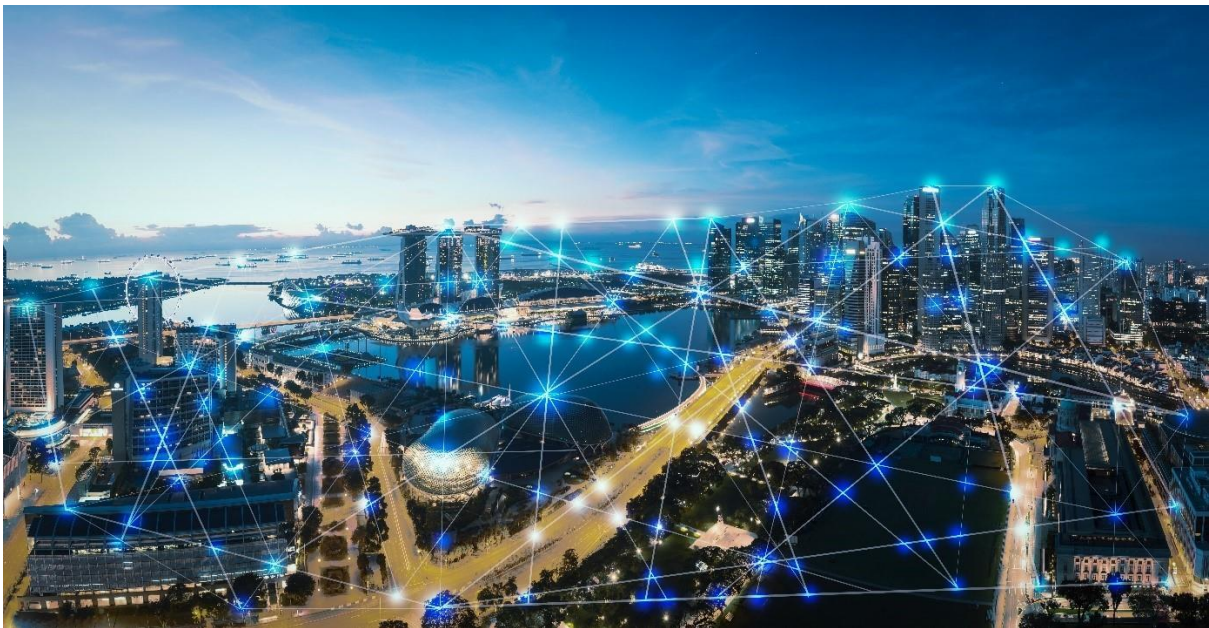




NestCloud Bridge Specification



* * * * *

Nestwave SAS

114 avenue Charles De Gaulle

92200 Neuilly-sur-Seine, France



nestwave

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



nestwave

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
2.2 NestCloud bridge to NestCloud communication.....	7
2.2.1 Authentication API	7
2.2.2 Token renewal API.....	7
2.2.3 Assistance API.....	7
2.2.4 Navigation API	8
2.3 Overall sequence	9
3 Customer communication protocol	11
3.1 Devices to customer cloud data	11
3.1.1 API version.....	11
3.1.2 Assistance data	11
3.1.3 Navigation data	11
3.2 Customer cloud to NestCloud communication	12
3.2.1 Authentication API	12
3.2.2 Token renewal API.....	12
3.2.3 Assistance API.....	12
3.2.4 Navigation API	12
3.3 Overall sequence	13



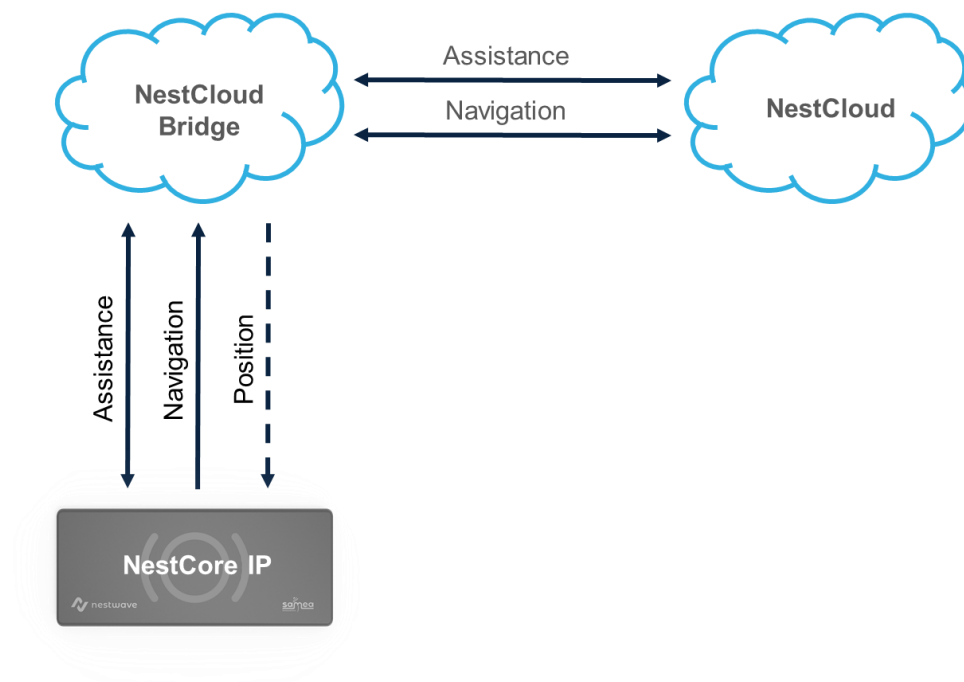
nestwave

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 both devices and NestCloud.

The overall diagram is shown below:





In order to further describe in details the messages exchanged between the devices, NestCloud bridge and NestCloud, we need to consider 2 use cases:

- Devices to NestCloud bridge communication is using NestCore IP protocol (CoAP)
- Devices to NestCloud bridge communication is using 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 2 CoAP APIs: assistance API and navigation API.

2.1.1 Assistance API

Resource	Value
URL	coap://<bridge_hostname>:5683/vx.y/gnssAssistance
Request type	POST
Payload	<assistance request payload to be forwarded to NestCloud assistance API as is>
Response	<assistance response payload received from NestCloud assistance API>

The assistance API is used to get recent assistance data by using AT+LPGNSSASSISTANCE command (see [1] for detailed syntax).

The response contains the assistance data to be used by NestCore IP.

2.1.2 Navigation API

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

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

The response contains a navigation payload (including the device position) that is used by NestCore IP. The device position is returned to the device by the +LPGNSSSENDRAW or +LPGNSSCLOUDNAV URCs (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 with NestCloud by using the following authentication API:

Resource	Value
URL	https://cloud.nestwave.com:9443/authenticate
Request type	POST
Body (raw type, json format)	{"username":"<user@company.com>","password":"<pwd>"} where <user@company.com> and <pwd> are the credentials used during NestCloud account creation
Response	{"token":"<access_token>"} 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
URL	https://cloud.nestwave.com:9443/authenticate/renew
Request type	POST
Header key=Authorization	Bearer <old_token> where <old_token> is the old authorization token previously received in the authentication phase
Response	{"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

Any assistance message payload received by NestCloud CoAP server from the devices must be forwarded as is to NestCloud by using the following assistance API:

Resource	Value
URL	https://cloud.nestwave.com:8443/vx.y/gnssAssistance where vx.y is the API version received by NestCloud CoAP server in the assistance API URL
Request type	POST
Header key=Authorization Content-Type=application/octet-stream	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase

Body (binary type)	<assistance request payload received by NestCloud bridge CoAP server on assistance API>
Response	<assistance response payload 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

Any navigation message payload received by NestCloud bridge CoAP server from the devices must be forwarded as is to NestCloud by calling the following navigation API:

Resource	Value
URL	https://cloud.nestwave.com:8443/vx.y/gnssPosition where vx.y is the API version received by NestCloud bridge CoAP server in the navigation API URL
Request type	POST
Header key=Authorization Content-Type=application/octet-stream	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
Body (binary type)	<navigation request payload received by NestCloud bridge CoAP server on navigation API>
Response	<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:

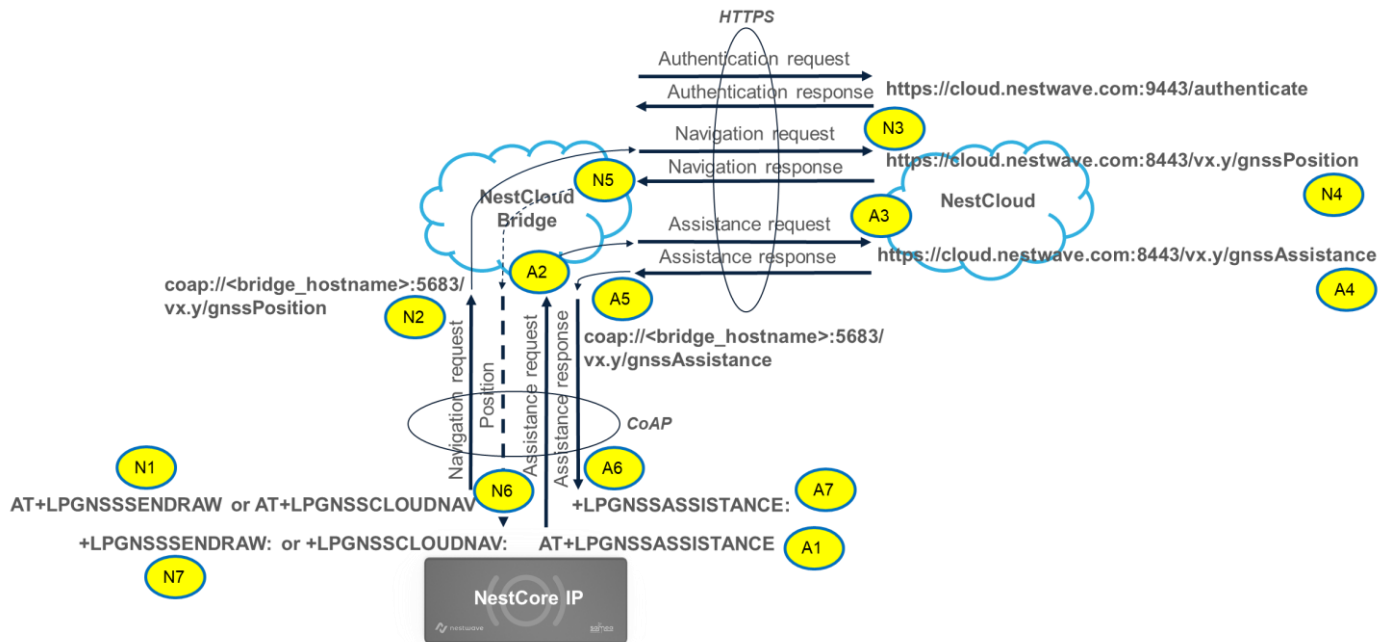
```
{
  "gpsTime":1332162589,
  "confidence":11.193524,
  "position":[2.348045,48.853610,27.083687],
  "velocity":[-0.095578,0.031254,0.303247],
  "gps":{
    "prn":[29,31,25,18,2,26,22],
    "cn0":[44,40,42,38,40,31,33]
  },
  "payload":"MMEflqNKSEdcd5yPGIAEwNkq/kl/vsO9wgMAPSRDmz6tGDNBAQcdMGdPXQEdPgEfTgEZLQESRAEC9wAaCwEW"
}
```

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



nestwave

2.3 Overall sequence



Prerequisites:

- NestCloud bridge hostname is set in NestCore IP using `AT+LPGNSSCLOUDSEL` command
- NestCloud bridge authentication is done using NestCloud authenticate API

For assistance requests:

- A1. The device checks NestCore IP assistance validity using `AT+LPGNSSASSISTANCE?` Command. If the assistance has expired, then the device requests an assistance update using `AT+LPGNSSASSISTANCE` command
- A2. This triggers an uplink CoAP assistance request from NestCore IP to NestCloud bridge `gnssAssistance` API
- A3. The CoAP assistance request payload is forwarded as is in an HTTPS assistance request to NestCloud `gnssAssistance` API
- A4. NestCloud calculates the requested assistance data
- A5. The assistance data is sent back to NestCloud bridge in the `gnssAssistance` API response payload
- A6. This payload is forwarded as is to NestCore IP in the downlink CoAP assistance response payload
- A7. NestCore IP updates the assistance data and generates a `+LPGNSSASSISTANCE` URC to the device

For navigation requests (in case device is configured for on-cloud navigation by using `AT+LPGNSSCFG` command):

- N1. A position is requested by the device using `AT+LPGNSSFIXPROG` command. The device receives the `+LPGNSSFIXREADY` URC when the GNSS satellites measurements are ready. The device requests an on-cloud navigation using `AT+LPGNSSSENDRAW` or `AT+LPGNSSCLOUDNAV` command and setting the `<return_position>` parameter to 1 if the

device position is needed on the device (see [1] for detailed syntax). Please note that it is preferable to send back the position to the device when possible as this is used as a starting point for the next fix which will help to reduce TTFF

- N2. This triggers an uplink CoAP navigation request from NestCore IP to NestCloud bridge gnssPosition API
- N3. The CoAP navigation request payload is forwarded as is in an HTTPS navigation request to NestCloud gnssPosition API
- N4. NestCloud calculates the device position
- N5. The device position information is sent back to NestCloud bridge in the gnssPosition API response payload
- N6. If the navigation response payload contains the “payload” parameter, then it is forwarded as is to NestCore IP in the downlink CoAP navigation response payload
- N7. NestCore IP updates the device approximate position with the received position and generates a +LPGNSSSENDRAW or +LPGNSSCLOUDNAV URC to the device

3 Customer communication protocol

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

3.1 Devices to customer cloud data

2 kinds of GNSS data can be piggybacked in customer data transfer packets: assistance data or 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 (see [1] for detailed syntax).

3.1.2 Assistance data

NestCore assistance request data to be piggybacked in uplink packets from the devices to customer cloud is obtained using AT+LPGNSSASSISTANCEPAYLOAD command (see [1] for detailed syntax).

The assistance response data to be piggybacked in 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 data is then passed to NestCore IP using AT+LPGNSSSTOREASSISTANCE command (see [1] for detailed syntax).

3.1.3 Navigation data

NestCore GNSS satellites measurement data to be piggybacked in uplink packets from the devices to customer cloud is obtained using the <raw_meas> parameter in AT+LPGNSSGETFIX command (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 (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

Any 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	https://cloud.nestwave.com:8443/vx.y/gnssAssistance where vx.y is the API version piggybacked from device to customer cloud (see section 3.1.1)
Request type	POST
Header key=Authorization Content-Type=application/json	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
Body (json type)	{"deviceId": "<device unique ID>", "assistancePayload": "<GNSS assistance data piggybacked from device to customer cloud, see section 3.1.2>"}
Response	<assistance response payload 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>

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
URL	https://cloud.nestwave.com:8443/vx.y/gnssPosition where vx.y is the API version piggybacked from device to customer cloud (see section 3.1.1)
Request type	POST
Header key=Authorization Content-Type=application/json	Bearer <access_token> where <access_token> is the authorization token received in the authentication phase
Body (json type)	{"deviceId": "<device unique ID>", "rawMeas": "<GNSS navigation data piggybacked from device to customer cloud, see section 3.1.3>"}
Response	<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>

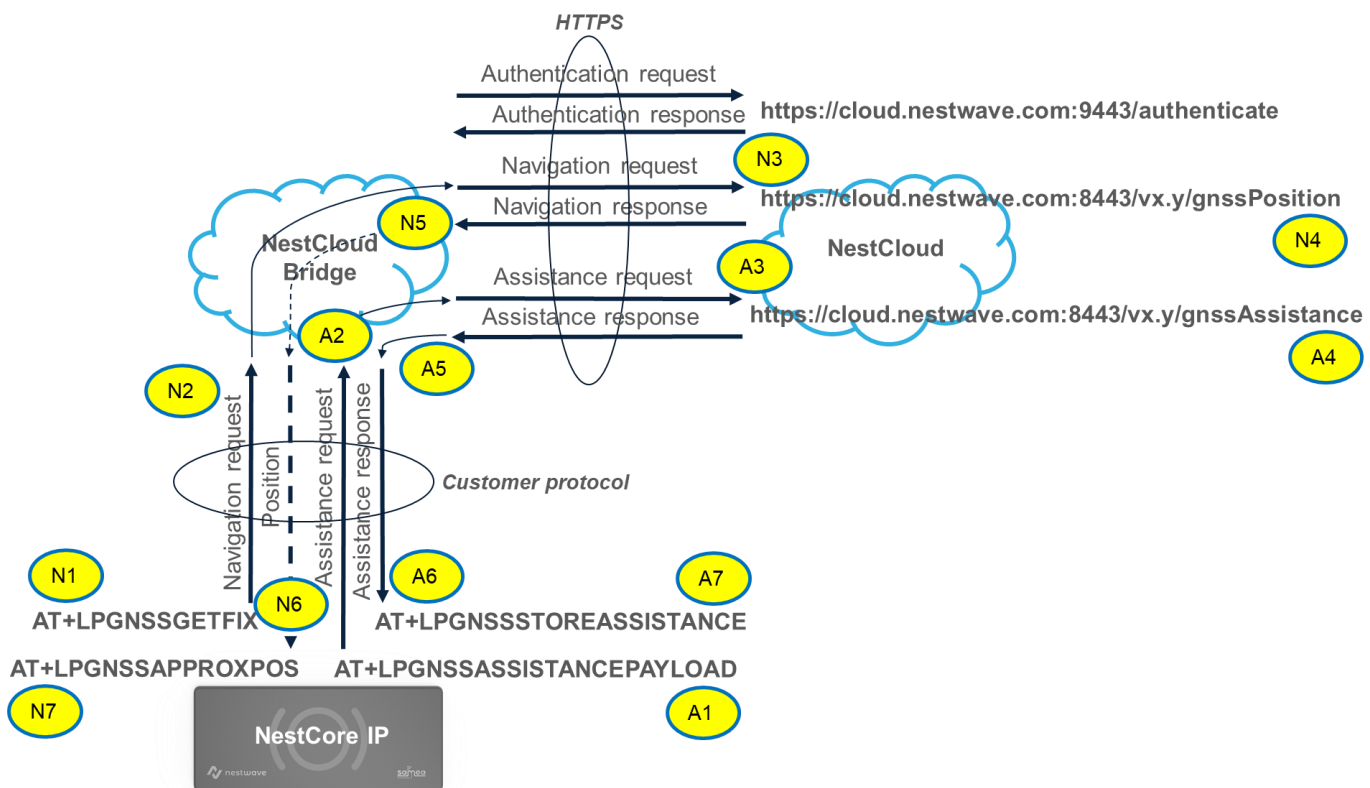


nestwave

Example of Navigation API response payload:

```
{
  "gpsTime":1332162589,
  "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]},
  }
```

3.3 Overall sequence



Prerequisites:

- NestCloud API version is obtained using AT+LPGNSSCLOUDSEL? command
- NestCloud bridge authentication is done using NestCloud authenticate API

For assistance requests:

- A1. The device checks NestCore IP assistance validity using AT+LPGNSSASSISTANCE? Command. If the assistance has expired, then the device gets the assistance request payload using AT+LPGNSSASSISTANCEPAYLOAD command
- A2. The device piggybacks the assistance request payload in an uplink customer message to the customer cloud

- A3. The assistance request payload is forwarded as is by the customer cloud in an HTTPS assistance request to NestCloud gnssAssistance API
- A4. NestCloud calculates the requested assistance data
- A5. The assistance data is sent back to the customer cloud in the gnssAssistance API response payload
- A6. This payload is piggybacked as is in a downlink customer message to the device
- A7. The device updates NestCore IP assistance data using AT+LPGNSSSTOREASSISTANCE command

For navigation requests:

- N1. A position is requested by the device using AT+LPGNSSFIXPROG command. The device receives the +LPGNSSFIXREADY URC when the GNSS satellites measurements are ready. The device gets the navigation request payload using AT+LPGNSSGETFIX command
- N2. The device piggybacks the navigation request payload in an uplink customer message to the customer cloud
- N3. The navigation request payload is forwarded as is by the customer cloud in an HTTPS navigation request to NestCloud gnssPosition API
- N4. NestCloud calculates the device position
- N5. The device position information is sent back to the customer cloud in the gnssPosition API response payload
- N6. If the device position is needed by the device and if the position information sent by NestCloud is valid, then the customer cloud sends the position information such as latitude, longitude, altitude, velocity... to the device application using customer communication protocol
- N7. The device updates NestCore IP approximate position with the received position information using AT+LPGNSSAPPROXPOS command