Summary 3](#_Toc109057514)

[1.1 Project Description 3](#_Toc109057515)

[1.2 Project Plan 3](#_Toc109057516)

[1.3 Project Team & Role 4](#_Toc109057517)

[1.4 Project Team Member 4](#_Toc109057518)

[2 Architecture Drivers 5](#_Toc109057519)

[2.1 Functional Requirements 5](#_Toc109057520)

[2.1.1 FR - Laptop Application 5](#_Toc109057521)

[2.1.2 FR – UI Display 6](#_Toc109057522)

[2.1.3 FR – Server Application 6](#_Toc109057523)

[2.2 Quality Attributes 7](#_Toc109057524)

[2.2.1 Quality Attributes Table 7](#_Toc109057525)

[2.2.2 Architectural Approach / Analysis 8](#_Toc109057526)

[2.2.2.1 Architectural Approach / Analysis - QA#1 8](#_Toc109057527)

[2.2.2.2 Architectural Approach / Analysis - QA#5 9](#_Toc109057528)

[2.2.2.3 Architectural Approach / Analysis - QA#7 10](#_Toc109057529)

[2.3 Design Constraints 11](#_Toc109057530)

[2.4 Planned Experiments 12](#_Toc109057531)

[3 SW Architecture 13](#_Toc109057532)

[3.1 Project Context 13](#_Toc109057533)

[3.1.1 Context Diagram 13](#_Toc109057534)

[3.2 ALPR Overall SW Architecture 13](#_Toc109057535)

[3.3 Module View 15](#_Toc109057536)

[3.3.1 Overall 15](#_Toc109057537)

[3.3.2 Laptop Application 17](#_Toc109057538)

[3.3.3 Server Application 19](#_Toc109057539)

[3.3.4 Mobile Application 21](#_Toc109057540)

[3.4 Runtime (Component & Connector) View 22](#_Toc109057541)

[3.4.1 Overall ALPR Runtime View 22](#_Toc109057542)

[3.4.2 ALPR Frame Processing Detail View 23](#_Toc109057543)

[3.4.3 ALPR Plate Manager Detail View 24](#_Toc109057544)

[3.4.4 Best Matching Processing Detail View 26](#_Toc109057545)

[3.5 Allocation View 27](#_Toc109057546)

[4 Architecture Decision 28](#_Toc109057547)

[4.1 Update tesseract library for safe multi-thread processing. 28](#_Toc109057548)

[4.2 Applying hybrid sampling rate control 30](#_Toc109057549)

[4.3 Choose alternatives for selecting best plate in plates of group. 33](#_Toc109057550)

# Project Summary

## Project Description

The ALPR project is a team project within the CMU course, and will proceed based on what is learned within the CMU course.

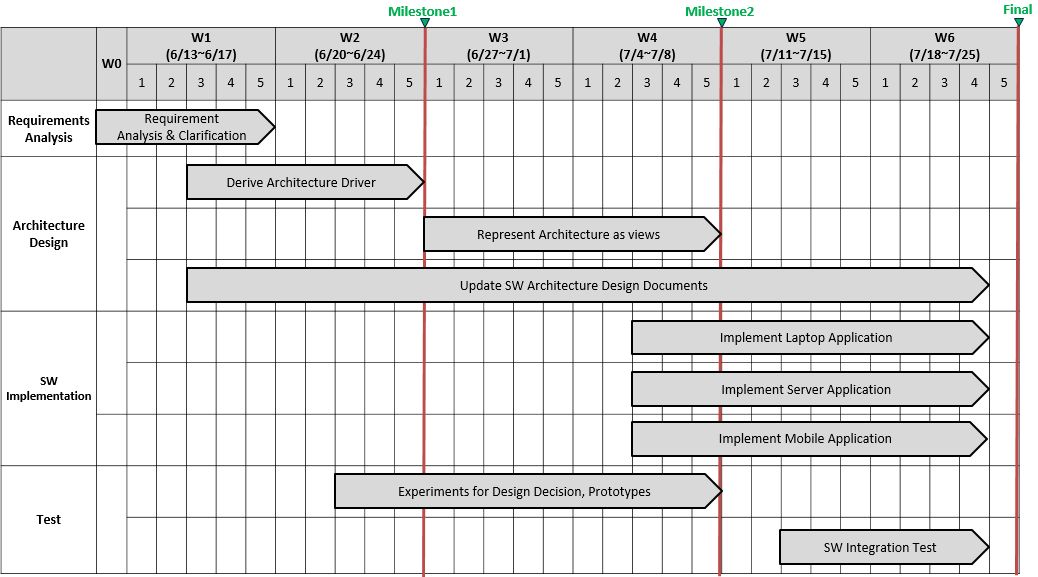
The basic function is to recognize the plate for each frame through image file recognition, etc.

Depending on the status, functions such as alert notifications are performed. Quality attributes such as maintaining frame rate, accuracy and secure information must be satisfied.

Various tactics will be used to improve the corresponding Quality Attribute.

## Project Plan

ALPR Project is planned as a project of 6 weeks, and the detailed plan is as follows:



## Project Team & Role

The roles of the ALPR Project are divided as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | | **Output** | **Owner** |
| Requirements Analysis & Clarification | | NA | All |
| Design | Derive Architecture Drivers and  perform ATAM | Functional Requirements  Quality Attribute requirements,  Design constraints  Risk Assessment/Planned Experiments | All |
| Represent Architecture as views | Module view  C&C view  Deployment view  Architectural Decisions and Reasoning | All |
| Update SW Architecture Design Documents | SW Architecture Design Documents | All |
| Implementation | Laptop Application | Code implementation | Dangwoo Choi  Bonhun Koo  Jungkwan Son  Sejoon Pyo |
| Backend server Application | Code implementation | Eunjung Lim  Tran Anh Tai |
| Mobile Application | Code implementation | Daemyeong Park |
| Test | Experiments list for Design Decision | Test Case & Result | All |
| SW Integration Test | Test Case & Result | All |

## Project Team Member

1. Dangwoo Choi
2. Bonhun Koo
3. Jungkwan Son
4. Eunjung Lim
5. Daemyeong Park
6. Sejoon Pyo
7. Tran Anh Tai

# Architecture Drivers

## Functional Requirements

### FR - Laptop Application

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Req.** | **Description** |
| FR-01 | Login / Authenticate | Laptop application shall support to authenticate user to access to the backend license plate database look up system. |
| FR-02 | Secure Communication | Laptop application shall provide secure communication channel between the laptop application and the backend license plate database lookup system. |
| FR-03 | Read Images | Laptop application shall support to read images from a playback file ( .avi, .wav, .mp4, etc.) |
| FR-04 | ALPR function | Laptop application shall support to ALPR function to identify license plates. |
| FR-05 | Alert Target | Laptop application shall provide alerts to users and operators.  1. Users - for data about details of queries.  2. Operators - any communication error or failures. |
| FR-06 | Alert Components | Laptop application needs to provide the following information of alerts:  1. Alert reason  2. Vehicle owner name  3. Vehicle owner address  4. Vehicle make  5. Vehicle model  6. Vehicle color  7. Vehicles are Stolen / Owner of the vehicle Wanted, or criminal /  Vehicle of interest (e.g. expired registration, unpaid tickets, owner is missing) |
| FR-07 | Monitoring Network | Laptop application shall monitor network status. The laptop application can detect network connectivity issues with the backend server if it can not communicate with the backend server. |

### FR – UI Display

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Req.** | **Description** |
| FR-08 | UI - Login | Laptop application shall provide user login UI. |
| FR-09 | UI - Logout | Laptop application should provide user logout UI |
| FR-10 | UI – Display last recognized | If a license plate does not generate an alert, then the laptop application UI needs to display the following things the last recognized:  1. plate image  2. license plate number  3. vehicle make  4. vehicle model  5. vehicle color |
| FR-11 | UI - Playback view Display | Laptop application UI shall display the following things:  1. camera/playback fps (frames per second).  2. average time per frame  3. jitter  4. frame number |

### FR – Server Application

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Req.** | **Description** |
| FR-12 | Response for Queries | Server application shall provide responses for license plate quires from laptop application. |
| FR-13 | Communication | Server application shall provide secure communicate with laptop applications. |
| FR-14 | Multiple User Access | Server application shall support multiple user accesses. |
| FR-15 | Multiple Queries | Server application shall support multiple quires from multiple users, and server application shall response each query for each user. |
| FR-16 | Partial Matching | Server application shall return the best match license if there is not an exact match.  1. Supports to configurable minimum confidence threshold |
| FR-17 | Number of Queries | Server application shall track the following things of queries:  1. Avg. number of Queries per second, for each user.  2. Overall number of Queries per second, for all users. |
| FR-18 | Number of partial matches and no matches | Server application shall track the number partial matches and no matches for each user and all users. |
| FR-19 | Configuration File | Server application shall support configurable values using configuration file. |

## Quality Attributes

### Quality Attributes Table

The QAs marked with a Red rectangle are the highest priority of QAs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Req. Name** | **Quality Attributes** | **Description** | **Difficulty (Technical)** | **Importance**  **(Business)** |
| QA-01 | Frame Rate | Performance | Users are playing the video/playback file for recognizing licensing plate. And then ALPR system is processing video data, which can cause peak (high) loads. Nevertheless, the ALPR system needs to keep the frame rate of that video at least 25 fps. | High | High |
| QA-02 | Detect Errors | Availability | Users request for querying the information which is in the backend server to ALPR system. When a network connectivity issues with the backend server occurs during the request and response process, the ALPR system detects it within 5 seconds. | Mid | High |
| QA-03 | Configurable data | Integrability | Users want to update configurable value in server application of ALRP system on Run-time. Server application can be applying updated value for configuration file and start to operate with the updated configurable data within 1 second. | Low | Mid |
| QA-04 | Authorized User Role | Security | To access the ALPR system, user shall login the system on run-time. The ALPR system check the login data and check the user’s authority. The ALPR system can deny access to unauthorized users 100% of the time. | Mid | Mid |
| QA-05 | Authentication process | Security | To access the ALPR system, user shall login the system on run-time. For enhanced user authentication, the ALPR system shall sequentially proceed at least 2 authentication methods. | High | High |
| QA-06 | Response time | Performance | In case of high load traffic, which means there are multiple requests from Laptop application on different devices at same time, Server application should return response for each request within 1 second. | High | Mid |
| QA-07 | Accurate Matching | Accuracy | The ALPR laptop application will extract the recognized plate license number from the video being played at runtime. If laptop application sends all of these extracted names to the Server, laptop application may arrive with incorrect number information. The ALPR laptop application must select the optimal license plate number from among the license plate number that appears to be the same vehicle.  At this time, the reliability of the selected plate information consistent with the actual vehicle information should be at least 85%. | High | High |

### Architectural Approach / Analysis

This section covered the highest priority Quality Attributes architectural approach and analysis.   
The rest of QAs are covered in the Appendix Section.

#### Architectural Approach / Analysis - QA#1

|  |  |
| --- | --- |
| **Parts** | **Description** |
| ID | QA - 01 |
| Scenario | Users are playing the video/playback file for recognizing licensing plate. And then ALPR system is processing video data, which can cause peak (high) loads. Nevertheless, the ALPR system needs to keep the frame rate of that video at least 25 fps. |
| Quality Attribute | Performance |
| Stimulus | Playing video / playback file for recognizing license plate. |
| Source of Stimulus | Users |
| Artifact | Automatic License Plate Recognition System - Laptop Application |
| Environment / Context | At peak(high) load(overloading video processing) |
| Response | Displaying a frame |
| Response Measure | Maintaining frame rate at least 25fps. |
| Architectural Approaches | Approach 1(#A1). Applying for the “**Maintain multiple copies of computations**” tactic.  - If ALPR system process using multiple instances that perform the same operation, “Performance” can be improved by processing the ALPR function for multiple frames in parallel.  Approach 2(#A2). Applying for the “**Manage work requests (Manage sampling rate)**” tactic.  - By adjusting the sampling frequency of the frame, it is possible to reduce the ALPR operation for each frame and improve the performance because of reducing the duplicated request/response.  Approach 3(#A3). Applying for the “**Introduce concurrency**” tactic.  - By separating the threads, it is possible to improve the performance of video processing by separating the thread that recognizes the license plate and the thread that displays video frame. |
| Risks | Risk 1: Is it possible that we display License Plates data sequentially in order from the video recognition?  Risk 2: Accuracy for recognition  Risk 3: Time gap of playback view and last recognized license plate image |
| Tradeoffs | Modifiability - Increased Design complexity.  Accuracy - Reduce the number of samples to be queried. It causes for low accuracy recognition. |

#### Architectural Approach / Analysis - QA#5

|  |  |
| --- | --- |
| **Parts** | **Description** |
| ID | QA - 05 |
| Scenario | To access the ALPR system, user shall login the system on run-time. For enhanced user authentication, the ALPR system shall sequentially proceed at least 2 authentication methods. |
| Quality Attribute | Security |
| Stimulus | Request login the system |
| Source of Stimulus | User |
| Artifact | ALPR system |
| Environment/Context | on Run-time |
| Response | Provide authentication process |
| Response Measure | at least 2 authentication methods |
| Architectural Approaches | Approach 1(#A1). Applying for the “**MFA (2FA) - Authenticate factor**” tactic.  - To support improved authentication, MFA (2FA) tactic is applied. (ID/PW, biometric authentication through user mobile phone). The backend server performs the role of the identity provider, and is responsible for storing user authentication information managing the effective time, and allocating security tokens. |
| Risks | Risk 1: Biometric Authentication - Accuracy, Is it possible to use what is supported by the platform?  Risk 2: How many authentication methods to be used? Are two methods okay or more methods are needed? |
| Tradeoffs | Modifiability(Complexity), Usability |

#### Architectural Approach / Analysis - QA#7

|  |  |
| --- | --- |
| **Parts** | **Description** |
| ID | QA - 07 |
| Scenario | The ALPR laptop application will extract the recognized plate license number from the video being played at runtime. If laptop application sends all of these extracted names to the Server, laptop application may arrive with incorrect number information. The ALPR laptop application must select the optimal license plate number from among the license plate number that appears to be the same vehicle.  At this time, the reliability of the selected plate information consistent with the actual vehicle information should be at least 85%. |
| Quality Attribute | Accuracy |
| Stimulus | License plate numbers which are recognized in the playing video. |
| Source of Stimulus | Video file |
| Artifact | ALPR Laptop Application |
| Environment/Context | on runtime |
| Response | Extracted plate number |
| Response Measure | The confidence ratio of the extracted plate number matching with the actual vehicle information shall be at least 85%. |
| Architectural Approaches | Approach 1(#A1). Applying for the “**Manage work requests**” tactic.  - A license plate group is created by grouping similar license plates of cars shown in the video, and a license plate that can be the best match among the license plate groups is selected and searched on the server for managing work requests.  Since the number of server queries can be reduced, incorrect license plate information can be purified in advance, and accuracy can be increased. |
| Risks | Risk 1: miss-picked plate number due to pre-purification process  Risk 2: How to choose the best number from among the grouped license plate candidates? |
| Tradeoffs | Performance - Slowing down due to plate grouping and pre-refining process (filter) |

## Design Constraints

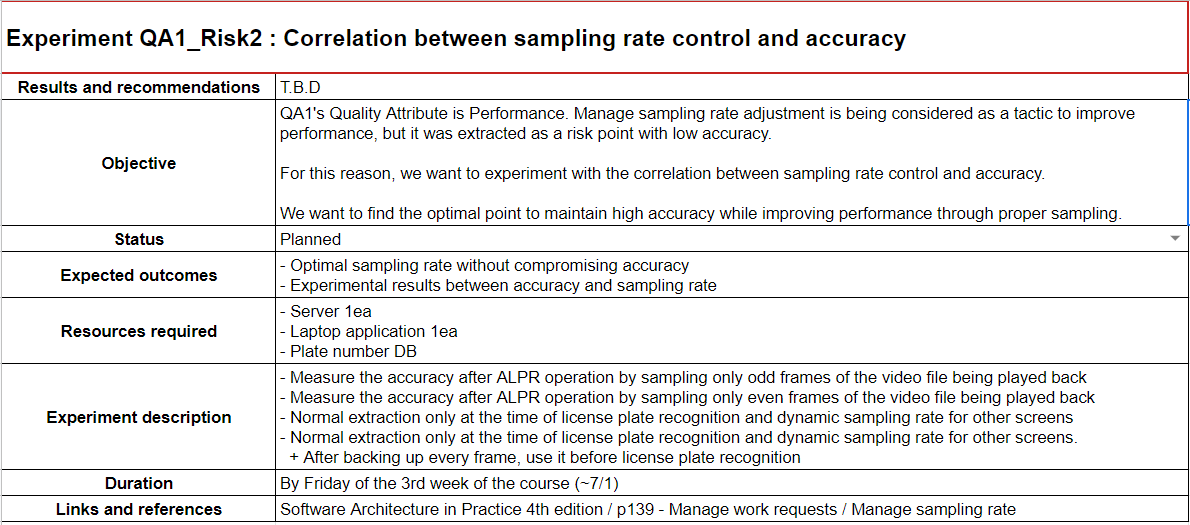
|  |  |  |
| --- | --- | --- |
| **ID** | **Constraint Type** | **Description** |
| CST-01 | Technical | System (Laptop Application & Server Application) needs to be developed using C++ Language. |
| CST-02 | Technical | Laptop Application needs to be running on Windows OS 10/11. |
| CST-03 | Technical | Laptop application and backend license plate database server needs to be run on separate devices. (Prototype Project: Uses two or more separate laptops.) |
| CST-04 | Technical | Laptop application communicates with backend database server application via 5G cellular network.  (Prototype Project: communicates via Wi-Fi of laptops.) |
| CST-05 | Technical | Backend server needs to be running on Windows OS 10/11. |
| CST-06 | Technical | Applications will be developed using Visual Studio 2022 Community Edition with OpenCV 4.5.5 |
| CST-07 | Technical | Backend Server will be implemented using Berkeley DB |
| CST-08 | Business | The number of plate using this application is up to millions unit. |
| CST-09 | Business | The number of development person is limited to 10, and the prototype must be developed for 6 weeks during the project period, and applied to the actual vehicle must be completed in this year. |

## Planned Experiments

Refer to following URL for all planned Experiment lists.

(https://docs.google.com/spreadsheets/d/1saaV-iwNsuR7OsR792jW3BRw4hqMHjeEyJ\_aBnxMQwM/)

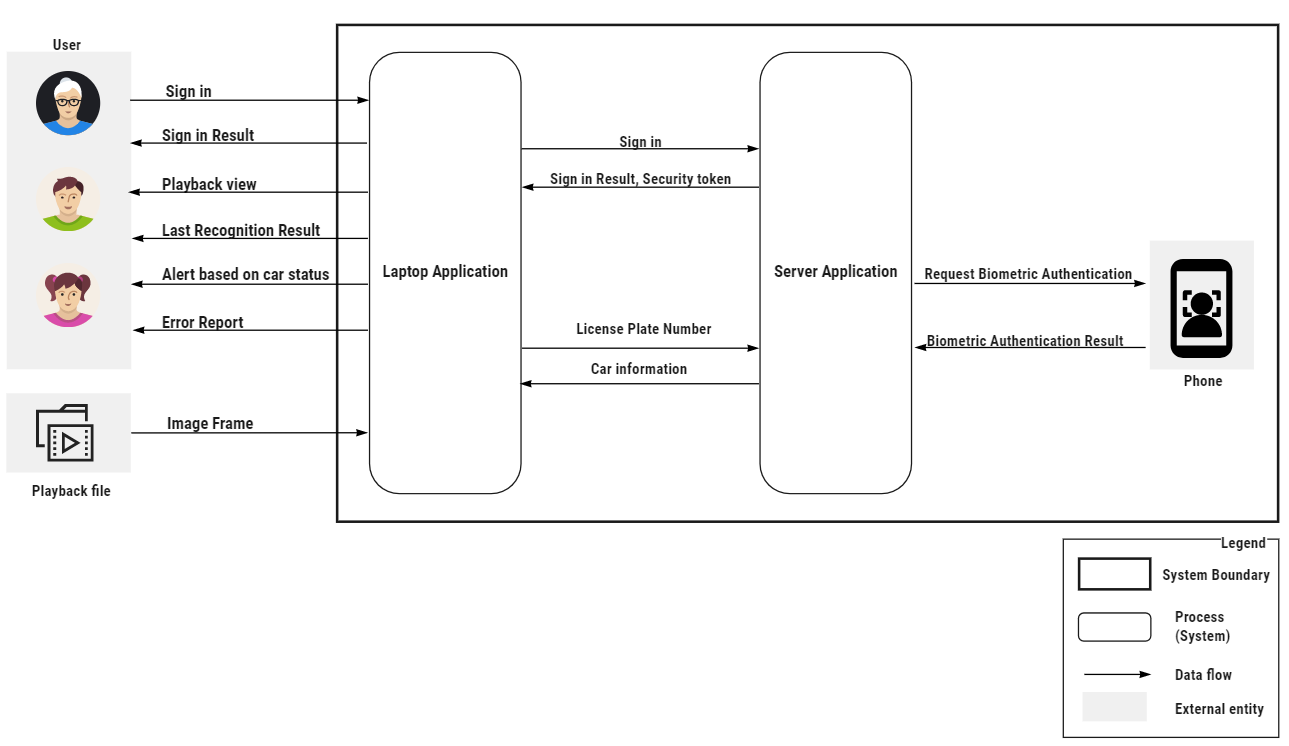
<Example>



# SW Architecture

## Project Context

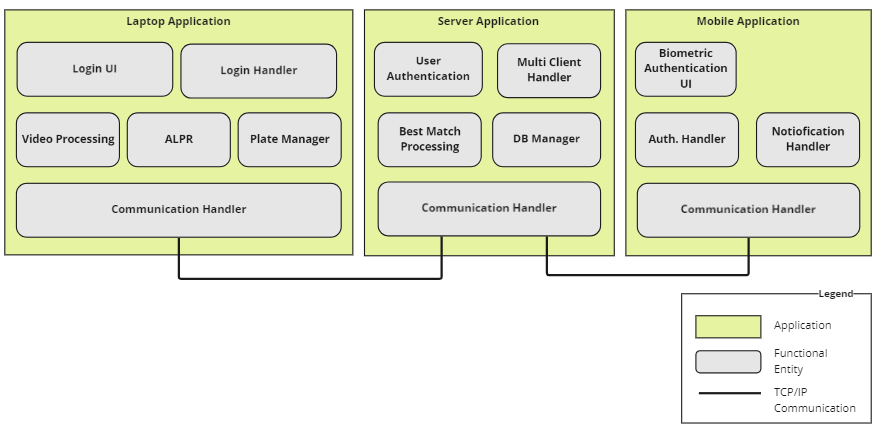
### Context Diagram



**Figure 3-1. Context Diagram**

## ALPR Overall SW Architecture

The ALPR System consists of three applications, and each application consists of functional entities to provide Functional Requirements and Quality Attributes.

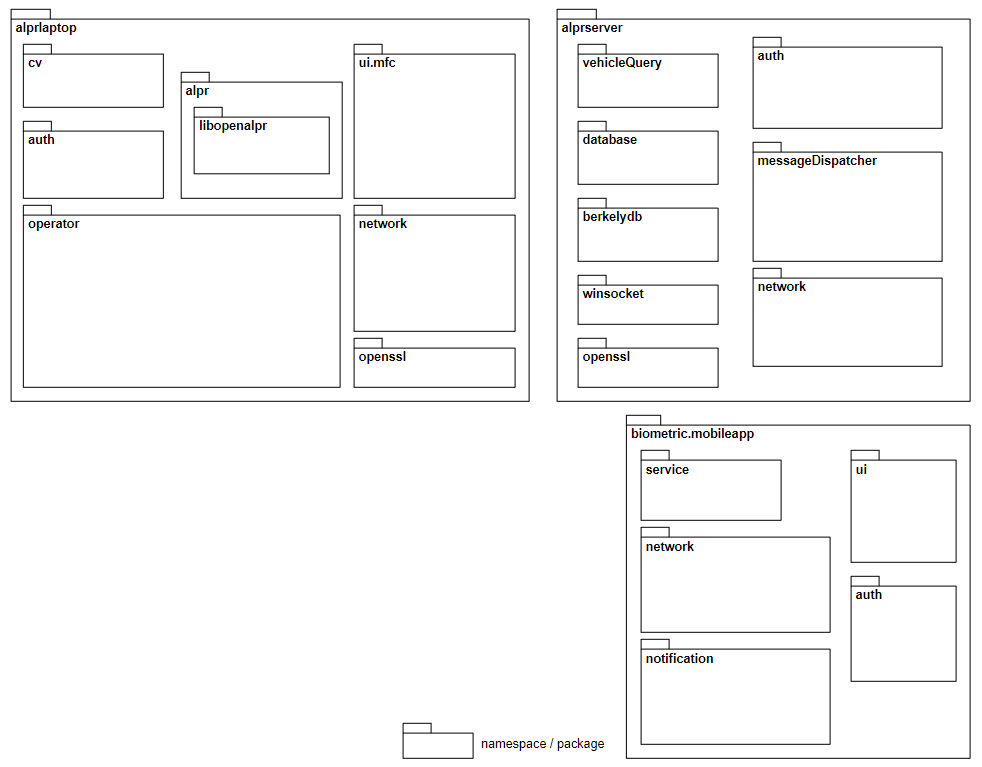


**Figure 3-2. ALPR Overall SW Architecture**

| **Functional Entity** | | **Description** |
| --- | --- | --- |
| Laptop Application | Login UI | Responsible for UI processing to provide user login |
| Login UI Handler | When a user requests a login, it passes it to the server and processes the response. |
| Video Processing | Image preprocessing to improve the recognition rate or performance of ALPR |
| ALPR  (Automated License Plate Recognition) | License plate recognition within input frame using AI model |
| Plate Manager | Post-processing to improve accuracy for license plates recognized by ALPR |
| Communication Handler | Responsible for TCP communication with Server Application and message processing |
| Server Application | User Authentication | Handling user login requests coming from Laptop Application. Saving and managing user login information |
| Multi-Client Handler | Information storage and connection management for multiple clients |
| Best Match Processing | Preprocessing to find the best match for the license plate queried by the Laptop Application |
| DB Manager | Manages DB that stores license plate and vehicle information, and returns vehicle information for license plate query |
| Communication Handler | Responsible for TCP communication with Server/Mobile Application and message processing |
| Mobile Application | Biometric Authentication UI | Process Biometric Authentication information entered through UI |
| Authentication Handler | Process the authentication result and pass it to the server application |
| Notification Handler | Processing to notify the user when authentication is requested from the server application |
| Communication Handler | Responsible for TCP communication with Server Application and message processing |

## Module View

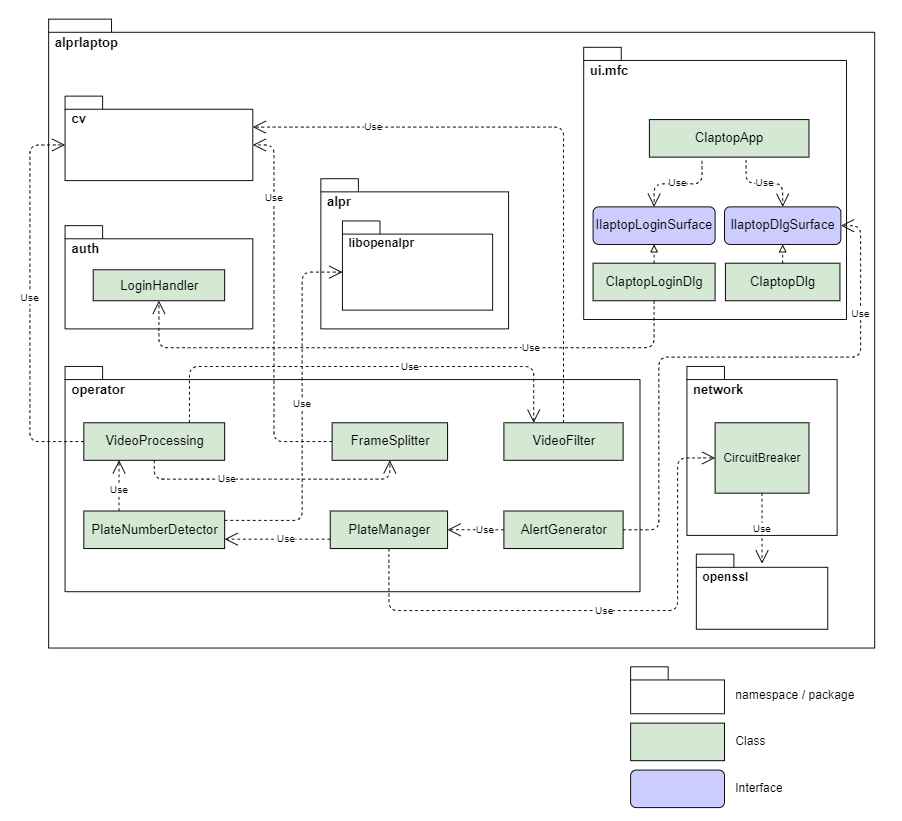
### Overall



**<Description>**

|  |  |  |
| --- | --- | --- |
| **Category** | **Package / Namespace Name** | **Description** |
| laptop application | alprlaptop | namespace for laptop application |
| cv | package for opencv library |
| alpr | package for openalpr library and other video extension libraries |
| libopenalpr | package for openalpr library |
| ui.mfc | package for UI using MFC Library of Windows application |
| auth | namespace for Login/Authentication |
| operator | namespace for alpr functional features |
| network | namespace for network connection to server |
| openssl | package for openSSL library |
| server application | alprserver | namespace for server application |
| vehicleQuery | namespace to handle license plate query received from laptop application |
| database | namespace to manage vehicle information database and user authentication information |
| berkeleydb | package for berkeley DB library |
| auth | namespace to handle user authentication |
| messageDispatcher | namespace to handle event received from client asynchronously |
| network | package for network connection to client |
| winsocket | package for TCP/IP socket library |
| openssl | package for openSSL library |
| mobile application | biometric.mobileapp | package for biometric authentication mobile application |
| service | package for biometric authentication service |
| network | package for network connection to server |
| notification | package for notifying authentication result |
| ui | package for mobile application UI |
| auth | package for biometric Authentication Handler |

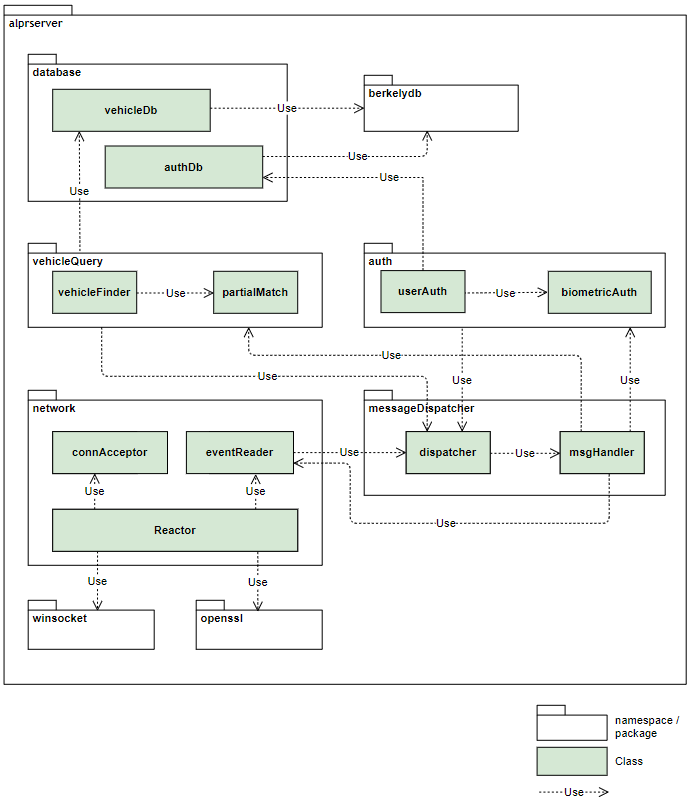
### Laptop Application



**<Description>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Package** | **Module Name** | **Description** | **FR Traceability** |
| ui.mfc | ClatopApp | MFC Wrapper Class | - |
| ClaptopLoginDlg | MFC UI for user login view | #08 |
| ClaptopDlg | MFC UI for displaying License Plate Information view. | #09, #10, #11 |
| IlaptopLoginSurface | Interface for Login UI | #08 |
| IlaptopDlgSurface | Interface for displaying License Plate information view. | #09, #10, #11 |
| auth | LoginHandler | Manage for Login / Authentication. | #01 |
| network | Circuit Breaker | Monitoring Network connection. | #07 |
| openssl | - | Provide secure communication with client | #02 |
| operator | VideoProcessing | Open video file and set configuration.  Reordering video frames from each queue.  Play a video stream with accurate frame rate. | #04 |
| FrameSplitter | Extract frames from input video  Distribute frames to each queue. | #04 |
| VideoFilter | Apply image filter to each frame.  (Grayscale, Auto level, etc.) | #04 |
| PlateNumberDetector | Recognition for a plate number using alpr libraries. | #04 |
| PlateManager | Frame grouping where a license plate is displayed for each vehicle.  Select the plate number with the best confidence in the grouped frames. Receive vehicle information by querying the plate number to the server. | #04 |
| AlertGenerator | When the status of vehicle information received from the server is illegal, displaying alert notification and showing vehicle information. | #05, #06 |

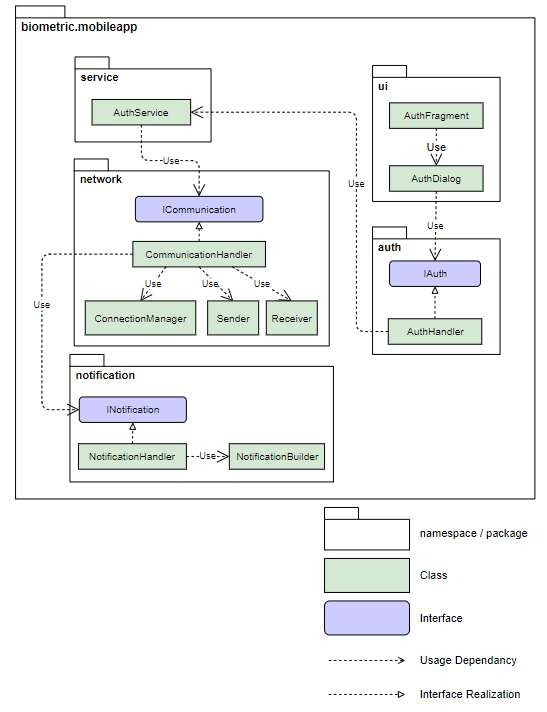
### Server Application



**<Description>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Package** | **Module Name** | **Description** | **FR Traceability** |
| vehicleQuery | vehicleQuery | Handle license query received from Laptop application and provide response after finding best match result.  Track following things of queries:  1. Avg. number of Queries per second, for each user.  2. Overall number of Queries per second, for all users.  3. Number of partial matches and no matches | #12  #17  #18 |
| partialMatch | Return the best match license if there is not an exact match  Support to configurable minimum confidence threshold | #16  #19 |
| openssl | - | Provide secure communication with client | #13 |
| network | connAcceptor | Accept connect request from multiple user | #14 |
| messageDispatcher | dispatcher | receive event and push them to msgHandler to process asynchronously | #15 |
| msgHandler | process events in parallel | #15 |

### Mobile Application

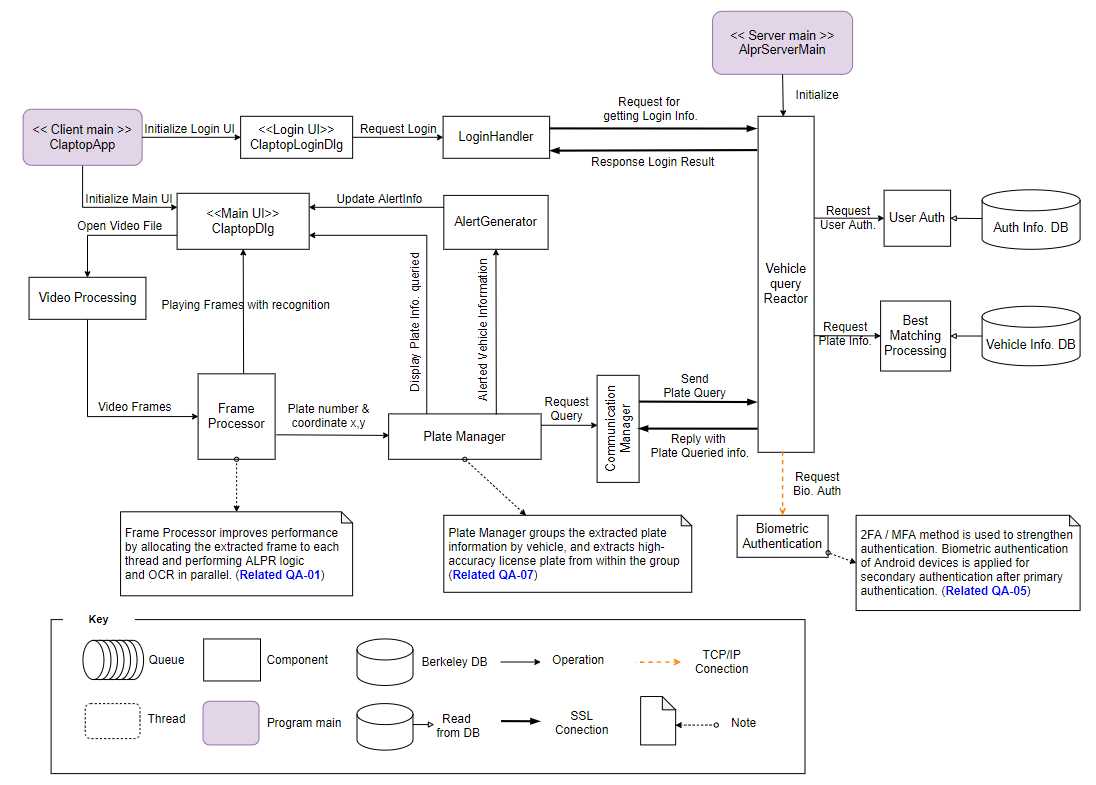


**<Description>**

|  |  |  |  |
| --- | --- | --- | --- |
| **Package** | **Module Name** | **Description** | **FR Traceability** |
| auth | AuthHandler | Manage for Login / Authentication | #01 |

## Runtime (Component & Connector) View

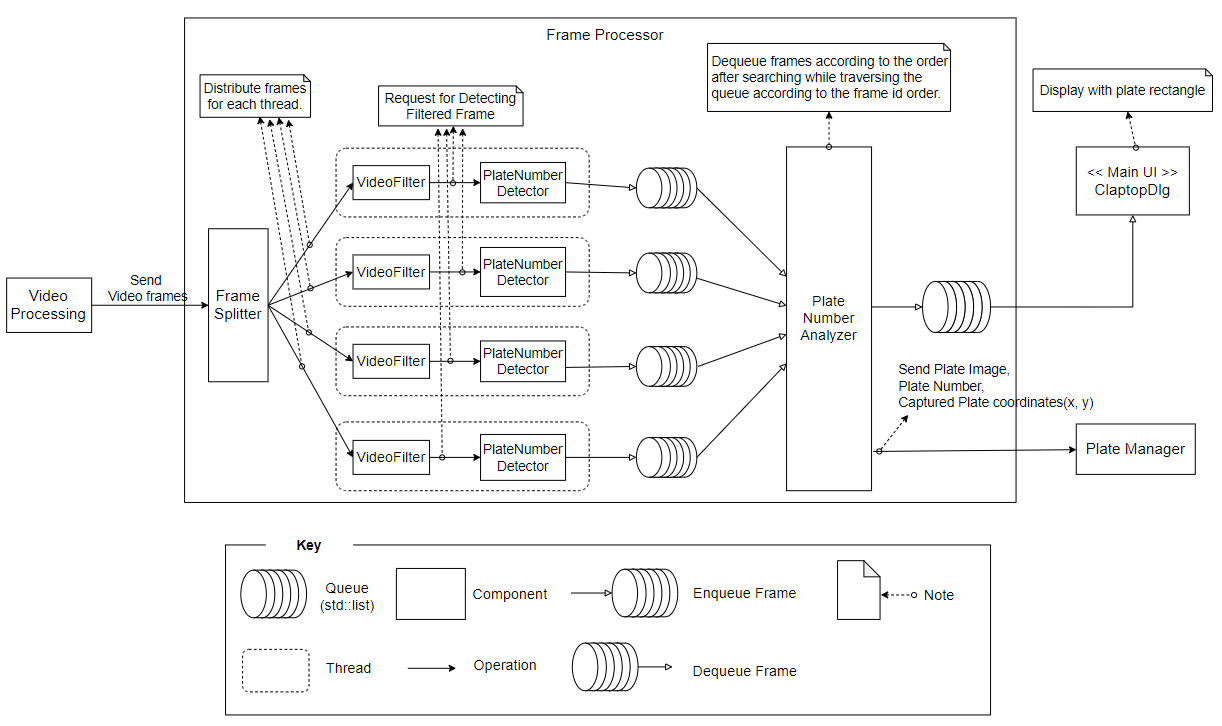
### Overall ALPR Runtime View



**<Component Reference>**

|  |  |
| --- | --- |
| **Component Name** | **Description** |
| ClaptopLoginDlg | Displaying Login UI and user can enter login id/pw. |
| ClaptopDlg | If login authentication is success, ClaptopApp invoke this component showing Main UI and initializing it.  User can open the video file which they want to do operate alpr. |
| Login Handler | It takes the information entered in the UI and actually requests login information to the server through the SSL communication channel, and requests the login request data entered in the UI to the “Vehicle query Reactor” of the server and receives a response. |
| Video Processing | It plays the role of delivering the video file desired by the user to the “Frame Processor” in units of frames. |
| Frame Processor | Creates a multi-thread, distributes the video frames received from video processing to each thread, and applies a video filter to each thread, stores the detected plate number by frame, and collects each frame in the order of the frame id. At this time, the processed frames are delivered to the UI of ClaptopDlg in order.  At the same time, the cropped plate image, the plate number saved for each frame, and the cropped graphical coordinates (x, y) are delivered to the Plate Manager.  ※ More details are described in section 3.4.2 ALPR Frame Processing Detail View. |
| Plate Manager |  |
| Connection Manager |  |
| Vehicle query Reactor |  |
| User Auth. |  |
| Best Matching Processing |  |
| Biometric Authentication |  |

### ALPR Frame Processing Detail View

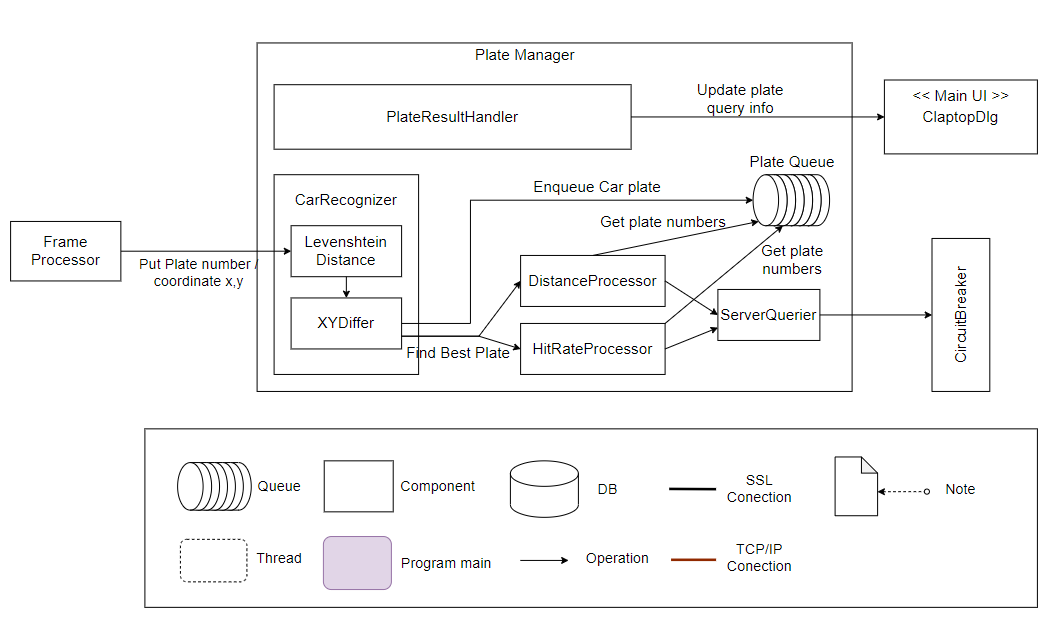


The diagram above is the multi-thread plate detection model used for QA-01. This model manages data resources by using 'Queue' for serialized video frames. Threads in charge of various filter and detection algorithms for each frame are allocated according to the operation speed. The purpose of this model is to reduce the recognition response time of a limited application and to perform more operations in a limited time.

**<Component Reference>**

|  |  |
| --- | --- |
| **Component Name** | **Description** |
| Frame Splitter |  |
| VideoFilter |  |
| PlateNumberDetector |  |
| Plate Number Analyzer |  |

### ALPR Plate Manager Detail View

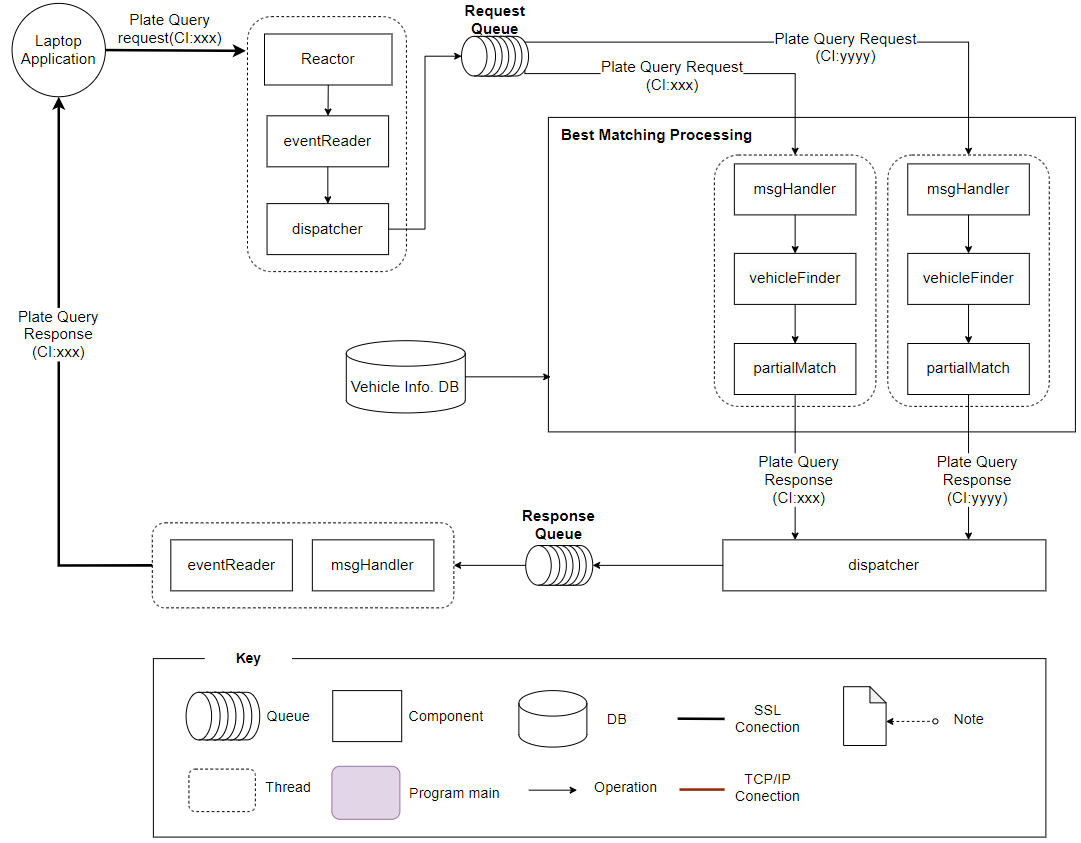


The above diagram is a query data optimization model used for QA #01 and QA #07. This model estimates the best number for each vehicle using the recognized information of the license plate and the coordinate information of the cropped image and optimizes the query requested to the server. The purpose of this model is to improve performance by reducing the number of queries requested to the server.

**<Component Reference>**

|  |  |
| --- | --- |
| **Component Name** | **Description** |
| LevenshteinDistance | LevenshteinDistance is used to recognize that it is the same car.  Recognizing the difference in number between plates, if it falls within a certain difference,  it is judged to be the same number plate.  (As a result of the experiment, the optimal number difference value is 3.)  Ex) ABC1 - ABC123 are recognized as the same vehicle with distance 2  AB123 - AB138 are recognized as the same vehicle with distance 2 |
| XYDiffer | In addition to the LevenshteinDistance above, we are also measuring the coordinates XY diff to recognize that it is the same car.  The x and y coordinates of the plate number plate are extracted from the video frame, and the coordinate values ​​are compared between plates.  If it is within a certain range, it will be judged to be the same number plate.  (As a result of the experiment, the optimal x, y difference value is 50.)  Ex) If x : 50 / y : 50 and the coordinates of the next number plate x : 90 / y : 90 are all within 50, it is judged to be the same number plate.  If x : 50 / y : 50 and the coordinates of the next license plate are x : 150 / y : 60, the x coordinate is out of the allowable value (50), so it is judged as a new vehicle |
| DistanceProcessor | LevenshteinDistance was used to select the optimal license plate from each candidate group among the bundles of the same license plate.  After calculating the distance between candidates, the number plate with the smallest number is adopted.  Ex) If there is a grouped candidate of ABC123 - ABC1 - ABC23,  The difference of (ABC123 and ABC1) + the difference of (ABC123 and ABC23) is calculated and summed, respectively, and the license plate with the smallest difference among these values ​​is adopted. |
| HitRateProcessor | Among the bundles of the same license plate, the license plate recognized the most repeatedly is adopted.  Ex) ABC123 repeats 10 times within frame  ABC23 repeats 5 times within frame  In the above situation, ABC123 is detected the most, and the server is searched for ABC123. |
| ServerQuerier | Before the server inquiry, the collection is done inside the Plate Manager, and the inquiry request is made from the external CommunicationManager. |
| PlateResultHandler | The plate image is managed to output the server inquiry result to the Main UI, and the plate image to be printed is delivered. |

### Best Matching Processing Detail View

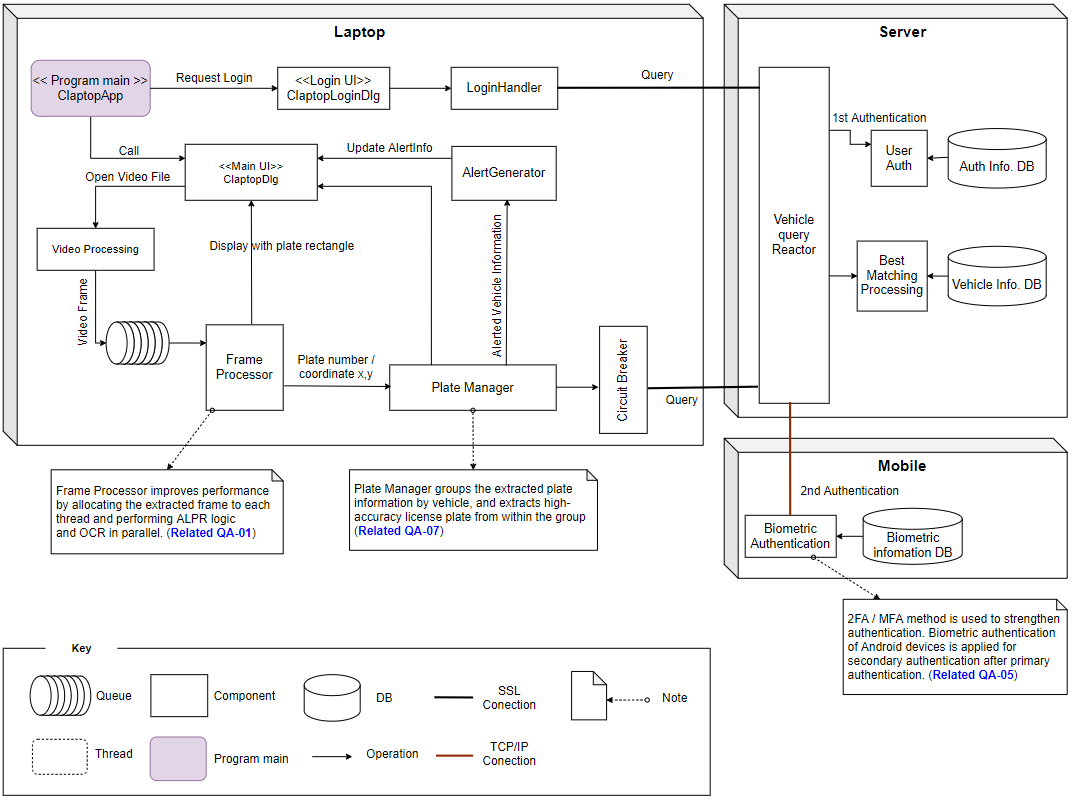


The server application receives the plate query request from the laptop application and transmits the vehicle information matching the license plate number in the database as a response. If the database does not have an exact matching license plate number, it should be able to send the best match result through partial match support.

If the partial match operation takes a long time, there is a delay in response transmission, and the next plate query request cannot be processed and is pending.

This problem can be solved by concurrently processing multiple plate query requests by running multiple-threads. At this time, there may be a problem that plate query requests cannot be processed sequentially. For this, CI is used for messages sent and received between the laptop application and the server application.

## Allocation View



# Architecture Decision

## Update tesseract library for safe multi-thread processing.

**Status**: ***Proposed***

Current version of tesseract library which is doing character recognition using OpenALPR is v3.04 and it is quiet old version. Benchmarking with newest version v5.1 and propose to design decision based on test result.

**Purpose**:

To improve ALPR performance, we suggest for using multi-threaded model for plate recognition module. But tesseract v3.14 library is not thread-safe and has memory leak issue, so it causes program crash. To decrease crash probability, we tested for multi-threaded with using tesseract libraries comparing old version and recent version.

**Test Results:**

The following is the test results:

1) Crash Rate

- Environment: running ALPR with 16 thread

- Result:

|  |  |  |
| --- | --- | --- |
|  | tesseract v3.04 | tesseract v5.1 |
| Crash Rates  (number of crash occurred  / number of test) | 8% (4 times / 50 tests) | 2% (1 time / 50 tests) |

- Analysis: tesseract v5.1 has improved stability, but crash by memory leak still occurs.

2) Accuracy

- Environment: create DB with only shown vehicles on beaver1.avi and beaver3.avi, and using ALPR for license plate of each video showing.

- Result:

|  |  |  |
| --- | --- | --- |
| Video File | Tesseract v3.04 | Tesseract v5.1 |
| Beaver1 | 31 plates recognized | 32 plates recognized |
| Beaver2 | 27 plates recognized | 28 plates recognized |

- Analysis: tesseract v5.1 can recognize more plates than tesseract v3.04

3) Performance

- Environment: test with 1 thread and 2/4/8/16 threads with no delay between frames. (Laptop with Ryzen7 5825U, processor, 8 core 16 thread)

- Analysis: The tesseract v3.04 has far better performance than v5.1, but tesseract v3.04 is less dramatic but it’s already fast enough with only 2 threads. The tesseract v5.1 has low performance per single thread recognition but has logarithmic scale performance by increasing thread counts. At least 8 threads is required for safe multi-thread processing on plate recognition module and design decision should be needed with considering H/W device availability.

**Consequences of Tests:**

|  |  |
| --- | --- |
| Pros. Using v5.1 tesseract | Cons. Using v5.1 tesseract |
| - Slightly better stability  - better accuracy  - Performance on 8 threads are adaptable. | - Additional image processing may impact performance even with 16 threads.  - Low single recognition performance. |

**Derive risks :**

Not verified much than tesseract v3.2

Much effort is needed to fix or apply work around to newer version of tesseract

**Trade Offs :**

There’s limitation due to low performance of tesseract v5.1, we can’t add more additional image processed frame to be recognized (e.g, add color-inverse image for better recognition)

**Alternatives** :

Applying intermediate version of tesseract v4.0 but from our simple test result v4.0 has no benefits either performance or stability

## Applying hybrid sampling rate control

**Status**: ***Proposed***

**Purpose**:

To increase ALPR performance, several sampling frame methods to be processed by ALPR is considered and hybrid sampling rate is now proposed to apply to ALPR system.

**Test Results:**

Result from comparing no sampling vs odd/even frame sampling demands new method for frame sampling.

The following is the test results:

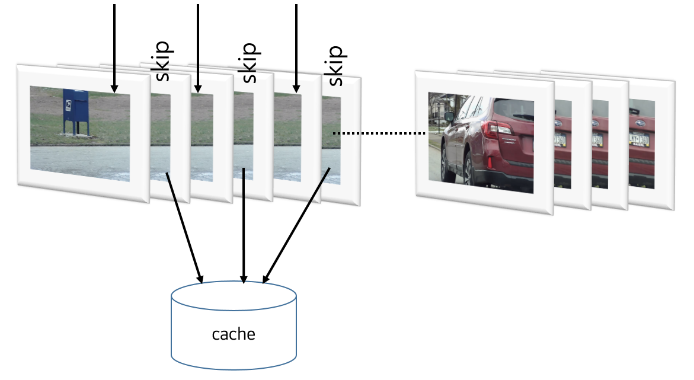
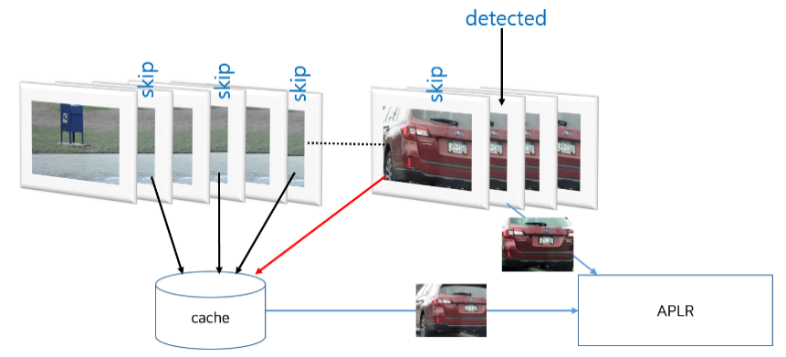
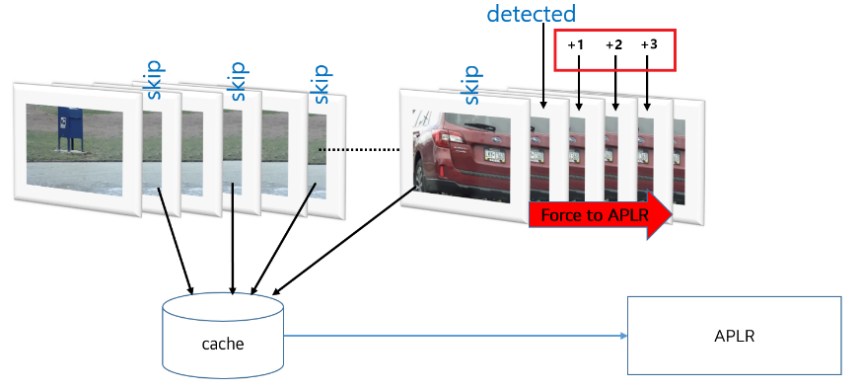
- Test environment: Fill DB with only known vehicle plates, counting number of correct plate recognition.

- Test result:

Count number of correct plate recognition

|  |  |  |  |
| --- | --- | --- | --- |
|  | no sampling | odd frame | even frame |
| beaver1.avi  beaver2.avi  beaver3.avi  beaver4.avi | 31  9  27  5 | 28  8  26  5 | 30  8  25  5 |

Proposed hybrid sampling is processing frames with following rules

1. If no plates are detected, push N+1 frame to cache and skip to N+2 frame  
   
2. When plate is detected, process ALPR with cached and current frame  
   
3. When plate is detected, set burst count to process ALPR continuously  
   

Test result of comparing no sampling with hybrid sampling

- Test environment: Fill DB with only known vehicle plates, and count number of correct plate recognition. No thread model is used (main loop), and processing time is calculated with no frame display delay.

|  |  |  |
| --- | --- | --- |
| plate recognition | no sampling | Hybrid Sampling |
| beaver1.avi  beaver2.avi  beaver3.avi  beaver4.avi | 31  9  27  5 | 31  9  27  5 |
|  |  |  |

**Consequences of Tests:**

|  |  |
| --- | --- |
| Pros. Using hybrid sampling | Cons. Using hybrid sampling |
| - Avg. 30% of performance increase.  - No computing overhead. | - Cannot detect about a very rare case(OXOXOXOXO pattern). - High code complexity for cache handling when used in multi-threaded model. |

**Derive risks :**

Unexpected behavior due to high code complexity

Spend hours to verify when code is modified

**Trade Offs :**

If first plate recognition is failed though there’s a plate in image, it will lost 1 frame.

**Alternatives** :

Skip N frames after put N-1 frames to cache, if detected it needs to perform recognition all N-1 frames.

From our test results, It doesn’t make big different from current methods

## Choose alternatives for selecting best plate in plates of group.

**Status: *Proposed***

As a result of the experiment, it is judged that Alternatives #4 is effective. If a reliable percentage (threshold) is not obtained for repeated license plates, this can be supplemented by selecting additional plates that can be supplemented. Therefore, we suggest to apply Alternative #4.

**Context:**

The laptop recognizes the license plate and searches the server for this license plate. If laptop queries the server, laptop will receive a reply with license plates that partially match the license plate. However, there is a high possibility that unwanted license plates will be extracted from among them.

In order to increase the accuracy, it is necessary to refining the license plate in advance on the laptop side rather than sending all recognized license plates to the server. If laptop selects a license plate with high accuracy by refining it in advance and sends it to the server, it will perform a server inquiry with relatively high accuracy. For this, the following alternatives should be considered.

1) Investigate each frame and bundle the license plate by each vehicle

2) Search for the best license plate among the bundled license plates and request a query to the server

For this, the following alternatives have been proposed.

**Alternative #1)** Priority is given to the number plate that appears repeatedly in the license plate group

Example)

A. PlateInfo (CPT MGN, 50, 50, 1)

B. PlateInfo (CPT MGN2, 49, 49, 2)

C. PlateInfo (CPT MG, 48, 48, 1)

We can check gathered plate list as the above example.

B. is the best plate number. Because repeated count is 2. So we can query plate number, which is “CPT MGN2”, to server.

**Alternative #2)** Measure the **Levenshtein distance** between license plates among license plate groups and give priority to license plates with a small distance

Assuming we have 5 candidate plates.

Assume F(i,j) is Levenshtein distance between i(th) candidate plate and j(th) candidate plate.

* so F(i,j) = F(j,i)

We calculate sum of Levenshtein distances from first candidate plate to other plates.

S1 = F(1,2) + F(1,3) + F(1,4) + F(1,5)

Similarity, calculate sum of Levenshtein distances from second candidate plate to others, and so on.

S2 = F(2,1) + F(2,3) + F(2,4) + F(2,5)

S3 = F(3,1) + F(3,2) + F(3,4) + F(3,5)

S4 = F(4,1) + F(4,2) + F(4,3) + F(4,5)

S5 = F(5,1) + F(5,2) + F(5,3) + F(5,4)

* We find minimum of S1, S2, S3, S4, S5. Assuming that is S3, so third candidate plate is selected.

**Alternative #3)** #1 and #2 candidates are all server query (2 times query)

**Alternative #4)** Alternative #2 is used when accurate judgment is not possible because the threshold value of alternative #1 is not exceeded.

Ex) pre-condition: threshold - 50%

1111 (repeated count: 2): Total count: 5 -> 2/5

1112 (repeated count: 1): Total count: 5 -> 1/5

1113 (repeated count: 1): Total count: 5 -> 1/5

1114 (repeated count: 1): Total count: 5 -> 1/5

* A total of 5 plates were recognized, but 1111 (repeated count: 2) occupies 40% (2/5): not satisfied considering the threshold value of 50%. Extracting additional candidates with #2 in this situation

**Consequences of Tests:**

|  |  |
| --- | --- |
| Pros. Using #4 alternative | Cons. Using #4 alternative |
| - Improved accuracy by adopting license plate with higher priority of license plate.  - Reduced server-side load due to reduced number of queries on the server.  - Because multiple queries do not occur, server-side asynchronous processing is easy | - If laptop adopts the wrong license plate, laptop may lose information on the correct plate.  - License plate selection algorithm is relatively important. |

**Derive risks:**

Group the Plates and pick a plate from among them that has a high probability of matching the actual plate, but still has a chance of missing the correct plate.

To prevent this part, various methods (Alternatives) have been tried, and among them, a license plate with a high probability will be adopted. Also, we will adjust the diff tolerance of XY Differ and the tolerance of Levenshtein distance appropriately to find the optimal license plate.

**Trade-offs:**

Query requests have been reduced to prevent incorrect plate inquiry on the server side, but there is a trade-off in which accurate plate information is not available. To improve this part, we plan to adjust the improvement algorithm and various setting parameters.