



Moving Europe towards a sustainable and
safe railway system without frontiers.

Silvia De Rocchi
Direttore affari regolatori e istituzionali
Captrain Italia SRL
Viale Milanofiori, Strada 3, Palazzo B10
20057, Assago (MI)
Italy

Valenciennes, 4/7/2024

**IRAIL Project Compliance Test – Captrain Italia SRL: CEF Grant Agreement INEA/CEF/TRAN/M2018/
1777054 – Milestone 05, 13, 14, 15, 16**

Dear Silvia De Rocchi,

I confirm and certify herewith that the following TAF TSI functionalities implemented by the company Captrain Italia SRL (Italy), within the project “Captrain Italia SRL: CEF Grant Agreement INEA/CEF/TRAN/M2018/ 1777054 – Milestone 05, 13, 14, 15, 16”, is soft compliant with the appropriate provisions of chapter “4.2.9.2. The Rolling Stock Reference Databases”, “4.2.11.5. Common Interface” and “4.2.10.2. Wagon and Intermodal unit Operational Database” of the [TAF-TSI Regulation \(EU\) 2021/541](#) and its [Technical Documents](#) (XSD version 3.4.0) for the message quoted below. In the messages provided by Captrain Italia SRL it is furthermore demonstrated that the used Reference Files for locations and companies and the Unique Train Identifiers are compliant with above [TAF-TSI Regulation \(EU\) 2021/541](#) and its [Technical Documents](#).

In this milestone the company Captrain Italia SRL had to implement the Rolling Stock Reference Database function. To enable the ERA verification, Captrain Italia SRL has sent on 07/03/2023 to ERA TAF TSI compliant Rolling Stock Reference Database message and a description of its Common Interface implemented and used in operation in its own IT system (see Annex). Above *soft* compliance is based on the fact that above message carries all mandatory elements from the TAF TSI XSD (see chapter 6.2.1 of the TAF TSI).

Mickael VARGA
European Union Agency for Railways

Operational Data Unit
Project Officer
Email: mickael.varga@era.europa.eu**Annex – Rolling Stock Reference Database Message**

```

<?xml version="1.0" encoding="utf-8"?>
<RollingStockDatasetMessage xmlns="http://www.era.europa.eu/schemes/TAFTSI/3.4"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.era.europa.eu/schemes/TAFTSI/3.4
file:///C:/Users/vargami/OneDrive%20-
%20European%20Union%20Agency%20for%20Railways%20(ERA)/Desktop/Telematics/Compliance%2
0mapping/IRAIL/Captrain%20Italy/XSD%203.4.0/taf_cat_complete.xsd">
  <MessageHeader>
    <MessageReference>
      <MessageType>6003</MessageType>
      <MessageTypeVersion>3.4.0.0</MessageTypeVersion>
      <MessageIdentifier>6357e00e-4f0b-479c-93b8-
990e3fa9c9b6</MessageIdentifier>
      <MessageDateTime>2024-03-06T16:19:22</MessageDateTime>
    </MessageReference>
    <SenderReference>XXXXX</SenderReference>
    <Sender>2287</Sender>
    <MessageDateTimeCreated>2024-03-06T16:19:22</MessageDateTimeCreated>
    <Recipient>9876</Recipient>
  </MessageHeader>
  <RollingStockDataset>
    <AdministrativeDataSet>
      <WagonNumberFreight>111111111111</WagonNumberFreight>
      <RegistrationCountry>CY</RegistrationCountry>
      <DatePutIntoService>2020-04-27</DatePutIntoService>
      <AuthorisationValidUntil>2022-04-05</AuthorisationValidUntil>
      <SuspensionOfAuthorisation>false</SuspensionOfAuthorisation>
      <KeeperShortNameVKM>1111111</KeeperShortNameVKM>
      <ECM>XXXYYY</ECM>
      <ECMCertificate>
        <EINNumber>
          <CountryCodeISO>11</CountryCodeISO>
          <TypeDocumentEIN>01</TypeDocumentEIN>
        </EINNumber>
        <CounterAccreditedRecognizedBody>01</CounterAccreditedRecognizedBody>
        <EINYear>99</EINYear>
        <EINCounter>01</EINCounter>
      </EINNumber>
      <ECMCertificateValidFrom>2020-04-
27</ECMCertificateValidFrom>
      <ECMCertificateValidTo>2022-04-05</ECMCertificateValidTo>
    </AdministrativeDataSet>
    <CoversTankWagonsForDangerousGoods>false</CoversTankWagonsForDangerousGoods>
    <CoversNonTankWagonsForDangerousGoods>false</CoversNonTankWagonsForDangerousGo
ods>

```

```
        <ECMCertificateSuspended>false</ECMCertificateSuspended>
    </ECMCertificate>
    <InteropCapability>01</InteropCapability>
    <GCUWagon>true</GCUWagon>
</AdministrativeDataSet>
<DesignDataSet>
    <LetterMarking>1111111</LetterMarking>
    <WagonNumberOfAxles>4</WagonNumberOfAxles>
    <InnerWheelbase>2500</InnerWheelbase>
    <MinCurveRadius>35</MinCurveRadius>
    <WagonWeightEmpty>99999</WagonWeightEmpty>
    <LengthOverBuffers>17900</LengthOverBuffers>
    <MaxAxleWeight>48</MaxAxleWeight>
    <MaxDesignSpeed>110</MaxDesignSpeed>
    <AirBrake>
        <NumberOfBrakes>2</NumberOfBrakes>
        <AirBrakeType>3</AirBrakeType>
    </AirBrake>
    <BrakingPowerVariationDevice>1</BrakingPowerVariationDevice>
        <AirBrakedMass>24</AirBrakedMass>
    </BrakingPowerVariationDevice>
    <BrakeSpecialCharacteristics>1</BrakeSpecialCharacteristics>
        </AirBrake>
        <HandBrake>
            <HandBrakeType>0</HandBrakeType>
            <ParkingBrakeForce>24000</ParkingBrakeForce>
        </HandBrake>
        <LoadingCapacity>66600</LoadingCapacity>
        <MaxGrossWeight>58600</MaxGrossWeight>
        <DateLastOverhaul>2020-04-27</DateLastOverhaul>
        <OverhaulValidityPeriod>1</OverhaulValidityPeriod>
        <PermittedTolerance>4</PermittedTolerance>
    </DesignDataSet>
</RollingStockDataset>
</RollingStockDatasetMessage>
```


Movements:

operation_id	source_car_list_id	destination_car_list_id	created_at	updated_at	source_car_list_index	destination_car_list_index	typology	relative_movement_id	destination_railroad_update_token	operation_addonable_id	discarded_at	performed_at
10535	311	2021-10-12T10:13:12.121Z	2021-10-12T10:13:12.121Z				7					2021-10-13T10:09:00.000Z
10535	311	2021-10-12T10:17:33.045Z	2021-10-12T10:17:33.045Z				10		26230			2021-10-12T10:17:33.044Z
25557		2022-03-02T11:06:22.928Z	2022-03-02T11:06:22.928Z				6					
10535	311	2021-10-12T10:24:00.647Z	2021-10-12T10:24:00.647Z									
10535	311	2021-10-12T10:23:53.443Z	2021-10-12T10:23:53.443Z				11					2021-10-12T10:23:53.442Z
25557	580	2022-03-02T11:06:22.910Z	2022-03-02T11:06:23.084Z		5	5	3	542e49229d62e2d2ae13801a4657b596				
10535		2021-10-12T10:38:00.301Z	2021-10-12T10:38:00.301Z				6					
10461	313	2021-10-12T10:43:55.681Z	2021-10-12T10:43:55.681Z			1	7					2021-10-12T10:43:00.000Z
28291	1024	2022-03-30T13:05:25.661Z	2022-03-30T13:05:25.661Z				2					
22677		2022-03-02T11:40:40.532Z	2022-03-02T11:40:40.532Z				6					
22677	315	2022-03-02T11:40:40.719Z	2022-03-02T11:40:40.796Z		4	4		542e49229d62e2d2ae13801a4657b596				
26782	594	2022-03-03T08:25:01.775Z	2022-03-03T08:25:01.775Z				10		99632			2022-03-03T08:25:01.775Z
26122	597	2022-03-03T09:31:39.209Z	2022-03-03T09:31:39.209Z		2				97113			
10784		2021-10-12T11:10:04.037Z	2021-10-12T11:10:04.037Z				6					
10784		2021-10-12T11:10:35.402Z	2021-10-12T11:10:35.402Z				5					
10784		2021-10-12T11:11:48.490Z	2021-10-12T11:11:48.490Z				6					
10784	312	2021-10-12T11:11:48.460Z	2021-10-12T11:11:48.618Z				3	ca82b7facd0934a3a6bc3a52774e02b1				
10546	313	2021-10-12T11:15:12.362Z	2021-10-12T11:15:12.524Z		1	1	3	ce3df58f64001f40c38f4ebd561e434d				
10546		2021-10-12T11:15:12.389Z	2021-10-12T11:24:04.653Z				6			2021-10-12T11:24:04.652Z		
10546		2021-10-12T11:24:04.674Z	2021-10-12T11:24:04.674Z				5					
11038	315	2021-10-12T16:22:55.948Z	2021-10-12T16:22:55.948Z				7					2021-10-12T16:22:00.000Z
11125	316	2021-10-19T19:33:06.866Z	2021-10-19T19:33:06.866Z				7					2021-10-18T23:10:00.000Z
11125	316	2021-10-19T19:34:04.970Z	2021-10-19T19:34:04.970Z				8					2021-10-19T19:34:04.969Z
11125	316	2021-10-19T19:34:44.232Z	2021-10-19T19:34:44.232Z				9					2021-10-19T19:34:44.232Z
25790	606	2022-03-04T15:16:12.916Z	2022-03-04T15:16:12.916Z		2	7						2022-03-04T15:15:00.000Z
11283	317	2021-10-27T14:00:56.183Z	2021-10-27T14:00:56.183Z				11		30256			2021-10-27T14:00:56.182Z
11283	317	2021-10-27T14:01:09.127Z	2021-10-27T14:01:09.127Z						30256			
11283	327	2021-10-27T14:03:26.675Z	2021-10-27T14:03:26.675Z						30255			

Annex – Common Interface and Reference files

The Common Interface (CI) is a TAF/TAP-TSI message exchange peer to peer application. All application modules of the SoftRail suite share a CI implementation developed by Binary System based on the TAF-TSI specifications. Such implementation is developed in Node Js, which is a JavaScript runtime built on Chrome V8 JavaScript engine, taking advantage of its asynchronous nature to better handle traffic load and I/O from database. V8 is Google open-source high-performance JavaScript and Web Assembly engine, written in C++. It is used in Chrome and in Node.js, among others.

Design and Message Flow

The Binary System Common Interface is implemented according to an event-driven design; after a message is received or requested to be sent, a series of events is triggered, representing the various sub-steps of the elaboration, i.e. creating a sort of set of queues of events which can be consumed in the right order and in parallel with others.

Since the standard operates with the SOAP protocol, the format of the messages must be changed from XML to JSON, this mainly for two reasons:

- Node JS is blazingly fast with JSON.
- Binary System ecosystem predominantly uses REST interfaces.

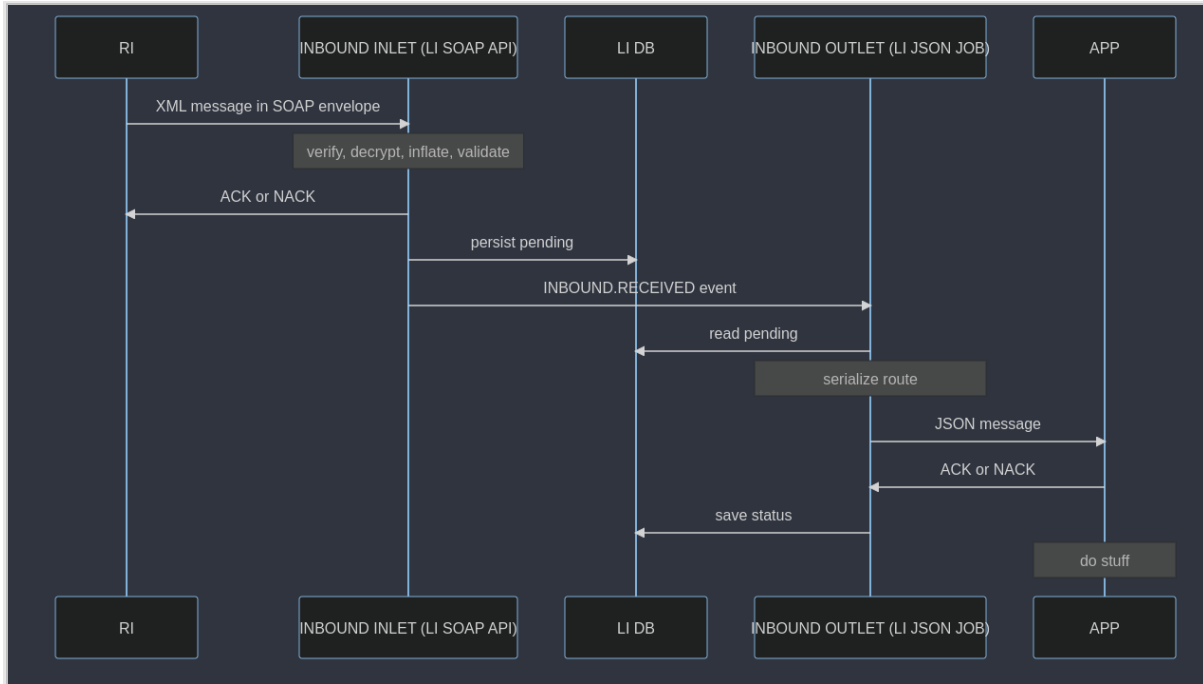
Messages received from other Common Interfaces are first validated against the appropriate XSD; if validation succeeds, they are converted into JSON and forwarded to SoftRail modules.

For outbound messages, the JSON received from SoftRail modules is converted into XML and then

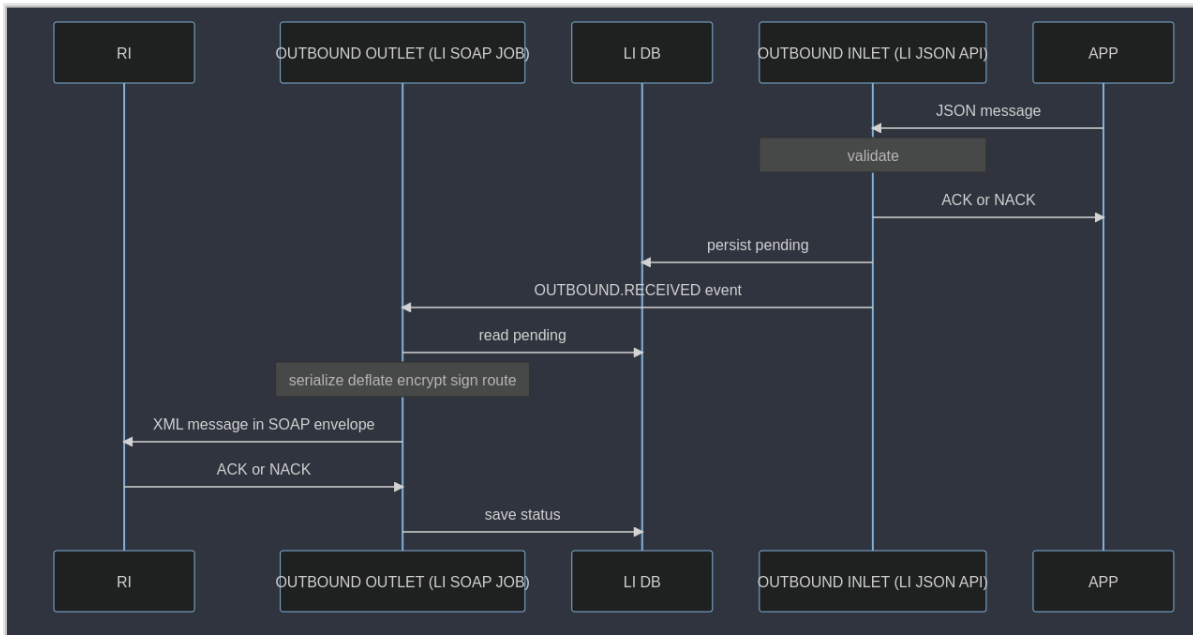
validated against the appropriate XSD. Through this validation process, only properly-formatted messages are sent to other Common Interfaces.

The diagrams below illustrate the two inbound/outbound flows:

- RI = Remote Common Interfaces.
- LI = Local Common Interface (Binary System implementation).
- APP = Applications from the SoftRail suite (Binary System modules).



Common Interface - inbound messages

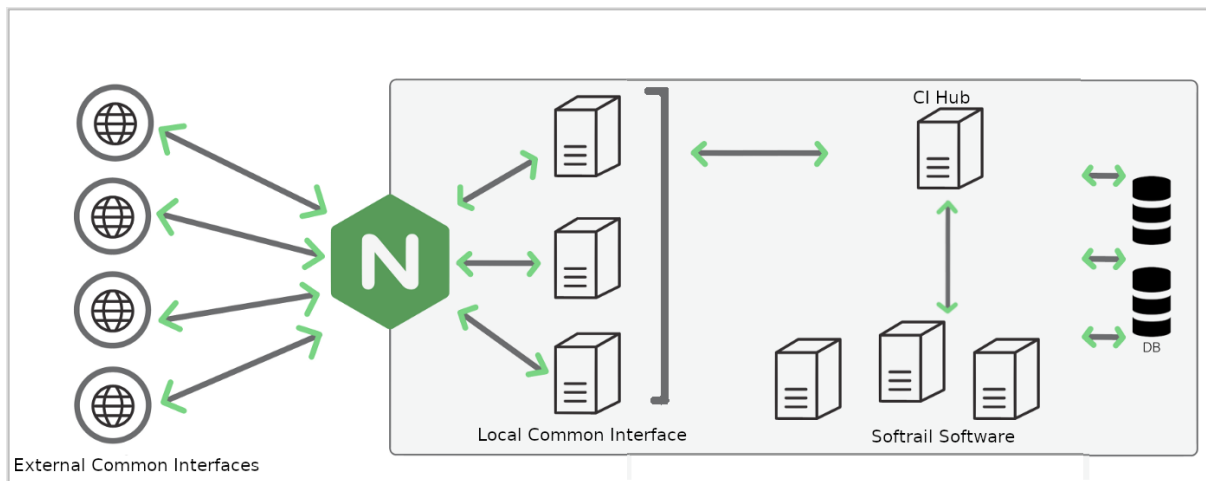


Common Interface - outbound messages

Since the SoftRail suite is a complex software environment including several microservices, it has been decided, as architectural choice, to create an intermediary application which works as a sort of proxy, to leave the CI implementation lean. This intermediary application is called “CI Hub”, which acts, as the name suggests, as a hub for all the requests that go from the CI to the SoftRail applications or vice versa. The Common Interface communicates (in JSON) with the CI Hub, which elaborates and produces the data by retrieving the information from the involved modules; by doing so, from the perspective of the Common Interface, the information from the modules is just a payload to be sent, and when in reception, the messages do not have to be elaborated, but just passed to the CI Hub, which will handle the content.

The separation of transportation and content management allows more precise integrations with fewer problems, as new developments can be added without modifying other information flows.

The Common Interface is not directly exposed to the internet: a Nginx reverse proxy is placed right in front of the node server. The choice lies in the fact that Nginx is a very run-in web server (and reverse proxy) which can manage high traffic loads, can load balance requests, and can directly manage https certificates. This way, every part of the flow handles only what is meant to be handled and nothing else.



Common Interface - isolation through reverse proxy.

Database

Both the Common Interface and CI Hub use a Postgres Database as data storage. PostgreSQL is a powerful, open-source, object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

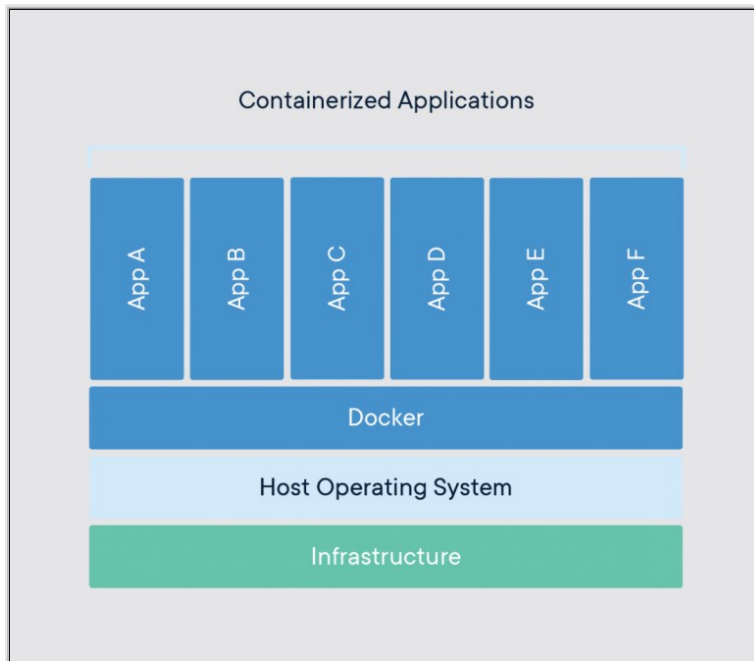
Postgres is a powerful union between a relational database, with tables, columns, and rows, and a “No-SQL” database: it gives easy access to JSON data, while maintaining its relational architecture, via its custom type JSONB.

PostgreSQL was chosen due to the high quantity of logs in both JSON and XML formats and the high performance which has in handling this type of data.

Deployment Architecture

The Common Interface is normally deployed in cloud, although in some special cases it can be deployed on-premises. The deployment architecture is based on the container technology, more specifically upon Docker. A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another. A Docker container

image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime, and in the case of Docker containers images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences in instances.



Containerized deployment scheme

On Linux-based machines, Docker containers run almost as if they were native processes, thanks to how the new kernels implement the containerization. It is planned to migrate, where possible, the Common Interface and the CI Hub on Kubernetes by the end of 2024; Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management, further improving availability and scalability.

Internal Architecture

Binary System Common Interface implements an arc-based message routing. This system routes messages between local nodes and remote nodes (which are isolated entities with specific characteristics) via arcs, which can be imagined as the connections in a graph, so that it is possible to decide whether two nodes are connected or not.

The input messages have an additional routing “rule”, meaning that they are routed through a list of endpoints specified within the local nodes metadata, per local node (which usually point to the CI Hub); this approach enables the dynamic change of the configuration.

For every node, other information can be specified; for instance:

- If communication has mutual authentication.
- If messages are to be compressed, signed, or encrypted through a specified public/private key.
- The standard to be used

All this information can be changed at run time.

Annex – Unique Train Identifier

```
<TrainOperationalIdentification>
  <TransportOperationalIdentifiers>
    <ObjectType>TR</ObjectType>
    <Company>2287</Company>
    <Core>048202174626</Core>
    <Variant>00</Variant>
    <TimetableYear>2023</TimetableYear>
    <StartDate>2023-02-06</StartDate>
  </TransportOperationalIdentifiers>
  <RelatedTransportOperationalIdentifiers>
    <ObjectType>TR</ObjectType>
    <Company>2287</Company>
    <Core>048202174626</Core>
    <Variant>00</Variant>
    <TimetableYear>2023</TimetableYear>
    <StartDate>2023-02-06</StartDate>
  </RelatedTransportOperationalIdentifiers>
</TrainOperationalIdentification>
```