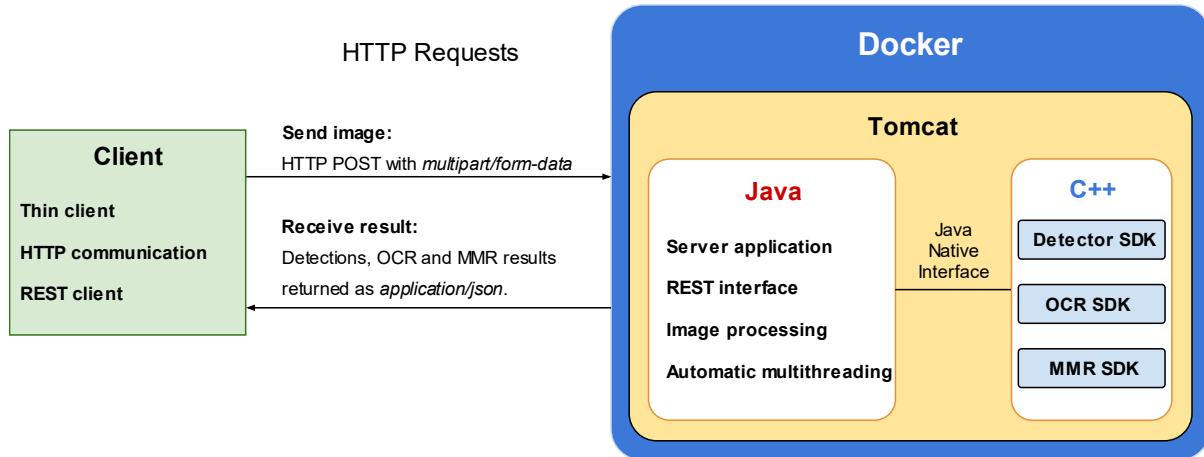


# MMR+ANPR REST Server

## Technical Specification



### Description

- **Client-Server** architecture with **REST API** written in **Java**.
- The server accepts input images (**JPG, PNG, BMP**) and returns processing results:
  - Detected license plates
  - OCR texts
  - Vehicle classifications.
- **Input** image is supplied to the server using **HTTP POST** request:
  - The content type is **multipart/form-data** and the image file data is contained as a parameter.
  - Optionally, the request may contain a parameter with license plate positions
- **Output** license plate detections, OCR texts and vehicle classifications:
  - Returned as **application/json** media type.

### Technologies

- System running as **Docker** image allows scalability.
- Application running on **Tomcat** web server.
- Application itself is written in **Java** and uses **JNI** (Java Native Interface) for communication with image processing SDKs.
- Image processing SDKs are written in **C/C++**.

### Contained SDKs

- **LPM SDK**
  - License plate detector running on input images (support for GPU computation for certain modules).
  - License plate OCR running on detected license plates (support for GPU computation).
- **MMR SDK**
  - Vehicle classification (view, category, make, model, generation, variation, and color recognition) running on detected license plates.
  - Support for GPU computation.

## Supported Operating Systems

- **Linux**
  - Ubuntu 16.04 and higher – x86\_64 platform

## Minimal Hardware Requirements

- Processor: 2 GHz, 2 cores (e.g., Intel Core i5)
- RAM: 4 GB
- Hard disk: 4 GB free space
- GPU (optional): NVIDIA Driver version  $\geq$  410.48 compatible (e.g., GeForce GTX 1050 Ti)

## Performance

For the performance test, the following configuration was used:

- CPU – processor Intel® Core™ i5-9400F @ 2.90 GHz, 32 GB RAM
- GPU – graphical card NVIDIA® GeForce® GTX 1660, 6GB GDDR5
- 1000 images, Full HD resolution (1920 x 1080 pixels)
- 1 processing thread per SDK
- Default SDK configuration (LPM module 800, precise MMR VCMM + Color models)

The following table summarizes the average processing time of the input file depending on the processing unit.

Used SDKs	CPU [ms]	GPU [ms]
MMR	380	13
Detector	50	10
Detector + OCR	52	9
Detector + OCR + MMR	417	18

Note: Only supported SDK combinations are listed.