Appendix to **IRS 90940**

2nd edition, September 2022

Last update 01/07/2022

Original

Data exchange with Driver Advisory Systems (DAS) following the SFERA protocol

Appendix B

SFERA Communication Protocol

This appendix:

- is updated regularly,
- corresponds to the above date of update.

INTERNATIONAL UNION OF RAILWAYS

Table of contents

Appendix B - SFERA Communication Protocol

Chapter 6 of the IRS contains a description of the three different layers (presentation, authorisation and communication) of the SFERA protocol. In the communication between DAS-OB and IM DAS-TS or RU DAS-TS, all layers are implemented to be fully SFERA compliant. In the communication between IM DAS-TS and RU DAS-TS, the application layer is the same but the authorisation and communication layers are implemented by the Common Interface (CI).

This appendix covers the detailed specification for both cases:

- The first clause covers the SFERA communication protocol, designed for communication between the DAS-OB and the DAS-TS. It contains a detailed specification of the authorisation layer based on the JWT web tokens and the communication layer with Message Queuing Telemetry Transport (MQTT) Version 5 over a secure web socket connection (TLS 1.3).
- The second clause explains the way SFERA integrates with the CI protocol (from the TAF/TAP TSI) which will support the exchange of SFERA messages between the IM DAS-TS and the RU DAS-TS.

1 DAS-OB - IM DAS-TS/RU DAS-TS communication

1.1 Motivation for the choice of MQTT

This paragraph gives the motivation for choosing MQTT instead of other alternatives (e.g. AMQP) or a simpler model such as REST API. The starting point is provided by the requirements of the communication protocol as described in Section 2.5 of the IRS.

The main arguments for choosing MQTT in the SFERA protocol are the following:

- *Interoperability:* to support interoperability for the IM-Train and IM-RU setups, it is not possible to leave the choice for technology open, so a choice has to be made for at least the technology that is used to physically exchange data. Otherwise, the DAS implementation would become unnecessarily complex.
- *Both guaranteed delivery and clean start:* SFERA chose to use guaranteed delivery, especially for short interruptions in an Internet connection (a few seconds to a few minutes) and, where necessary, only new messages are seen (clean start). With any messaging system, this comes ‘out of the box’. To implement this in a REST API solution, for instance, all the functionality has to be built into the application layer.
- *Support of multiple devices (1:n):* with the publish-and-subscribe message exchange and topic tree model, it is easy to support multiple devices (e.g for driver change) out of the box. So, the distribution logic regarding both one-to-one and one-to-many communications is solved ‘for free’ with the topic tree model.
- *Platform-independent and vendor-independent:* MQTT is widely used and has many implementations (e.g. C#, Java, .Net) and has more implementation variants than, for example, AMQP. MQTT is designed more for networks with high latency (low footprint) and can be effectively built into embedded systems. Both commercial and open-source implementations of the MQTT broker are available.

1.2 Implementing the DAS-OB - DAS-TS communication

The design of the DAS-OB - DAS-TS communications is based on the following principles:

- The SFERA service can be accessed through public Internet connections.
- A DAS-OB has a logical 1-to-1 communication channel with the DAS-TS, and the DAS-TS may have a logical 1-to-many communication channel with the DAS-OB for a given train.
- A technical connection between a DAS-OB and the DAS-TS may be disconnected and reconnected at any time.
- A single technical connection between a DAS-OB and the DAS-TS can be used during many train movements.
- A DAS-OB may have connections with multiple IMs at the same time. SFERA messages contain information on when to connect to another IM to get the required follow-up information.

MQTT is a message broker system that implements a so called publish-subscribe pattern. Using this pattern, the message is not sent directly to specific receivers, called subscribers. Instead, the messages are published to a broker (man in the middle) who handles their delivery to the connected subscribers, even when they are not connected for a moment. This way messages are delivered asynchronously, the sender (publisher) does not have to know anything about the connected receivers and will publish them once to the broker. The broker will guarantee that messages are delivered to all devices.

Fig. 1 illustrates this pattern using the example of a solar panel publishing data when it produces 1 kW. The solar panel only publishes to the broker (and not to the single devices). The devices (laptop and mobile) can subscribe to the broker to retrieve this data. The solar panel has no knowledge of the connected devices.

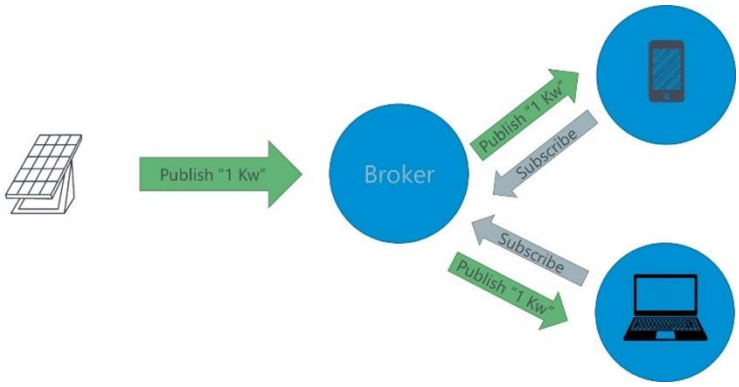

Figure 1: Example of publish-subscribe pattern

The DAS-TS shall be made available as an MQTT V5 service (the SFERA service); service end points are communicated with contracted RUs. This MQTT service shall have a predefined topic structure. This structure is organised around RUs, train movements and individual DAS devices to accommodate the one-to-many and one-to-one communications.

1.2.1 Topic tree for distribution logic

The MQTT messaging system is basically an organised topic tree (hierarchy) which can contain multiple levels. It can be compared with a folder structure on a hard drive. A topic is a name (string) where levels are separated by a slash (/). For example, a topic structure for the solar panel example in Fig. 1 could be home/room/topfloor/solarpanel/panel1.

1.2.2 Topic tree for the SFERA protocol

SFERA defines a mandatory topic tree organisation as illustrated in Fig. 2, and the details of the topic structure are described in the implementation in Section 1.4. The topic structure is defined such that:

- multiple major versions of SFERA messages can be processed within one single MQTT deployment;
- each DAS device can have individual request/response (G2B and B2G segments) interactions with the DAS-TS;
- each DAS device can send individual event messages to the DAS-TS (B2G segment);

  the DAS-TS can publish messages to trains, i.e. one or more DAS devices associated with a single train movement (event segment).

The RU grouping is for transparency reasons only.

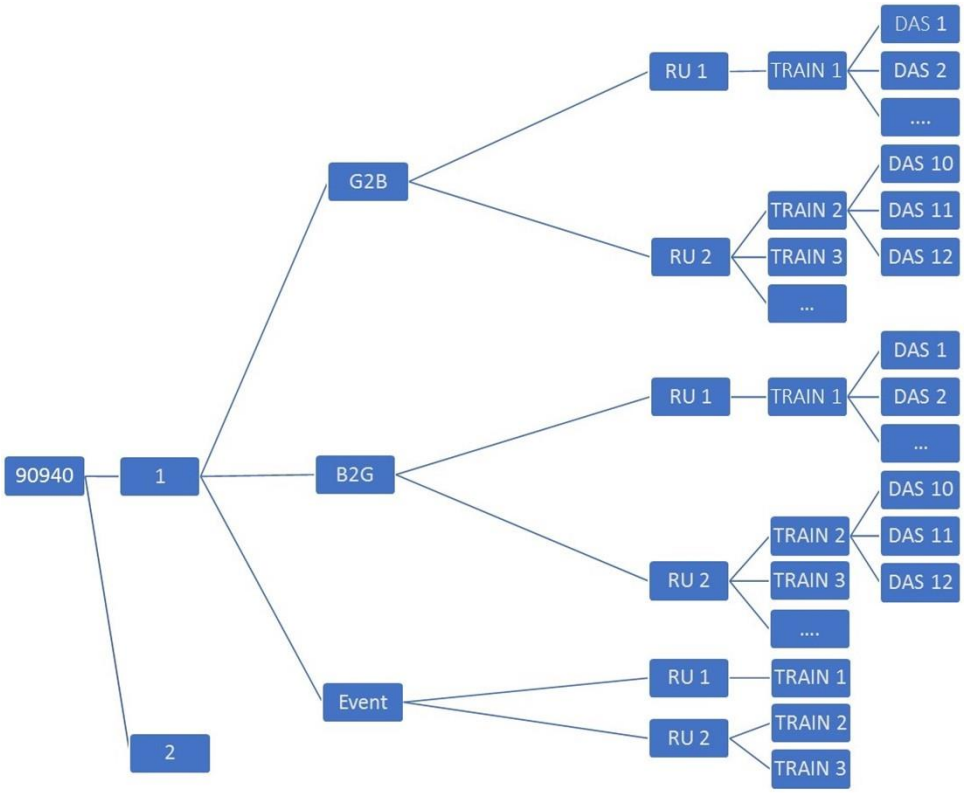

Figure 2: SFERA Topic Tree

1.3 Authorisation and security

The authorisation model is based on the following:

- A signed JSON web token (JWT) is used to carry authorisation information, *IETF RFC 7519 standard*.
- A token contains claims for:
    - identification of issuing system
    - identification of RU
    - identification of DAS device
    - identification of SFERA service that provides the SFERA data
    - identification of the train movement that is to be operated and time validity of token
- An RU is responsible for the token issuing process; in the ERTMS/ATO setup the IM is responsible for setting up a token issuing process that maps the authorisations obtained through the ETCS processes to SFERA authorisation tokens.
- Each DAS-TS maintains an administration that holds the public key(s) required to verify tokens issued by that RU for all recognised RUs.
- With all MQTT operations a DAS-OB performs to interact with the MQTT broker, it shall send a valid token.
- Any DAS-OB from a recognised RU is allowed to connect with the MQTT broker.
- A DAS-OB can only exchange messages with the MQTT broker through subscribe and publish operations; these are restricted to the train movement that the DAS instance has been authorised for by means of the authorisation token.
- To ensure integrity of tokens during transmission, public/private key encryption is used to sign tokens.

The IRS does not specify:

- the authentication/authorisation mechanism that is used to connect the TMS to the DAS-TS service;
- how public key information is exchanged between RUs and IMs (e.g. X509 certificates);
- whether device authentication is required.

NB:
Any OAUTH2-based authorisation procedure can be used to produce the required tokens, but any custom process that produces tokens according to the above specifications will work. The implementation depends on local requirements.

1.3.1 Security/integrity implementation specifications

The security/integrity specifications for DAS-OB - DAS-TS links are listed in Table 1.

Table 1: Security/integrity specifications for DAS-OB - DAS-TS links

Specification	Remark
The MQTT broker shall accept only TLS (Transport Layer Security) or TLS over web socket connections.	HTTP/plain socketconnections are insecure.
The MQTT broker shall refuse connection requests not bearing a token at all or not bearing a valid token.	The MQTT broker refuses the connect operation with status unauthorised.
The MQTT broker shall only accept messages on a topic that matches the train identifier and RU claims in the associated token.	The MQTT broker shall disconnect the connection with status unauthorised in case of a mismatch.
The MQTT broker shall not accept messages with an invalid or missing token.	The MQTT broker shall disconnect the connection with status unauthorised in this case.
A DAS-OB shall apply server certificate validation.	Verify server identity
DAS-TS and DAS-OB shall only send data that complies with the referenced XML schema version and shall not accept data that does not comply with the referenced XML schema.	Data integrity
Tokens shall be treated/processed by all parties as confidential information.	
Tokens should be issued for relatively short validity periods and be re-issued (refreshed) frequently.	Typical token validity is an hour.

JWT tokens are not part of the MQTT standard and there is no MQTT product that offers the required authorisation logic out of the box. The authorisation rules shall be implemented as a custom authorisation module in the MQTT service. All major MQTT brokers support a plugin mechanism for authentication and authorisation. The authorisation module shall ensure that publish/subscribe operations from individual DAS devices only access the topic tree sections that match the RU and train movement claims in the authorisation token.

The details of the token format are given in Section 1.5.

1.4 MQTT operations

The implementation and technology of this publish-subscribe pattern in SFERA is formalised and described. It shall be used for the IM-Train setup and it is recommended for the IM-RU setup.

In the IM-Train setup, the IM DAS-TS acts as MQTT broker and in the IM-RU setup, the RU DAS-TS acts as MQTT broker. In both setups, the RU DAS-OB acts as client for the broker. To achieve interoperability, SFERA defines precise rules on how to organise the topic trees and messaging operation attributes in each scenario. These are described in the basic operations of the MQTT protocol, which are:

- *Connect:* to set up a connection with a broker (DAS-TS);
- *Subscribe:* to start listening to messages (receive);
- *Publish:* to send a message to the other party. This will result in the receipt of the message by the subscribed client(s);
- *Disconnect:* to disconnect from a broker.

The four basic communication patterns described in Section 6.2 of the IRS are all mapped onto the Subscribe and Publish operations. The Connect and Disconnect operations are used to start/end a physical connection with the broker such that messages can actually be exchanged. One physical connection can be used for a number of subsequent communications from different use cases, depending on the network connections.

This section gives more detailed definitions for the communication layer. It describes details of the different communication principles as well as requirements on the topic structure within the message broker.

1.4.1 Connect

For a DAS-OB to start communication with the DAS-TS MQTT broker, there are a few parameters which are mandatory. These are described in Table 2.

Table 2: Mandatory parameters for MQTT communication

Parameter	Value / Description
Username	JWT
Password	The JSON Web Token (JWT), authorisation with JSON web tokens, refer to Section 1.5
Clean Start	This parameter depends on whether it is the start of a train drive (initial connection) or a reconnection. If it is an initial connection, this means no information has been requested from board to ground for this train on this specific date and so this parameter is true. When the DAS is disconnected and reconnected again (e.g. lost Internet connection), this parameter is false. This means that any messages that are sent in the period between disconnect and reconnect shall still be delivered to the client (receiver).
Session expiry interval	This value shall be set to a reasonable value to help the server clean up state information as soon as possible. A reasonable value is (remaining) journey duration + a margin for dealing with operational delays that may occur. This

Parameter	Value / Description
	is situation-dependent. It shall be set when Clean Start is set to true and also when it is set to false.
Client ID	Unique identifier of the device. When installing the software, a unique identifier (UUID V4[1]) will be generated for this device (once); this shall be used as the client ID. On reinstalling the software, a new UUID can be generated.

Remember that the IRS does not specify how a TMS or RU DAS-TS that actually produces or consumes the SFERA messages connects with the broker. All MQTT products offer options to separate external message exchange (with DAS-OB clients) from internal message exchange (TMS/RU Common Interface gateway) from a network point of view and apply different security mechanisms for internal connections.

1.4.2 Subscribe

As explained, to receive information (listen) from MQTT, an on-board system or a trackside system shall subscribe to the right topics. Also, to achieve guaranteed delivery some parameters on the subscribe operation have to be set. To make SFERA work in an interoperable way (cross-IM), the names of these topics and the parameters shall be used as described in this paragraph.

Table 3 shows the only mandatory parameter for subscribing to a topic. For authorisation, a JWT token is required. This allows the IM to validate if the client is allowed to retrieve information for a certain train by validating the requested topic with the JWT Token.

Table 3: Mandatory parameters for subscribing to a topic

Parameter	Description
User property	User properties are so-called key/value pairs for each MQTT operation; here it will be used to send the JWT Token. The name of the key is “bearer” (naming convention) and the value is the actual JWT Token.
QoS	2 (Highest possible level). QoS stands for “Quality of Service”

On Board

A DAS-OB client subscribes to two topics based on the type of interaction:

- Topic for the request-response interactions
- Topic for train-based events

[1] UUID=Universally Unique Identifier, see IETF RFC 4122.

These two topics are described in the subsections below. The naming of topics contains variables explained in Table 4.

Table 4: Variables used in the naming of topics

Variable	Value / Description
Major version of SFERA	The major version of IRS 90940, for example 1
Company code of RU	The UIC company (RICS) code, see https://uic.org/rics
Train identifier	Unique identifier of the train trip. This identifier depends on whether a TAF/TAP or an OTN identifier is used.- TAF/TAP: "{Core} + _ + {Variant} + _ + {StartDate}"- OTN: "{TrainNumber} + _ + {StartDate} + _ + {AdditionalTrainNumber}". The _ and AdditionalTrainNumber are only added when it is available (filled)
Client ID	The unique identifier of the device, as described in Table 3

Topic for the request-response interactions

The pattern for the topic name is the following: (names in brackets {} are variables described in Table 4)

```
90940/{Major version of SFERA}/G2B/{Company code of RU}/{Train identifier}/{Clientid}
```

For example, take a DAS-OB with Clientid 'fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9' which is in use by the RU NS (Company code 1084) and is starting a trip with train 512 on 21 September 2019. Let's also assume that SFERA Version 1.2 is used. The topic structure is then as follows:

```
90940/1/G2B/1084/2019-03-21_512/fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9
```

The following messages (see Appendix E for details on SFERA messages) will be received by the RU DAS-OB on this topic:

- SFERA_G2B_RequestMessage
- SFERA_G2B_ReplyMessage

Topic for train-based events

The pattern for the topic name is the following:

```
90940/{Major version of SFERA}/event/{Company code of RU}/{Train identifier}
```

Using the example described in Section 1.4.2, the topic name will be the following:

```
90940/1/event/1084/2019-03-21\_512
```

The following message will be received by the RU DAS-OB on this topic:

- SFERA_G2B_EventMessage

Trackside

The DAS-TS (TMS for example) subscribes to one node in the topic tree to process messages from all DAS devices, the topic for the request-response interactions.

The template of the topic naming is:

`90940/{Major version of SFERA}/B2G/#`

Names in brackets {} are variables described in Table 4. A dash (#) is a multi-level wildcard; in practice it means that it will receive all requests from all clients.

The following messages will be received by the DAS-OB on this topic:

- SFERA_B2G_RequestMessage
- SFERA_B2G_EventMessage (Status report)
- SFERA_B2G_ReplyMessage

1.4.3 Publish to topics

To send information with MQTT from a DAS-OB to the DAS-TS or vice versa, information is published to certain topics. The naming of the topics shall be used as prescribed here.

The names in brackets {} are the same variables described in the Subscribe part in Table 4. There are a few parameters that are mandatory for publishing a message to MQTT: these are described in Table 5.

Table 5: Mandatory parameters for publishing a message to MQTT

Parameter	Value / Description
Retain	False
QoS	2 (Highest possible level). QoS stands for “Quality of Service”
User property	User properties are so-called key/value pairs for each MQTT operation; here this will be used to send the JWT Token. The name of the key is “bearer” (naming convention) and the value is the actual JWT Token.

The naming of the topics is described below and is also split into an on-board and a trackside part.

On Board

A DAS-OB client publishes its request or event information to one single topic with the following pattern:

```
90940/{Major version of SFERA}/B2G/{Company code of RU}/{Train identifier}/{Clientid}
```

Names between brackets {} are variables described in Table 4.

The following messages are published by the DAS-OB on this topic:

- SFERA_B2G_EventMessage
- SFERA_B2G_RequestMessage
- SFERA_B2G_ReplyMessage

Trackside

The DAS-TS publishes its response on a request or event to the following topics, depending on the type of the message:

- Topic for the request-response interactions
- Topic for the event-based events

These two topics are described in the subsections below. The naming of topics contains variables explained in Table 4.

Topic for the request-response interactions

The template for the request-response interaction is:

```
90940/{Major version of SFERA}/G2B/{Company code of RU}/{Train identifier}/{Clientid}
```

The following message is published by the IM DAS-TS to this topic:

- SFERA_G2B_ReplyMessage

Topic for the event-based events

The template for the event interaction is:

```
90940/{Major version of SFERA}/event/{Company code of RU}/{Train identifier}
```

The following messages are published by the IM DAS-TS to this topic:

- SFERA_G2B_EventMessage (for example text message)
- SFERA_G2B_RequestMessage

1.5 Token format

This section contains additional specifications for the use of authorisation tokens.

The JSON web token standard is IETF RFC 7519. This standard defines a set of names for properties of a token that are used for basic token management. Table 6 lists the official RFC7519 claims that are also **mandatory** in the IRS 90940 protocol and define their meaning.

Table 6: RFC7519 claims mandatory in IRS 90940

Claim name	Meaning	90940 interpretation
iss	Issuer of token	Name/identification of token-issuing system
sub	Subject of claims	RU-specific identifier of the DAS instance. The RU is responsible for defining a unique identifier for each DAS instance within its fleet.
aud	Audience, recipient of token	Identification of IM SFERA service with which the DAS-OB wants to communicate. The value depends on the specific token issuer service and needs to be part of the trust settlement between IM and RU.
exp	Expiration Time	UNIX Epoch time stamp of the last moment in time the token is allowed to be used/accepted by the DAS-TS.
nbf	Not before	UNIX Epoch time stamp of the first moment in time the token is allowed to be used/accepted by DAS-TS.
iat	issued at	UNIX Epoch time stamp of the moment the token is issued
jti	token unique identifier	Token-service-dependent unique token identifier.

Some additional (application-specific) claims have been defined to accommodate the SFERA-specific authorisation model. The names of these claims are “private names” and are prefixed with an IRS 90940-specific URI to prevent collision of names. This is needed to ensure compatibility with tokens issued according to the OpenID Connect standard. All these claims have to be considered as **mandatory** except http://uic.org/90940/role.

Table 7: Application-specific claims

Claim name	Meaning	90940 interpretation
http://uic.org/90940/ru	UIC unique RU identifier	Official RU identifier for which this token is issued. This shall be an RICS Company code (see https://uic.org/rics)
http://uic.org/90940/train	Definition of train the client is authorised for	Train identifier for which authorisation is given. This train identifier is limited to those within the scope of the RU, refer to Table 4 for the format of train identifier.
http://uic.org/90940/role	Allowed role for this DAS driver or ATO GoA2 status	“read-only” or “active”. In read-only role, the DAS is not allowed to send status reports on this train and the DAS-TS to which the DAS-OB is connected shall not accept the reports from a client with read-only role. If this claim is omitted, the DAS-TS shall assume the “read-only” role.

An example: a JWT token body in SFERA looks like the following:

```
{
"aud":                                                          "0084",
"iss":      [https://ns-service.auth0.com/](https://ns-service.auth0.com/),
"sub":                                              "1084-etd-sng2345-cab1",
"iat":                                                          1553548860,
"exp":                                                          1553552460,
"nbf":                                                          1553548860,
"jti":                                      "147-sodsojbdhgq-1651239064300",
"http://uic.org/90940/ru":                                       "1084",
"http://uic.org/90940/train":                                "1728_2019-03-25",
"http://uic.org/90940/role":                                       "active",
}
```

The token has been issued by the Auth0 token service, stating that the DAS-OB with identifier “1084-etd-sng2345-cab1” is authorised to drive NS (1084) Train 1728 on 25 March 2019 and the token is intended to be used with the ProRail (0084) DAS-TS system.

For the token header, the values applied are shown in Table 8.

Table 8: Token header values

Header attribute name	Meaning	90940 value
typ	token content type	Value is always ‘JWT’
alg	signing algorithm used	An asymmetric algorithm shall be used.
kid	signing key identifier	Optional parameter to help identify the public key used for signing the JWT

Header attribute name	Meaning	90940 value
x5t#S256	SHA-256 hash of DER encoded X.509 certificate	hash of certificate containing public key that has been used to sign the JWT

Nota bene:

- The choice of signing algorithm and the signing key identification method will change over time as encryption standards evolve. The baseline for SFERA implementation will be set after formal approval of the IRS by the UIC just before the public announcement of the IRS so that it will match the latest security standard insights.

1.6 Implementation example

This section will illustrate the mechanisms of authentication, authorisation and communication by retrieving a Journey Profile in the IM-RU setup. The description contains the 'happy flow' scenario and does not include all the error scenarios. The numbers in the sequence diagram represented in Fig. 3 reference the steps described below.

Pre-conditions:

- In this example it is assumed that a Client ID (UUID) is created on installing the software, in this case it's UUID fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9;
- The IM DAS-TS is connected to the MQTT broker as well and is subscribed to topics for receiving incoming messages, for example /90940/1/B2G/#;
- All communication goes through the public Internet (4G for example) and DAS-OB has a proper connection.

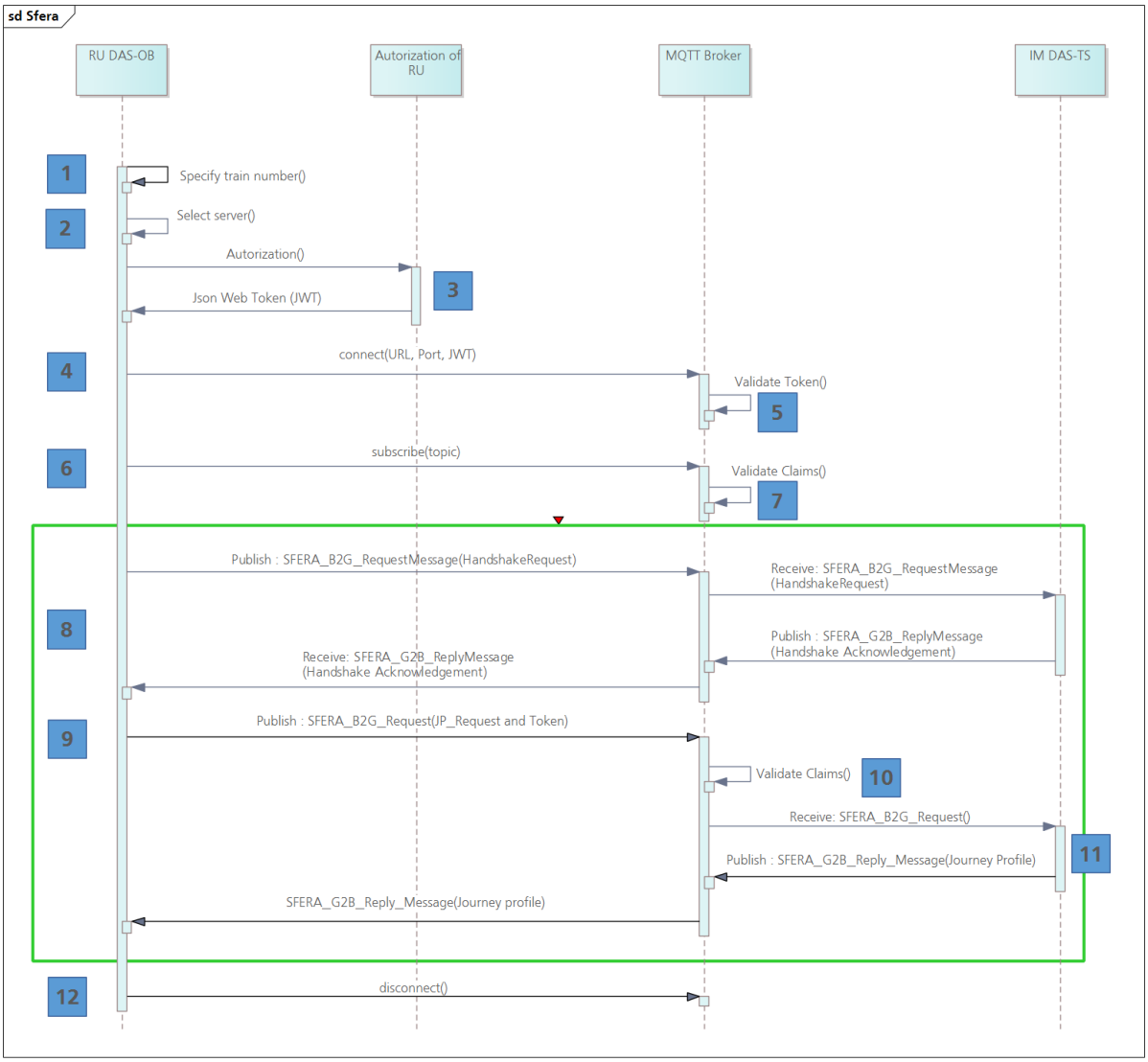

Figure 3: Sequence diagram for example implementation scenario

Steps:

1. **Specification of trip**

The user of the DAS-OB specifies for which trip the DAS is going to operate.

For example, the driver inputs the data in a device, as shown in Fig. 4:

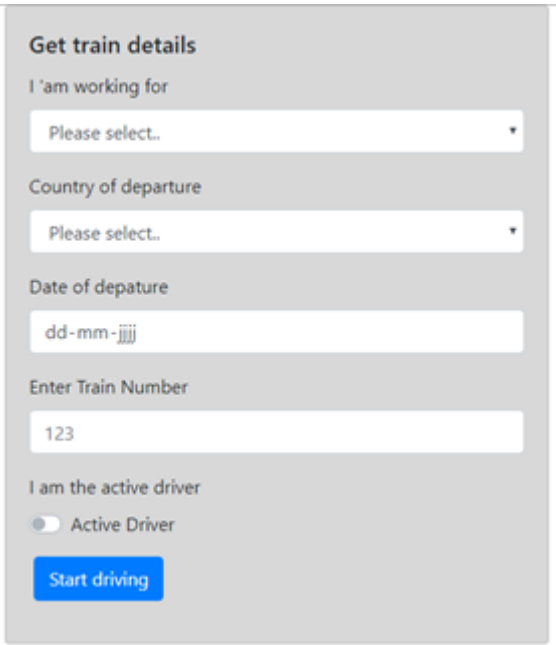

Figure 4: Example of an input mask for train details on a DAS-OB

2. **Server selection**

The DAS-OB selects the end points of the IM or the RU, depending on the setup. The DAS-OB should know the proper DNS names of relevant servers; this is done by means of a preliminary agreement.

3. **Authorisation**

The DAS-OB connects to the RU authorisation server and gets a JWT Token from the RU. This can be a different authorisation flow, depending on the configuration of the RU. The main idea is that every RU has some existing way of authentication and will want to use that. For example, this can be an OAUTH2-based flow.

It is up to the RU to handle this flow, as long as the return value to DAS-OB is a signed token conform to the SFERA standard. The token is signed asymmetrically and the key exchange is achieved by bilateral agreement (this administrative step is a separate IM-RU-specific process).

Example of token (partial):

```
"http://uic.org/90940/ru":     "1084",     //     ("allowed     for     NS")
"http://uic.org/90940/train": "512_2019-03-21", // ("allowed for Train 512
on 21 March 2019")
```

4. **DAS-OB connects to the MQTT broker with JSON web token over secure web socket connection**

After the authentication, the DAS-OB connects to the MQTT broker using a secure web socket connection (TLS 1.3) and uses the JWT token as a password. The TLS handshake is in the MQTT connect and handled in the MQTT client within the DAS-OB.

5. **Validation of claims (Authentication)**

The MQTT broker validates the signed token against the public key of the RU (which is exchanged). As described, the validation of the JWT token is not part of the MQTT standard, so there is an extra piece of software to validate the token. Common implementations are to extend MQTT with a plugin which calls an API to validate the token or extend/overwrite the authentication part of the MQTT Broker client (own implementation).

The validation steps for the connect operation are as follows:

- perform the basic signature validation with the key information in the token header
- verify that the referenced key is the one that the IM has registered for the RU indicated by the http://uic.org/90940/ru claim value
- verify that the aud and iss claim values correspond to the ones registered for this RU; preferably the aud claim should be assigned the RICS code of the IM, but if this is not possible, any other value agreed on between IM and RU can be used
- verify that the token is valid at time of processing by checking the timestamps in the exp and nbf claims

In every next step when a DAS-OB wants to subscribe or publish to a topic, the token bound to this session during connect will be verified against the requested topic name.

6. **RU DAS-OB subscribes to a topic as specified in Section 1.4.2**

For example, a DAS-OB with UUID fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9 which is going to the driver of the train with number 512 from NS (Company Code 1084) on 21 March 2019 subscribes to the following topic to receive data from the trackside:

```
90940/1/G2B/1084/2019-03-21_512/fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9
```

7. **MQTT Broker validates the claims in the JWT Token (authorisation)**

The broker validates if the DAS-OB is allowed to receive information for certain topics with the information (claims) in the JWT Token. This means the token received in Step 5. is validated against the topic name that is in the subscribe operation. The train should correspond with the values of the claims http://uic.org/90940/ru and http://uic.org/90940/train. In this case the http://uic.org/90940/ru claim should have value 1084 and the http://uic.org/90940/train claim should have value 2019-03-21_512. In that case the client is correctly authorised for the given train.

In this example, the JWT token of Step 2 corresponds to the topic of Step 6:

Token (extract)

```
"http://uic.org/90940/ru":     "1084",     //     ("allowed     for     NS")
"http://uic.org/90940/train": "512_2019-03-21", // ("allowed for Train 512
on 21 March 2019")
```

Topic

`90940/1/G2B/1084/2019-03-21_512/fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9`

8. **Feature exchange**

DAS-OB publishes a handshake message to the topic to exchange features and set status of the driver.

9. **Request for a Journey Profile**

As specified in Use Case JP2 (see Appendix A), the RU DAS-OB publishes a SFERA_B2G_RequestMessage to the topic, for requesting a JP.

For example a DAS-OB with UUID fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9 which is going to the driver of the train with number 512 from NS (Company Code 1084) on 21 March 2019 publishes a SFERA_B2G_RequestMessage on the topic:

`90940/1/B2G/1084/2019-03-21_512/fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9`

10. **DAS-OB is allowed to request information (authorisation)**

The MQTT broker validates if the DAS-OB is allowed to receive information for this specific topic with the information (claims) in the JWT Token. This means the token received in Step 5. is validated against the topic name in the publish operation in the same way as in Step 7. Additionally, the train identification in the JourneyProfile request (operational number) shall also match the values of http://uic.org/90940/ru and http://uic.org/90940/train claims.

11. **DAS-TS sends Journey Profile back from IM DAS-TS to DAS-OB**

The DAS-TS publishes a SFERA_G2B_ReplyMessage to the MQTT broker and the MQTT broker will send it to the DAS-OB.

The trackside will publish the reply to the topic which has been subscribed in Step 6, in this case:

`90940/1/G2B/1084/2019-03-21_512/fa6e0e68-63b6-4b13-8e9c-74e9a66dd1f9`

Any other request can be sent now.

12. **Disconnect** At the end of the journey, the DAS-OB disconnects from the MQTT Broker.

2 Common Interface: exchanging SFERA messages between IM DAS-TS and RU DAS-TS

2.1 Presentation of the Common Interface protocol

The CI is a communication tool that exchanges messages in XML format. As stated in the TAF/TAP TSI, the CI needs to support a minimum number of functionalities to be able to exchange such messages, as mentioned in Annex D.2 - Appendix E of TAF TSI:

"Consequently, chapter 4.2.12.6 of the TAF TSI document that the Common Interface is mandatory for each actor in order to join the TAF TSI rail interoperability community and must have the following capabilities: • message formatting of outgoing messages according to the metadata, • signing and encryption of outgoing messages, • addressing of the outgoing messages, • authenticity verification of the incoming messages, • decryption of incoming messages, • conformity checks of incoming messages according to metadata, • handling the single common access to various databases."

The CI is positioned at the boundary of a railway company's IT system, and will be used to establish a standardised interface with the CI of other railway companies. The outside interface of the CI shall be implemented according to the standard specifications established in the TAF TSI. The inside interface is conceived to be able to address the different systems that need to exchange data with the outside. The choice of the protocol(s) established is up to individual implementations.

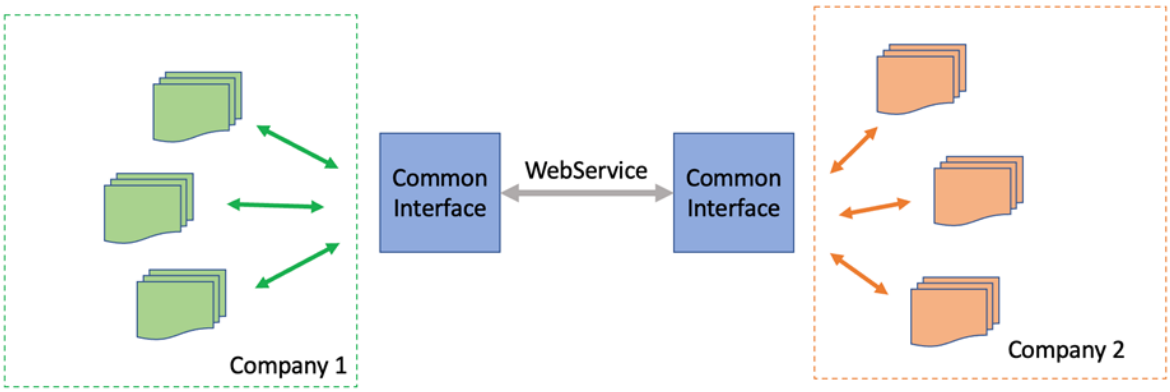

Figure 5: Position of the Common Interface in inter-company communication

2.2 Motivation for the Common Interface

This paragraph gives the motivation for introducing the CI as the communication layer for SFERA exchanges on the IM DAS-TS / RU DAS-TS link.

The main arguments for choosing the CI in the SFERA protocol are the following:

- *Existing interoperability:* the TAF/TAP TSI has been defined at a European level to define a standard communication protocol for inter-server exchanges for all actors in the railway logistics chain. This includes IM-RU communication between planning/regulation systems and RU IT systems. All European IMs and the majority of RUs have deployed a CI and exchange operational data through them.

- *Standard protocols for server-to-server communication:* the CI uses standard protocols implemented in a specific way adapted to server-server exchange of XML messages. This is the use case for IM DAS-TS / RU DAS-TS communication through SFERA.
- *Long-term solution and mutualisation potential*: the TAF/TAP TSI is defined at a European level and is legally applicable to EU-Member railway companies. This framework (legal and technical level) has been designed with long-term stability in mind. IMs and RUs that have implemented the CI will be able to reuse this investment for the C-DAS use case.
- *Platform-independent and vendor-independent*: the CI uses standard technical solutions (https webservices, SSL/TLS certificates…), which are widely used and freely implementable by any company.
- *"Off-the-shelf" solutions:* CI products exist on the European market that can be licensed by companies wishing to accelerate the deployment of a C-DAS solution.

2.3 Integration between SFERA messages and the Common Interface

The SFERA messages are designed to be exchanged through CIs with a very limited impact compared with standard TAF/TAP TSI implementations. Two main aspects have been identified:

- The use of the SFERA header shall adhere to CI specifications;
- The SFERA/C-DAS use case may introduce the need for companies to implement multiple CIs.

2.3.1 Using the header

To exchange SFERA messages through the CI, the use of some elements of the SFERA message header will need to follow certain rules. Fig. 6 identifies the elements impacted by these rules.

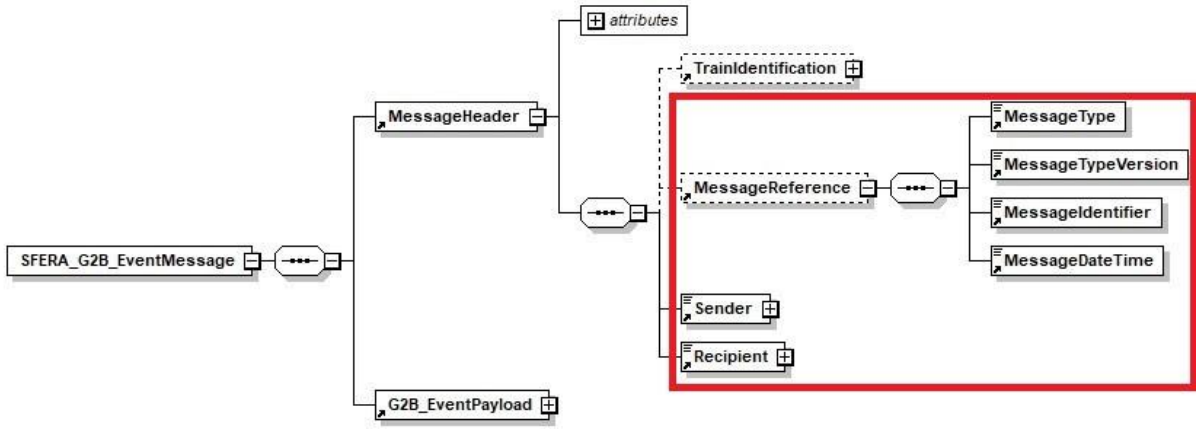

Figure 6: The SFERA MessageHeader elements relevant to CI communication

The constraints for these elements are presented in Table 9.

Table 9: Constraints for MessageHeader elements relevant to CI communication

Header field	Secondary header field	Constraint
<MessageReference>		
	<MessageType>	The messageType code is used by the CI to identify the right XSD structure for the message that is exchanged. The standard values to be used have been included in the annotation of the 6 SFERA messages.
	<MessageTypeVersion>	The MessageTypeVersion is used by the CI in conjunction with the MessageType to identify the right XSD structure for the message that is exchanged. Consistency is advised with the “version” field at the top of the SFERA XSD, even though some CI implementations do not require this.
	<MessageIdentifier>	This information is generated by the CI before sending the message to the recipient. If the CI and the DAS-TS are separate components, the emitting DAS-TS shall not generate this identifier.
	<MessageDateTime>	This information is generated by the CI before sending the message to the recipient. If the CI and the DAS-TS are separate components, the emitting DAS-TS shall not generate this element.
<\MessageReference>		
<Sender>		This information is the Company Code of the company that is sending the

Header field	Secondary header field	Constraint
		message. It is used by the CI to route the messages.
	<CI_InstanceNumber>	This element helps the CI to route messages. It is only to be used in situations where a company has implemented multiple CIs. More details on this use case are presented in Section 2.3.3.
<\Sender>		
<Recipient>		This information is the company code of the company at which the message is aimed. It is used by the CI to route the messages. This element can be filled directly by the CI. It may be filled using TrainIdentification, in which case, the element TrainIdentification shall be specified.
	<CI_InstanceNumber>	This element helps the CI to route messages. It is only to be used in situations where a company has implemented multiple CIs. More details on this use case are presented in Section 2.3.3.
<\Recipient>		

The company code that is used in the Recipient and Sender elements is the RICS company code for an IM or RU. For example, the CI (and possibly the DAS-TS associated with it) could be provided by a third party. The CI shall use the RU's RICS code. Also, if this service is provided for multiple RUs by the DAS supplier, the IMs shall address distinct CIs using the respective RICS codes of the different RUs.

Examples of SFERA Messages instantiated for communication through a CI (with single or multiple instances, see next section) are available in Appendix F.

2.3.2 RU DAS-TS and IM DAS-TS responsibilities

When using the CI in the IM-RU setup, the IM and RU exchange data regard the “train” object only. The RU will not expose to the IM details regarding the different devices used by different users.

Regarding authorisation, this means that the responsibilities are attributed as follows: - the RU shall handle the authorisation of the different devices on the RU DAS-OB/RU DAS-TS link according to the authorisation layers defined in the application of the MQTT + JWT protocol for SFERA; - the CI establishes a link with an authorisation that is established at a company level only. Once the CI connection is established, it is implicitly authorised to share information regarding C-DAS for trains for which it has a “Responsible RU” (RRU) role. The RRU is responsible for making sure that the data transferred from a device to the IM respects the rules established for SFERA (e.g. respecting the status reports that shall only be sent from the “active” device). The IM DAS-TS will only be able to implement a control loop to make sure that the information sent and requested by the RU concerns paths for which it has the RRU role.

2.3.3 Using multiple Common Interface instances

In the general use case, railway companies implement a unique CI to exchange data with other companies. When this company implements a DAS-TS inside its IT eco-system, the connection can be established with this preexisting CI, as represented in Fig. 7.

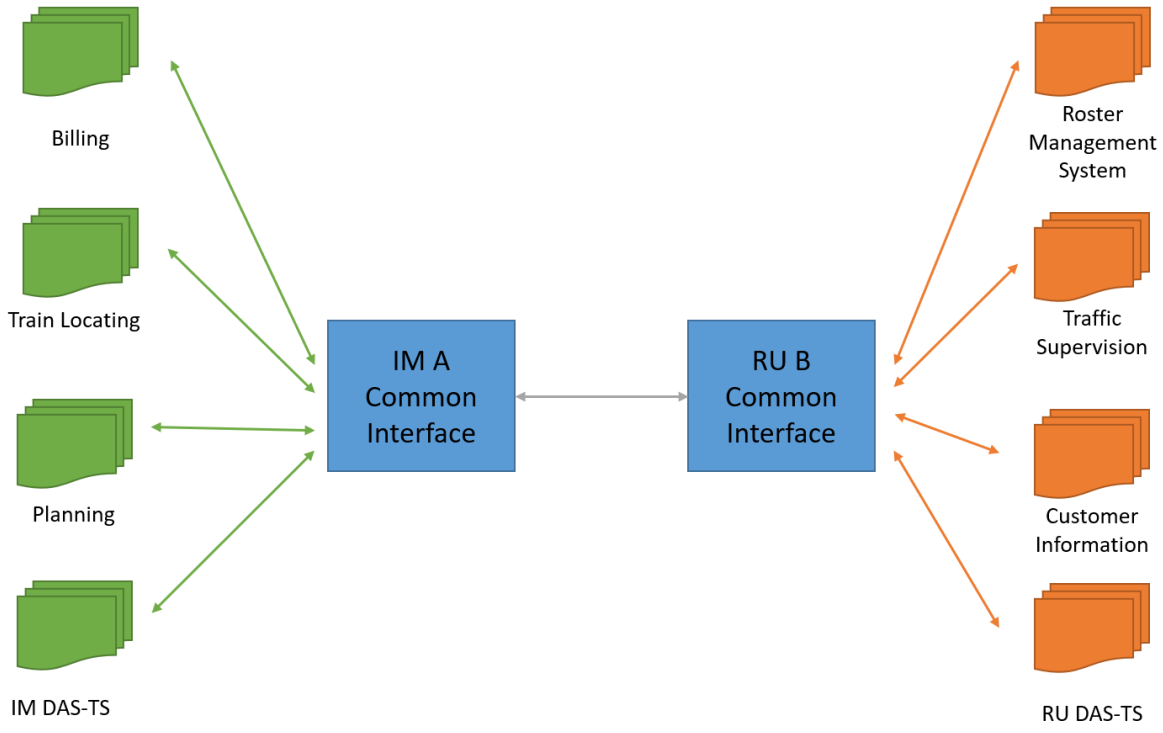

Figure 7: DAS-TS in a company with a unique CI

However, it is foreseen that some C-DAS suppliers will provide the RU DAS components in a specific environment (e.g. provided as a SaaS solution). This means that two CIs will coexist using the same RICS company code. In this situation, the company shall attribute distinctive instance identifiers to both CIs. These instance identifiers will then be communicated to all other companies which establish communications with either one of the CIs. Fig. 8 gives an example of this configuration.

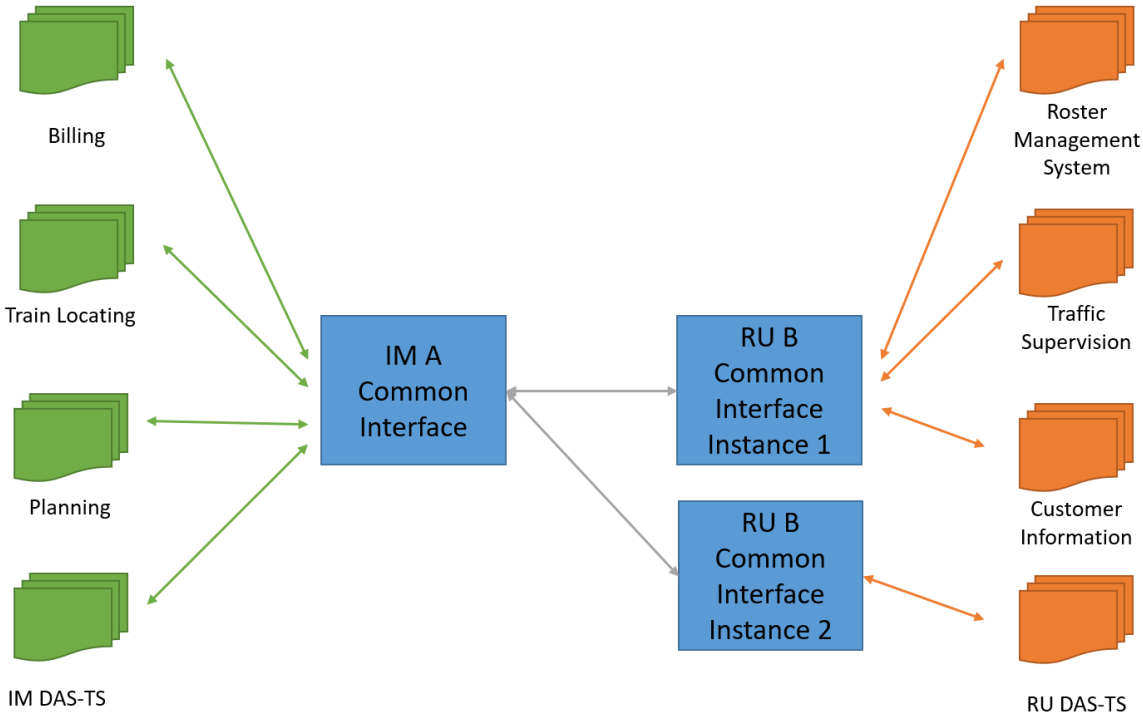

Figure 8: DAS-TS in a company with multiple CIs