# D-FLIGHT

## U-BOX and UTM Interface Control Document (ICD)

**VERSION 1.2**
**15/02/2021**

## Amendment Status Sheet

| Issue | Date | Reason |
|---|---|---|
| 1.0 | 17/07/2020 | First Release - internal |
| 1.1 | 22/12/2020 | Alignement to  D-Flight ICD v1 compliance |
| 1.2 | 15/02/2021 | Connection Details provided |

| Issue | Date | Description of Change |
|---|---|---|
| 1.0 | 17/07/2020 | Document Created |
| 1.1 | 22/12/2020 | • TRACKING protocol updated (sec 3.1)<br>• Device-type Header added (sec 3.1)<br>• mandatory column Y/N added (sec 4.1)<br>• Authentication Token methods updated (sec 5.1)<br>• Device Type allowed values updated (sec 5.4)<br>• dev values updated (sec  4.1.2)<br>• Added figure captions (sec 2)<br>• Statedata example updated (sec 4.1.3)<br>• Updated Architecture/Interface pictures (sec 2) |
| 1.2 | 15/02/2021 | • Added email for retrieving client/secred credentials (sec 5,2)<br>• Added minimal example for tracking messge (sec 4.1.10)<br>• Added Example section with endpoints and parameters details (sec 5.5.3)<br>• Added Test Platform (sec 5.5.4) |

## Document Change Record History

## Index

# 1 Introduction

## 1.1 Purpose and Scope

This document provides description of the external interfaces of DFLIGHT UTM Platform System. It specifies the input/output messages to be sent to UTM platform in order to implement and properly manage the tracking functionality. Following sections describes possible available scenarios, including involved components, required interfaces, and exchanged messages. Applicable and Reference Documents

| ID | TITOLO | CODICE |
|---|---|---|
| ~~R1~~ | ~~PropostaTecnica_D-Flight_v0.12~~ | |
| R2 | WG-105 SG32 MOPS eID ET_20200120_d_for Peer Review | |

## 2 Interface Context

Following figures reports the context diagram of the system, showing all high level involved components.

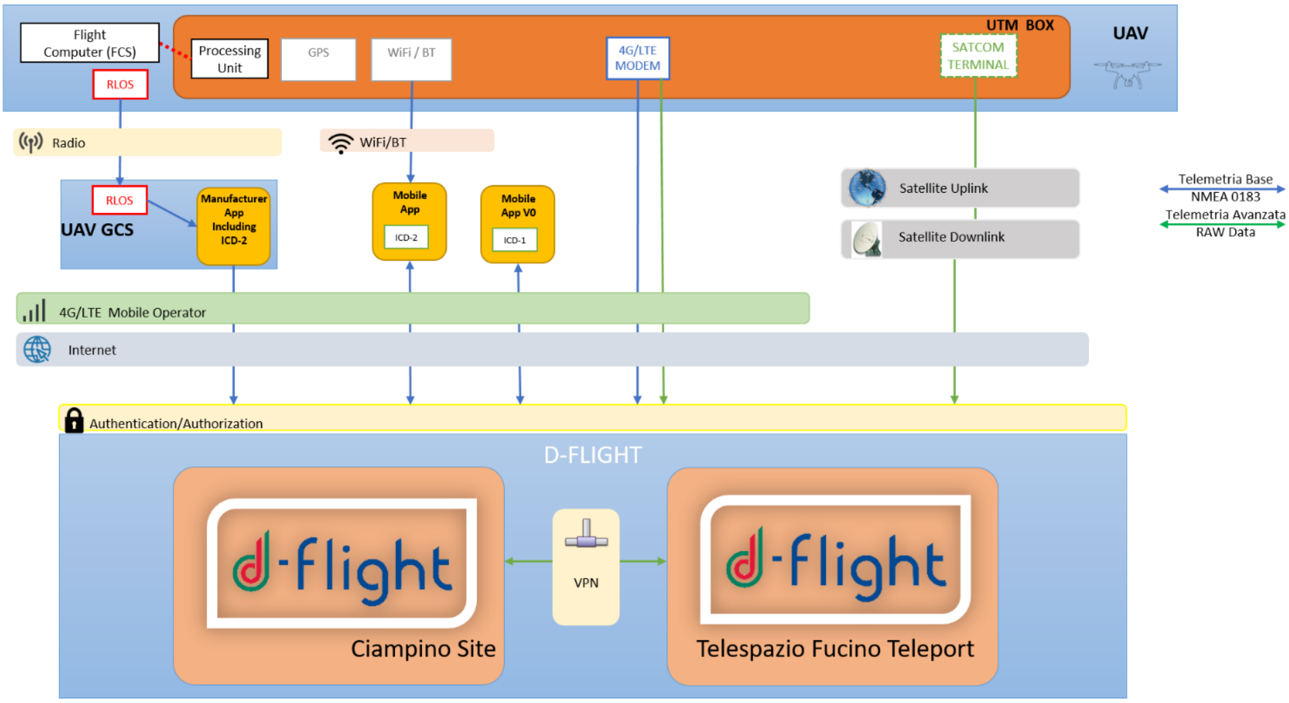

**Figure 1: High Level components interactions**

There are two different and possible scenarios, depending on the project phase: the possibility to dispatch only basic tracking information (Project Phase2), and/or the availability of sending RAW positioning data through a Satellite connection (Project Phase3).

## *2.1 Phase 2: Base Telemetry Scenario*

In this scenario, the most common one, only basic telemetry/position information is dispatched, no Satellite connection is available from the drone.

Following image depicts all involved components and their interfaces/connections.

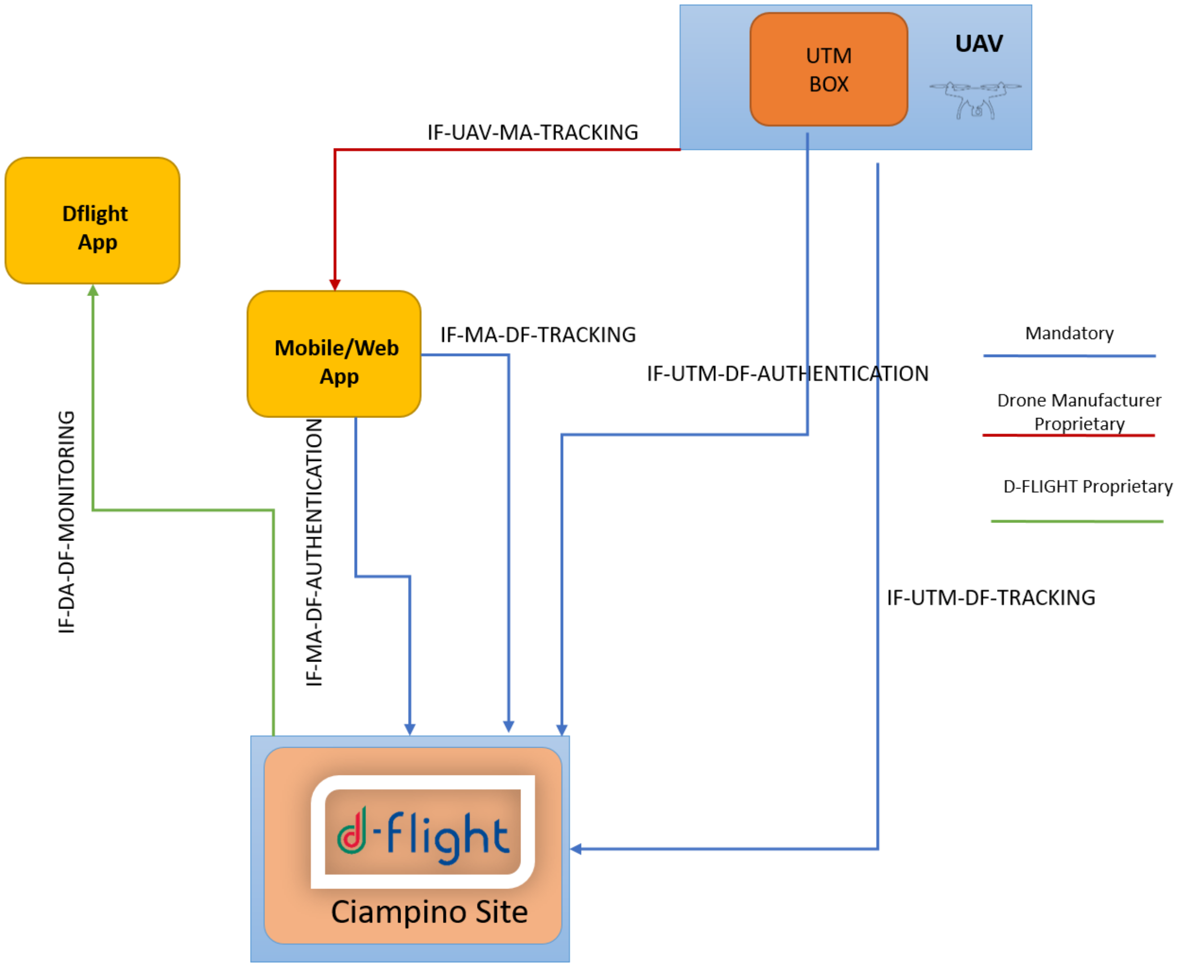

**Figure 2: Phase 2 (ICD V1 compliance) interface context**

Tracking information can be either sent by the UTM Box physically located inside the drone, or by a Mobile (or web) Application developed by the Drone manufacturer.

In this latter case, such tracking information shall be retrieved from the drone by the Mobile/Web application (UAV-Tracking interface in the figure): the content of this message, providing that it contains all needed data, is proprietary of the drone manufacturer and it is out of the scope of this document.

The interfaces in the figure are detailed in section 3

## *2.2 Phase 3: Advanced Telemetry Scenario*

This scenario is to be considered as an alternative to scenario described in previous section.

In this scenario, advanced telemetry/position is available and – possibly – also a Satellite connection is available from the drone.

Following image depicts all involved components and their interfaces/connections.

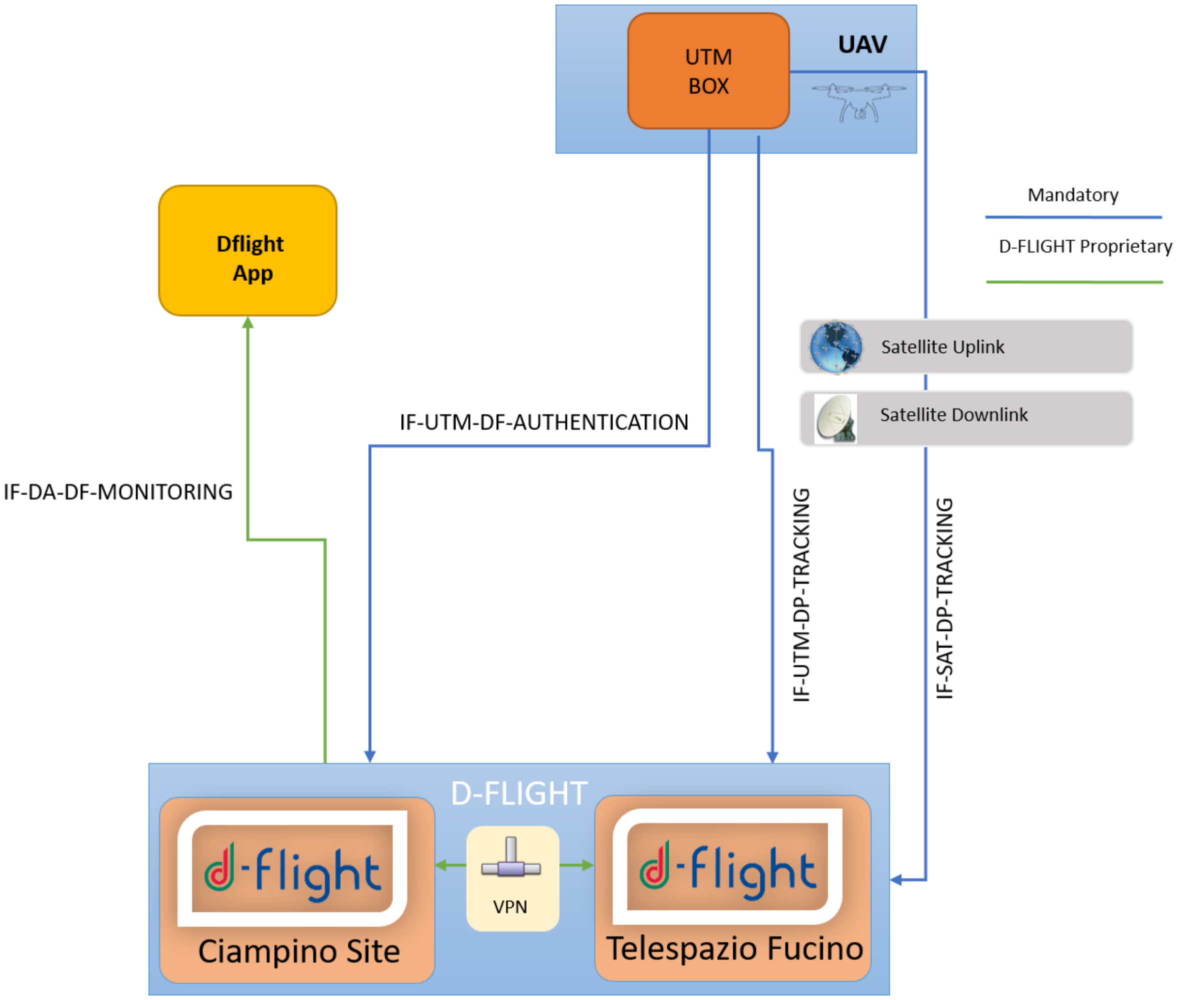

**Figure 3: Phase 3 (ICD V2 compliance) interface context**

Tracking information contains additional fields specifying Satellite Raw positioning Data. Such information is sent directly from the UTM-Box of the UAV through a satellite link connection. A dedicated module (DFLIGHT-Proxy in the figure) will receive and process the information.

Optionally, the Mobile/Web Application can establish a Monitoring connection to D-Flight server, in order to receive the positions of other UAVs in the flying area and – if necessary – display them on a map.

The listed interfaces are detailed in section 3

# 3 Interface Description

This section describes the details of the interface listed in previous section.

### 3.1.1 IF-UAV-MA-TRACKING

This interface is proprietary of Drone manufacturer and it is out of the scope of this document

### 3.1.2 IF-MA-DF-AUTHENTICATION

This interface allows a client perform an authentication to D-Flight system. It is based on openID v1.0 protocol and it is described more in details on section 5.1

### 3.1.3 IF-MA-DF-TRACKING

This interface allows third party Mobile Apps to send positioning/tracking information to DFLIGHT system.

- Protocol: STOMP over WebSocket
- Authentication HTTP or STOMP Header: it shall contain a valid access_token (for more details refer on section 5.1)
- (optional) Device type HTTP or STOMP Header: it should contain the type of device being connected to the system. (for more details refer on section 5.4)
- Payload
  - Format: both JSON and ASTERIX***** formats will be accepted
  - Content: Tracking Message payload (limited to: Identification, State Data, Status, Intent, Application, Geo Fencing, Augmentation). Refer to Tracking Message section for more details.
- Ouptut: N/A
- Mandatory: yes

* ASTERIX NOT CURRENTLY SUPPORTED – REFER TO 4.2 SECTION FOR MORE DETAILS

### 3.1.4 IF-DA-DF-MONITORING

This interface is proprietary of DFLIGHT mobile/web app, and it is out of the scope of this document.

### 3.1.5 IF-UTM-DF-AUTHENTICATION

This interface is alternative to IF-MA-DF-AUTHENTICATION. Refer to such interface description for details.

### 3.1.6 IF-UTM-DF-TRACKING

This interface is alternative to IF-MA-DF-TRACKING. Refer to such interface description for all details.

### 3.1.7 IF-UTM-DP-TRACKING

This interface allows to send advanced positioning/tracking information to DFLIGHT system through DFLIGHT-Proxy module. It is a one-way message.

Note that IF-UTM-DP-TRACKING and IF-SAT-DP-TRACKING messages will be sent in parallel.

- Protocol: STOMP over WebSocket
- Authentication HTTP or STOMP Header: it shall contain a valid access_token (for more details refer on section 5.1)
- (optional) Device type HTTP or STOMP Header: it should contain the type of device being connected to the system. (for more details refer on section 5.4).
- Payload
    - Format: JSON, JSON/ZIP and ASTERIX***** formats will be accepted
    - Content: Tracking Message payload (limited to: Identification, State Data, Status, Intent, Application, Geo Fencing, Augmentation, Raw Data). Refer to Tracking Message section for more details.
- Mandatory: yes

* ASTERIX NOT CURRENTLY SUPPORTED – REFER TO 4.2 SECTION FOR MORE DETAILS

### 3.1.8 IF-SAT-DP- TRACKING

This interface, only available if the APR is equipped with a SATCOM terminal, it allows to send advanced positioning/tracking information to DFLIGHT system through a satellite connection to

DFLIGHT-Proxy module. The type and content of this message is equals to IF-UTM-DP-TRACKING. Refer to such interface for all details.

### 3.1.9 IF-DP-DF-TRACKING

This interface allows to propagate DFLIGHT-Proxy aggregated information to D-Flight Data center. It is a one-way message.

- Protocol: STOMP over WebSocket
- Authentication HTTP or STOMP Header: it shall contain a valid access_token (for more details refer on section 5.1)
- (optional) Device type HTTP or STOMP Header: it should contain the type of device being connected to the system. (for more details refer on section 5.4).
- Payload
  - Format: both JSON and ASTERIX***** formats will be accepted
  - Content: Tracking Message payload (limited to: Identification, State Data, Status, Intent, Application, Geo Fencing, Augmentation). Refer to Tracking Message section for more details.
- Mandatory: yes

* ASTERIX NOT CURRENTLY SUPPORTED – REFERT TO 4.2 SECTION FOR MORE DETAILS

# 4 Tracking Message Specification

## 4.1 Tracking Message Specification (JSON)

The payload forwarded to the UTM platform shall have a JSON format composed from different basic sections:

1. IDENTIFICATION: it shall contain the UA and operator identifications;
2. STATEDATA: it shall contain the position data of the UA at a given time
3. STATUS: it shall contain the health of the main devices and the accuracy of the position and speed values reported in the STATEDATA section
4. INTENT: it shall contain the future intention of the UA as the next waypoint position and altitude
5. APPLICATION: it shall contain data that is required only for certain purposes or applications, and specifically the take-off and the RPS position
6. GFENCING: it shall contain the time of last geofencing database update
7. AUGMENTATION: it shall contain information about GNSS Augmentation system used by the RPAS (if any) to improve and validate positioning data
8. RAW DATA: it shall contain satellite raw data information retrieved from GPS receiver

A single payload shall be composed from:

- IDENTIFICATION section
- One or more of the remaining basic sections, according to interface description (section 3 )

This composition makes easy to specify different rates of transmission for each different section and permits to reduce the throughput required.

### 4.1.1 Transmission rates

The frequency of transmission shall be (recommendation WG-105):

| | |
|---|---|
| STATEDATA | At a 2 Hz rate as a minimum when airborne<br>At a 1 Hz otherwise |
| STATUS | As soon as a change occurs<br>At a 0.1 Hz rate as a minimum otherwise |
| INTENT | 2 Hz rate as a minimum when the UA is flown manually<br>0.1 Hz rate as a minimum when the UA is flown in automatic mode |
| APPLICATION | At a 2 Hz rate as a minimum when airborne<br>At a 0.1 Hz rate as a minimum otherwise |
| GFENCING | As soon as a change occurs (TBC)<br>At a 0.1 Hz rate as a minimum otherwise |
| AUGMENTATION | At a 2 Hz rate as a minimum when airborne<br>At a 1 Hz otherwise |

### 4.1.2 Identification section layout

| Element | Description | Mandatory |
|---|---|---|
| UAId | Identification of the UA according *ANSI/CTA-2063* | Y |
| OpId | Operator Identification | N |
| src | Type of data channel. It can be:<br>0=Bluetooth,<br>1=WIFI,<br>2=Lora,<br>3=4G/LTE,<br>4=Satellite | Y |
| dev | Type of device source. It can be:<br>0=U-Box On board<br>1=U-Box c/o GCS,<br>2=Virtual U-Box<br>3=Reserved<br>4=Drone Operation Area | Y |

Example:

```
identification:
{
"UAId": "1234567890AB",
"OpId": "Operator2020",
"src": 1,
"dev": 1
}
```

### 4.1.3 State Data section layout

| Element | Description | Mandatory |
|---|---|---|
| time | Timestamp of position update<br>Time of Day in UTC.<br>When transmitted as string, use 3 decimal digits, at least | Y |
| lat | WGS-84 latitude<br>Latitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | Y |
| lon | WGS-84 longitude<br>Longitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | Y |
| height | WGS-84 height<br>Height in decimal format. Unit of measure m<br>When transmitted as string, use 1 decimal digits, at least | Y |
| altitudeMSL | Altitude above Mean Sea Level<br>Altitude in decimal format. Unit of measure m<br>When transmitted as string, use 1 decimal digits, at least | N |
| speedNS | Ground speed North axis<br>Speed in decimal format. Unit of measure m/s<br>When transmitted as string, use 2 decimal digits, at least | N |
| speedEW | Ground speed East axis<br>Speed in decimal format. Unit of measure m/s<br>When transmitted as string, use 2 decimal digits, at least | N |
| VRate | Climb/descent rate<br>Vertical Speed in decimal format. Unit of measure m/s<br>When transmitted as string, use 2 decimal digits, at least | N |

Example:

```
statedata:
{
"time": "17:22:26.711",
"lat": 42.123451,
"lon": 11.123451,
"height": 129,
"altitudeMSL": 42.781,
"speedNS": 67.33,
"speedEW": 12.35,
"VRate": 4.25
}
```

### 4.1.4 Status section layout

| Element | Description | Mandatory |
|---|---|---|
| AGMode | Air/Ground mode.<br>0: On Ground<br>1: Airborne | N |
| Payload | Nature of the UA payload<br>0: Unknown<br>1: Sensors (e.g. imagery)<br>2: Goods (e.g. parcels)<br>3: Medical goods (e.g. organs for transplantation)<br>4: Dangerous goods<br>5: Passengers<br>6: Others | N |
| Priority | Priority reports the urgency of the UA operation<br>Priority shall be encoded as follows:<br>0: Unknown<br>1: Low<br>2-6: To be defined<br>7: High | N |
| UAHealth | UA health status<br>0: Unknown<br>1: Nominal (no failure)<br>2: Degraded (failure detected but flight can still continue)<br>3: Emergency (failure detected and flight shall be terminated) | N |
| RPSHealth | RPS health status<br>0: Unknown<br>1: Nominal (no failure)<br>2: Degraded (failure detected but flight can still continue)<br>3: Emergency (failure detected and flight shall be terminated) | N |
| LinkHealth | C2 link health status<br>1: Nominal (no failure)<br>2: Degraded (degradation in performance)<br>3: Lost (total loss of capability) | N |
| FCSHealth | Flight control system health status<br>1: Nominal (no failure)<br>2: Degraded (degradation in performance)<br>3: Lost (total loss of capability) | N |
| EngHealth | UA engine(s) status<br>1: Nominal (no failure)<br>2: Degraded (degradation in performance)<br>3: Lost (failure detected on at least one engine/motor and associated emergency procedure is engaged) | N |

| | | |
|---|---|---|
| PwrStatus | Power (fuel/battery) level status<br>0: Unknown<br>1: Nominal (no failure)<br>2: Degraded (power low but flight can still continue)<br>3: Low (power low and power low procedure is engaged) | N |
| CDAAStatus | Cooperative DAA health status<br>0: Unknown<br>1: Nominal (no failure)<br>2: Degraded (degradation in performances)<br>3: Lost (total loss of capability) | N |
| NDAAStatus | Non-cooperative DAA health status<br>0: Unknown<br>1: Nominal (no failure)<br>2: Degraded (degradation in performances)<br>3: Lost (total loss of capability) | N |
| TrjStatus | UA trajectory/mode engaged<br>0: Unknown (or not valid)<br>1: Nominal (following pre-planned trajectory)<br>2: De-confliction (following U-space tactical de-confliction or RWC manoeuvre requested by the RP)<br>3: Collision Avoidance (following a Collision Avoidance trajectory-avoidance manoeuvre till clear of conflict (CoC))<br>4: Emergency (following a trajectory engaged by an emergency procedure)<br>5: Out of Control (following an undefined trajectory due to failure) | N |
| TimeValidity | Time (timestamp) validity<br>0: Time information is invalid/not available or exceed the requested accuracy<br>1: Time information is valid and requested accuracy can be satisfied | N |
| IDValidity | UA identification validity<br>0: Identification information is not available<br>1: Identification information is valid<br>2: Identification information is NOT valid | N |
| PosValidity | UA position validity<br>0: Position sources are /not available or exceed the requested accuracy<br>1: Position sources are valid and requested accuracy can be satisfied<br>2: Identification information is NOT valid | N |

| | | |
|---|---|---|
| AltValidity | UA altitude / height validity<br>0: Altitude/Height sources are /not available or exceed the requested accuracy<br>1: Altitude/Height sources are valid and requested accuracy can be satisfied<br>2: Identification information is NOT valid | N |
| GndValidity | UA ground speed validity<br>0: Ground Speed sources are /not available or exceed the requested accuracy<br>1: Ground Speed sources are valid and requested accuracy can be satisfied<br>2: Identification information is NOT valid | N |
| VRateValidity | UA vertical rate validity<br>0: Vertical Speed sources are /not available or exceed the requested accuracy<br>1: Vertical Speed sources are valid and requested accuracy can be satisfied<br>2: Identification information is NOT valid | N |
| IntValidity | UA intent information validity / availability<br>0: Intent data elements sources are /not available or exceed the requested accuracy<br>1: Intent data elements sources are valid and requested accuracy can be satisfied2: Identification information is NOT valid | N |
| GFValidity | UA Geo Fencing information validity / availability<br>0: Geo Fencing database is /not available<br>1: Geo Fencing database is valid<br>2: Identification information is NOT valid | N |
| PosFOM | UA position uncertainty (FOM)<br>Estimated Position Uncertainty (EPU) as follows:<br>0: Unknown<br>1-9: Reserved for compatibility<br>10: EPU < 10 m<br>11: EPU < 3 m<br>12: EPU < 1m<br>13-15: Reserved for future | N |

| | | |
|---|---|---|
| AltFOM | UA altitude/height uncertainty (FOM)<br>Vertical Estimated Position Uncertainty (VEPU) as follows:<br>0: Unknown<br>1-9: Reserved for compatibility<br>10: VEPU < 15 m<br>11: VEPU < 4 m<br>12: VEPU < 1 m<br>13-15: Reserved for future | N |
| GndSFOM | UA ground speed uncertainty (FOM)<br>Horizontal Figure of Merit Reported (HFOMR) as follows:<br>0: Unknown<br>1: Reserved for compatibility<br>2: HFOMR < 3 m/s<br>3: HFOMR < 1 m/s<br>4: HFOMR < 0.3 m/s | N |
| VRateFOM | UA vertical rate uncertainty (FOM)<br>Vertical Figure of Merit Reported (VFOMR) as follows:<br>0: Unknown<br>1: Reserved for compatibility<br>2: VFOMR < 4.5 m/s<br>3: VFOMR < 1.52 m/s<br>4: VFOMR < 0.46 m/s | N |

Example:

```
status:
{
"AGMode": 1,
"Payload": 3,
"Priority”: 7,
"UAHealth": 1,
"RPSHealth": 1,
"LinkStatus": 1,
"FCSStatus": 1,
"EngStatus": 1,
"PwrStatus”: 1,
"CDAAStatus": 1,
"NDAAStatus": 0,
"TrjStatus": 0,
"TimeValidity": 1,
"IDValidity": 1,
"PosValidity": 1,
"AltValidity": 1,
"GndSValidity": 1,
"VRateValidity": 1,
"IntValidity": 1,
"GFValidity": 1,
"PosFOM": 12,
"AltFOM": 12,
"GndSFOM": 3,
"VRateFOM": 3

}
```

### 4.1.5 Intent section layout

| Element | Description | Mandatory |
|---|---|---|
| time | Timestamp of next position<br>Time of Day in UTC.<br>When transmitted as string, use 3 decimal digits, at least | N |
| lat | WGS-84 latitude<br>Latitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | N |
| lon | WGS-84 longitude<br>Longitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | N |
| height | WGS-84 height<br>Height in decimal format. Unit of measure m<br>When transmitted as string, use 1 decimal digits, at least | N |
| altitudeMSL | Altitude above Mean Sea Level<br>Altitude in decimal format. Unit of measure m<br>When transmitted as string, use 1 decimal digits, at least | N |
| dataOrigin | Origin of the Intent data<br>0: Unknown<br>1: UA<br>2: Remote Pilot Station (RPS)<br>3: Extrapolated by RPS | N |

Example:

```
intent:
{
"time": "17:22:26.711",
"lat": 42.123451,
"lon": 11.123451,
"height": 29,
"altitudeMSL": 42.781,
"dataOrigin": 1
}
```

### 4.1.6 Application Section layout

| Element | Description | Mandatory |
|---|---|---|
| time | Timestamp of RPS position<br>Time of Day in UTC.<br>When transmitted as string, use 3 decimal digits, at least | N |
| TOLat | WGS-84 latitude of take-off location<br>Latitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | N |
| TOLon | WGS-84 longitude of take-off location<br>Longitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | N |
| TOAlt | Altitude above Mean Sea Level<br>Altitude in decimal format. Unit of measure m<br>When transmitted as string, use 1 decimal digits, at least | N |
| RPSLat | WGS-84 latitude of RPS location<br>Latitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | N |
| RPSLon | WGS-84 longitude of RPS location<br>Longitude in decimal format. Unit of measure deg<br>When transmitted as string, use 5 decimal digits, at least | N |
| RPSAlt | Altitude above Mean Sea Level<br>Altitude in decimal format. Unit of measure m<br>When transmitted as string, use 1 decimal digits, at least | N |
| RPSAG | RPS Air/Ground mode<br>0: On Ground<br>1: Airborne | N |

Example:

```
application:
{
"time": "17:22:26.711",
"TOLat": 42.123451,
"TOLong": 11.123451,
"TOAlt": 29,
"RPSLat": 42.123451,
"RPSLong": 11.123451,
"RPSAlt": 29,
"RPSAG": 0
}
```

### 4.1.7 Geo Fencing Section layout

| Element | Description | Mandatory |
|---|---|---|
| Date | Date of last update<br>Format dd-mm-yyyy | N |
| time | Time of last geo fencing database update<br>Time of Day in UTC | N |

Example:

```
gfencing:
{
"Date": "01-12-2019",
"time": "17:22:26.000"
}
```

### 4.1.8 Augmentation availability layout

| Element | Description | Mandatory |
|---|---|---|
| POSAUGMode | GNSS Service Positioning Augmentation Mode<br>When greater than 0, positioning fields depicted in "State Data" section are output by the augmentation system<br>0: Service Unavailable<br>1: EGNOS with no integrity available<br>2: EGNOS<br>3: RTK<br>4: RTK+Integrity<br>5: PPP<br>6: PPP+Integrity<br>7: PPP+RTK<br>8: PPP+RTK+Integrity | N |
| HPL | Estimated Horizontal Protection Level Integrity field<br>HPL in decimal format. Unit of measure meters | N |
| VPL | Estimated Vertical Protection Level Integrity field<br>VPL in decimal format. Unit of measure meters | N |
| NavStatusValidity | Positioning fields status depicted in "Status Layout" section validated by augmentation system<br>0: Positioning data fields validated by drone itself<br>1: Positioning data fields validated by augmentation system | N |
| ASFlag | Antispoofing Flag<br>0: Spoofing on GNSS signals not detected<br>1: Spoofing on GNSS signals detected | N |

Example:

```
"augmentation":
{
```

```
"POSAUGMode": 1,
"HPL": 7.8,
"VPL": 12.5,
"NavStatusValidity”: 1,
"ASFlag”: 0
}
```

### 4.1.9 Raw Data layout

| Element | | Definition | Mandatory |
|---|---|---|---|
| GPSTOW | | This is the GPS Time of the measurements and shall be provided in milliseconds from the beginning of the GPS week, which begins at midnight GMT on Saturday night/Sunday morning, measured in GPS time (as opposed to UTC). See RTCM DF004 for further details. | N |
| GNSSData | | | N |
| Array containing data for Tracked Satellites | SatelliteId | Constellation Identifier Char followed by constellation PRN<br>‘G’- for GPS<br>‘E’- for Galileo<br>‘R’- for Glonass | N |
| | PseudoRange | This field has been derived from the RTCM GPS DF011 field to support all GNSS measurements. In particular this field provides the full raw L1 pseudorange measurement in meters, on L1 C/A in case of GPS; on E1 B/C in case of Galileo; on L1OF in case of Glonass | N |
| | CNR | This field represents an estimate of the carrier-to noise ratio of the satellite's signal in dB-Hz. A value of “0” means that the CNR measurement is not computed, or not available | N |

Example:

```
"rawdata":
{
"GPSTOW": 220456100,
"GNSSData": [
       { "SatelliteId": ‘E12’, "PseudoRange": 25678321.345, "CNR": 45.3 },
       { "SatelliteId": ‘E27’, "PseudoRange": 26678321.345, "CNR": 41.3 },
       { "SatelliteId": ‘E23’, "PseudoRange": 24678321.345, "CNR": 42.3 },
       { "SatelliteId": ‘E21’, "PseudoRange": 23678321.345, "CNR": 43.3 },
       { "SatelliteId": ‘E16’, "PseudoRange": 22678321.345, "CNR": 44.3 },
       { "SatelliteId": ‘G13’, "PseudoRange": 23678321.345, "CNR": 40.3 },
       { "SatelliteId": ‘G23’, "PseudoRange": 22678321.345, "CNR": 35.3 },
       { "SatelliteId": ‘G24’, "PseudoRange": 21678321.345, "CNR": 36.3 },
       { "SatelliteId": ‘G25’, "PseudoRange": 20678321.345, "CNR": 37.3 },
       { "SatelliteId": ‘R25’, "PseudoRange": 21008321.345, "CNR": 38.3 }
]
}
```

### 4.1.10 Tracking Message Example

A simple Message Example containing minimal sections for a valid tracking message sending procedure:

```
{
   "identification":{
      "UAId":"1234567890AB",
      "src":1,
      "dev":1
   },
   "statedata":{
      "time":"17:22:26.711",
      "lat":42.123451,
      "lon":11.123451,
      "height":29,
   },
   "application":{
      "time": "17:22:26.711",
      "TOLat": 42.123451,
      "TOLong": 11.123451,
      "TOAlt": 29,
      "RPSLat": 42.123451,
      "RPSLong": 11.123451,
      "RPSAlt": 29,
      "RPSAG": 0
      }
}
```

An example of the whole message including advanced telemetry follows.

```
{
   "identification":{
      "UAId":"1234567890AB",
      "OpId":"Operator2020",
      "src":1,
      "dev":1
   },
   "statedata":{
      "time":"17:22:26.711",
      "lat":42.123451,
      "lon":11.123451,
      "height":29,
      "altitudeMSL":42.781,
      "speedNS":67.33,
      "speedEW":12.35,
      "VRate":4.25
   },
   "status":{
      "AGMode":1,
      "Payload":3,
      "Priority":7,
      "UAHealth":1,
      "RPSHealth":1,
      "LinkStatus":1,
      "FCSStatus":1,
      "EngStatus":1,
      "PwrStatus":1,
      "CDAAStatus":1,
      "NDAAStatus":0,
      "TrjStatus":0,
      "TimeValidity":1,
      "IDValidity":1,
      "PosValidity":1,
      "AltValidity":1,
      "GndSValidity":1,
      "VRateValidity":1,
      "IntValidity":1,
      "GFValidity":1,
      "PosFOM":12,
      "AltFOM":12,
      "GndSFOM":3,
      "VRateFOM":3
   },
   "intent":{
      "time":"17:22:26.711",
      "lat":42.123451,
      "lon":11.123451,
      "height":29,
      "altitudeMSL":42.781,
      "dataOrigin":1
   },
   "application":{
      "time":"17:22:26.711",
      "TOLat":42.123451,
      "TOLong":11.123451,
      "TOAlt":29,
      "RPSLat":42.123451,
      "RPSLong":11.123451,
      "RPSAlt":29,
      "RPSAG":0
   },
   "gfencing":{
     "Date": "01-12-2019",
     "time": "17:22:26.000"
   }
   "rawdata":
   {
      "GPSTOW": 220456100,
      "GNSSData": [
        { "SatelliteId": ‘E12’, "PseudoRange": 25678321.345, "CNR": 45.3 },
        { "SatelliteId": ‘E27’, "PseudoRange": 26678321.345, "CNR": 41.3 },
        { "SatelliteId": ‘E23’, "PseudoRange": 24678321.345, "CNR": 42.3 },
        { "SatelliteId": ‘E21’, "PseudoRange": 23678321.345, "CNR": 43.3 },
        { "SatelliteId": ‘E16’, "PseudoRange": 22678321.345, "CNR": 44.3 },
        { "SatelliteId": ‘G13’, "PseudoRange": 23678321.345, "CNR": 40.3 },
        { "SatelliteId": ‘G23’, "PseudoRange": 22678321.345, "CNR": 35.3 },
        { "SatelliteId": ‘G24’, "PseudoRange": 21678321.345, "CNR": 36.3 },
        { "SatelliteId": ‘G25’, "PseudoRange": 20678321.345, "CNR": 37.3 },
        { "SatelliteId": ‘R25’, "PseudoRange": 21008321.345, "CNR": 38.3 }
      ]
   }
}
```

### 4.1.11 Size of messages

Following the above specification, the Json message can have an approximate size of 1.4-2.2k (depending whether the advanced telemetry information is part of the message).

### 4.1.12 JSON/ZIP format

Compressed data (zip/tgz formats accepted) is supported when sending messages on specific interfaces (check section 3 for more details). Compression rates show to be around 55-66%, resulting in a file size of the message around 600-750 Bytes.

## *4.2 Tracking Message Specification (ASTERIX)*

PLEASE NOTE: AT THE TIME OF WRITING THIS DOCUMENT, CURRENT AVAILABLE SPECIFICATION FOR ASTERIX FORMAT IS NOT MATURE ENOUGH TO PROPERLY TRANSMIT ALL NEEDED FIELDS. FOR THIS REASON, FOR THE TIME BEING, ONLY JSON FORMAT WILL BE

ACCEPTED BY D-FLIGHT SYSTEM, LEAVING THE EXTENSION TO ASTERIX FORMAT FOR FUTURE RELEASES.

ASTERIX message format is alternative to JSON and it is used to reduce bandwidth when sending tracking messages.

The category and part identified for best fitting the Tracking message specification is ASTERIX part 29 category 129 (https://www.eurocontrol.int/publication/cat129-eurocontrol-specification-surveillance-data-exchange-asterix-part-29-category)

Following table displays currently available fields in UAP for UAS Identification and Target Reports

| FRN | Data Item | Information | Length |
|---|---|---|---|
| 1 | I129/010 | Data Source Identification | 2 |
| 2 | I129/015 | Data Destination Identification | 2 |
| 3 | I129/020 | UAS Manufacturer Identifier | 3 |
| 4 | I129/030 | UAS Model Identifier | 3 |
| 5 | I129/040 | UAS Serial Number | 12 |
| 6 | I129/050 | UAS Office Registration Country | 2 |
| 7 | I129/070 | Time of Day | 3 |
| FX | - | Field Extension Indicator | - |
| 8 | I129/080 | Position in WGS-84 Coordinates | 8 |
| 9 | I129/090 | Altitude above Mean Sea Level | 3 |
| 10 | I129/100 | Altitude above Ground Level | 3 |
| 11 | I129/110 | GNSS Signal Accuracy | 2 |
| 12 | I129/120 | Operational Risk Levels | 1 |
| 13 | SP | Special Purpose Field | 1+ |
| 14 | I129/185 | Horizontal Velocity (Cartesian) | 5 |
| FX | - | Field Extension Indicator | - |
| 8 | I129/220 | Vertical Velocity | 3 |
| 9 | - | Reserved for Future Use | - |
| 10 | - | Reserved for Future Use | - |
| 11 | - | Reserved for Future Use | - |
| 12 | - | Reserved for Future Use | - |
| 13 | - | Reserved for Future Use | - |
| 14 | - | Reserved for Future Use | - |

| | | | |
|---|---|---|---|
| FX | - | Field Extension Indicator | - |

# 5 APPENDIX

## *5.1 Authentication Token*

Two (alternative) types of authentication method are available

- **OpenId connect**: based on Bearer authentication, it is meant for mobile and third party applications
- **Authentication code:** based on a unique identifier, it is meant for UTM-boxs and lightweight device authentication cases

Authenticaion type shall be specified in HTTP or STOMP Header as follows:

- Header name: "Authorization" (the name is case insensitive)
- Hear value: "<type> <value>" where <value> is the authentication info and <type> the authentication info type. Following authentication info types are supported:
  - "Bearer" (case insensitive) for bearer (OpenId connect) access token.
  - "Authentication-code" (case insensitive) for authentication code.

Sections 5.2 and 5.3 provide implementation details on the above mentioned available solutions.

## *5.2 OpenID connect authentication workflow*

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

This section briefly describes the steps needed to perform a successful authentication process.

### 5.2.1 Client Id and Client Secret retrieval

Before being able to perform any actual authentication, each client MUST obtain following parameters:

- client_id: an identifier of the type of application being used. It is used by D-Flight system to validate the client_secret value. Each drone manufacturer shall be assigned to –at least - a different client_id.
- client_secret: the unique password assigned to the client_id performing the call. Such value MUST not be accessible from end users: it shall be hidden inside the mobile application (or inside the firmware of the UTM-box)

Both parameters are static (once generated, they never change) and shall be requested offline by the drone manufacturers/third parties to D-Flight support, by sending an email to protocollogenerale@pec.d-flight.it, explaining and justifying third party needs.

Once such parameters are received, they will be embedded into the application and/or UTM-box provided to the end users.

### 5.2.2 Token Retrieval

#### *5.2.2.1 First Token Retrieval*

In order to establish a successfully authentication with D-Flight system, the client (mobile app or UTM-box) MUST perform a Rest (post) call to D-Flight authentication endpoint, providing following parameters:

- **username**: the username of a valid operator registered to D-Flight system
- **password**: the password corresponding to the username described above
- **cliend_id**: the identifier of the client id as described in previous section
- **client_secred**: the secret password as described in previous section

Token Retrieval input example (values in **bold** being placeholders):

client_id=**drone-company-name**&grant_type=password&username=**operator-dflight-username**&password=**operator-dflight-password**&scope=openid user-data pilot-license&client_secret=**drone-company-client-secret**

If the input parameters are successfully validated, the system replies with a json containing – at least – following parameters:

- **access_token**: the security authentication string to be used in following calls
- **expires_in**: validity of the access_token (in seconds)
- **refresh_token**: the security authentication string to be used for refreshing the access token

Token Retrieval output example:

```
{"access_token":"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJhcm0zR
zVnX0U1RjNZejJJQkRMZlh1WWtCeE9GUWlmYUt1VnJsOXNJRExzIn0.eyJqd
GkiOiI0MjQxMDUzOS0wYmQyLTRjMDMtYjIxZC1mYTUwMmVkNjI4ZTYiLCJleHAiOjE1Nzk2MDE
4NjQsIm5iZiI6MCwiaWF0IjoxNTc5NjAxNTY0LCJpc3MiOiJodHRwOi8vYXV0aC5
kZmxpZ2h0Lml0OjgwODAvYXV0aC9yZWFsbXMvREZsaWdodCIsImF1ZCI6IndlYi1hcHAiLCJzd
WIiOiJmOjk4NWU1NDE5LWJhMGMtNGRhMS1iMGRjLTA2OThmMzIyZGI2YjoxM2Y1O
DA4Ny1jZGQ5LTQxNGMtYjRlMC1hODdmYWI4N2E5N2YiLCJ0eXAiOiJCZWFyZXIiLCJhenAiOiJ
3ZWItYXBwIiwiYXV0aF90aW1lIjowLCJzZXNzaW9uX3N0YXRlIjoiMWNmYjA0NGE
tMzA5Yy00MTBkLTk3OTYtZjM2ODM3M2NkMDkzIiwiYWNyIjoiMSIsInJlYWxtX2FjY2VzcyI6e
yJyb2xlcyI6WyJTVVBFUlZJU09SIl19LCJzY29wZSI6Im9wZW5pZCBwcm9maWxlI
iwidXNlcl9uYW1lIjoiYW50b25lbGxvLmNlZHJvbmUiLCJuYW1lIjoiQW50b25lbGxvIENlZHJ
vbmUiLCJwcmVmZXJyZWRfdXNlcm5hbWUiOiJhbnRvbmVsbG8uY2Vkcm9uZSIsImd
pdmVuX25hbWUiOiJBbnRvbmVsbG8iLCJmYW1pbHlfbmFtZSI6IkNlZHJvbmUiLCJhdXRob3Jpd
GllcyI6WyJST0xFX1NVUEVSVklTT1IiXX0.SxJVsS6T15tixAz9KHBEfUJge55JA
LU5VktpInLiMEbILH7XeEhTGmQg0WVBDdzyfCESNuZS2jfDIMDSy9kKlHVVbQFaJVoi_ILdvUT
MFrZWSyP8uOunlSGKcBkZ6b6lcGL17vFwBEijHbb4JyqFok1dMGv9O-3XyNgjvdW
75xE2DNvNQXG5JzLpK9Xej8BYLrFWbugnvFT88fEpmf0iCTt9wv2PHF3yN2DsP0tBoVzZVpr6D
qNmwZiiJ5bUjDpcrbeo6MNziElSRcjrqyET9NyliSEC5ltuaOTpr_CPePlgUsrf3
EoFfTZbSGyenStIr5RL3obTzzgXFyA-z-m5jQ",
    "expires_in": 300,
    "refresh_expires_in": 1800,
    "refresh_token":
"eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICIyNGNiMDFjNC04ODQyLTQ2M
zQtOTJiMC1iMzRiM2I1ZTM0ZjYifQ.eyJqdGkiOiI0N
jJkOTY3MS1hYWRiLTQwZjMtOGRkOS0wZTk2MTE5NWMyNDIiLCJleHAiOjE1Nzk2MDMzNjQsIm5
iZiI6MCwiaWF0IjoxNTc5NjAxNTY0LCJpc3MiOiJodHRwOi8vYXV0aC5kZmxpZ2h
0Lml0OjgwODAvYXV0aC9yZWFsbXMvREZsaWdodCIsImF1ZCI6Imh0dHA6Ly9hdXRoLmRmbGlna
HQuaXQ6ODA4MC9hdXRoL3JlYWxtcy9ERmxpZ2h0Iiwic3ViIjoiZjo5ODVlNTQxO
S1iYTBjLTRkYTEtYjBkYy0wNjk4ZjMyMmRiNmI6MTNmNTgwODctY2RkOS00MTRjLWI0ZTAtYTg
3ZmFiODdhOTdmIiwidHlwIjoiUmVmcmVzaCIsImF6cCI6IndlYi1hcHAiLCJhdXR
oX3RpbWUiOjAsInNlc3Npb25fc3RhdGUiOiIxY2ZiMDQ0YS0zMDljLTQxMGQtOTc5Ni1mMzY4M
zczY2QwOTMiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsiU1VQRVJWSVNPUiJdf
Swic2NvcGUiOiJvcGVuaWQgcHJvZmlsZSJ9.bKRHBlTapHEUXatjz2ax6-
umqLOWPaoVn5xg6RDKRBE",
    "token_type": "bearer",
    "id_token":
"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJhcm0zRzVnX0U1RjNZejJJQ
kRMZlh1WWtCeE9GUWlmYUt1VnJsOXNJRExzIn0.eyJqdGkiO
iJjZDYxMjU4OC00ODg5LTQxNzItOWM5My0wZDcwMjA4MTg2YWMiLCJleHAiOjE1Nzk2MDE4NjQ
sIm5iZiI6MCwiaWF0IjoxNTc5NjAxNTY0LCJpc3MiOiJodHRwOi8vYXV0aC5kZmx
pZ2h0Lml0OjgwODAvYXV0aC9yZWFsbXMvREZsaWdodCIsImF1ZCI6IndlYi1hcHAiLCJzdWIiO
iJmOjk4NWU1NDE5LWJhMGMtNGRhMS1iMGRjLTA2OThmMzIyZGI2YjoxM2Y1ODA4N
y1jZGQ5LTQxNGMtYjRlMC1hODdmYWI4N2E5N2YiLCJ0eXAiOiJJRCIsImF6cCI6IndlYi1hcHA
iLCJhdXRoX3RpbWUiOjAsInNlc3Npb25fc3RhdGUiOiIxY2ZiMDQ0YS0zMDljLTQ
xMGQtOTc5Ni1mMzY4MzczY2QwOTMiLCJhY3IiOiIxIiwicmVhbG1fYWNjZXNzIjp7InJvbGVzI
jpbIlNVUEVSVklTT1IiXX0sIm5hbWUiOiJBbnRvbmVsbG8gQ2Vkcm9uZSIsInByZ
WZlcnJlZF91c2VybmFtZSI6ImFudG9uZWxsby5jZWRyb25lIiwiZ2l2ZW5fbmFtZSI6IkFudG9
uZWxsbyIsImZhbWlseV9uYW1lIjoiQ2Vkcm9uZSJ9.S7FrYDS5N_1Ta-nFvX8pue
EYJZckcpCtDm8vaezvbd4i6U10hCgiMtCtUmrNdHhn91NuUQeSz_nvPlhP-
gpScX02HCqsWWWTOdkuP6DNh6qzQGisbGKAMiDITVh7vru0aQAvYAyZ34Bv6rQE0q-p1R_I4-
yWTu9k
MWKXUDkCgCDtzWG9xqua1lR9ILm7GdUs22-R1RVywVgcuDsBIIPI9lONU8T2Wx9akw74SDdj-
YYhipWDLR4vKdF8cndB5Ea3Xim7E06qX4roV2IRdalPtMl-Gl0iVLOYMafrYtEkOh
T0m8oOPTnC7J5cQURRgWztN8f5hIBza7zG3nvRBgIcpg",
    "not-before-policy": 0,
    "session_state": "1cfb044a-309c-410d-9796-f368373cd093",
    "scope": "openid profile"
}
```

### 5.2.2.2 *Next Token Retrievals*

The same token can be used multiple times until it expires (see expires_in parameter in previous section). Once expired, a new valid token can be retrieved by sending a token retrieval request with following parameter:

- refresh_token: the value of the refresh token retrieved during first token retrieval procedure

Token Refresh input example:

```
"refresh_token=eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICIyNGNiMDF
jNC04ODQyLTQ2MzQtOTJiMC1iMzRiM2I1ZTM0ZjYifQ.eyJqdGkiOiI0N
jJkOTY3MS1hYWRiLTQwZjMtOGRkOS0wZTk2MTE5NWMyNDIiLCJleHAiOjE1Nzk2MDMzNjQsIm5
iZiI6MCwiaWF0IjoxNTc5NjAxNTY0LCJpc3MiOiJodHRwOi8vYXV0aC5kZmxpZ2h
0Lml0OjgwODAvYXV0aC9yZWFsbXMvREZsaWdodCIsImF1ZCI6Imh0dHA6Ly9hdXRoLmRmbGlna
HQuaXQ6ODA4MC9hdXRoL3JlYWxtcy9ERmxpZ2h0Iiwic3ViIjoiZjo5ODVlNTQxO
S1iYTBjLTRkYTEtYjBkYy0wNjk4ZjMyMmRiNmI6MTNmNTgwODctY2RkOS00MTRjLWI0ZTAtYTg
3ZmFiODdhOTdmIiwidHlwIjoiUmVmcmVzaCIsImF6cCI6IndlYi1hcHAiLCJhdXR
oX3RpbWUiOjAsInNlc3Npb25fc3RhdGUiOiIxY2ZiMDQ0YS0zMDljLTQxMGQtOTc5Ni1mMzY4M
zczY2QwOTMiLCJyZWFsbV9hY2Nlc3MiOnsicm9sZXMiOlsiU1VQRVJWSVNPUiJdf
Swic2NvcGUiOiJvcGVuaWQgcHJvZmlsZSJ9.bKRHBlTapHEUXatjz2ax6-
umqLOWPaoVn5xg6RDKRBE"
```

The output of the refresh token call, contains the same information (a new access_token and a new refresh_token) as the first request token.

### 5.2.3 Service Access

All further requests and access to D-Flight services shall specify a valid access_token in order to be validated and processed. Such token must be specified in the HTTP header of the request with following syntax:

```
Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJhcm0zRzVnX0U1RjNZejJJQk
RMZlh1WWtCeE9GUWlmYUt1VnJsOXNJRExzIn0.eyJqdGkiOiI0MjQxMDUzOS0wYmQyLTRjMDMt
YjIxZC1mYTUwMmVkNjI4ZTYiLCJleHAiOjE1Nzk2MDE4NjQsIm5iZiI6MCwiaWF0IjoxNTc5Nj
AxNTY0LCJpc3MiOiJodHRwOi8vYXV0aC5
kZmxpZ2h0Lml0OjgwODAvYXV0aC9yZWFsbXMvREZsaWdodCIsImF1ZCI6IndlYi1hcHAiLCJzd
WIiOiJmOjk4NWU1NDE5LWJhMGMtNGRhMS1iMGRjLTA2OThmMzIyZGI2YjoxM2Y1ODA4Ny1jZGQ
5LTQxNGMtYjRlMC1hODdmYWI4N2E5N2YiLCJ0eXAiOiJCZWFyZXIiLCJhenAiOiJ3ZWItYXBwI
iwiYXV0aF90aW1lIjowLCJzZXNzaW9uX3N0YXRlIjoiMWNmYjA0NGE
tMzA5Yy00MTBkLTk3OTYtZjM2ODM3M2NkMDkzIiwiYWNyIjoiMSIsInJlYWxtX2FjY2VzcyI6e
yJyb2xlcyI6WyJTVVBFUlZJU09SIl19LCJzY29wZSI6Im9wZW5pZCBwcm9maWxlIiwidXNlcl9
uYW1lIjoiYW50b25lbGxvLmNlZHJvbmUiLCJuYW1lIjoiQW50b25lbGxvIENlZHJvbmUiLCJwc
mVmZXJyZWRfdXNlcm5hbWUiOiJhbnRvbmVsbG8uY2Vkcm9uZSIsImd
pdmVuX25hbWUiOiJBbnRvbmVsbG8iLCJmYW1pbHlfbmFtZSI6IkNlZHJvbmUiLCJhdXRob3Jpd
GllcyI6WyJST0xFX1NVUEVSVklTT1IiXX0.SxJVsS6T15tixAz9KHBEfUJge55JALU5VktpInL
iMEbILH7XeEhTGmQg0WVBDdzyfCESNuZS2jfDIMDSy9kKlHVVbQFaJVoi_ILdvUTMFrZWSyP8u
OunlSGKcBkZ6b6lcGL17vFwBEijHbb4JyqFok1dMGv9O-
3XyNgjvdW75xE2DNvNQXG5JzLpK9Xej8BYLrFWbugnvFT88fEpmf0iCTt9wv2PHF3yN2DsP0tB
oVzZVpr6DqNmwZiiJ5bUjDpcrbeo6MNziElSRcjrqyET9NyliSEC5ltuaOTpr_CPePlgUsrf3E
oFfTZbSGyenStIr5RL3obTzzgXFyA-z-m5jQ"
eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUI
```

For more details refer to the official website: https://openid.net/connect/

## *5.3 Authentication Code workflow*

This functionality will be supported starting from version 2.0 of this document.

## *5.4 Device Type*

Device type shall be specified in HTTP or STOMP Header as follows:

- ○ Header name: “x-device-type” (the name is case insensitive).
- ○ Hear value: “<positive integer>”: refer to below  table for available values

| DEVICE TYPE | DESCRIPTION | VALUE |
|---|---|---|
| *U-Box On board* | UTM-box phisically located inside/within the Drone | 0 |
| *U-Box c/o GCS* | UTM-box phisically located into the GCS, receiving actual tracking information from the Drone | 1 |
| *Virtual U-Box* | Third party (es Mobile app) sending pilot position | 2 |

### *5.5 Connection Details and Examples*

This section provides technical details on how to set-up and establish a secure authentication connection with D-Flight, to be able to eventually send tracking information to the system.
Please note: there are several and multi-language opensource API already available. This section provides example for python language.

### 5.5.1 Token Retrieval

Token retrieval shall be performed using an OpenId client/library to access D-Flight IAM Identity and Access Manager).

OpenId Client Configuration:

- Server Url: provided offline by D-Flight (see section 5.2.1 for credential request procedure)
- Realm Name: DFlight
- Client Id: provided offline by D-Flight (see section 5.2.1 for credential request procedure)
- Client Secret Key: provided offline by D-Flight (see section 5.2.1 for credential request procedure)

Token Retrieval Configuration:

- User: a valid username (or email) registered to D-Flight portal (https://www.d-flight.it/web-app/)
- Password: password related to the above mentioned user

Snippet:

```
# OpenId Connection
provider_openid = MyProviderForOpenID(
        server_url='my-dflight-authentication-url',
        client_id='my-client-id',
        client_secret_key='my-client-secret-key',
        realm_name='DFlight',
        verify=False)
# OpenId Token Retrieval
token = provider_openid.token('my-dflight-username', 'my-dflight-password')
```

### 5.5.2 Stomper Connect: Session Id Retrieval

Once a Token is successfully retrieved as described during previous section, it can be used to obtain a valid session id. Session Id shall be retrieved by sending a stomper request via websocket.

Reference libraries:

- Stomper: [https://pypi.org/project/stomper/](https://pypi.org/project/stomper/)
- Websocket: [https://pypi.org/project/websocket_client/](https://pypi.org/project/websocket_client/)

WebSocket Configuration:

- Websocket url: provided offline by D-Flight (see section 5.2.1 for credential request procedure)

Stomper Configuration:

- Command: 'CONNECT'
- Headers:
  - Authorization: 'Bearer ' + Access Token
  - Version: 1.1

Snippet:

```
# Websocket Creation
ws = create_connection('my-dflight-messaging-url',
      sslopt={"cert_reqs": ssl.CERT_NONE})

# Stomper Creation
msg = stomper.Frame()
msg.cmd = 'CONNECT'
msg.headers = {'Authorization': 'Bearer ' + tk["access_token"],
      "accept-version": "1.1",
      'x-device-type': 0}

# Sending Connection Stomper via Websocket
ws.send(msg.pack())
# Receive Stomper and reading Session id
d = ws.recv()
# parse for 'session:' inside the received message
```

### 5.5.3 Stomper Send: Tracking Message

Once a session Id is successfully retrieved as described during previous section, it can be used to send Tracking Messages to D-Flight endpoint via web socket.

Reference libraries:

- Stomper: [https://pypi.org/project/stomper/](https://pypi.org/project/stomper/)
- Websocket: [https://pypi.org/project/websocket_client/](https://pypi.org/project/websocket_client/)

WebSocket Configuration:

- Websocket url: provided offline by D-Flight (see section 5.2.1 for credential request procedure)

Stomper Configuration:

- Command: 'SEND'
- Headers:

- session: session Id retrieved during previous step
- originTime: current datetime
- destination: /exchange/input_position_reports
- Body: the json representation of the tracking position according to this ICD (see examples on section 4.1.10)

Snippet:

```
# Stomper Creation
dt = int(round(time.time() * 1000))
msg = stomper.Frame()
msg.cmd = 'SEND'
msg.headers = {
    "session": sessionid,
    "originTime": dt,
    "destination": "/exchange/input_position_reports"}

msg.body = json.loads(my_json_tracking_message)

# Sending Tracking Stomper via Websocket
ws.send(msg.pack())
```

### 5.5.4 Test Platform

A test platform (D-Flight pre-production platform) is available for performing integration tests before going operational.

With regard to endpoint and configuration detailed in previous section, Pre-Production platform can be reached by specific configuration connection parameters provided offline as detailed below.

Please note that, in order to access the pre-production web platform (therefore to be able to see the positions on the map), a personal certificate shall be created and used when connecting via browser/mobile.

In order to receive connection parameters and to grant access to the platform, please send an email to protocollogenerale@pec.d-flight.it, listing all people that shall have access to the system.