

Interface Design Document (IDD)

JSON Push interface

for Technolution's MobiMaestro



CONFIDENTIAL

This document is strictly confidential. Neither the plan nor any of the information contained in this plan may be reproduced or disclosed to any person without the written permission of the author.

Left intentionally blank

Table of Content

Document History	1
1 Introduction.....	2
1.1 Purpose and scope.....	2
1.2 Assumptions and dependencies	2
2 Client-server connection	3
2.1 Client and server certificates	3
2.2 HTTPS connection details	4
2.3 API key	4
2.4 Certificates.....	4
3 Interface description	5
3.1 Generic JSON information	5
3.2 Application specific JSON part	6
3.2.1 TrafficCount.....	6
3.2.2 Parking.....	9
3.2.3 AirQuality.....	11
4 Appendix I – Abbreviations	12

Document History

Version	Changes	Modified by	Date
Draft A to F	- Feedback from Technolution incorporated - Timestamps in epoch Unix (milliseconds) format - Technolution connection details (URL, port) added - Air quality - o Air quality measurements PM1 and PM2,5 included in json - o Name 'PM10' changed to 'PM1' - o JSON Air Quality always nested - o Numbers pushed as numbers instead of string in JSON - o Period from/to changed to timestamp Unix epoch - o Measurement units described - o measurement_humidity_perc added to interface description - o Examples updated - Parking - o Possible parking types defined - o lotId defined - o LastUpdate field added - o Examples updated - Traffic count - o Examples updated - Interval of sending described per application - Contact information added - JSON generic status description updated	Jeroen van Bussel Laurens de Graaff	23 rd of August 2019 to 16 th of October 2019
1.0	Mutually agreed document between JENG and Technolution (Tom de Groot) for the JSON push interface towards Technolution.	Laurens de Graaff	23 rd of October 2019
1.1	On request of Technolution (Johan Wunker) the destination URL was changed to <a href="https://devis.mobimaestro.nl/jengsensor/<datastream>">https://devis.mobimaestro.nl/jengsensor/<datastream> (was <a href="https://devis.deventer.nl:60443/jengsensor/<datastream>">https://devis.deventer.nl:60443/jengsensor/<datastream>) (see JENG ticket #1619)	Jeroen van Bussel	18 th of December 2023

In case you have questions, suggestions or want to check if this is the latest version of the IDD document please do contact us personally or at info@jeng.io.

1 Introduction

1.1 Purpose and scope

The purpose of this document is to provide the JSON interface description between JENG and a 3rd party. The JSON messages are pushed towards the 3rd party. When and how often the JSON messages are pushed depends on the application logic and is beyond the scope of this document.

In the Figure below a generic overview is given of the interface connection as described in this document.

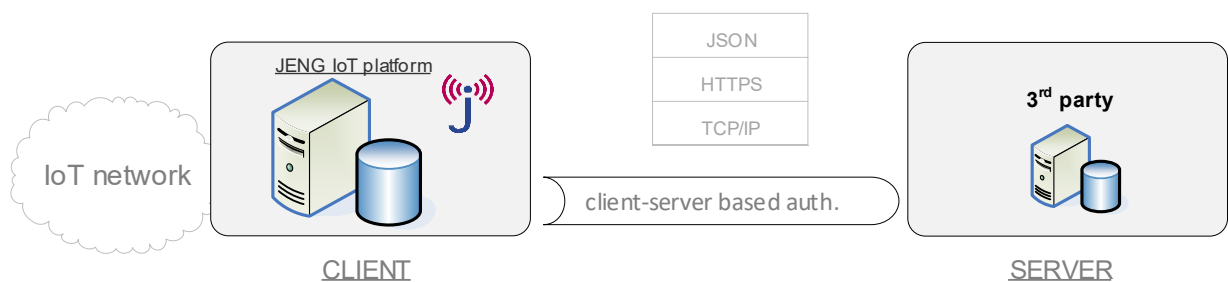


Figure 1: Interface overview.

1.2 Assumptions and dependencies

The following assumptions are made:

- The calibration data for the air quality sensors is not send via the JSON interface. Correction is done in the JENG platform by JENG. The sensor has the following calibration values:
 - WEOffset
 - AUXOffset
 - Sensitivity
- Measured information is sent raw, without any corrections, unless mentioned otherwise.
- JENG may optimize traffic intensive application in collaboration with the 3rd party.

2 Client-server connection

The connection must be established between the JENG platform and a 3rd party. The client (JENG IoT platform) will always initiate the connection establishment towards the 3rd party.

2.1 Client and server certificates

The authentication between the client and the server is based upon client-server authentication. Both the server and the client will check each other certificate.

In the Figure below an illustrative overview is provided how the connection is established. Here both client and server certificates are exchanged (see next paragraph) for obvious reasons (security, data integrity).

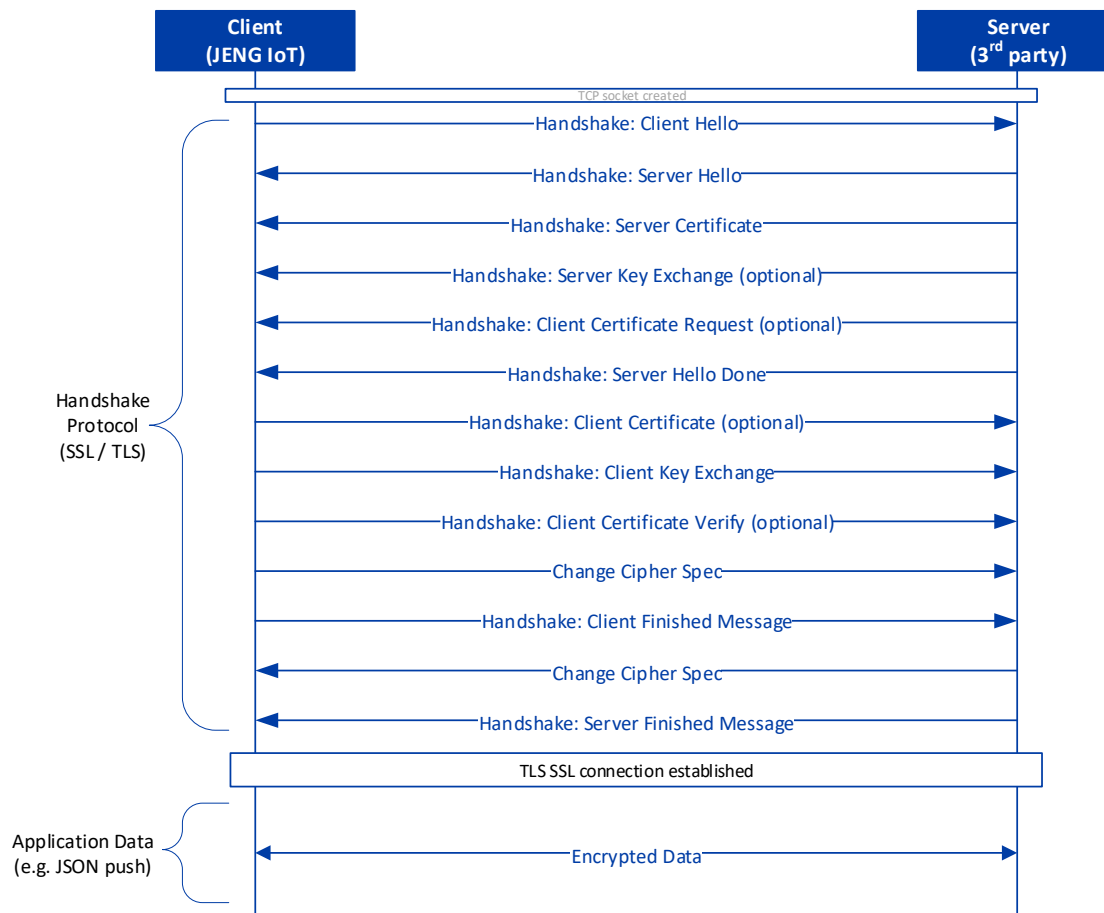


Figure 2: Overview of client – server connection establishment.

2.2 HTTPS connection details

The details to connect to Technolution are shown below:

- **URL:** `https://devis.mobimaestro.nl/jengsensor/<name_datastream>`.
 - `<name_datastream>`: as specified in Chapter 3.

2.3 API key

Optionally API keys can be used. This is project dependent and is not in scope of this IDD.

After the connection between the client server is established information the information can be pushed towards the 3rd party. In the next paragraph the information and the JSON format is described.

2.4 Certificates

Certificates information and validity is shown below (see #1218):

- not valid after September 21 2032 (GMT).

JENG will issue a new certificate 3 months before this date.

3 Interface description

The information is pushed in JSON format (JavaScript Object Notation) towards the 3rd party. The interface is designed to be able to handle information from multiple IoT applications.

In order to be able to expand the JSON interface there is a generic JSON part and an application specific JSON part as described in the next section.

3.1 Generic JSON information

The information in the Table below is independent from the application.

JSON field	Description	Format	Mandatory?
Interface	The application type from which the information in this JSON messages originates from.	TEXT	✓
Timestamp	Date and time when this message was pushed out. Time is presented in Unix Epoch in milliseconds (e.g. 1567153993).	EPOCH (ms)	✓
Status	Status field: 0 = OK, 1 .. 99 = Information is provided in the description field.	NUMERIC	✓
Description	Field with additional information – e.g. with respect to the sensor type / vendor etc.	TEXT	✓

Table 1: Overview of application independent information.

In the Table below an overview of the applications that are supported in this interface are depicted.

Nr.	Application	Description
1	trafficcount	Provides information from the traffic sensors. The sensors are able to detect different kind of traffic (bikes, cars, trucks etc.)
2	parking	Provides parking information.
3	airquality	Provides information on the Air Quality measurement

Table 2: Overview of supported applications in JSON interface.

The information and related JSON format used for the different applications as shown in the Table are described in the next paragraphs.

3.2 Application specific JSON part

In the sections below the JSON information is provided per application

3.2.1 TrafficCount

In the Table below the used fields are described for the TrafficCount application. The data is pushed directly on receiving sensor data.

JSON field	Description	Format	Mandatory?
locations[]	<i>Nested</i>		
locationId	Unique identifier for this location.	TEXT	✓
Lat	Latitude of the location	NUMERIC	-
Long	Longitude of the location	NUMERIC	-
roadType	Type of the road: 0 = one-directional, 1 = bi-directional	NUMERIC	✓
Method	Indicates which measurement method is used.	TEXT	-
Quality	Provides information about the quality of the measurement (as specified by CROW).	TEXT	-
Period_from	Measurement starting time. Time is presented in Unix Epoch in milliseconds (e.g. 1567153993).	EPOCH (ms)	✓
Period_to	Measurement ending time. Time is presented in Unix Epoch in milliseconds (e.g. 1567153993).	EPOCH (ms)	✓
Types[]	<i>Nested / nested</i>		-
typeId	Type of traffic (e.g. bike, scooter, car, truck)	TEXT	-
totalTraffic	Total number of this type of traffic detected in defined period	NUMERIC	-
trafficLR	Total traffic detected from Left to Right	NUMERIC	-
speedLR_kmh	Average speed of the traffic going from Left to Right	NUMERIC	-
trafficRL	Total traffic detected from Right to Left	NUMERIC	-
speedRL_kmh	Average speed of the traffic going from Right to Left	NUMERIC	-

Table 3: TrafficCount specific JSON information.

An example with single location and 1 type of traffic identified:

```
{
  "interface": "trafficcount",
  "timestamp": 1570022000990,
  "locations": [
    {
      "locationId": "steenbokstraat_1",
      "lat": 0,
      "lng": 0,
      "roadType": 1,
      "method": "JENG sensor",
      "quality": "",
      "roadType": 1,
      "period_from": 1570021663089,
```

```
"period_to": 1570022013000,
"types": [
  {
    "typeId": "bikes",
    "totalTraffic": 5,
    "trafficLR": 3,
    "speedLR_kmh": 0,
    "trafficRL": 2,
    "speedRL_kmh": 0
  }
],
"status": "OK",
"description": "No remarks"
}
```

Example with multiple locations and different types of traffic identified:

```
{
  "interface" : "trafficcount",
  "timestamp" : "1567153993",
  "locations" :
  [
    {
      "locationId" : "Apeldoorn-123",
      "lat" : 52.237610,
      "long" : 5.977827,
      "roadType" : 1,
      "method" : "JENG_sensor",
      "quality" : "",
      "period_from" : 1567153993,
      "period_to" : 1567154993,
      "types" :
      [
        {
          "typeId" : "bikes",
          "totalTraffic" : 100,
          "trafficLR" : 40,
          "speedLR_kmh" : 15.6,
          "trafficRL" : 60,
          "speedRL_kmh" : 18
        },
        {
          "typeId" : "cars",
          "totalTraffic" : 350,
          "trafficLR" : 180,
          "speedLR_kmh" : 48,
          "trafficRL" : 170,
          "speedRL_kmh" : 76
        },
        {
          "typeId" : "trucks",
          "totalTraffic" : 60,
          "trafficLR" : 20,
          "speedLR_kmh" : 35,
          "trafficRL" : 40,
          "speedRL_kmh" : 60
        }
      ]
    }
  ]
}
```

```
]
},
{
  "locationId" : "Deventer-158",
  "lat" : 52.253523,
  "long" : 6.169669,
  "roadType" : "0",
  "method" : "JENG_sensor",
  "quality" : "",
  "period_from" : 1567153993,
  "period_to" : 15671549930,
  "types" :
  [
    {
      "typeId" : "cars",
      "totalTraffic" : 234,
      "trafficLR" : 234,
      "speedLR_kmh" : 60
    },
    {
      "typeId" : "trucks",
      "totalTraffic" : 12,
      "trafficLR" : 12,
      "speedLR_kmh" : 43
    }
  ]
}
],
"status" : 0,
"description" : "No remarks"
}
```

3.2.2 Parking

In the Table below the used fields are described for the parking application. The data is collected and pushed on a configured interval. The configuration is as follows:

- Every 1 minute only changes.
- Every 2 hours all data.

JSON field	Description	Format	Mandatory?
lotAreas[]	Nested		
lotAreaId	Unique identifier for this location. This can e.g. refer to a street, parking area.	TEXT	✓
lotName	Name of the lot (e.g. area, street, square)	TEXT	✓
lotType	Type of the lot. Following possible values: • gemengd • langsparkeren • parkeerplein • parkeergarage • laad en los • e-parking • NP • illegaal parkeren • minder valide • Max x-uur parkeren • normaal vlak • kort parkeren • deel auto	TEXT	✓
totalLots	Total lots available	NUMERIC	✓
occupiedLots	Number of lots in this area which are occupied.	NUMERIC	✓
availableLots	Number of lots in this area which are available.	NUMERIC	✓
individualLots[]	Nested / nested		-
lotId	Identification of the lot in the following format: <u>city_street_number</u> Every field contains 3 letters. Example: cit_str_001	TEXT	✓
Status	Status of the lots: • 0 = available, • 1 = occupied	NUMERIC	✓
lat	Latitude of the location	NUMERIC	-
long	Longitude of the location	NUMERIC	-
lastUpdate	Timestamp Unix Epoch in milliseconds of the lastUpdate from the sensor.	EPOCH(msec)	✓

Table 4: Parking specific JSON information.

An example with individual lot information:

```
{
  "interface": "parking",
  "timestamp": 1569308746460,
  "lotAreas": [
    {
      "lotAreaId": "id-0000-0001",
      "lotName": "Nieuwe Markt",
      "lotType": "gemengd",
      "totalLots": 21,
      "occupiedLots": 14,
      "availableLots": 7,
      "individualLots": [
        {
          "lotId": "dev_nie_020",
          "status": 0,
          "lastUpdate": 1571078111000
        }
      ]
    },
    {
      "lotAreaId": "id-0000-0002",
      "lotName": "Hofstraat",
      "lotType": "gemengd",
      "totalLots": 8,
      "occupiedLots": 6,
      "availableLots": 2,
      "individualLots": [
        {
          "lotId": "dev_hof_001",
          "status": 1,
          "lastUpdate": 1571046132000
        },
        {
          "lotId": "dev_hof_003",
          "status": 1,
          "lastUpdate": 1571046132000
        },
        {
          "lotId": "dev_hof_004",
          "status": 1,
          "lastUpdate": 1571046132000
        },
        {
          "lotId": "dev_hof_007",
          "status": 1,
          "lastUpdate": 1571046132000
        }
      ]
    }
  ]
  "status": "OK",
  "description": "No remarks"
}
```

3.2.3 AirQuality

In the Table below the used fields are described for the AirQuality application. The data is pushed directly on receiving sensor data.

JSON field	Description	Format	Mandatory?
Locations[]	Nested		
locationId	Unique identifier for this location.	TEXT	✓
lat	Latitude of the location	NUMERIC	-
long	Longitude of the location	NUMERIC	-
Timestamp	Timestamp of measurement. Time is presented in Unix Epoch in milliseconds)	EPOCH (ms)	✓
measurement_PM1	Density of particles over 1 µm. pcs/283ml (particles per 283ml)	NUMERIC	-
measurement_PM25	Density of particles over 2.5 µm. pcs/283ml (particles per 283ml)	NUMERIC	-
measurement_CO	CO ppm (parts per million)	NUMERIC	-
measurement_CO2	CO2 ppb (parts per billion)	NUMERIC	-
measurement_NO2	NO2 ppb (parts per billion)	NUMERIC	-
measurement_O3	O3 ppb (parts per billion)	NUMERIC	-
measurement_temp_celcius	Temperature in Celcius	NUMERIC	-
measurement_humidity_perc	Relative humidity in percentage	NUMERIC	

Table 5: Air Quality specific JSON information.

An example with AirQuality measurement information.

```
{
  "interface": "airquality",
  "timestamp": 1569309603556,
  "locations": [
    {
      "locationId": "aqm_003",
      "lat": 0,
      "lng": 0,
      "timestamp": 1569309611104,
      "measurement_PM1": 17,
      "measurement_PM25": 0,
      "measurement_CO": 69.3001,
      "measurement_CO2": 625,
      "measurement_NO2": 0,
      "measurement_O3": 0,
      "measurement_temp_celcius": 20.2722111,
      "measurement_humidity_perc": 60.33521
    }
  ],
  "status": "OK",
  "description": "No remarks"
}
```

4 Appendix I – Abbreviations

Abbreviation	Meaning
API	Application Program Interface
CO	Carbon Monoxide
CO2	Carbon Dioxide
CROW	“Centrum voor Regelgeving en Onderzoek in de Grond-, Water- en Wegenbouw en de Verkeerstechniek”
HTTPS	Hypertext Transfer Protocol over SSL
IDD	Interface Design Document
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
LT	Local Time
NO2	Nitrogen Dioxide
O3	Ozone
PM	Particulate Matter
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security

- END OF DOCUMENT -