v1.1 DIS

Endringslogg / Changelog since v1.0 DIS

- Remove `messageNumber` from alle event topics. This is no longer in use.
- Added `callSequenceNumber` to v1.1 Observed left-behind vehicles, v1.1 Observed left-behind passengers, v1.1 Ferry Fare Count and v1.1 Activation/deactivation of flexible stop on route
- Added missing events from v1.1 Message requirements guide
- Clarify that apiVersion in topic path is *major* version only
- Added limitations on fractional second formatting for [vNext] ISO 8601 with timezone
- Added `operatingDate` to v1.1 Sign on and v1.1 Sign off messages
- Added callSequenceNumber to v1.1 Passenger Count

v1.1 MQTT API

This is the description of the MQTT (Message Queuing Telemetry Transport) API for communication between PTOs and Skyss

General requirements

- MQTT must be of version 5.0 or higher.
- Data made available on the MQTT API must be JSON formatted and UTF-8 encoded
- The communication from Operator's API must be encrypted with TLS 1.2 or higher

In the pages below you will find specification of both topics as well as the *structure* of the messages. As far as possible there will be both a spec as well as code-examples.

All APIs are on the format:
`skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/[events/sensors]/flexstop`

Example:

Data for the [vNext] Location event for API Version 1.1 should be sent on:

`skyss/{operatorID}/{vehicleID}/v1/sensors/location`

v1.1 Topics

Events

> Event data are either data that are created at irregular times, for example on user input, or aggregated data that is only available at certain points in time. E.g. a combined passenger count event after the vehicle departs from a stop.

Sensors

> These are messages that are created/read from a sensor onboard the vehicle.
>
> All messages should contain a `timestamp` in the message. This should represent, as close as possible, the time the sensor data was ***measured*** by the source system, and **not** be the time the MQTT message was created.

Full list in alphabetical order:

- v1.1 Events
  - v1.1 Activation/deactivation of flexible stop on route — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/flexstop
  - v1.1 Ferry Fare Count — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/ferryfarecount
  - v1.1 Observed left-behind passengers — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/olbp
  - v1.1 Observed left-behind vehicles — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/olbv
  - v1.1 Operating mode — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/opmode
  - v1.1 Passenger Count — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/passengerCount
  - v1.1 Sign off — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/signoff
  - v1.1 Sign on — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/signon
- v1.1 Sensors
  - v1.1 Anemometer — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/anemometer
  - v1.1 Charging status detection — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/csd
  - v1.1 Distance — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/distance
  - v1.1 Energy - battery consumption — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/energy/battery
  - v1.1 Energy - fuel consumption — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/energy/fuel
  - v1.1 Inside temperature — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/intemp
  - v1.1 Location — skyss/{operatorID}/{VehicleID}/{majorAPIVersion}/sensors/location
  - v1.1 Outside temperature — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/outtemp
  - v1.1 State of Charge (SOC) — skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/soc
  - v1.1 Water temperature — skyss/{operatorID}/{VehicleID}/{majorAPIVersion}/sensors/watertemp

v1.1 Events

v1.1 Activation/deactivation of flexible stop on route

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/flexstop

v1.1 Operating mode

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/opmode

v1.1 Observed left-behind passengers

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/olbp

v1.1 Observed left-behind vehicles

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/olbv

v1.1 Passenger Count

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/passengerCount

v1.1 Ferry Fare Count

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/ferryfarecount

v1.1 Sign on

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/signon

v1.1 Sign off

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/signoff

v1.1 Activation/deactivation of flexible stop on route

Topic definition

Topic	`skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/flexstop`
Retain	v1.1 Retain
QoS	v1.1 Quality of Service (QoS) - 2
Type	Event
Frequency	Event triggered / manual input

This message describes the activation or deactivation of a flexible stop point on route. Flexible stop points are normally omitted unless activated. Activation of a flexible stop point would typically be triggered by a ticket sale, an electronic order, or some other form of notification to either the vessel, or to a back-office system. When the flexible stop is activated, the stop is visited by the vessel.

Deactivation of a flexible stop would normally follow an activation of a flexible stop.

If no flexible stops are activated or deactivated, no message is required.

Message structure

Name	Data type	Unit / Type specification	Description
timestamp	string	v1.1 ISO 8601 with timezone	Timestamp of the reading
flexibleStopActive	boolean	N/A	Whether the flexible stop is activated. (True = flexible stop is activated, False = flexible stop is deactivated)
nsrQuayName	string	N/A	The name of the flexible stop quay/dock according to NSR
nsrQuayRef	string	v1.1 NSR Quay Reference	The NSR id of the flexible stop quay/dock

callSequenceNumber	int	N/A	Order of the stop within the trip/servicejourney. This must match the underlying planning data for the trip.

Code example

```
{
    "timestamp": "2024-03-13T11:21Z",
    "flexibleStopActive": true,
    "nsrQuayName": "Skjerheim kai",
    "nsrQuayRef": "NSR:Quay:12345",
    "callSequenceNumber": 2
}
```

v1.1 Passenger Count

Topic definition

Topic	`skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/passengerCount`
Retain	v1.1 Retain
QoS	v1.1 Quality of Service (QoS) - 2
Type	Event
Frequency	Event triggered

This message describes the **accumulated** number of passengers from a specified quay/dock, after the vehicle has left the quay/dock and sailed. The count should not include the members of the crew or the driver of a vehicle. The count should include how many people alighted and how many boarded the vehicle at the quay/dock.

The message can also be sent after a manual counting has been performed and entered. A flag determines whether the count is automatic or manual.

Only categories “Adult” and “child” will contribute to accumulated “Count” values.

Message structure

Name	Data type	Unit / Type information	Description
timestamp	string	v1.1 ISO 8601 with timezone	Timestamp of the event. Typically as close as possible to the departure time.
manualCount	bool	N/A	True/False whether the count was partially or completely manual.
nsrQuayRef	string	v1.1 NSR Quay Reference	NSR Quay reference
alighting	int	N/A	Number of people that alighted the vehicle at the stop.
boarding	int	N/A	Number of people that boarded the vehicle at the stop.

callSequenceNumber	int	N/A	Order of the stop within the trip/servicejourney. This must match the underlying planning data for the trip.

Code example

```
{
    "timestamp": "2024-03-13T11:21Z",
    "manualCount": true,
    "nsrQuayRef": "NSR:Quay:49976",
    "alighting": 23,
    "boarding": 34,
    "callSequenceNumber": 3
}
```

v1.1 Sign on

Topic definition

Topic	`skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/signon`
Retain	v1.1 Retain
QoS	v1.1 Quality of Service (QoS) - 2
Type	Event
Frequency	Event triggered / manual input

This event should be sent as soon as the vehicle signs on to a journey. All messages sent in between a sign-on and a sign-off message is logically connected to the journey specified in these messages.

See v1.1 Sign on / Sign off flow for more information on how this is intended to work.

Message structure

Name	Type	Unit	Description
timestamp	string	v1.1 ISO 8601 with timezone	Timestamp of the reading
serviceJourneyId	string	v1.1 ServiceJourneyId	ServiceJourneyId
operatingDate	date	v1.1 ISO 8601 date	Operating date of the service journey. This should match what is found in plan-data for the service journey. Note that trips happening shortly after midnight is often assigned to the previous day when considering operating dates.

Code example

```
{
    "timestamp": "2024-03-13T11:21Z",
    "serviceJourneyId": "SOF:ServiceJourney:9559_578_1415_69"
    "operatingDate": "2024-03-13"
```

```
}
```

v1.1 Sign off

Topic definition

Topic	`skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/events/signoff`
Retain	v1.1 Retain
QoS	v1.1 Quality of Service (QoS) - 2
Type	Event
Frequency	Event triggered / manual input

This event should be sent as soon as the vehicle signs off from a journey. All messages sent in between a sign-on and a sign-off message is logically connected to the journey specified in these messages.

See <link to description> for more information on how this is intended to work.

Message structure

Name	Type	Unit	Description
timestamp	string	v1.1 ISO 8601 with timezone	Timestamp of the reading
serviceJourneyId	string	v1.1 ServiceJourneyId	ServiceJourneyId
operatingDate	date	v1.1 ISO 8601 date	Operating date of the service journey. This should match what is found in plan-data for the service journey. Note that trips happening shortly after midnight is often assigned to the previous day when considering operating dates.

Code example

```
{
    "timestamp": "2024-03-13T11:21Z",
    "serviceJourneyId": "SOF:ServiceJourney:9559_578_1415_69",
    "operatingDate": "2024-03-13"
```

```
}
```

v1.1 Sensors

v1.1 Anemometer

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/anemometer

v1.1 Charging status detection

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/csd

v1.1 Distance

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/distance

v1.1 Energy - battery consumption

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/energy/battery

v1.1 Energy - fuel consumption

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/energy/fuel

v1.1 Inside temperature

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/intemp

v1.1 Location

skyss/{operatorID}/{VehicleID}/{majorAPIVersion}/sensors/location

v1.1 Outside temperature

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/outtemp

v1.1 State of Charge (SOC)

skyss/{operatorID}/{vehicleID}/{majorAPIVersion}/sensors/soc

v1.1 Water temperature

skyss/{operatorID}/{VehicleID}/{majorAPIVersion}/sensors/watertemp

v1.1 Location

Topic definition

Topic	`skyss/{operatorID}/{VehicleID}/{majorAPIVersion}/sensors/location`
Retain	No
QoS	v1.1 Quality of Service (QoS) - 0
Type	Sensor
Frequency	< 10 seconds

This message describes the GPS location of the vessel/vehicle, and relevant meta-data attached to the reporting of its position.

Message structure

Name	Type	Unit	Description
timestamp	string	v1.1 ISO 8601 with timezone	Reflects the UTC time provided by the GPS equipment for position fix. This is the point in time the location applies to.
latitudeDegree	float	v1.1 Decimal Degree	Latitude coordinate in decimal degrees
longitudeDegree	float	v1.1 Decimal Degree	Longitude coordinate in decimal degrees
altitude	float	n/a	Altitude value (in meters) above mean sea level.
trackDegreeTrue	float	v1.1 Heading / direction	Heading of the vehicle.
numberOfSatellites	integer	n/a	Number of satellites used
speedOverGround	float	m/s	GNSS based speed over ground (m/s)

Code example

```
{
    "timestamp": "2024-03-13T11:26:53Z",
    "latitudeDegree": 63.214599609375,
    "longitudeDegree": 7.98093318939209,
    "altitude": 0.0,
    "trackDegreeTrue": 38.0,
    "numberOfSatellites": 0,
    "speedOverGround": 18.05,
}
```

v1.1 Topic Specs

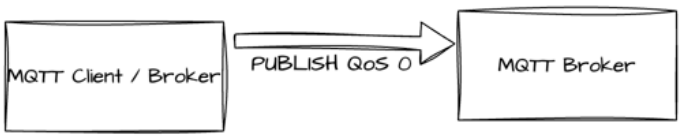
The pages below define in detail requirements set on a **topic** level. They are all referenced from the topic definitions.

v1.1 Quality of Service (QoS) - 0

"At most once", where messages are delivered according to the **best efforts** of the operating environment, also known as “fire and forget”.

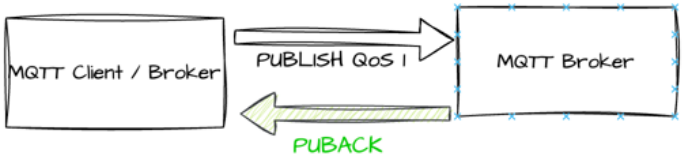
This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.

Message loss can occur at this quality of service level and the message will not be stored or re-transmitted.



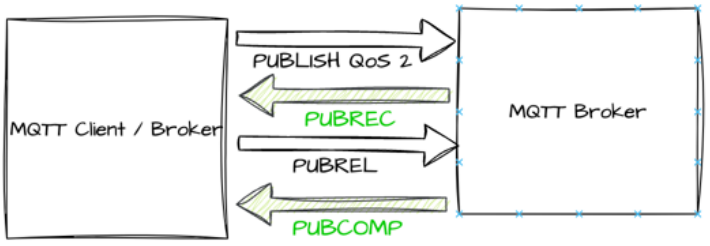
v1.1 Quality of Service (QoS) - 1

"At least once", where messages are assured to arrive but **duplicates** can occur.



v1.1 Quality of Service (QoS) - 2

"Exactly once", where messages are assured to arrive exactly once. This level could be used, for example, with billing systems where duplicate or lost messages could lead to incorrect charges being applied.



v1.1 Retain

An MQTT retained message is a standard message that includes the retained flag set to true, signifying its importance. When a client publishes a retained message, the broker stores it along with the corresponding Quality of Service (QoS) for a specific topic. Subscribing clients immediately receive the retained message when they subscribe to a topic pattern that matches the retained message's topic.

v1.1 Units / Type specifications

- v1.1 Celsius
- v1.1 Decimal Degree
- v1.1 ISO 8601 date
- v1.1 ISO 8601 with timezone
- v1.1 Percentage - bounded
- v1.1 Meter per second (m/s)
- v1.1 Message Number
- v1.1 Heading / direction
- v1.1 NSR Quay Reference
- v1.1 ServiceJourneyId
- v1.1 AutoPASS vehicle category
- v1.1 Operating Mode categories

v1.1 Celsius

Measure of temperatur using Celcius. This should be a `float` value.

Examples

`12.2` - ✅ Valid

`-2.3` - ✅ Valid

`21.2C` - ❌ Not Valid - No unit should be included

`"21.2"` - ❌ Not Valid - should be a `float` and not a `string`

v1.1 Decimal Degree

Latitude or longitude represented as a decimal value. See Decimal degrees. Skyss requires at least `4` decimal places in the value. From Wikipedia:

decimal places	decimal degrees	DMS	Object that can be *unambiguously* recognized at this scale	N/S or E/W at equator	E/W at 23N/S	E/W at 45N/S	E/W at 67N/S
0	1.0	1° 00′ 0″	country or large region	111 km	102 km	78.7 km	43.5 km
1	0.1	0° 06′ 0″	large city or district	11.1 km	10.2 km	7.87 km	4.35 km
2	0.01	0° 00′ 36″	town or village	1.11 km	1.02 km	0.787 km	0.435 km
3	0.001	0° 00′ 3.6″	neighborhood, street	111 m	102 m	78.7 m	43.5 m
4	0.0001	0° 00′ 0.36″	individual street, large buildings	11.1 m	10.2 m	7.87 m	4.35 m
5	0.00001	0° 00′ 0.036″	individual trees, houses	1.11 m	1.02 m	0.787 m	0.435 m
6	0.000001	0° 00′ 0.0036″	individual humans	111 mm	102 mm	78.7 mm	43.5 mm
7	0.0000001	0° 00′ 0.000036″	practical limit of commercial surveying	11.1 mm	10.2 mm	7.87 mm	4.35 mm
8	0.00000001	0° 00′ 0.000036″	specialized surveying	1.11 mm	1.02 mm	0.787 mm	0.435 mm

> Longitude ranges from `-180` westward to `+180` eastward, while Latitude ranges from `-90` to `+90`.

Examples

`63.214599609375` - ✅ Valid

`"N 60° 22' 22.103'"` - ❌ Not Valid - Not on decimal format

v1.1 ISO 8601 date

String representation of a *date* using the W ISO 8601 standard.

Examples

`2024-03-13` ✅ Valid (UTC timezone)

`2024.03.13` ❌ Not Valid - wrong separating character

v1.1 ISO 8601 with timezone

String representation of a datetime using the W ISO 8601 standard, with the following exceptions:

> ⚠ **Important - deviation from standard**
>
> - Fractional seconds must be represented with `.` and not `,`. e.g. `2024-10-07T13:06:00.954+02:00` and not `2024-10-07T13:06:00,954+02:00`
> - Timezone information **must** be included.

Examples

`2024-03-13T10:48:55Z` ✅ Valid (UTC timezone)

`2024-10-07T13:06:00.954+02:00` ✅ Valid (with both fractional seconds and timezone info)

`2024-03-13T10:48:55` ❌ Not Valid - no timezone information included

`2024-03-13T10:48:55,123Z` ❌ Not Valid - fractional seconds must be represented with `.`

v1.1 Percentage - bounded

A `float` value representing a percentage value ranging from `0` to `100`. `%` sign should *NOT* be included in the value.

Optionally it can include decimals depending on specification in message structure.

Examples

`5` - ✅ Valid - represents 5%

`23.23` - ✅ Valid - represents 23.23%

`102` - ❌ Not Valid - this field is bounded to 0 - 100

`-23` - ❌ Not Valid - This field is bounded to 0 - 100

`3%` - ❌ Not Valid - Percentage sign should not be included

```
{ "value": 23.23 } # Valid
{ "value": "23.23" } # Not Valid - Should be a float
```

v1.1 Meter per second (m/s)

Here used as a measurement of ***speed***.

Should be a `float` value and **positive**.

Examples

`5` - ✅ Valid - represents 5 m/s

`23.23` - ✅ Valid - represents 23.2 m/s

`-23` - ❌ Not Valid - Speed cannot be negative.

`"3m/s"` - ❌ Not Valid - Should not be string

```
{ "value": 23.23 } # Valid
{ "value": "12.23" } # Not Valid - Should be a float
```

v1.1 Message Number

All events messages should contain a message number. This should be:

- an `integer` value
- monotonically increasing by `1` (one) for each event being created by the vechicle/event combination

Examples

`123` - ✅ VALID

`"123"` - ❌ NOT VALID - should be an `integer` value

v1.1 Heading / direction

Skyss uses this both for **vessels** and to indicated direction of **wind**.

A `float` value ranging from 0 to 360 (inclusive). It cannot be negative.

Heading is typically based on cardinal directions, so 0° (or 360°) indicates a direction toward true north, 90° true east, 180° true south, and 270° true west.

For a vessel the heading *may not necessarily be the direction that the vehicle actually travels*, which is known as its *course* or *track*. Any difference between the heading and course is due to the motion of the underlying medium, the air or water, or other effects like skidding or slipping. See more info at: Heading (navigation)

Examples

`123` - ✅ Valid - represents 123° from true north

`180` - ✅ Valid - heading directly south

`-12` - ❌ Not Valid - Heading cannot be negative

`23.1` - ✅ Valid - 23.1°

`"90"` - ❌ Not Valid - Should be a `float` value

v1.1 NSR Quay Reference

This should be a `string` value containing a *valid* NSR Quay Id. EnTur manages a national list of stops National Stop Register | Intro. The operator should normally receive this data as a part of the planning data received regularly from Skyss.

Examples:

`"NSR:Quay:49976"` - ✅ VALID

`499976` - ❌ NOT VALID - not a string

v1.1 ServiceJourneyId

This is a unique `string` identifier found in the NetEx format. It represents a journey with its associated timetable and stop-patterns. See Data models | NeTEx conceptual model for ServiceJourney for more information.

Examples:

`"SOF:ServiceJourney:9559_578_1415_69"` - ✅ VALID

`499976` - ❌ NOT VALID - not a string

v1.1 AutoPASS vehicle category

Should contain a string value with a valid “Takstgruppe”.

See: Informasjon | Ferjedatabanken At the date of publication the following categories exists:

TAKSTGRUPPE	KATEGORI	BESKRIVELSE
AP1	Kjøretøy	Kjøretøy 0 - 6m
AP2	Kjøretøy	Kjøretøy 6.01 - 8m
AP3	Kjøretøy	Kjøretøy 8.01 - 10m eller 0 -6m med tilhenger u/10m
AP4	Kjøretøy	Kjøretøy 10.01 - 12.50m eller 0 -6m med tilhenger o/10m
AP5	Kjøretøy	Kjøretøy 12.51 - 14.50m
AP6	Kjøretøy	Kjøretøy 14.51 - 17.50m
AP7	Kjøretøy	Kjøretøy 17.51 - 19.50m
AP8	Kjøretøy	Kjøretøy 19.51 - 22m
AP9	Kjøretøy	Kjøretøy over 22m
MC	MC	Motorsykkel
AP1FR	Kjøretøy	Kjøretøy 0 - 6m (gjelder for Flakk - Rørvik sambandet)
AP2FR	Kjøretøy	Kjøretøy 6.01 - 12m (gjelder for Flakk - Rørvik sambandet)
AP3FR	Kjøretøy	Kjøretøy over 12m (gjelder for Flakk - Rørvik sambandet)
D1	Annet	Straffegebyr
D2	Annet	Gods
D3	Annet	Beredskapstur
D4	Annet	Lugar
D5	Annet	Forhåndsbestillingsgebyr

D6	Annet	Kaiavgift
D7	Annet	Gebyr for kjøretøy uten fører
D8	Annet	Ventegebyr < 30 min
A40	Person	Telling person (bilfører, bilpassasjer og reisende uten bil)
GB1	Annet	Telling gjenstående biler

v1.1 Operating Mode categories

Assignment categories for vessel operation

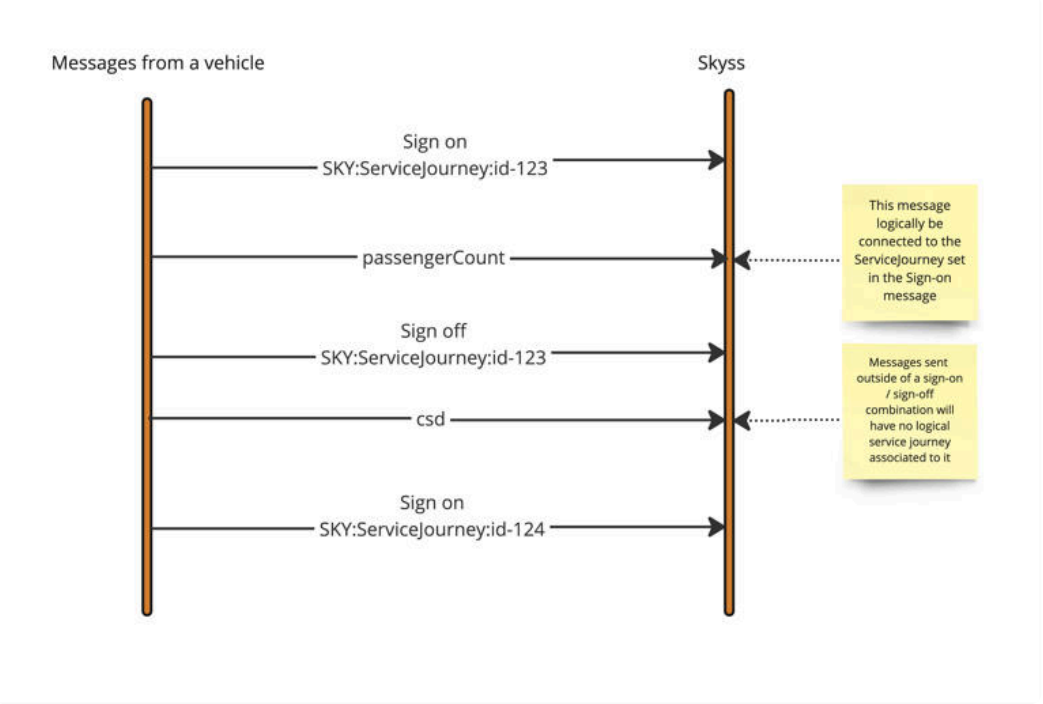
1 =Position driving
2 = Maintenance driving
3 = 3rd party assignment
4 = Short-term change

v1.1 Sign on / Sign off flow

In order to correlate the MQTT events to actual planned trips Skyss requires explicit messages whenever a vehicle starts and ends a ServiceJourney. All messages that flow between a v1.1 Sign on message and an Sign off messages are logically assigned to that ServiceJourney.

All messages sent outside of a sign-on / sign-off combination will have no service journey logically associate to it. This is expected for all operations happening outside scheduled passenger traffic.

Due to the importance of these messages they should always be sent with v1.1 Quality of Service (QoS) - 2.



v1.1 SIRI Service description

About the SIRI standard

SIRI is a CEN standard for exchanging real-time information from public transportation operations. The standard contains both an information model and a web service.

Homepage

An Norwegian content interpretation (profile) of the SIRI standard has been made by Entur

Nordic SIRI Profile

Consuming relevant SIRI web services from an Operator is Skyss' desired method for automatically measuring and monitoring traffic, punctuality and regularity from a transport contract. A successfully implemented SIRI-service based on a accurate real-time system have the prerequisites for replacing manual reporting.

The following SIRI-services are supported by the Skyss SIRI consumer application:

- Estimated Timetable (ET)
- Vehicle Monitoring (VM)
- Stop Monitoring (SM)
- Situation Exchange (SX)
- Production Timetable (PT)

A SIRI service from an Operator will normally be based on content and business logic in an Operator's real-time system/real-time solution. For that reason, the real-time system is also sometimes referred to in this description.

Robustness and communication

The connection between the Operator and the Consumer client is established through a subscription in a publish/subscribe communication pattern, where the Operator's service keeps track of what the Client has received and what the Client has requested in the subscription request when the subscription is established or if the subscription is re-established. The Operator must ensure that the Client has all required accesses to the service.

In the event of communication failure between the Operator and the Client, the Operator's service must provide robustness mechanisms that ensures persistence of data which

ensures that the Client receives all information which occurred, but was not sent during the downtime.

Business logic

The trip or route the vessel/vehicle is on must be represented in the SIRI service as a timeline, from start to finish, together with all relevant references to route data / scheduled data. Historical times, i.e. times for arrivals or departures that lies back in time, must be stated as measured/actual times - not as the planned / scheduled times. The measured times must not be rounded, and must be given in whole seconds.

The service and communication with the Client must be event-driven, so that an actual event triggers an update that is sent. Examples on events which should be included, as a minimum:

- Arrival or departure derived by the real-time system
- Cancellation, partial or full
- Change in route
- Publication of deviation information
- Updating of deviation information
- (Significant) change in derived delay value
- Login of the vessel in the real-time system before the start of the route/trip or start of operating day
- Logout of the vessel in the real-time system after completion of the route/trip or end of operating day

In order to automatically derive arrivals and departures, the real-time system must use specific measurement points. These can, for example, be based on geography, position, sensors, telemetry or a combination of these. Arrivals and departures must always be added to a time-date object.

Between calls, it is expected that the real-time system checks the actual position and any eventual alternative pathway against the planned position and regular pathway - and derives updated forecasts based on a comparison, so that any delays can be detected and communicated in sufficient time before the call.

Deviations

Deviations from the scheduled timetable that includes information about the deviation to the public and/or information to the Skyss must be sent via the Situation Exchange (SX) service. The `reason` for the deviation should always be communicated and in accordance with categories provided by Skyss.

Profile compliance

“Håndbok N801” describes ID's and information elements that are specific to the Norwegian/Nordic profile. The service must comply with these. The full name and ID's of quays included in the route data must be in accordance with the conventions used by “NSR - Norsk stoppesteds register”

Examples of best practice in some of the various SIRI services are also provided in “Håndbok N801” by Entur.

Documentation

A comprehensive documentation of the SIRI service, how the service reflects business logic in the real-time system along with explanations of how the service meets the quality requirements must be documented in the progress plan according to the Contract - and be available to Skyss before test consumption