Ruter#

OTA Messages

18 October, 2018

Introduction

This page is an update of an outdated OTA Messages wiki page that was based on version 1.0 of the requirements. Version 1.1 of the requirements introduced some changes and after the development process began, new needs were identified as documented here, partly for use in formulating a change order for the operators selected for routes in Romerike and the electric bus contract.

Ruter has agreed with the operators to use only JSON in MQTT messages and not support Protobuf. The contents of the messages will therefore be defined by JSON schemas and these will be made available to the operators.

Summary

The summary refers to topics on board the buses. See below for mapping of the local topic to the bridged topic.

Changes noted in status are based on a comparison with v. 1.1 of the OTA Messages document. Direction is in/out of the bus.

Local topic	Change	Direction	Name	Comments
signon/json		Out	Start block	
signoff/json		Out	Complete block	
avl/json		Out	Vehicle position	
apc/+/json		Out	Passenger count	"+" is the placeholder for the door number
stopsignal/json	New	Out	Stop signal status	This signal does not exist in FMS and must be read directly from the electrical signal system on board
telemetry/+/json	New	Out	Vehicle telemetry	"+" is the placeholder for an ID, such as PGN which is defined in the FMS standard; We specify which PGN or other telemetry data should be sent. **Used to be fmstoip but has now been generalized.**
infohub/dpi/diagnostics/json	New	Out	Diagnostics for screens	DPI diagnostics data to make sure the displays are set up correctly and are running as expected
tsp/json	New	In	Signal prioritization	Message to be sent directly from the bus to facilitate signal priority at traffic lights
madt/notification/json		In	Message to driver	Unclear who is producing and consuming this message

Local topic	Change	Direction	Name	Comments
infohub/dpi/journey/json	Modified	In	Vehicle journey	Coordinates for the stop are now included; stopPointRef was changed to stopPlaceId in v. 1.1
infohub/dpi/nextstop/json		In	Next stop	stopPointRef was changed to stopPlaceId in v. 1.1
infohub/dpi/eta/json	Modified	In	Estimated arrivals	The text for the time to be displayed on the screen is now included; stopPointRef was changed to stopPlaceId in v. 1.1
infohub/dpi/externaldisplay/json		In	Information on sign box	
infohub/dpi/arriving/json	Modified	In	Arriving	Public announcement of the stop; adds expiryTimestamp and zoneId field; createTimestamp was deleted in v. 1.1
infohub/dpi/deviation/json	Modified	In	Deviation	Multi-lingual, textual deviation messages. With references to stops/journal/lines etc.
infohub/dpi/announcement/json	Modified	In	Other announcement	Multi-lingual textual messages
infohub/dpi/audio/json	New	In	Audio message	Audio messages to be played on the bus. Can contain an array of messages with different target speakers and codecs.
infohub/dpi/c2/json	New	In	DPI command and control messages	Command and control messages to be used by DPI, from Ruter backend.
infohub/dpi/connections/json	Planned	In	Information about connections	Real-time data for connections before arrival at the stop
infohub/dpi/digitalsignage/json	Planned	In	Multimedia control	Message that controls the multimedia surfaces on board

Topics

signon/json

Name	Start block
Description	Notify PTA BO that the bus is starting a block
Local topic	signon/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/signon/json
Schema	signon.json

JSON example

```
{
  "eventTimestamp": "2017-10-31T12:45:50Z",
  "vehicleNumber": "12345",
  "blockId": "1234:34",
  "vehicleJourneyId": "35:ABC"
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
vehicleNumber	integer	same as for the bridged topic
blockId	string	A series of journeys
vehicleJourneyId	string	The actual journey started

Vehicle journey is required and helps track cases where blocks are interrupted and replacement vehicles take over.

signoff/json

Name	Complete block
Description	Notify PTA BO that the bus has completed a block
Local topic	signoff/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/signoff/json
Schema	signon.json

The content of signoff is the same as signon and uses the same schema.

avl/json

Name	Vehcile position
Description	Reporting of the bus's position, course and speed to PTA BO
Local topic	avl/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/avl/json
Schema	avl.json

JSON example

```
{
  "eventTimestamp": "2017-10-31T12:45:50Z",
  "seqNumber": 0,
  "latitude": 60.25255,
  "longitude": 11.0567,
  "heading": 0.5,
  "speedOverGround": 34.5,
  "signalQuality": "AGPS_QUALITY",
  "numberOfSatellites": 4,
  "gnssType": "GPS",
  "gnssCoordinateSystem": "WGS84",
  "deadReckoning": false,
  "positionIsSimulated": false
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
seqNumber	integer	used to ensure that the messages are processed in the correct order
latitude	float	
longitude	float	
heading	float	
speedOverGround	float	
signalQuality	string	enum SignalQuality
numberOfSatellites	integer	
gnssType	string	enum GNSSType
gnssCoordinateSystem	string	enum GNSSCoordinateSystem
deadReckoning	boolean	
positionIsSimulated	boolean	true when a block is simulated

Enum SignalQuality

Name	Description
AGPS_QUALITY	
DGPS_QUALITY	
ESTIMATED_QUALITY	
GPS_QUALITY	
NOT_VALID_QUALITY	
UNKNOWN_QUALITY	

Enum GNSSType

Name	Description
GPS	
GLONASS	
GALILEO	
BEIDOU	
IRNSS	
OTHER	
DEAD_RECKONING	
MIXED_GNSS_TYPES	

Enum GNSSCoordinateSystem

Name	Description
WGS84	
AGPS	
DGPS	
ESTIMATED	
GPS_COORDINATE_SYSTEM	

apc/+/json

Name	Passenger count
Description	Report of passenger count per door to PTA BO
Local topic	apc/<doorid>/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/apc/<doorid>/json
Schema	apc.json

JSON example

```
{
  "eventTimestamp": "2017-10-31T12:45:50Z",
  "doorId": 1,
  "passengerCounting": [{
    "objectClass": "ADULT",
```

```
    "doorPassengerIn": 1,
    "doorPassengerOut": 1
  }, {
    "objectClass": "CHILD",
    "doorPassengerIn": 0,
    "doorPassengerOut": 0
  }, {
    "objectClass": "PRAM",
    "doorPassengerIn": 0,
    "doorPassengerOut": 0
  }, {
    "objectClass": "WHEELCHAIR",
    "doorPassengerIn": 0,
    "doorPassengerOut": 0
  }],
  "doorCountQuality": "REGULAR"
}
```


Fields

APC

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
doorId	integer	Data is reported only per door
passengerCounting	array of PassengerCount	One PassengerCount per ObjectClass
doorCountQuality	string	enum CountQuality

PassengerCount

Name	Type	Description
objectClass	string	enum ObjectClass
doorPassengerIn	integer	
doorPassengerOut	integer	

Enum ObjectClass

Name	Description
ABSENT	
ADULT	
CHILD	
PRAM	
BIKE	
WHEELCHAIR	
OTHER	

Enum CountQuality

Name	Description
ABSENT	
REGULAR	
DEFECT	
OTHER	

stopsignal/json

Name	Stop signal status
Description	Stop signal status when changed by using stop button
Local topic	stopsignal/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/stopsignal/json
Schema	stopsignal.json

JSON example

```
{
  "eventTimestamp": "2017-10-31T12:45:50Z",
  "stopSignalled": true
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
stopSignalled	boolean	

telemetry/+/json

Name	Vehicle telemetry
Description	Vehicle telemetry from systems on the bus
Local topic	telemetry/<id>/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/telemetry/<id>/json
Schema	telemetry.json

Several different kinds of telemetry are available varying by vehicle type. For traditional busses, FMS is the standard defines what data about the vehicle is published on the FMS bus and out by FMStoIP. For trams VehicleToIP defines a a different and more limited set of data. In addition, electrical and hydrogen have proprietary data that is not captured by either.

This topic generalizes all such data as telemetry defined by a unique id. We will use FMS PGN ids (4-digit hex values, i.e.) but otherwise use unique ids to identify data from other sources.

If we define the ids along the lines of FMS we can have 32-bit ids that use the following pattern:

Bytes	Description
1	Source identifier
2-4	Source-specific id, e.g. FMS PGNs

Source identifiers will be:

Id	Description
0x00	FMS
0x01	non-FMS
...	new sources

Therefore all FMS PGNs become "0000" + 4-digit hex PGN.

FMS

Each of the available data points is defined with a Parameter Group Number (PGN) with fields defined by Suspect Parameter Number (SPN).

The FMS-to-IP service in the ITxPT standard makes the desired data available on the bus's own network over UDP (multicast broadcast). It sends out only the PGNs previously requested by calling a service. The messages are sent every second, according to ITxPT, and includes all the requested PGNs. The format is in XML.

We therefore define which PGNs are needed by Ruter and that the XML message should be broken up per PGN and forwarded as JSON.

FMS Parameter Group Number (PGN)

This is the list of required PGNs. It should be possible to expand this over time.

Kode	Description	PGN	SPN	Formål	ID	Local topic
DC1	Door Control 1	FE4E	3411 Status 2 of doors 1820 Ramp/Wheel chairlift 1821 Position of doors	Doors open / closed	0000FE4E	telemetry/0000fe4e/json
DC2	Door Control 2	FDA5	XXXX Lock Status Door N XXXX Enable Status Door N XXXX Open Status Door N	Alternative to DC1?	0000FDA5	telemetry/0000fda5/json
VDHR	High Resolution Vehicle Distance	FEC1	917 High resolution total vehicle distance	Supplements position	0000FEC1	telemetry/0000fec1/json

ITxPT

These PGNs must be subscribed to when the buses are started (unless the subscription is persistent) through the AddPGN call:

```
<PGNReq>FE4E</PGNReq>
<PGNReq>FDA5</PGNReq>
<PGNReq>FEC1</PGNReq>
```

A service must be implemented by the operator's vendor that listens the UDP messages and converts them to MQTT messages.

XML format (source)

The message for all subscribed PGNs is sent to UDP as XML.

```
<FMStoIPDelivery>
    <FMStoIP FMSVersion="03">
        <frame Status="OK">
            <PGN>FEF1</PGN>
            <data>F35237C403501FFF</data>
            <relativetime>5000</relativetime>
            <SPN id="84">
                <name>Wheel-Based Vehicle Speed</name>
                <unit>km/h</unit>
                <value>55.3</value>
            </SPN>
            <SPN id="597">
                <name>Brake Switch</name>
                <value>released</value>
            </SPN>
        </frame>
        <frame Status="OK">
            <PGN>F004</PGN>
            <data>F35237C403501FFF</data>
            <relativetime>5000</relativetime>
            <SPN id="190">
                <name>Engine Speed</name>
                <unit>rpm</unit>
                <value>854</value>
            </SPN>
        </frame>
    </FMStoIP>
</FMStoIPDelivery>
```

PGN FEF1 og F004 shown here are the examples found in the ITxPT spec.

JSON format (MQTT)

The message containing several PGNs is split up into several MQTT messages. SPNs are mapped as subids.

telemetry/0000fef1/json

```
{
    "eventTimestamp": "2017-10-31T12:45:50Z",
    "id": "0000FEF1",
    "payloads": [
        {
            "subid": 84,
            "name": "Wheel-Based Vehicle Speed",
            "unit": "km/h",
            "value": 55.3
        },
        {
            "subid": 597,
            "name": "Wheel-Based Vehicle Speed",
            "value": "released"
        }
    ]
}
```

telemetry/0000f004/json

```
{
    "eventTimestamp": "2017-10-31T12:45:50Z",
    "id": "0000F004",
    "payloads": [
        {
            "subid": 190,
            "name": "Engine Speed",
            "unit": "rpm",
            "value": 854
        }
    ]
}
```

Fields

Telemetry

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
id	string	eight-digit hex value of the telemetry provided
payloads	array of Payload	one or more payloads

Payload

Name	Type	Description
subid	int	subid such as SPN, if appropriate (optional)
name	string	optional
unit	string	
value	any	

tsp/json

Name	Signal prioritization
Description	The message to be sent to VHF to ensure that the bus is prioritized at the traffic lights
Local topic	tsp/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/tsp/json
Schema	tsp.json

This message is generated by Ruter when approaching an intersection or, when a stop is just before an intersection, after the doors have closed.

JSON example

```
{
    "eventTimestamp": "2017-10-31T08:38:02.749Z",
    "encodedMessage": ""
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
encodedMessage	string	String of characters that represent the message to transmit

madt/notification/json

Name	Message to driver
Description	Messages directly to the bus driver
Local topic	madt/notification/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/madt/notification/json
Schema	notification.json

JSON example

```
{
    "eventTimestamp": "2017-10-31T08:38:02.749Z",
    "urgency": "MEDIUM",
    "subject": "Lorem ipsum",
    "content": "Duis aute irure dolor"
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
urgency	string	enum Urgency
subject	string	
content	string	

Enum Urgency

Name	Description
LOW	
MEDIUM	
HIGH	

infohub/dpi/journey/json

Name	Vehicle journey
Description	The stops included in the bus route, with connections to other lines
Local topic	infohub/dpi/journey/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/journey/json
Schema	journey.json

The coordinates of the stop have been added to facilitate backup calculations for stop announcements and possibly other messages.

JSON example


```
{
  "eventTimestamp": "2017-10-31T08:38:02.749Z",
  "route": {
    "id": "RUT:Route:31-1041",
    "name": "Fornebu vest-Tonsenhagen",
    "line": {
      "id": "RUT:Line:31",
      "name": "Snarøya - Fornebu - Tonsenhagen - Grorud",
      "publicCode": "31"
    },
    "stopPlaces": [
      {
        "id": "RUT:StopPlace:02190017",
        "name": "Fornebu vest",
        "connections": [],
        "location": {
            "latitude": 12.33345,
            "longitude": 12.33345
        }
      },
      {
        "id": "RUT:StopPlace:03010013",
        "name": "Jernbanetorget",
        "connections": [
          {
            "line": {
              "id": "RUT:Line:30",
              "name": "Bygdøy via Bygdøynes",
              "publicCode": "30"
            },
            "type": "BUS",
            "color": "e60000"
          },
          {
            "line": {
              "id": "RUT:Line:12",
              "name": "Majorstuen",
              "publicCode": "12"
            },
            "type": "TRAM",
            "color": "0b91ef"
          }
        ],
        "location": {
            "latitude": 12.33345,
            "longitude": 12.33345
        }
      }
    ]
  }
}
```


Fields

Journey

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
route	Route	

Route

Name	Type	Description
id	string	NSR code
name	string	
line	Line	
stopPlaces	array of StopPlace	

Line

Name	Type	Description
id	string	NSR code
name	string	
publicCode	string	

StopPlace

Name	Type	Description
id	string	NSR code
name	string	
connections	array of Connection	
location	Location	

Connection

Name	Type	Description
line	Line	
type	string	enum TransportType
colour	string	RGB code in hex; originally spelled color, but British spelling is preferred in European standards

Location

Name	Type	Description
longitude	float	
latitude	float	

Enum TransportType

Name	Description
BUS	This was originally listed as BUSS
TRAM	
OTHER	

infohub/dpi/nextstop/json

Name	Next stop
Description	Next stop on the bus's route after leaving a stop
Local topic	infohub/dpi/nextstop/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/nextstop/json
Schema	nextstop.json

StopPointRef was replaced by StopPlaceId in v. 1.1.

JSON example

```
{
    "eventTimestamp": "2017-10-31T08:38:02.749Z",
    "stopPlaceId": "RUT:StopPlace:03012453"
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
stopPlaceId	string	NSR StopPlace code

infohub/dpi/eta/json

Name	ETA
Description	Estimated arrival at the remaining stops
Local topic	infohub/dpi/eta/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/eta/json
Schema	eta.json

StopPointRef was replaced by StopPlaceId in v. 1.1. The field "text" was added to send the desired display text for the arrival time.

JSON example

```
{
          "eventTimestamp": "2017-10-31T08:38:02.749Z",
          "estimatedCalls": [
                    {
                              "eta": "2017-10-13T12:27:04Z",
                              "stopPlaceId": "RUT:StopPlace:03010510",
                              "text": "Nå"
          },
                    {
                              "eta": "2017-10-13T12:27:04Z",
                              "stopPlaceId": "RUT:StopPlace:03010511",
                              "text": "5 min"
          }
          ]
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
estimatedCalls	array of EstimatedCall	

EstimatedCall

Name	Type	Description
eta	string	ISO 8601, UTC
stopPlaceId	string	NSR StopPlace code
text	string	display text for passengers

infohub/dpi/externaldisplay/json

Name	Information for sign boxes
Description	Message to display on signposts. Usually line number (publicCode) and routeName, with support for alternative message
Local topic	infohub/dpi/externaldisplay/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/externaldisplay/json
Schema	externaldisplay.json

Define what "alternativeMessage" can be and rules for displaying the field!

JSON example

```
{
  "eventTimestamp": "2017-10-31T08:38:02.749Z",
  "publicCode": "31",
  "destination": "Lorem ipsum",
  "alternativeMessage": "Duis aute irure dolor"
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
publicCode	string	Publicly known bus, tram or subway line number
destination	string	Usually the final stop
alternativeMessage	string	The use of this field must be clarified

infohub/dpi/arriving/json

Name	Arrival
Description	Notice to passengers that the bus is approaching a stop
Local topic	infohub/dpi/arriving/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/arriving/json
Schema	arriving.json

StopPointRef was replaced by StopPlaceId in v. 1.1. The field createTimestamp was removed in v. 1.1. Multi-lingual suoport added in v.1.2

The field expiryTimestamp has been added to to prevent delayed messages from being played after a certain amount of time.

JSON example

```
{
  "eventTimestamp": "2017-10-31T08:38:02.749Z",
  "expiryTimestamp": "2017-10-31T08:38:47.749Z",
  "ref": "RUT:StopPlace:03012453",
  "zoneId": "2b-vest",
  "message": {
    "no": {
      "title": "Ankommer"
      "text": "Oslo sentralstasjon"
    },
    "en": {
      "title": "Arriving at"
      "text": "Oslo Central Station"
    }
  }
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
expiryTimestamp	string	ISO 8601, UTC, audio should not be played after this time
ref	string	
zoneID	string	Id of payment zone.
message	dictionary of MultilingualMessage	

MultilingualMessage

Name	Type	Description
title	string	title to be display to passengers (optional)
text	string	text to be display to passengers

infohub/dpi/deviation/json

Name	Deviation
Description	Notice to passengers of a deviation
Local topic	infohub/dpi/deviation/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/deviation/json
Schema	deviation.json

The affects field has been added to specify the scope of the deviation. This should be an NSR ID (where the type is included in the ID) such as StopPlaceId, QuayId, LineId, RouteId, JourneyId, OperatorId or AuthorityId. It is possible additional identifiers may be included.

JSON example

```
{
    "eventTimestamp": "2017-10-31T08:38:02.749Z",
    "expiryTimestamp": "2017-10-31T08:38:47.749Z",
    "ref": [
        "NSR:StopPlace:2561",
                    "NSR:StopPlace:2562"
    ],
    "message": {
        "no": {
            "title": "afsdf",
            "text": "Lorem ipsum dolor sit amet"
        },
        "en": {
            "title": "afsdf",
            "text": "Lorem ipsum dolor sit amet"
        }
    }
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
expiryTimestamp	string	ISO 8601, UTC, do not show after
ref	string or array of strings	List of affected entities, if empty or not included the deviation is general
message	dictionary of MultilingualMessage	

Responsibility for the use of the ref field lies entirely with the DPI application. When it has matching ids it can tailor the information display accordingly. An example of this is stopPlaceId deviations.

MultilingualMessage

Name	Type	Description
title	string	title to be displayed to the passengers (optional)
text	string	text to be displayed to the passengers

infohub/dpi/announcement/json

Name	Announcement
Description	Announcement to the passengers (ad hoc)
Local topic	infohub/dpi/announcement/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/announcement/json
Schema	announcement.json

Message contains a reference to the scope of the message, if applicable. Typically NSR stopplaceid / lineId or similar.

JSON example

```
{
    "eventTimestamp": "2017-10-31T08:38:02.749Z",
    "expiryTimestamp": "2017-10-31T08:38:47.749Z",
          "type": "INFO",
    "ref": [
        "NSR:StopPlace:2561",
                    "NSR:StopPlace:2562"
    ],
    "message": {
        "no": {
            "title": "afsdf",
            "text": "Lorem ipsum dolor sit amet"
        },
        "en": {
            "title": "afsdf",
            "text": "Lorem ipsum dolor sit amet"
        }
    }
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
expiryTimestamp	string	ISO 8601, UTC
type	string	Type melding: kan styre visning i DPI
ref	String or array of strings	Liste av id'er som f.eks. StopPlace, Line, Route, VehicleJourney. brukes til å knytte avvik til bestemte scopes, ikke påkrevd
message	dictionary of MultilingualMessage	Key is ISO 639-1 language code.

MultilingualMessage

Name	Type	Description
title	string	title to be displayed to the passengers (optional)
text	string	text to be displayed to the passengers

infohub/dpi/audio/json

Navn	Audio message
Beskrivelse	Audio message to be played by the speaker system on the bus.
Lokal topic	infohub/dpi/audio/json
Bridged topic	<recipient>/ruter/<vehicleId>/itxpt/ota/dpi/audio/json
Schema	audio.json

Topic used exclusively to transmit audio messages to be played by the speaker system. The audio messages may contain an array of sound bites, that are to be played in the sequence they have been received.

JSON example

```
{
  "eventTimestamp": "2017-10-31T08:38:02.749Z",
  "expiryTimestamp": "2017-10-31T08:38:47.749Z",
  "type": "ARRIVING",
  "ref": "RUT:StopPlace:03012453",
  "value": [
    {
      "encoding": "OPUS",
      "content": "ZkxhQwAAACIQABAAAAAUJABtAA+gA8AB+W8FZndQvQAyjv...",
      "speaker": "PASSENGERS",
      "volume": 70
    },
    {
      "encoding": "MP3",
      "content": "ZkxhQwAAACIQABAAAAAUJABtAA+gA8AB+W8FZndQvQAyjv...",
      "speaker": "DRIVER",
      "volume": 60
    },
    {
      "encoding": "MP3",
      "content": "ZkxhQwAAACIQABAAAAAUJABtAA+gA8AB+W8FZndQvQAyjv...",
      "speaker": "EXTERNAL",
      "volume": 80
    }
  ]
}
```

Fields

Navn	Type	Beskrivelse
eventTimestamp	string	ISO 8601, UTC
expiryTimestamp	string	ISO 8601, UTC
type	string	Type of audio message
key	string	key of the audio message
value	array of Audio	one or more sound bits to be played in sequence

Audio

Navn	Type	Beskrivelse
encoding	string	enum Encoding
content	string	Base 64 encoded audio data
speaker	string	enum of SpeakerType
volume	int	1-100

Enum Encoding

Navn	Beskrivelse
OPUS	Opus Audio Codec (rfc6716)
MP3	MP3 Codec

Enum SpeakerType

Name	Prio	Description
PASSENGERS	1	Internal speaker for passengers
DRIVER	2	internal speaker for the driver only
EXTERNAL	3	External speaker for announcements to waiting passengers

Speaker types corresponds to the ITxPT-standard, S01v2.0.1_2017, Vehicle Installation Requirements Specification, side 32.

infohub/dpi/c2/json

Name	Command and controls channel
Description	Command and control messages from Ruter Data Platform
Local topic	infohub/dpi/c2/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/dpi/c2/json
Schema	c2.json

The c2 channels is reserved for command and control messages originated by Ruter. Typical use cases include:

- Diagnostics / debugging
  - Trigger transfer of debug information
  - Trigger screenshot of DPI screen
  - Trigger clearing of cache and refresh of webpage
- Content
  - Trigger display of campaign

The payload is defined as an object with no structure to provide flexibility.

JSON example - DEBUG

```
{
    "eventTimestamp": "2018-10-31T12:45:50Z",
    "type": "DEBUG",
    "payload": {
                    "command": "LOG_TRANSFER",
                    "arg": {
                              "level": "ERROR",
                              "limit": 10,
                              "page": 0
                    }
    }
}
```

JSON example - SET_ITEM

```
{
    "eventTimestamp": "2018-10-31T12:45:50Z",
    "type": "WORKER_JOB",
    "payload": {
                    "command": "SET_ITEM",
                    "arg": {
                              "name": "enableFeatureX",
                              "value": true
                    }
    }
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
type	string	message type
payload	Object	

infohub/dpi/diagnostics/json

Name	Screen diagnostics
Description	Report to PTA BO about a screen
Local topic	infohub/dpi/diagnostics/json
Bridged topic	ruter/<sender>/<vehicleid>/itxpt/ota/dpi/diagnostics/json
Schema	diagnostics.json

It is expected that the DPI application itself will produce diagnostic messages.

The payload is defined as an object with no structure to provide flexibility.

The types illustrated below are example of possible messages. The types are under discussion but will be generated entirely by the DPI application and consumed by the PTA BO.

JSON example - STATUS

```
{
    "eventTimestamp": "2018-10-31T12:45:50Z",
    "screenId": "ad71dba8-c881-11e8-a8d5-f2801f1b9fd1",
    "type": "STATUS",
    "payload": {
        "version": {
            "application": "2018-10-03T12:45:50Z",
            "media": "2018-10-05T12:45:50Z",
        },
        "display": {
            "type": "1",
            "res": {
                "height": 360,
                "width": 1080
            }
        },
        "stats": {
            "logEntries": {
                "error": 0,
                "warning": 14,
                "info": 123
            },
            "lastLoaded": "2018-10-31T12:45:45Z",
                            "pingFreq": 3600
            "usedStorage": "124kb"
        }
    }
}
```

JSON example - HEARTBEAT

```
{
    "eventTimestamp": "2018-10-31T12:45:50Z",
    "type": "HEARTBEAT",
    "screenId": "ad71dba8-c881-11e8-a8d5-f2801f1b9fd1",
}
```

Fields

Name	Type	Description
eventTimestamp	string	ISO 8601, UTC
screenId	string	UUID produced and stored by the application per screen
type	string	message type, enum DiagnosticType
payload	dictionary of any	data with a blend possibly of standardized keys and PTA / PTO-specific

Enum DiagnosticType

Name	Description
STATUS	When the screen is turned on and the application application starts, this message is sent
HEARTBEAT	A regular message that the screen is alive; frequency every X period

Other types of messages may be defined later.

Planned Topics

infohub/dpi/connections/json

Name	Connections
Description	A snapshot of real-time data about the transit connections at the next stop
Local topic	infohub/dpi/connections/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/connections/json
Schema	connections.json

This message is intended to be sent before the bus comes to the next stop. It is possible that it is sent only once, but it may also be updated if the view changes.

infohub/dpi/digitalsignage/json

Name	Multimedia control
Description	Change what appears on the target surfaces of the screens
Local topic	infohub/dpi/digitalsignage/json
Bridged topic	<recipient>/ruter/<vehicleid>/itxpt/ota/dpi/digitalsignage/json
Schema	digitalsignage.json

When we begin to deliver packages of media to the buses, for example, in connection with campaigns, we must be able to trigger playlists as needed, for example at a stop, a time, etc.

Summary of Changes

Category	Topic	Description
added field	infohub/dpi/journey/json	added location (latitude and longitude) per stop place
added field	infohub/dpi/eta/json	the field text has been added for display text in DPI
changed field	infohub/dpi/eta/json	the field expectedArrivalTime has been renamed eta
changed field	infohub/dpi/externaldisplay/json	the field routeName has been renamed destination
removed field	infohub/dpi/arriving/json	audio has been removed; see new audio/json topic
changed field	infohub/dpi/arriving/json	the field message was made multilingual
added field	infohub/dpi/arriving/json	the field zoneId was added to support the sales system
removed field	infohub/dpi/deviation/json	audio has been removed; see new audio/json topic
changed field	infohub/dpi/deviation/json	the field message was made multilingual
added field	infohub/dpi/deviation/json	the field ref was added to indicate what the deviation affects
removed field	infohub/dpi/announcement/json	audio has been removed; see new audio/json topic
changed field	infohub/dpi/announcement/json	the field message was made multilingual
added field	infohub/dpi/announcement/json	the field ref was added to indicate what the deviation affects
added field	infohub/dpi/announcement/json	the field type was added to support DPI needs
new topic	stopsignal/json	new topic for stop signal status when changed by using stop button
new topic	telemetry/+/json	new topic for vehicle telemetry from systems on the bus, including FMS data
new topic	tsp/json	new topic for traffic signal pre-emption message
new topic	infohub/dpi/audio/json	new topic on which **all audio** will be sent
new topic	infohub/dpi/c2/json	new topic to send commands to DPI application
new topic	infohub/dpi/diagnostic/json	used by DPI only to send diagnostic info to BO