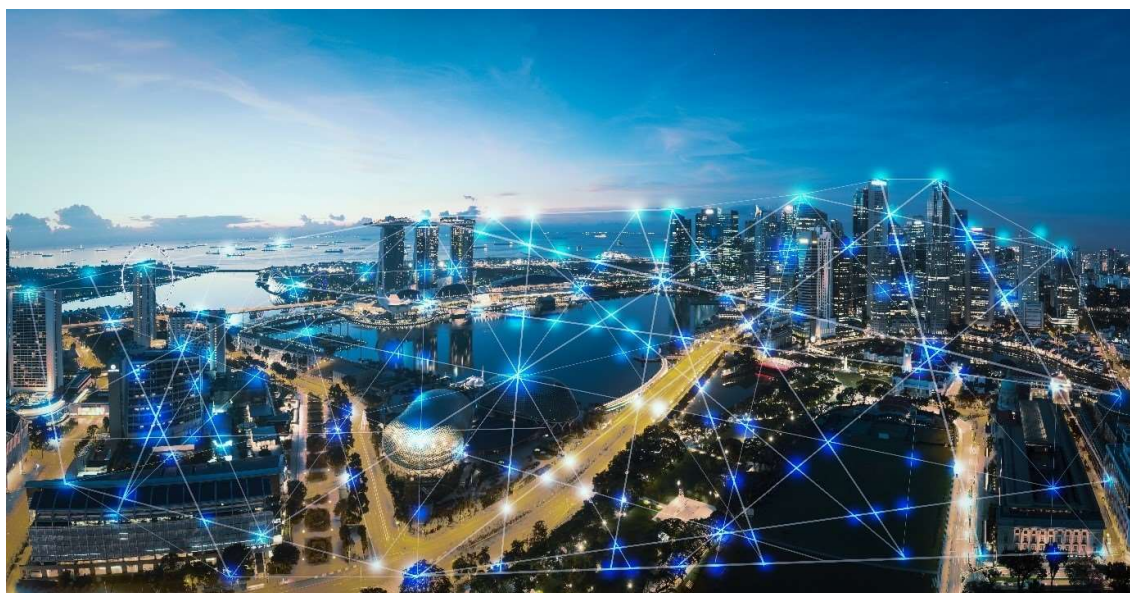




# Nestcloud Bridge Specification



\* \* \* \* \*

NextNav France

114 avenue Charles De Gaulle

92200 Neuilly-sur-Seine, France



## Revision

Revision	Date	Comments
1.00	February 2022	Initial revision
1.01	March 2022	Changed device picture
1.02	March 2022	Corrections after review
1.03	April 2022	Reworked for customer communication protocol usage
1.04	September 2022	Reworked for v1.7 API version
1.05	December 2023	Reworked for v1.9 API version and AWS migration



## Reference

[1] Nestcore AT command manual



## Table of Contents

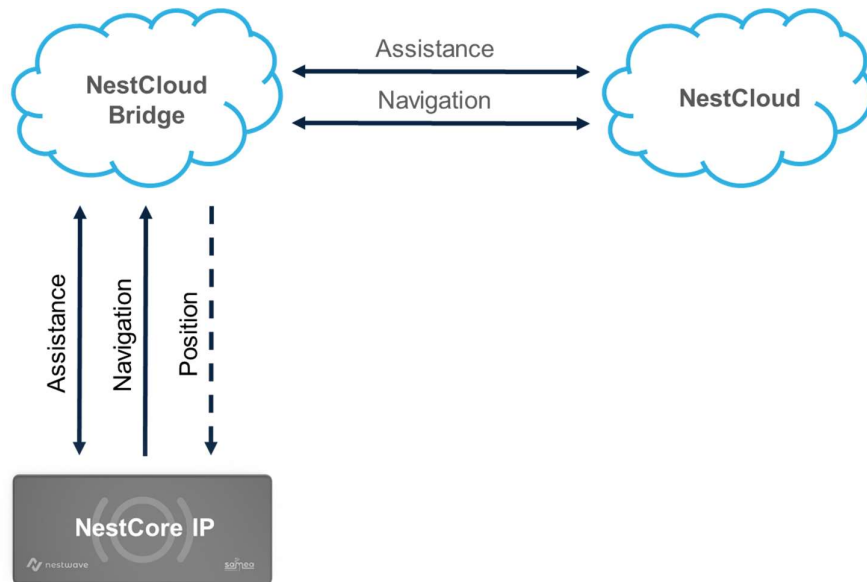
Revision .....	2
Reference .....	3
1 Introduction.....	5
2 Nestcore IP communication protocol .....	6
2.1 Devices to Nestcloud bridge communication.....	6
2.1.1 Assistance API.....	6
2.1.2 Navigation API .....	6
2.1.3 Hybrid navigation API .....	7
2.2 Nestcloud bridge to Nestcloud communication.....	7
2.2.1 Authentication API .....	7
2.2.2 Token renewal API.....	8
2.2.3 Assistance API.....	8
2.2.4 Navigation API .....	8
2.2.5 Hybrid navigation API .....	10
2.3 Overall sequence .....	12
3 Customer communication protocol .....	13
3.1 Devices to customer cloud data .....	13
3.1.1 API version.....	13
3.1.2 Assistance data.....	13
3.1.3 Navigation data .....	13
3.1.4 Hybrid navigation data .....	14
3.2 Customer cloud to Nestcloud communication.....	14
3.2.1 Authentication API .....	14
3.2.2 Token renewal API.....	14
3.2.3 Assistance API.....	14
3.2.4 Navigation API .....	15
3.2.5 Hybrid navigation API .....	16
3.3 Overall sequence .....	18

## 1 Introduction

This document is targeting Sequans GM02SP customers who need to design a cloud bridge to interface between Nestcloud and their devices embedding Nestcore IP.

It provides a detailed description of the communication methods to be supported on that bridge, in order to communicate with devices and Nestcloud.

The overall diagram is shown below:





In order to further describe the messages exchanged between the devices, Nestcloud bridge and Nestcloud in detail, 2 use cases need to be considered:

- Devices to Nestcloud bridge communication uses Nestcore IP protocol (CoAP)
- Devices to Nestcloud bridge communication uses customer own protocol. This typically allows bundling GNSS specific data with customer own data

## 2 Nestcore IP communication protocol

### 2.1 Devices to Nestcloud bridge communication

Nestcore IP uses CoAP as the communication protocol between the devices and Nestcloud bridge.

Nestcloud bridge must therefore embed a CoAP server configured as follows:

- CoAP port = 5683
- CoAP block size = 1024

The devices must be configured as follows:

- AT+LPGNSSCLOUDSEL=<bridge\_hostname> where <bridge\_hostname> is Nestcloud bridge hostname (example: my.company.com)

Nestcloud bridge must implement 3 CoAP APIs: assistance API and navigation API.

#### 2.1.1 Assistance API

Resource	Value
<b>URL</b>	coap://<bridge_hostname>:5683/vx.y/gnssAssistance
<b>Request type</b>	POST
<b>Payload</b>	<assistance request payload to be forwarded to Nestcloud assistance API, see 2.2.3>
<b>Response</b>	<assistance response payload received from Nestcloud assistance API>

The assistance API is used to send a GNSS assistance request to Nestcloud by using AT+LPGNSSASSISTANCE command on the device (see [1] for detailed syntax) and to get assistance data in return.

The assistance request payload received by the assistance API must be processed in Nestcloud bridge and sent to Nestcloud navigation API as explained in section 2.2.3.

The assistance response sent back by Nestcloud bridge contains the assistance data to be used by Nestcore IP.

#### 2.1.2 Navigation API

Resource	Value
<b>URL</b>	coap://<bridge_hostname>:5683/vx.y/gnssPosition
<b>Request type</b>	POST
<b>Payload</b>	<navigation request payload to be forwarded to Nestcloud navigation API, see 2.2.4>
<b>Response</b>	<navigation response payload received from Nestcloud navigation API, if any. Empty otherwise>



The navigation API is used to send Nestcore GNSS satellites measurements to Nestcloud by using AT+LPGNSSSENDRAW or AT+LPGNSSCLOUDNAV commands on the device (see [1] for detailed syntax) and to get the device cloud-based position in return.

The navigation payload received by the navigation API must be processed in Nestcloud bridge and sent to Nestcloud navigation API as explained in section 2.2.4.

The navigation response sent by Nestcloud bridge may contain a navigation payload (including the device position) that is used by Nestcore IP. The device position is returned to the host application by the +LPGNSSSENDRAW or +LPGNSSCLOUDNAV URCs (see [1] for detailed syntax).

### 2.1.3 Hybrid navigation API

Resource	Value
<b>URL</b>	coap://<bridge_hostname>:5683/vx.y/locate
<b>Request type</b>	POST
<b>Payload</b>	<hybrid navigation request payload to be forwarded to Nestcloud hybrid navigation API, see 2.2.5>
<b>Response</b>	<hybrid navigation response payload received from Nestcloud hybrid navigation API, if any. Empty otherwise>

The hybrid navigation API is used to send Nestcore GNSS satellites measurements and optionally cellular and pressure information to Nestcloud by using AT+LPGNSSSENDDATA command on the device (see [1] for detailed syntax) and to get the device cloud-based position in return.

The hybrid navigation payload received by the hybrid navigation API must be processed in Nestcloud bridge and sent to Nestcloud hybrid navigation API as explained in section 2.2.5.

The hybrid navigation response sent by Nestcloud bridge may contain a hybrid navigation payload (including the device position) that is used by Nestcore IP. The device position is returned to the device application by the +LPGNSSSENDDATA URC (see [1] for detailed syntax).

## 2.2 Nestcloud bridge to Nestcloud communication

### 2.2.1 Authentication API

In order to communicate with Nestcloud, Nestcloud bridge must first proceed to an authentication phase by using the following authentication API:

Resource	Value
<b>URL</b>	https://navigation.nextnav.io/authenticate
<b>Request type</b>	POST
<b>Body (raw type, json format)</b>	<pre>{   "username": "&lt;user@company.com&gt;",   "password": "&lt;pwd&gt;" }</pre> where <user@company.com> and <pwd> are the credentials used during Nestcloud account creation on <a href="https://gpsui.nextnav.io/">https://gpsui.nextnav.io/</a>
<b>Response</b>	<pre>{   "token": "&lt;access_token&gt;" }</pre> where <access_token> is the authorization token (valid for 30 days) to be used for further communication with Nestcloud



### 2.2.2 Token renewal API

When expired, Nestcloud bridge must renew the authorization token by using the following token renewal API:

Resource	Value
<b>URL</b>	<code>https://navigation.nextnav.io/authenticate/renew</code>
<b>Request type</b>	POST
<b>Header key=Authorization</b>	Bearer <old_token> where <old_token> is the old authorization token previously received in the authentication phase
<b>Response</b>	{ "token": "<new_token>" } where <new_token> is the new authorization token (valid for 30 days) to be used for further communication with Nestcloud

### 2.2.3 Assistance API

The assistance message payload received by Nestcloud CoAP server from the device must be processed as follows: remove 8 heading bytes and remove 4 tailing bytes.

The resulting bytes must be forwarded to Nestcloud by using the following assistance API:

Resource	Value
<b>URL</b>	<code>https://navigation.nextnav.io/vx.y/gnssAssistance</code> where vx.y is the API version received by Nestcloud CoAP server in the assistance API URL
<b>Request type</b>	POST
<b>Header key=Authorization Content-Type=application/json</b>	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
<b>Body (json type)</b>	{ "deviceId": "<unique device ID>", "assistancePayload": "<assistance request payload>" } where: <unique device ID> is a unique ID identifying the device <assistance request payload> is the payload received by Nestcloud bridge CoAP server on assistance API minus 8 heading bytes, minus 4 tailing bytes, then encoded in base64 format
<b>Response</b>	<assistance response payload in binary format to be forwarded as is by Nestcloud bridge CoAP server to the device in response to the CoAP assistance API>

### 2.2.4 Navigation API

The navigation message payload received by Nestcloud bridge CoAP server from the device must be processed as follows: remove 8 heading bytes and remove 4 tailing bytes.





The resulting bytes must be forwarded to Nestcloud by calling the following navigation API:

Resource	Value
<b>URL</b>	https://navigation.nextnav.io/vx.y/gnssPosition where vx.y is the API version received by Nestcloud bridge CoAP server in the navigation API URL
<b>Request type</b>	POST
<b>Header</b> <b>key=Authorization</b> <b>Content-Type=application/json</b>	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
<b>Body (json type)</b>	<pre>{   "deviceId": "&lt;device unique ID&gt;",   "rawMeas": "&lt;navigation request payload&gt;",   "features": [&lt;feature list&gt;],   "approxPos": [&lt;longitude, latitude, altitude&gt;] }</pre> <p>where:          &lt;unique device ID&gt; is a unique ID identifying the device          &lt;navigation request payload&gt; is the payload received by Nestcloud bridge CoAP server on navigation API minus 8 heading bytes, minus 4 tailing bytes, then encoded in base64 format          &lt;feature list&gt; is an optional coma-separated list of features to be activated on Nestcloud (example: "NoC", "AAN")          &lt;longitude, latitude, altitude&gt; is the optional device approximate position</p>
<b>Response</b>	<navigation response payload in json format containing the device position information, to be processed by Nestcloud bridge (example: map display, log in a database...)> NB: If the device position is requested by the device in the navigation request payload, then the navigation response payload contains an additional optional "payload" field. This payload must be forwarded as is by Nestcloud bridge CoAP server to the device in CoAP navigation API response>.

Example of navigation API response payload:

```
{
  "utcTime": "2023-11-07T14:29:18.150949001+00:00",
  "gpsTime": 1383402576.150949,
  "confidence": 11.193524,
  "position": [2.348045, 48.853610, 27.083687],
  "velocity": [-0.095578, 0.031254, 0.303247],
  "gps": {
    "prn": [29, 31, 25, 18, 2, 26, 22],
    "cn0": [44, 40, 42, 38, 40, 31, 33]
  },
  "payload":
  "MMEflqNKSEdcd5yPGlAEwNkq/kI/vsO9wgMAPSRDmz6tGDNBAQcdMGdPXQEdPgEfTgEZLQESRA
  EC9wAaCwEW",
  "HeightAboveTerrain": 22.439892425075655
}
```

Optional "payload" parameter to be forwarded to the device, if present.



### 2.2.5 Hybrid navigation API

The hybrid navigation message payload received by Nestcloud bridge CoAP server from the device must be processed as follows: remove 8 heading bytes and remove 4 tailing bytes.

The resulting bytes must then be parsed and forwarded to Nestcloud by calling the following hybrid navigation API:

Resource	Value
<b>URL</b>	https://navigation.nextnav.io/vx.y/locate where vx.y is the API version received by Nestcloud bridge CoAP server in the hybrid navigation API URL
<b>Request type</b>	POST
<b>Header</b> <b>key=Authorization</b> <b>Content-Type=application/json</b>	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
<b>Body (json type)</b>	<pre>{   "deviceId": "&lt;device unique ID&gt;",   "rawMeas": "&lt;GNSS navigation payload&gt;",   "features": [&lt;feature list&gt;],   "approxPos": [&lt;longitude, latitude, altitude&gt;],   "hybrid": {     "radioType": "LTE",     "cellTowers": [{       "cellId": &lt;cell identifier&gt;,       "lac": &lt;location area code&gt;,       "mcc": &lt;mobile country code&gt;,       "mnc": &lt;mobile network code&gt;,       "rsrp": &lt;received power&gt; }],     "wifiAccessPoints": [{       "mac": "&lt;MAC address&gt;",       "rssi": &lt;received power&gt; }],     "bluetoothBeacons": [{       "mac": "&lt;MAC address&gt;",       "rssi": &lt;received power&gt; }],     "pressure": {       "average": &lt;average pressure&gt;,       "variance": &lt;variance&gt;,       "count": &lt;count&gt;,       "min": &lt;minimum pressure&gt;,       "max": &lt;maximum pressure&gt; }} }</pre> <p>where:</p> <p>&lt;unique device ID&gt; is a unique ID identifying the device</p> <p>&lt;GNSS navigation payload&gt; is the GNSS navigation payload received by Nestcloud bridge CoAP server on hybrid navigation API encoded in base64 format</p> <p>&lt;feature list&gt; is an optional coma-separated list of features to be activated on Nestcloud ("NoC", "AAN", "PAAN")</p> <p>&lt;longitude, latitude, altitude&gt; is the optional device approximate position</p> <p>&lt;cell identifier&gt; is the 28-bit LTE cell identifier</p> <p>&lt;location area code&gt; is the LTE location area code</p>

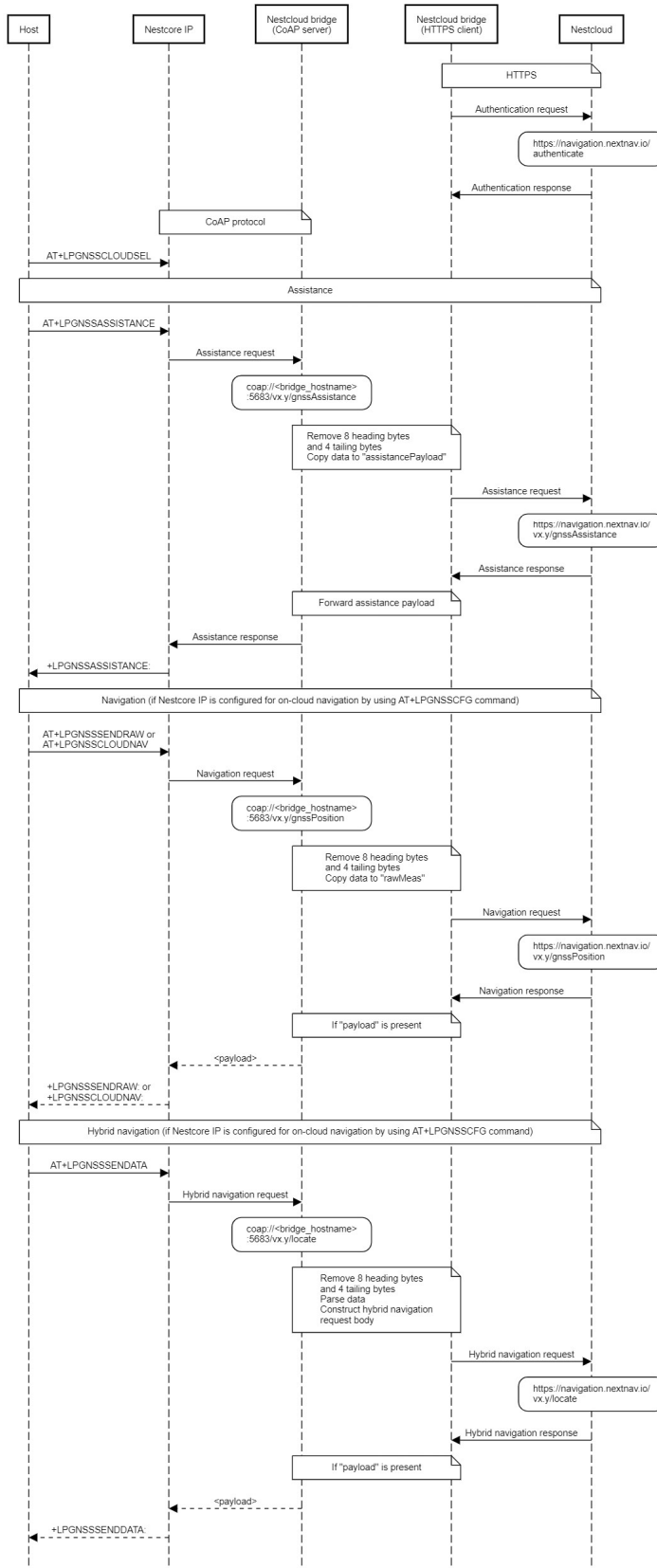
	<p>&lt;mobile country code&gt; is the LTE mobile country code</p> <p>&lt;mobile network code&gt; is the LTE mobile network code</p> <p>&lt;received power&gt; is the LTE or Wi-Fi or Bluetooth received power in dBm</p> <p>&lt;MAC address&gt; is the Wi-Fi access point or Bluetooth beacon MAC address</p> <p>&lt;average pressure&gt; is the average pressure in Pascal unit</p> <p>&lt;variance&gt; is the pressure variance</p> <p>&lt;count&gt; is the number samples used for average pressure computation</p> <p>&lt;minimum pressure&gt; is the minimum pressure value</p> <p>&lt;maximum pressure&gt; is the maximum pressure value</p>
<b>Response</b>	<p>&lt;hybrid navigation response payload in json format containing the device position information, to be processed by Nestcloud bridge (example: map display, log in a database...)&gt;</p> <p>NB: If the device position is requested by the device in the hybrid navigation request payload, then the hybrid navigation response payload contains an additional optional "payload" field. This payload must be forwarded as is by Nestcloud bridge CoAP server to the device in CoAP hybrid navigation API response&gt;.</p>

Example of hybrid navigation API response payload:

```
{
  "utcTime": "2023-11-07T14:29:18.150949001+00:00",
  "gpsTime": 1383402576.150949,
  "confidence": 11.193524,
  "position": [2.348045, 48.853610, 27.083687],
  "velocity": [-0.095578, 0.031254, 0.303247],
  "gps": {
    "prn": [29, 31, 25, 18, 2, 26, 22],
    "cn0": [44, 40, 42, 38, 40, 31, 33]
  },
  "payload":
  "MMEflqNKSEdcd5yPGlAEwNkq/kI/vsO9wgMAPSRDmz6tGDNBAQcdMGdPXQEdPgEfTgEZLQESRA
  EC9wAaCwEW",
  "HeightAboveTerrain": 22.439892425075655,
  "technology": "GNSS"
}
```

Optional "payload" parameter to be forwarded to the device, if present.

## 2.3 Overall sequence





### 3 Customer communication protocol

Customers may want to use their existing communication protocol between their devices and their cloud to piggyback GNSS data inside their existing data transfer packets.

#### 3.1 Devices to customer cloud data

3 kinds of GNSS data can be piggybacked inside customer data transfer packets: assistance data, navigation data or hybrid navigation data. Together with any GNSS data, the API version information must also be piggybacked.

##### 3.1.1 API version

The API version information to be piggybacked with any GNSS data is obtained using AT+LPGNSSCLOUDSEL? command on the device (see [1] for detailed syntax).

##### 3.1.2 Assistance data

Nestcore assistance request payload to be piggybacked inside uplink packets from the devices to customer cloud is obtained using AT+LPGNSSASSISTANCEPAYLOAD command on the device (see [1] for detailed syntax).

The assistance response payload to be piggybacked into devices downlink packets from customer cloud to devices is the data received from Nestcloud in assistance API response (see section 3.2.2 for more details).

This assistance payload is then passed to Nestcore IP using AT+LPGNSSSTOREASSISTANCE command on the device (see [1] for detailed syntax).

##### 3.1.3 Navigation data

Nestcore navigation request payload to be piggybacked into uplink packets from the devices to customer cloud is obtained by taking the <raw\_meas> parameter in the response to AT+LPGNSSGETFIX command on the device (see [1] for detailed syntax).

The navigation response data to be piggybacked in devices downlink packets from customer cloud to devices depends on device use case:

- If the device position is not needed by the device, then there is nothing to piggyback to the device
- If the device position is needed by the device, then the customer cloud needs to:
  - Determine the validity of the position information sent by Nestcloud:
    - $-1000 < \text{altitude} < 10000$
    - $-138.9 < \text{velocity} < 138.9$
    - $\text{Confidence} < 10000$
  - If the position information is valid then:
    - Send the position information such as latitude, longitude, altitude, velocity... (received from Nestcloud in navigation API response) to the device application using customer communication protocol
    - Update Nestcore IP approximate position of the device with latitude, longitude and altitude received from Nestcloud in navigation API response by using AT+LPGNSSAPPROXPOS command on the device (see [1] for detailed syntax)



- If the position information is invalid, then it is up to customer to notify their device of the invalid position or not

### 3.1.4 Hybrid navigation data

Nestcore hybrid navigation request payload to be piggybacked into uplink packets from the devices to customer cloud is obtained by taking the <raw\_meas> parameter in the response to AT+LPGNSSGETFIX command on the device (see [1] for detailed syntax). More location data such as cellular, Wi-Fi, Bluetooth information or atmospheric pressure can also be added to the hybrid navigation payload.

The navigation response data to be piggybacked in devices downlink packets from customer cloud to devices depends on device use case:

- If the device position is not needed by the device, then there is nothing to piggyback to the device
- If the device position is needed by the device, then the customer cloud needs to:
  - Determine the validity of the position information sent by Nestcloud:
    - $-1000 < \text{altitude} < 10000$
    - $-138.9 < \text{velocity} < 138.9$
    - Confidence < 10000
  - If the position information is valid then:
    - Send the position information such as latitude, longitude, altitude, velocity... (received from Nestcloud in navigation API response) to the device application using customer communication protocol
    - Update Nestcore IP approximate position of the device with latitude, longitude and altitude received from Nestcloud in navigation API response by using AT+LPGNSSAPPROXPOS command on the device (see [1] for detailed syntax)
  - If the position information is invalid, then it is up to customer to notify their device of the invalid position or not

## 3.2 Customer cloud to Nestcloud communication

### 3.2.1 Authentication API

In order to communicate with Nestcloud, customer cloud must first proceed to an authentication phase with Nestcloud by using the authentication API described in section 2.2.1.

### 3.2.2 Token renewal API

When expired, Nestcloud bridge must renew the authorization token by using the token renewal API described in section 2.2.2.

### 3.2.3 Assistance API

GNSS assistance data piggybacked from the devices to customer cloud must be forwarded as is to Nestcloud by using the following assistance API:

Resource	Value
URL	<code>https://navigation.nextnav.io/vx.y/gnssAssistance</code> where vx.y is the API version piggybacked from device to customer cloud (see section 3.1.1)
Request type	POST



<b>Header</b> <b>key=Authorization</b> <b>Content-Type=application/json</b>	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
<b>Body (json type)</b>	<pre>{   "deviceId": "&lt;device unique ID&gt;",   "assistancePayload": "&lt;assistance data&gt;" }</pre> <p>where:          &lt;unique device ID&gt; is a unique ID identifying the device          &lt;assistance data&gt; is the assistance data piggybacked from device to customer cloud, see section 3.1.2</p>
<b>Response</b>	<assistance response payload in binary format to be forwarded as is by customer cloud to the device in response to the uplink piggybacked GNSS assistance request data> see section 3.1.2 for more details

### 3.2.4 Navigation API

Any GNSS navigation data piggybacked from the devices to customer cloud must be forwarded as is to Nestcloud by using the following navigation API:

Resource	Value
<b>URL</b>	https://navigation.nextnav.io/vx.y/gnssPosition where vx.y is the API version piggybacked from device to customer cloud (see section 3.1.1)
<b>Request type</b>	POST
<b>Header</b> <b>key=Authorization</b> <b>Content-Type=application/json</b>	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
<b>Body (json type)</b>	<pre>{   "deviceId": "&lt;device unique ID&gt;",   "rawMeas": "&lt;navigation data&gt;",   "features": [&lt;feature list&gt;],   "approxPos": [&lt;longitude, latitude, altitude&gt;] }</pre> <p>where:          &lt;unique device ID&gt; is a unique ID identifying the device          &lt;navigation data&gt; is the navigation data piggybacked from device to customer cloud, see section 3.1.3          &lt;feature list&gt; is an optional coma-separated list of features to be activated on Nestcloud (example: "NoC", "AAN")          &lt;longitude, latitude, altitude&gt; is the optional device approximate position</p>
<b>Response</b>	<navigation response payload in json format containing the device position information, to be processed by customer cloud (example: map display, log in a database, forward to device...)> see section 3.1.3 for more details

Example of navigation API response payload:



```
{
  "utcTime": "2023-11-07T14:29:18.150949001+00:00",
  "gpsTime": 1383402576.150949,
  "confidence": 11.193524,
  "position": [2.348045, 48.853610, 27.083687],
  "velocity": [-0.095578, 0.031254, 0.303247],
  "gps": {
    "prn": [29, 31, 25, 18, 2, 26, 22],
    "cn0": [44, 40, 42, 38, 40, 31, 33]
  },
  "HeightAboveTerrain": 22.439892425075655
}
```

### 3.2.5 Hybrid navigation API

Any location data (GNSS, cellular, Wi-Fi, Bluetooth or atmospheric pressure) piggybacked from the devices to customer cloud must be forwarded to Nestcloud by using the following navigation API:

Resource	Value
<b>URL</b>	https://navigation.nextnav.io/vx.y/gnssPosition where vx.y is the API version piggybacked from device to customer cloud (see section 3.1.1)
<b>Request type</b>	POST
<b>Header</b> <b>key=Authorization</b> <b>Content-Type=application/json</b>	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
<b>Body (json type)</b>	{       "deviceId": "<device unique ID>",       "rawMeas": "<GNSS navigation payload>",       "features": [<feature list>],       "approxPos": [<longitude, latitude, altitude>],       "hybrid": {         "radioType": "LTE",         "cellTowers": [{           "cellId": <cell identifier>,           "lac": <location area code>,           "mcc": <mobile country code>,           "mnc": <mobile network code>,           "rsrp": <received power> }],         "wifiAccessPoints": [{           "mac": "<MAC address>",           "rssi": <received power> }],         "bluetoothBeacons": [{           "mac": "<MAC address>",           "rssi": <received power> }],         "pressure": {           "average": <average pressure>,           "variance": <variance>,           "count": <count>,           "min": <minimum pressure>,           "max": <maximum pressure> }}     }





	<p>where:</p> <p>&lt;unique device ID&gt; is a unique ID identifying the device</p> <p>&lt;GNSS navigation payload&gt; is the GNSS navigation payload received by Nestcloud bridge CoAP server on hybrid navigation API encoded in base64 format</p> <p>&lt;feature list&gt; is an optional coma-separated list of features to be activated on Nestcloud ("NoC", "AAN", "PAAN")</p> <p>&lt;longitude, latitude, altitude&gt; is the optional device approximate position</p> <p>&lt;cell identifier&gt; is the 28-bit LTE cell identifier</p> <p>&lt;location area code&gt; is the LTE location area code</p> <p>&lt;mobile country code&gt; is the LTE mobile country code</p> <p>&lt;mobile network code&gt; is the LTE mobile network code</p> <p>&lt;received power&gt; is the LTE or Wi-Fi or Bluetooth received power in dBm</p> <p>&lt;MAC address&gt; is the Wi-Fi access point or Bluetooth beacon MAC address</p> <p>&lt;average pressure&gt; is the average pressure in Pascal unit</p> <p>&lt;variance&gt; is the pressure variance</p> <p>&lt;count&gt; is the number samples used for average pressure computation</p> <p>&lt;minimum pressure&gt; is the minimum pressure value</p> <p>&lt;maximum pressure&gt; is the maximum pressure value</p>
<b>Response</b>	<navigation response payload in json format containing the device position information, to be processed by customer cloud (example: map display, log in a database, forward to device...) > see section 3.1.4 for more details

Example of hybrid navigation API response payload:

```
{
  "utcTime": "2023-11-07T14:29:18.150949001+00:00",
  "gpsTime": 1383402576.150949,
  "confidence": 11.193524,
  "position": [2.348045, 48.853610, 27.083687],
  "velocity": [-0.095578, 0.031254, 0.303247],
  "gps": {
    "prn": [29, 31, 25, 18, 2, 26, 22],
    "cn0": [44, 40, 42, 38, 40, 31, 33]
  },
  "HeightAboveTerrain": 22.439892425075655,
  "technology": "GNSS"
}
```

### 3.3 Overall sequence

