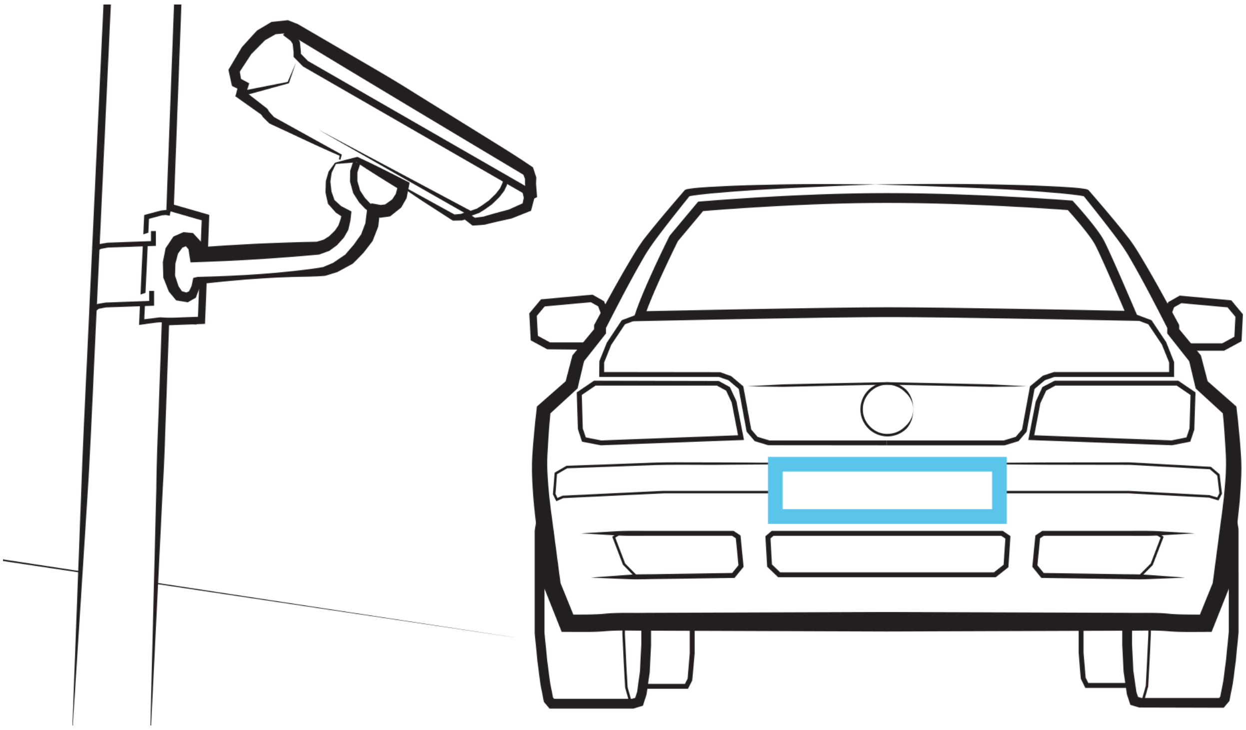
NeuroCar System VI - Technical specification

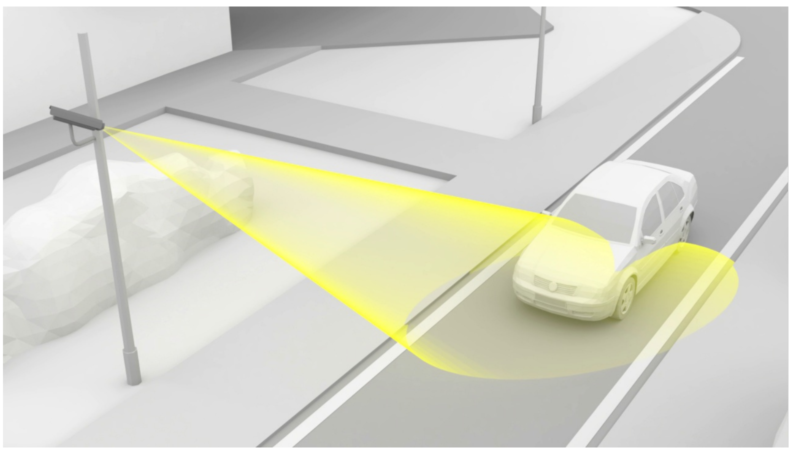
Version EN 4.0.5 released on Nov 12, 2022.

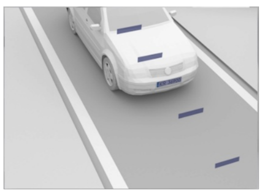


Design and operation of a system for vehicle detection and identification based on license plate recognition (ANPR) and vehicle manufacturer and model version recognition (MMR). For operation, the system uses only the video stream from a camera looking at the front or rear of the vehicle.

1. Operating principle

The **NeuroCar - Vehicle Identification** (*NeuroCar VI*) system is used for automatic detection and identification of vehicles by analyzing the video stream provided by the camera. The basic way of identification is recognition of the license plate - in the extended version it is also possible to automatically recognize the category of the vehicle, its make (model version), speed and country of origin.





COUNTRY / NUMBER	PL / KR 36900
CLASS	car (7)
MAKE/MODEL	VW / Passat
DIRECTION	approaching
SPEED	40 km/h

A vehicle driving within the field of view of the camera is automatically detected, and then the license plate is recognized and retrieved at various points (shots). As a result, a measurement record containing a picture of the vehicle and various types of data extracted from image analysis is stored in the system's memory.

The data collected by the measuring devices is stored in a central system (*BackOffice*) installed in the cloud, with the help of which users can comfortably search and analyze this data. The central system also has diagnostic functions to improve control of the correct operation of the measurement infrastructure.

Thanks to its modular design and the use of open protocols, easy integration is possible – users can add their own data sources or can attach their own processes to the system using information collected in the central system - through WEB API.

2. Functions

The functions delivered by the system can be divided into three groups:

Label	Function	Description
BASIC	**basic**	the main functions for which the system is built (*key functions*)
EXTENDED	**extension**	additional functions, which extend the operation of the basic functions; these functions may be activated optionally
AUXILIARY	**auxiliary**	auxiliary functions, using either the main functions or additional functions in an indirect way to improve the operation of the system; these functions can be activated optionally

2.1. Detection

BASIC

It consists of automatic detection of the vehicle passing within the camera's field of view (*free-flow*). The camera should be positioned in such a way that the license plate number can be read. Detection works for vehicles observed from both the front and rear, regardless of the driving direction. The detector selects from the video stream one shot (snapshot) in which the detected vehicle is most clearly visible → it has a legible license plate and the image of the driver can be seen.

EXTENDED

In the extended configuration, it is possible to detect vehicles without a license plate, or whose plate is unrecognizable - such a mechanism is called *VLOOP* → *virtual loop*.

2.2. Registration number

BASIC

It consists of automatic detection of the location of the license plate in the processed image and reading its content. The result of recognition is a string of alphanumeric characters, including special diacritics found in some countries – e.g. Ä, Ë, Ö, etc. The recognized string of characters also includes a space, which represents each separator. The result of the recognition is also information about the position and size of the array in the source image.

EXTENDED

In addition to the sequence of characters, the system can also determine the type of plate based on its syntax. This makes it possible to additionally mark the vehicle, for example, as a privileged vehicle (in Poland this is the appropriate prefix), or place of registration (city code). Individual plates are also recognized.

2.3. Class

EXTENDED

It consists of automatic recognition of the vehicle class based solely on the photo. The system can distinguish the following classes:

Code	Meaning
0	unknown (category determination was not possible)
3	truck
5	bus
7	car
10	motorcycle (only when *VLOOP* is active)
11	delivery van up to 3.5 t

Determining the vehicle class from the image requires that the silhouette of the front (or rear) of the vehicle is visible in the image. The problem can be night conditions, where the camera provides a picture of the plate taken in infrared - except the plate nothing is visible. If the prerequisite is to recognize the category also at night it is necessary to use additional special illumination.

2.4. Make and model

EXTENDED

It consists of automatically recognizing the manufacturer and its model version based solely on a photo of the vehicle. The algorithm works for photos taken from the front and rear of the vehicle - so that the license plate is visible. The recognition result is two text labels: *manufacturer name* + *model name*, for example *audi* + *q5*. The system recognizes ~140 different manufacturers and nearly 1,900 different models.

2.5. Color

EXTENDED

It involves automatically estimating the color of a vehicle based on the front image. Color determination works only for daytime images when the camera is in color mode. Color recognition can be affected by the “white balance” mechanism and depends on the type of camera used. The result of the recognition operation is the name of the primary color:

Code	Meaning
yellow	yellow
red	red
green	green
blue	blue
black	black
grey	grey

In addition, modifications `dark` and `light` are added to the base color.

2.6. Direction

`BASIC`

By analyzing image sequence, the direction of vehicle movement can be determined – the system determines the direction value by generating the following values:

Code	Meaning
+1	Conforms to the expected direction of traffic in that lane
0	Unspecified, possibly vehicle is standing
-1	Opposite of the expected direction of traffic in this lane - the vehicle is driving upstream

2.7. Lane

`BASIC`

Traffic lanes are determined by indicating the line of separation between lanes in the camera frame. At the configuration stage, you can define the number of lanes the camera observes (1÷4). A label is also attached to each lane: `L0`, `L1`, `L2` …, which is then used in the measurement metadata. The following convention is applied:

Code	Meaning
`L0`	shoulder (mostly non-existent)
`L1`	slow lane, in Europe the rightmost (always present)
`L2`	“overtaking” lane, in Europe to the left of the `L1` lane (may not be present)
`L3`	“overtaking” lane, in Europe to the left of the `L2` lane (may not be present)

Code	Meaning
…	

For countries with left-hand traffic, the convention is similar except that the `L1` lane is the leftmost lane.

2.8. Country

EXTENDED

The country of origin of the vehicle is determined based on geometric features and the contents of the license plate. The system is adapted to simultaneously recognize numbers from multiple countries, but the user must select one of the following regions – recognition is limited to countries from that region only:

Code	Meaning
AFRI	Africa [1]
AMEC	Central America [1]
AMEN	North America [2] (United States, Canada)
AMES	South America [1]
ASIA	Asia [2]
AUOC	Australia and Oceania [3]
EURO	Europe
MIDE	Middle East [1]

The country of origin is indicated using a two-letter country code according to ISO 3166-1 alpha-2 - for example, `PL` means *Poland*.

1(1,2,3,4)

only some countries

2(1,2)

in preparation

3

currently unavailable

2.9. Speed

EXTENDED

The speed of the vehicle is calculated on the basis of analysis of displacements of the position of the center of the license plate in successive frames of the video stream. The system has a mechanism for automatically determining, necessary for recalculation, parameters of the measurement system (camera position relative to, viewing angle, magnification, etc.) so that the user does not need to provide such data during installation.

The speed calculation accuracy in this way depends on many factors and is generally used for statistical purposes – it cannot be used, for example, for the purpose of violations enforcement.

2.10. Dangerous goods

`EXTENDED`

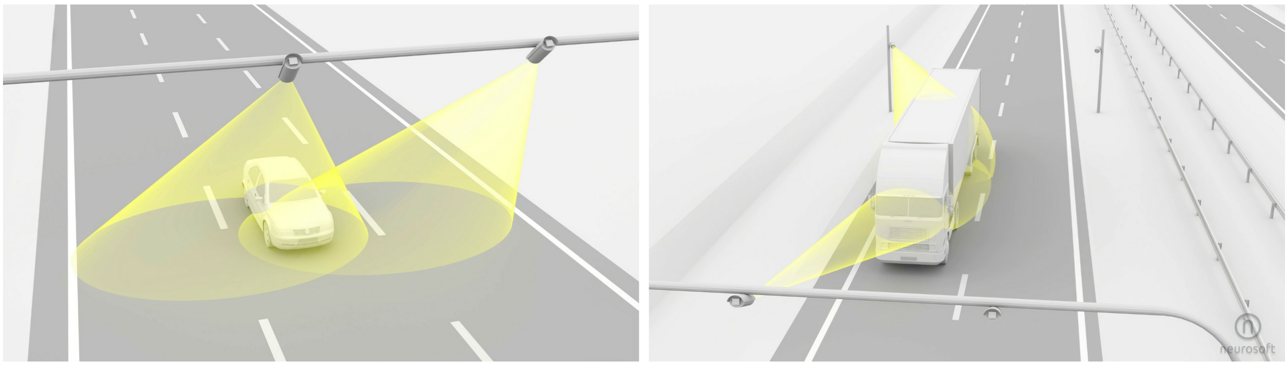
Vehicles carrying dangerous goods are identified based on detection and recognition of the contents of an ADR plate, which should be mounted on the front or rear of the vehicle. Such a placard is placed on vehicles in countries that are signatories to the International Convention on the Carriage of Dangerous Goods and Cargoes by Road (fr. L'Accord européen relatif au transport international des marchandises Dangereuses par Route).

The ADR plate contains two codes (recognized by the system), which specify the hazard identification number and the material identification number. In addition, the system detects the presence of so-called "blank ADR plates", as well as plates indicating the carriage of waste (the texts `Odpady` and `A`).

2.11. Multivision

`EXTENDED`

The system can consider measurement configurations in which multiple cameras observe a same detection location. It can then happen that the same vehicle is detected and identified by several cameras simultaneously. This applies to configurations in which, for example, two cameras observe exactly the same point, or two cameras observe both the front and rear of the same vehicle.



In both cases, the system is able to combine measurements from individual cameras and ultimately generates detection and identification information for only one vehicle.

2.12. Alerts

`AUXILIARY`

Based on the data on registered vehicles, it is possible to define alerts on the passage of a vehicle with the indicated characteristics, such as a wanted vehicle (indicated registration number) or a prohibited vehicle (i.e. vehicle of the "truck" class). Alerts can be generated on the terminal or in the central system.

The alert can be sent asynchronously to the host system as a notification immediately after being created. The form of the alert is configurable.

2.13. Monitoring

`AUXILIARY`

Monitoring the health of elements of the measurement infrastructure is a key sentence from the perspective of data reliability. A two-way health control mechanism is implemented in the system:

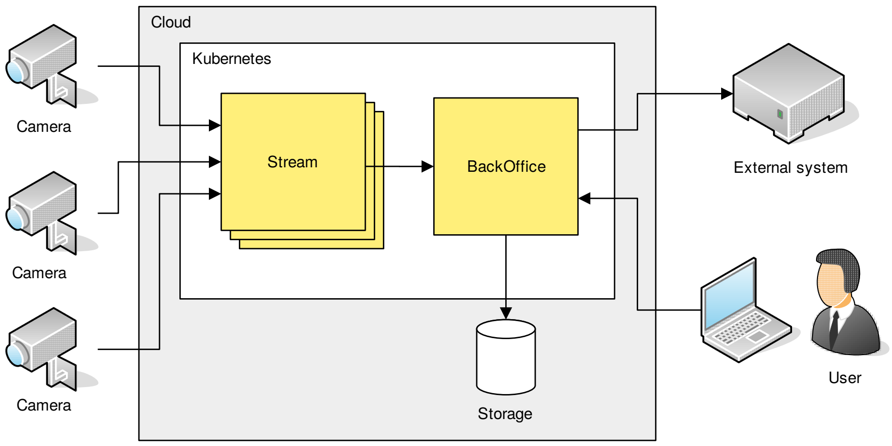
- **push** - periodic transmission by selected devices (terminals) of information about their status, to the central system → `status`,
- **pull** - periodic querying by the central system of the status of all devices impacting the health of the measurement infrastructure.

The aforementioned dual methodology makes it possible to react quickly to faults but also to review records of the system's operating history and detect disturbances caused by temporary malfunctions of measuring devices.

3. Design

The system can operate in various configurations, in which the key processing element - video stream analysis - is implemented in different locations. The user can choose the most convenient configuration from those described in the following sections.

3.1. Cloud computing

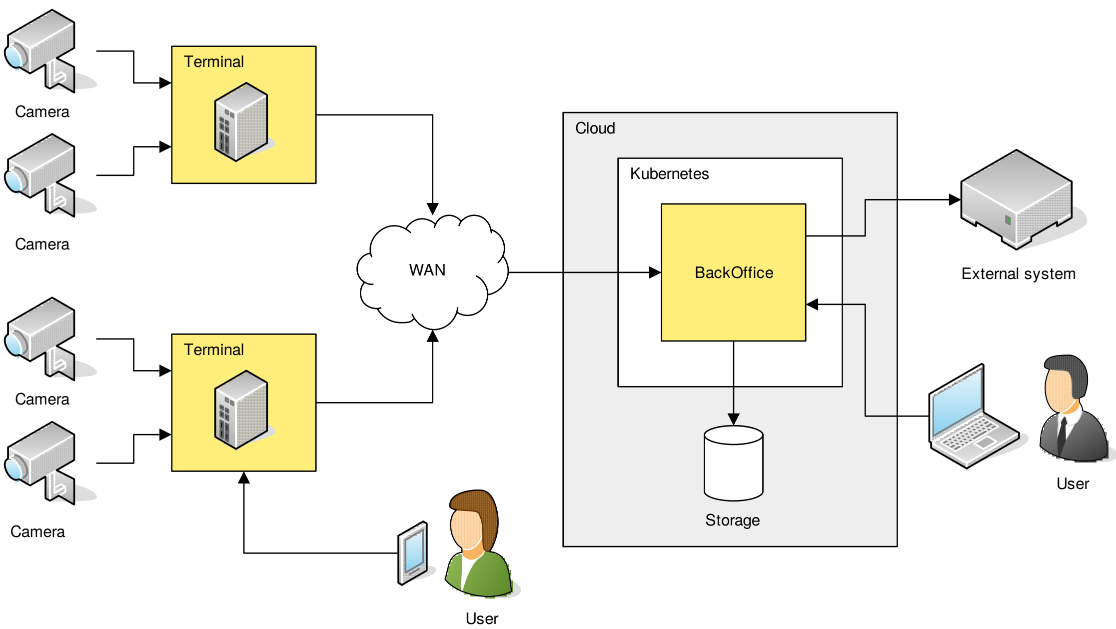


In such a configuration, cameras are mounted at the point of detection (on the road) to provide a video stream to the central system. These cameras are adapted for vehicle detection and identification (type: *ANPR camera*), however, since they do not analyze video, they must be connected to the central system with a relatively good and wide communication link – with a bandwidth of several Mbps.

The video stream is fed to a cloud-based application *NeuroCar Stream VI* – a separate instance of the *stream* engine is launched for each camera, which analyzes the stream and forwards (to *BackOffice*) only information about detected vehicles (*VehicleTrace* data record). The measurement data can only be accessed via *BackOffice*; either through the corresponding GUI application or by sending out notifications.

Thanks to the use of Kubernetes technology, such a solution can be run in virtually any cloud (either in a private → *on premise*, or with a third-party cloud provider). In addition, scaling is very easy – increasing the number of cameras is implemented only by adding working nodes in the cluster.

3.2. Fog computing



In such a configuration, cameras delivering the video stream are mounted at the point of detection (on the road), and computing units - *Terminals* - perform processing of the video stream “on the spot”. The terminal is equipped with an industrial CPU computing computer of sufficient power, a local storage medium (e.g. SSD) allowing data buffering, and communication devices (e.g. modem) allowing communication with the central system. The terminal can also collect other data such as environmental information such as temperature, power voltage, etc. The CPU computer, network equipment and power supply system are placed in a telecommunications cabinet mounted near the detection site. The camera signal is fed to the cabinet via a digital connection (UTP cable, fiber optic).

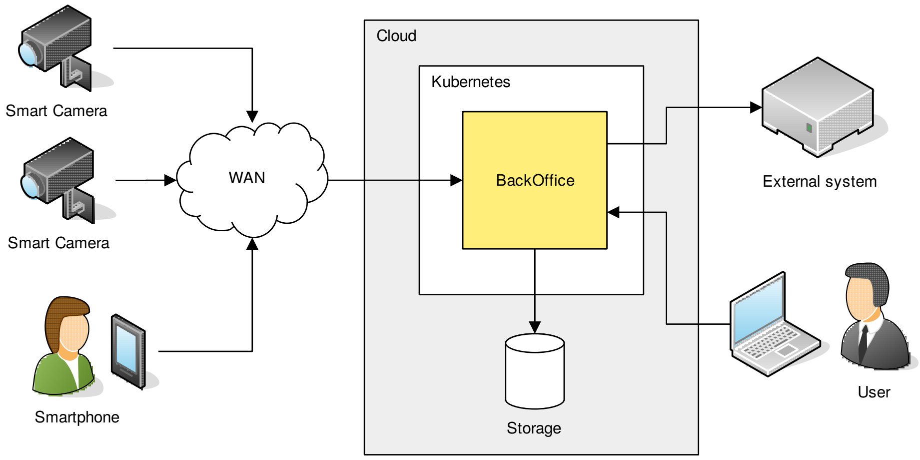
Fog processing allows a significant reduction in the size of the data that is sent over the WAN. The terminal sends only selected information about detected vehicles and status data to the BackOffice system. The system configured in this way is immune to disturbances in the data transmission network - a break in transmission does not cause loss of vehicle information. Buffered data can wait for delivery to the headquarters even many days.

An additional advantage of local processing is the ability to efficiently fuse data from multiple sensors, such as processing the stream from multiple cameras and detection from the induction loop controller. In this situation, it is the Terminal that is responsible for properly merging sensor data and forming a measurement record representing a single vehicle.

An important feature of fog processing is the ability of the user to work directly on the terminal, without the intermediation of the central system. Through this, the user on access to data immediately - without the potential delay generated by data transmission connections. To work with the terminal, the user must use an additional computer and a WEB browser.

All data generated by multiple terminals ultimately goes to the BackOffice central system. From this system, the user (or an external system) has access to the complete set of data, both current and archive.

3.3. Edge computing



In such a configuration, smart cameras are mounted at the point of detection (on the road), equipped with a motorized lens and a built-in infrared illuminator, provided with a built-in single-chip computer that processes the video stream. In addition, the smart camera has a built-in data storage device (SSD drive) for buffering measurements, and a modem - all in one compact enclosure.

The smart camera functionally corresponds to the terminal configuration, but it is more compact and smaller, and therefore easier to install, deploy and maintain. The disadvantage of a smart camera is the very limited possibility of creating multi-sensor measurement systems, which means that it is most often used when the user only needs the functionality of identifying vehicles “from a single shot”. The advantage of a smart camera is the ability to process images without distortion (e.g., without compression) which is due to the fact that the camera’s built-in CPU is physically integrated into the video path.

Another device that implements the instant processing concept is a smartphone that is both a camera and a processing unit. Such a phone, like a smart camera, analyzes the video stream, detects vehicles, identifies them and sends the result immediately to the central system.

3.4. Hybrid computing

In such a configuration, cloud computing, local computing and smart cameras can be used within a single system. The user receives at his disposal – from the central system – a unified, complete set of measurement data from all devices.

4. Devices

This chapter describes the key/specific devices for the vehicle identification system.

4.1. ANPR camera

The ANPR camera is a digital camera that provides a video stream suitable for license plate detection and license plate recognition. This camera has specific features:

- image resolution of at least HD (1280×720p)
- high sensitivity of the CMOS sensor allowing good (bright) images at shutter speed ≤1 millisecond,
- the ability to provide both color (day) and monochrome (night) images,
- switchable “day / night” modes – switchable, mechanical infrared filter,
- motorized lens or a motorized CMOS sensor position control system that allows remote focusing without physical access to the camera,
- a good-quality infrared-corrected lens with a focal length that allows the field of view to be set to 1÷3 lanes, with automatic iris control,
- built-in infrared illuminator that allows proper operation at night (invisible light in the range of 850÷940 nm),
- encryption of the connection to the client software, using SSL/TLS and X.509 certificates,
- the ability to enforce authentication (`user` | `pass`) by which it is protected from unauthorized use,
- H.264 video streaming with adjustable stream quality,
- built-in interface for diagnostics (SNMP),
- time synchronization capability, e.g. via NTP,
- specialized housing that reduces fogging of the windshield and has a shield against direct contact with sunlight,
- a suitable mounting bracket with precise adjustment of the camera position in at least 2 axes.

Attention

The ANPR camera does not process the video stream, i.e. it has no built-in processor for data analysis.

4.2. Smartcamera

The smart camera has all the features of an ANPR camera, plus:

- has a built-in CPU on which the video stream is processed,
- has terminal functions, i.e. generates highly processed data (information on captured vehicles),
- has a local storage drive (SSD) on which it can collect and buffer data,
- is equipped with dedicated software (firmware)

4.3. Smartphone

A standard smartphone equipped with *NeuroCar Stream VI* software running on the Android operating system can also be a data source.

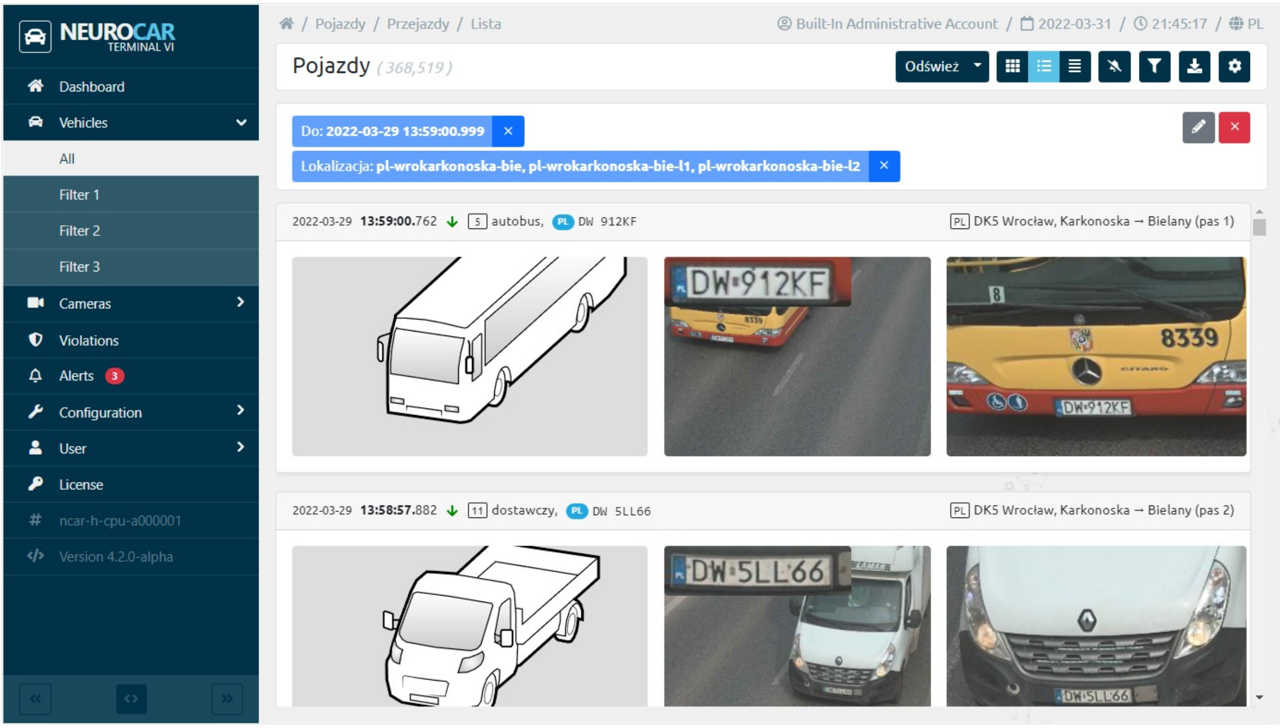
4.4. CPU

The CPU controller is an industrial computer, mounted in a telecommunications cabinet, whose task is to process the video stream from ANPR cameras, buffer the data, and then send it to the central BackOffice system. This computer has the following features:

- industrial design, without moving parts – no fans, convection heat dissipation,
- is ready to work in an extended temperature range,
- is resistant to vibration,
- is sufficiently powerful (Intel i5 or i7 class processor, appropriate RAM size),
- has built-in storage - disk (SSD)
- has a number of communication interfaces (ETH, RS-232/485, GPIO, USB2, USB3, HDMI/DVI),
- has the ability to extend with a communication modem (miniPCIe) or a dedicated PCIe expansion card,
- is powered by DC voltage in the range of 12÷24V DC.

5. Software

5.1. Terminal VI



This is a dedicated firmware for devices performing terminal functions – the software controls all functions of the device ensuring continuous 24/7 operation. In particular, it controls camera parameters, controls the status of storage media, controls time synchronization, manages communication modules and controls data transmission to the host system.

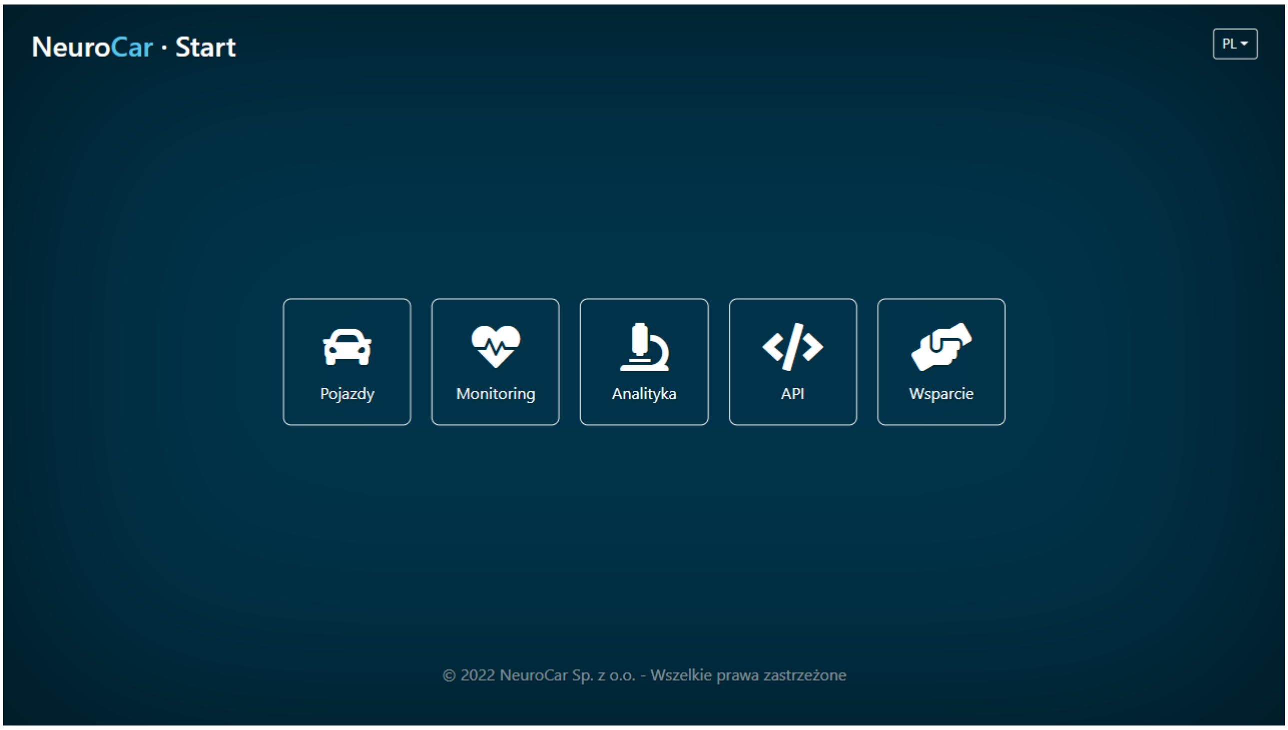
The firmware has a built-in graphical interface (GUI) accessible through any web browser supporting HTML-5. Using it, the operator can monitor the status of the device, can view measurement data (registered vehicles) in real time. The GUI also allows managing access privileges.

The firmware also has RestAPI implemented, allowing the device to be accessed and controlled by an automated higher-level system.

5.2. BackOffice

The software for the central system runs in the cloud under the control of the Kubernetes environment. The software has a modular design that allows the architecture to be flexibly adapted to the user's needs. The following sections list the most important applications.

5.2.1. Base

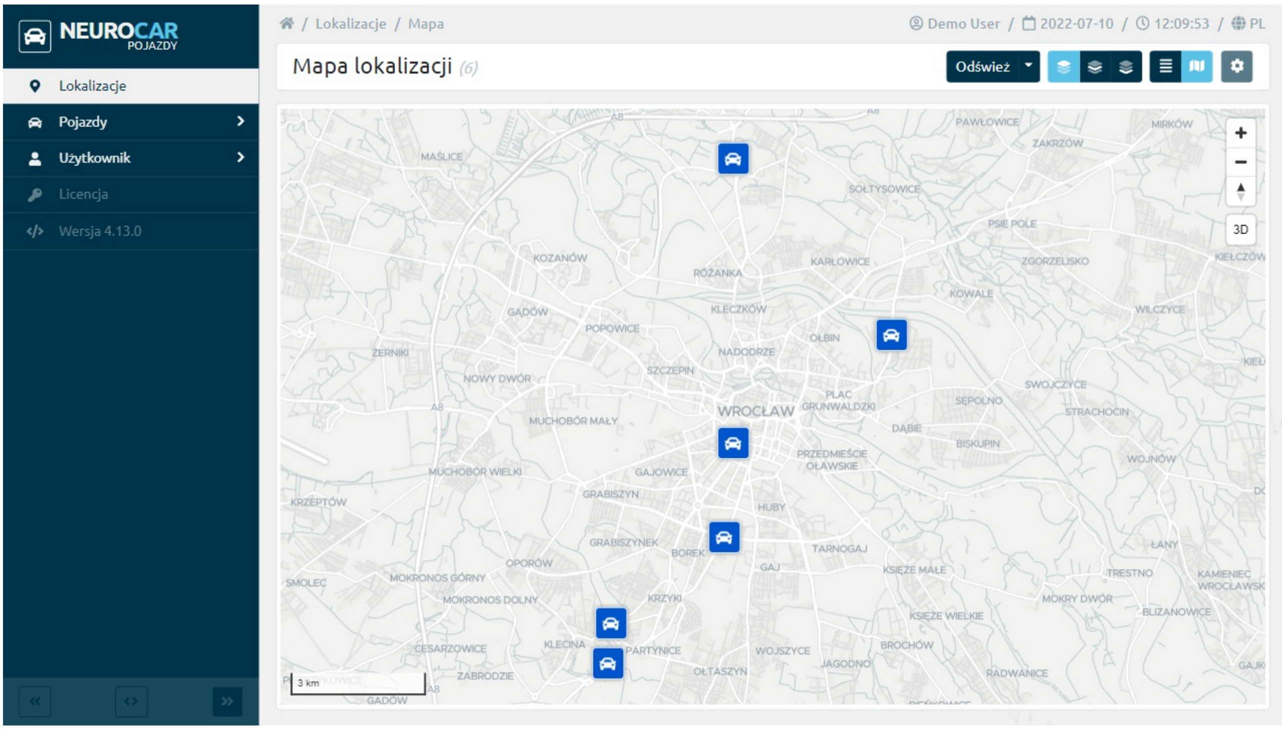


The **Base** application is the basis for the operation of the entire central system and consists of modules implementing the main functionalities:

- authentication
- providing API and GUI
- user/group management
- data storage and search management
- configuration management (*configuration*), including management of the list of measurement devices (*terminals*) and the list of measurement points (*locations*)
- management of data / statuses collection (*receivers*)
- managing the flow of data between system components (*message routing*)
- notification management – sending data to higher-level systems
- providing maps for other applications (in vector and raster form)

A component of the **Base** application is also the main system startup screen.

5.2.2. Vehicles

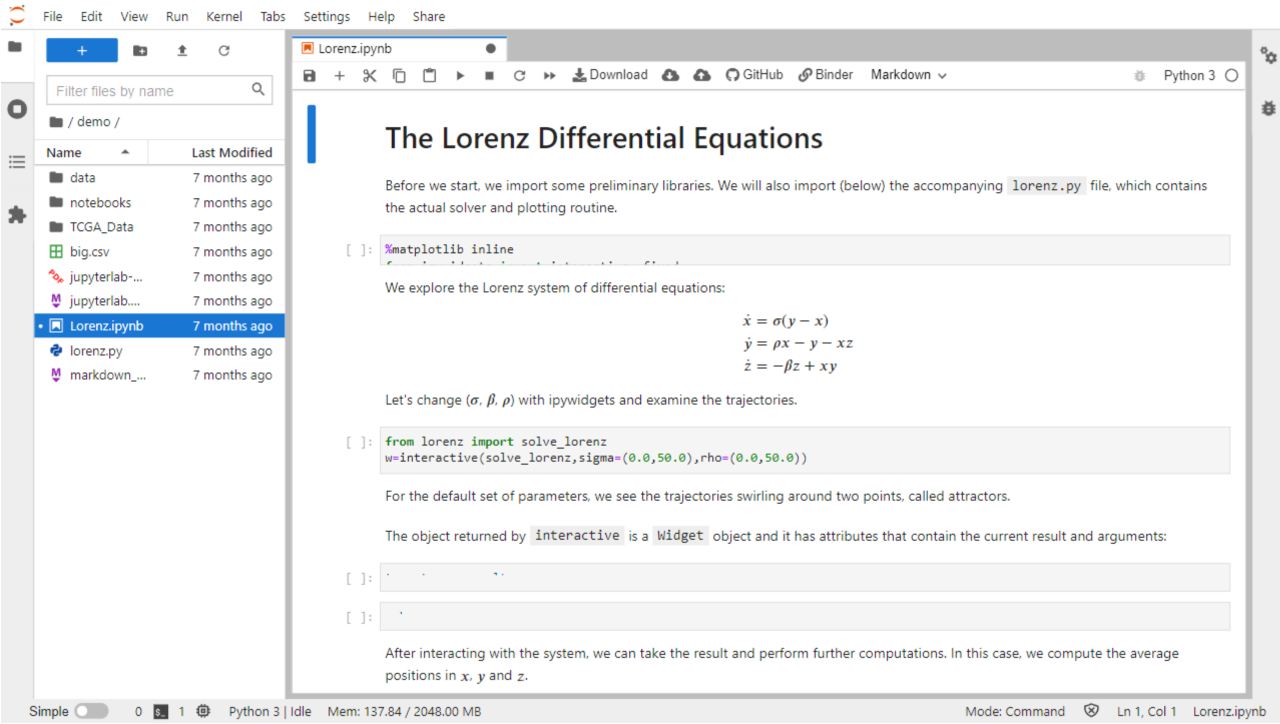


The **Vehicles** application is a dedicated module that allows the user to view data on registered vehicles. The application works as a WEB application. The user has access to data on the location of measurement points, can search for registered vehicles using an extensive filter mechanism, can reach a full list of vehicle information (pictures, metadata). There is also an option to download data in tabular format.

5.2.3. Monitor

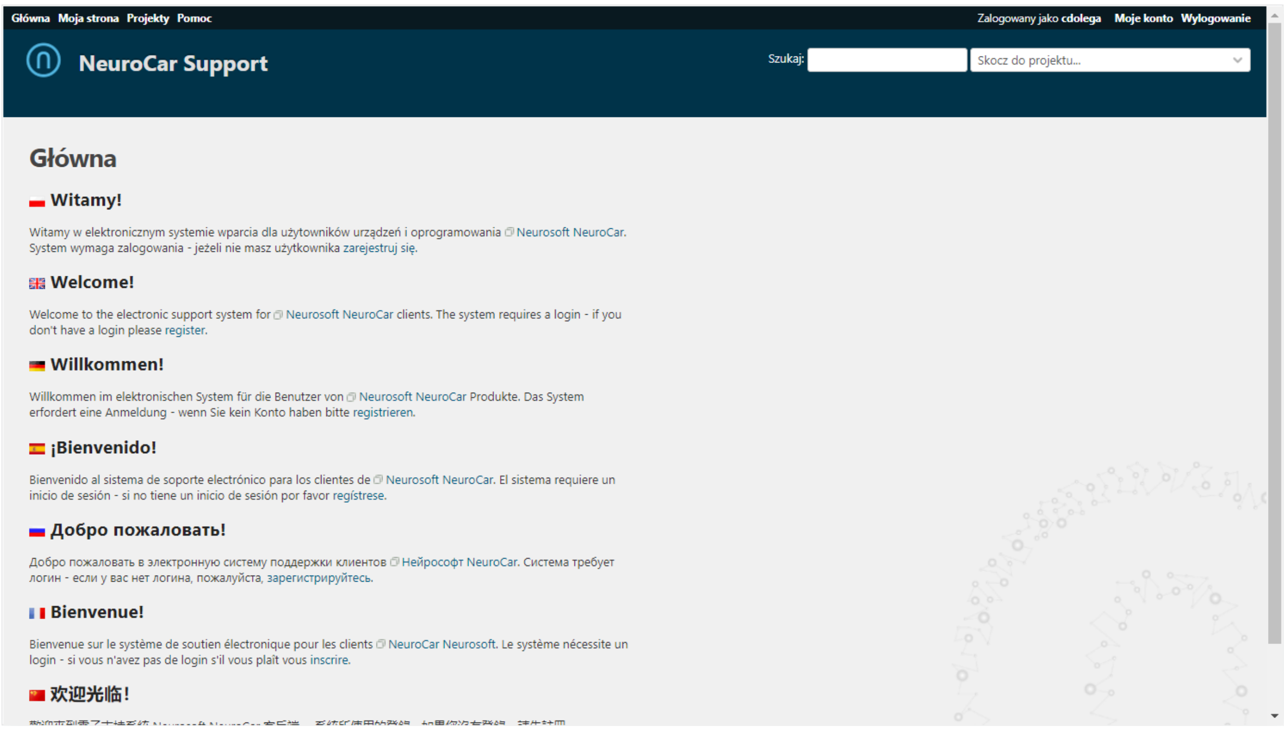
The **Monitor** application is used for pro-active supervision of measurement infrastructure using the Icinga package. The application controls the status of all measurement devices and the status of critical services running in the central system. In case of failure, the application provides the ability to send notifications (e.g. e-mail) to a designated group of operators.

5.2.4. Analytics



The **Analytics** application is used for extended data analysis using the JupyterLab package. The application is provided via GUI and allows you to run scripts prepared by NeuroCar downloadable from https://gitlab.com/ncar-tools/04

5.2.5. Support



The **Support** application runs in the cloud and is dedicated for the management of the ticket tracking process (Ticket tracking).

6. Data

This chapter describes the basic types of data records generated by measurement devices and stored in the system. This information can be used, for example, to estimate the size of data to be ultimately stored on storage media in the central system.

A detailed description of each of the data records presented can be found in the online documentation available free of charge at:

```
https://gitlab.com/ncar-tools/04/api
```

6.1. VehicleTrace

For each registered vehicle, the system generates a data record of type **VehicleTrace**. This record consists of a main element which is a detailed recognition result stored in the form of JSON.

Property	Value
Type	**VehicleTrace**
Format	TAR → JSON + JPEG's
Size	100 kB (estimated)
Transfer	up terminal → cloud
Frequency	for each detected vehicle

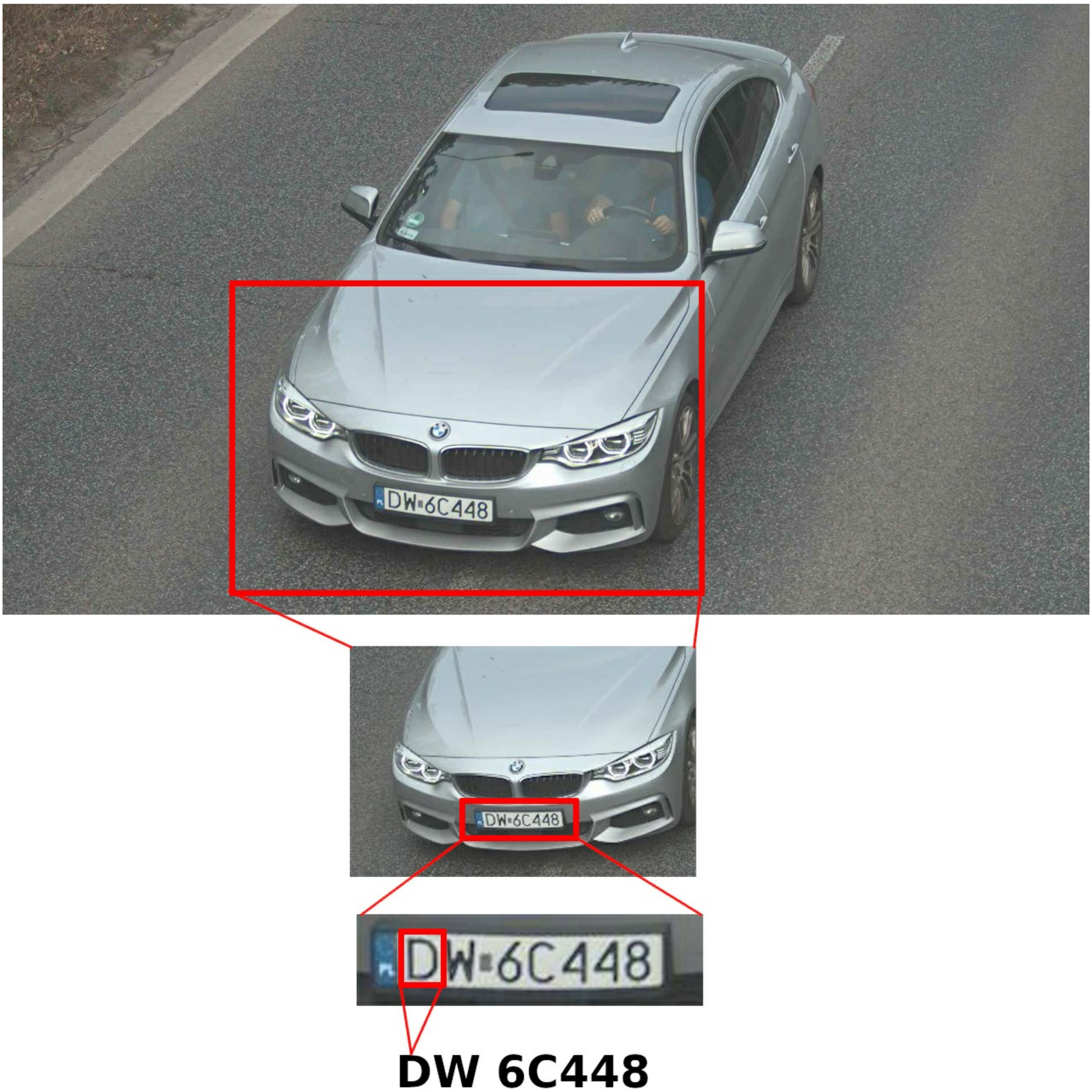
Example of metadata:

```
{
    "type": "VehicleTrace",
    "version": "4.3",
    "id": "VehicleTrace-20220629-122101-903-pl-wrokarkonoska-bie-l1",
    "time": "2022-06-29T12:21:01.903Z",
    "vehicle": {
        "class": 7,
        "color": "silver",
        "confidence": 96,
        "country": "PL",
        "direction": 1,
        "maker": "bmw",
        "model": "4",
        "number": "DW 6C448",
        "speed": 31.649284569475775
    },
...
}
```

In addition to metadata, the **VehicleTrace** record also contains images:

- **full** - a photo of the vehicle (front or rear) in full resolution / dimensions provided by the ANPR camera,
- **thumb** - reduced photo of the vehicle (front or rear) - multiple reduction of the photo *full*,
- **front** - close-up / photo "cutout" of the vehicle on the basis of which recognition of category, manufacturer and model version was made

- **plate** - photo of a cut-out license plate,
- **adr** - (optional) picture of a cut-out ADR plate for vehicles transporting dangerous goods.



The metadata (JSON) includes, in particular:

- information about the position and dimensions *(x,y),(s,w)* of the front of the vehicle in the image - the image of the front of the vehicle allows, for example, to identify the class, manufacturer and model version,
- information about the position and dimensions *(x,y),(s,w)* of the license plate - and the plate is then used to recognize its contents,
- information about the position and dimensions *(x,y),(s,w)* of each character,
- all recognized variants of the license plate, along with the indicated confidence level of a given variant,
- the result of recognizing each character together with the confidence level of this recognition,
- the result of recognizing the other features of the license plate, along with the confidence level such recognition.

Attention

Depending on the system configuration, the content of the data record may be reduced, for example, only to metadata or may not include some images. Similarly, depending on the configuration and properties of the camera, the image size may differ from the values given above.

6.2. CameraViews

The **CameraViews** measurement record contains a camera image taken at a specific time, which is transmitted from the camera to the central system for preview purposes. The size and quality of the transmitted image can be adjusted; the following parameters were adopted as standard:

Property	Value
Type	**CameraViews**
Format	TAR → JPEG (color) + JSON
Dimensions	1280 × 720 px (for HD camera)
Quality	70%
Size	120 kB (estimated)
Transfer	`up` camera → cloud

6.3. CameraRecording

The **CameraRecording** measurement record contains the video recording from a given camera, in the form of a `MP4` file compressed with `H.264` compression. The parameters of the video stream can be configured according to your needs.

Property	Value
Type	**CameraRecording**
Format	TAR → MPEG4 (color) + JSON
Dimensions	640 × 360 px
Quality	2 fps, H.264 MPEG-4 AVC
Size	2.0 MB / 2 min (estimated)
Transfer	`up` camera → cloud (optional)
Frequency	every 2 min

Attention

Creation of **CameraRecording** measurement records is disabled by default.

6.4. Status

The **Status** measurement record contains detailed information about the status of the terminal (camera) at a given time, stored as JSON metadata.

Property	Value
Type	**Status**
Format	TAR → JSON
Size	10 kB
Transfer	`up` camera → cloud
Frequency	every 5 min

7. Integration

7.1. API

Use of terminal (camera) as well as central system functions possible through the RestAPI interface, for which the full specification – *NeuroCar API* – is available free of charge at:

```
https://gitlab.com/ncar-tools/04/api
```

The various functions of the system (e.g., data access, measurement record format, user management) are described using the OpenAPI 3 standard.

7.2. Notifications

Both the terminal (camera) and the central system can actively transmit information to master systems using the notification mechanism (*push technology*) in asynchronous mode.

7.2.1. Data

All data records (measurements, camera images, status) can be sent via the HTTP(s) protocol using `GET`, `PUT` or `POST` commands. The content of the data record is appended to the message in the form of a TAR file, which includes all elements such as metadata along with images.

7.2.2. Notifications

The system can be configured to send notifications (e.g., information about events such as a wanted vehicle) using:

- e-mail (SMTP e-mail server must be available),
- SMS (an SMS gateway must be available).

7.3. Import

The BackOffice central system has the ability to connect external data sources (e.g., data from third-party ANPR cameras) via **Importer** (*pulling*) or **Receiver** (*pushing*) modules. The task of these modules is to transform the data into a unified form compliant with the API specification.

References

Changelog

4.0.5 `2022-11-11`

- added language version `EN`
- added language version `DE`

4.0.4 `2022-10-11`

- conversion to `RST` format

4.0.3 `2022-09-06`

- corrections

4.0.1 `2022-07-10`

- adding software information
- adding information about data structures
- adding integration information (API)

4.0.0 `2022-03-15`

- document initiation

License

Note

See also

This document is available online at https://docs.neurocar.pl/prod/ncar-sys-vi-4-spec/.