
EASYLOG PROJECT MICROSERVICES SPECIFICHE TECNICHE

v 1.0 - 28/04/21

Sommario

1	INTRODUZIONE AL DOCUMENTO	3
1.1	MODIFICHE E AGGIORNAMENTI	3
2	PROCESSO DI INTERCHANGE NOTARIZZATO	4
2.1	DESCRIZIONE DEL PROCESSO	4
2.2	DEFINIZIONE DEL PROCESSO	6
2.3	DEFINIZIONE ENDPOINT	8
2.4	DEFINIZIONE OGGETTI	10
3	PROCESSO APPLICAZIONE DI PESATURA	16
3.1	DESCRIZIONE DEL PROCESSO	16
3.2	MODULI	17
3.3	SERVIZIO DI PESATURA	18
3.4	CONTROLLORE PESA	22
3.5	GESTORE UTENTI	22
3.6	FILE SERVICE	26
4	INTERFACCIA UTENTE APPLICAZIONE DI PESATURA	28
4.1	INTRODUZIONE	28
4.2	WORKFLOW	28
4.3	VISTA RICERCA PRENOTAZIONE	29
4.4	VISTA NUOVA PRENOTAZIONE	30
4.5	VISTA SELEZIONE OPERAZIONE	31
4.6	VISTA OPERAZIONE DI PESA	32
4.7	VISTA INSERIMENTO MANUALE PESO	33
4.8	CONTROLLORE PESA	34
4.9	GESTIONE UTENTI	35
4.10	FILE SERVICE	38
5	ANTICIPO DOCUMENTALE	40
5.1	INTRODUZIONE	40
5.2	SERVIZI	40

1 INTRODUZIONE AL DOCUMENTO

Questo documento fornisce le specifiche tecniche relative ai tre microservizi parte del Progetto Easylog.

1.1 MODIFICHE E AGGIORNAMENTI

Rev. Nr.	Date	Description
1.0	28/04/2021	First revision

2 PROCESSO DI INTERCHANGE NOTARIZZATO

2.1 DESCRIZIONE DEL PROCESSO

Interchange Notarizzato è una delle funzionalità offerte dal servizio di Notarizzazione.

È ottenuto apponendo le firme digitali del terminal e del trasportatore al documento PDF creato dal terminal.

Le firme digitali sono apposte al documento attraverso due chiamate separate al sistema di Firma Digitale.

Il Pre-Interchange document (anticipo documentale) è una lista di operazioni si prevede vengano svolte dal trasportatore al terminal.

Il documento è normalmente creato quando il trasportatore entra al terminal (gate-in).

Il documento di interscambio è un rapporto delle operazioni eseguite dal trasportatore per una data missione nel terminal.

Il documento di interscambio viene solitamente creato quando l'autotrasportatore esce dal terminal (gate-out).

La gestione dei documenti nel servizio notarile è tracciata da un'entità di processo.

Il processo è identificato in modo univoco da un riferimento alla firma.

Il riferimento alla firma viene utilizzato per associare il processo, il documento di pre-scambio e il documento di interscambio. Inoltre tiene traccia degli attori coinvolti: il firmatario (il terminal) e il controfirmatario (il trasportatore).

Il riferimento della firma è costituito da:

- riferimento del documento
- data del documento
- codice firmatario
- codice controfirmatario

Il processo ha uno *status* ed un *type* firma.

Per l'interchange notarizzato type firma può essere:

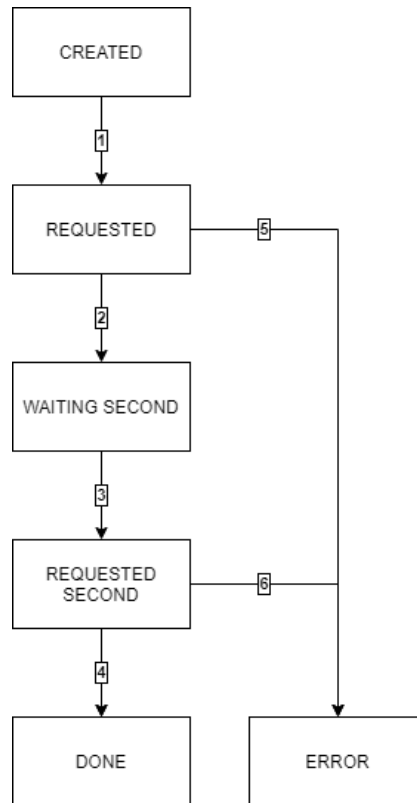
- DOUBLE, Doppio, quando il terminal prima produce l'interchange and poi il trasportatore accetta di firmare.
- PRE-ARMED DOUBLE, quando il trasportatore accetta di firmare automaticamente l'interchange se i dati sono gli stessi del pre-interchange.

Lo stato del processo può essere:

- CREATED
- REQUESTED
- WAITING SECOND
- REQUESTED SECOND
- DONE
- ERROR

I passaggi di stato del processo sono i seguenti (vedi anche l'immagine sottostante):

- 1) La richiesta di firma dell'interchange per il terminal è stata inviata al servizio di firma.
- 2) L'interchange è stato firmato dal terminal.
- 3) La richiesta di firma dell'interchante per il trasportatore è stata inviata al servizio di firma.
- 4) L'interchange è stato firmato sia dal terminal che dal trasportatore.



Il *firmatario* (signer) è una persona fisica per conto della quale il servizio appone la firma sul documento. I firmatari devono essere registrati prima di poter utilizzare il servizio. I firmatari possono nominare diversi delegati per richiedere la firma per loro conto.

Un *delegato* (delegate) è un soggetto con un account utente autorizzato da un firmatario a richiedere la firma per conto del firmatario stesso.

Il file PDF da firmare deve includere nel testo due ancore univoche per il posizionamento delle firme. Queste possono essere ad esempio "1. _____" e "2. _____".

2.2 DEFINIZIONE DEL PROCESSO

Il processo inizia quando il pre-scambio viene creato dal terminale dopo il gate-in. Il Terminale invia i dati di pre-intercambio al Servizio Notarile e questo li inoltra all'Autotrasportatore.

Quindi, a seconda delle impostazioni predefinite, il sistema Hauler può pre-attivare la firma o attendere la creazione dell'interscambio.

Se i dati di interscambio sono diversi dai dati di pre-intercambio, viene inviata una notifica all'autotrasportatore e deve essere inviata una richiesta di firma esplicita.

Quando sia il Terminal che l'Autotrasportatore hanno firmato lo scambio, una copia viene inviata alle parti.

Nei diagrammi di sequenza sottostanti sono rappresentati 4 casi a seconda dell'uso della firma pre-triggerata e della presenza di modifiche tra il pre-intercambio e i dati di interscambio.

	Pre-triggered signature	Variations in the interchange
<i>Case A</i>	YES	NO
<i>Case B</i>	NO	NO
<i>Case C</i>	YES	YES
<i>Case D</i>	NO	YES

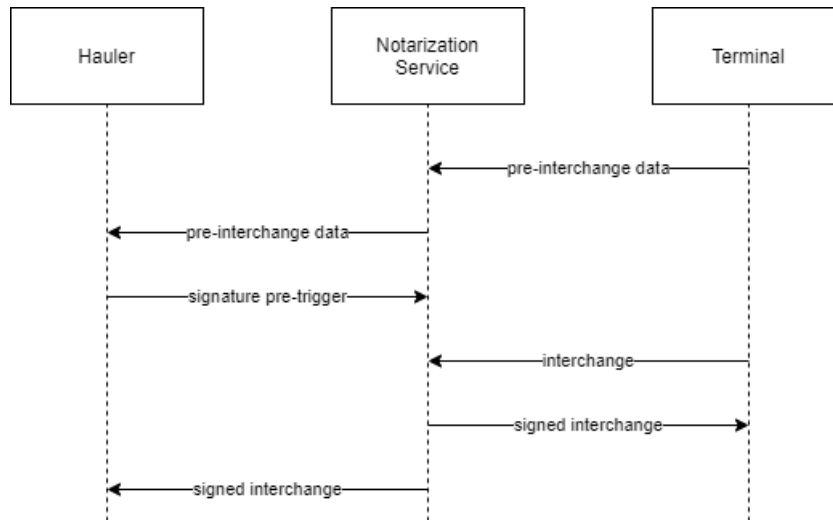


Figure 1 - Case A

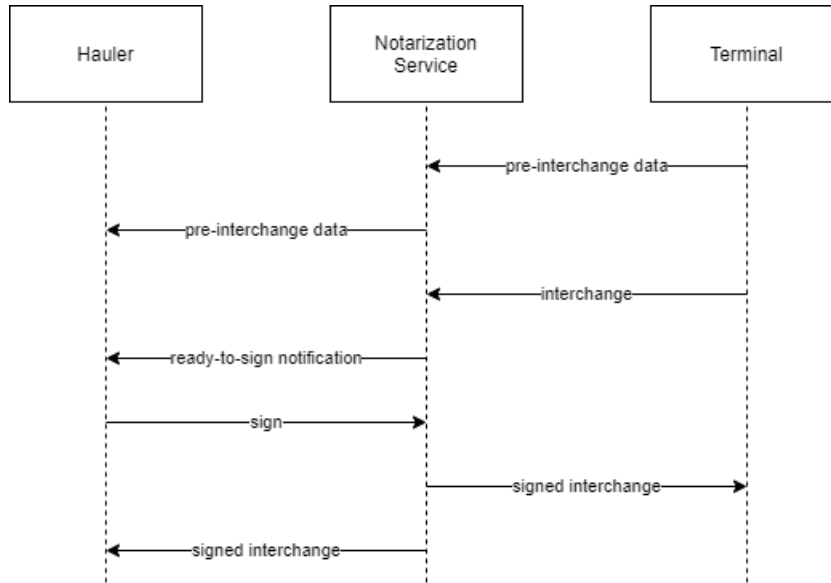


Figure 2 - Case B

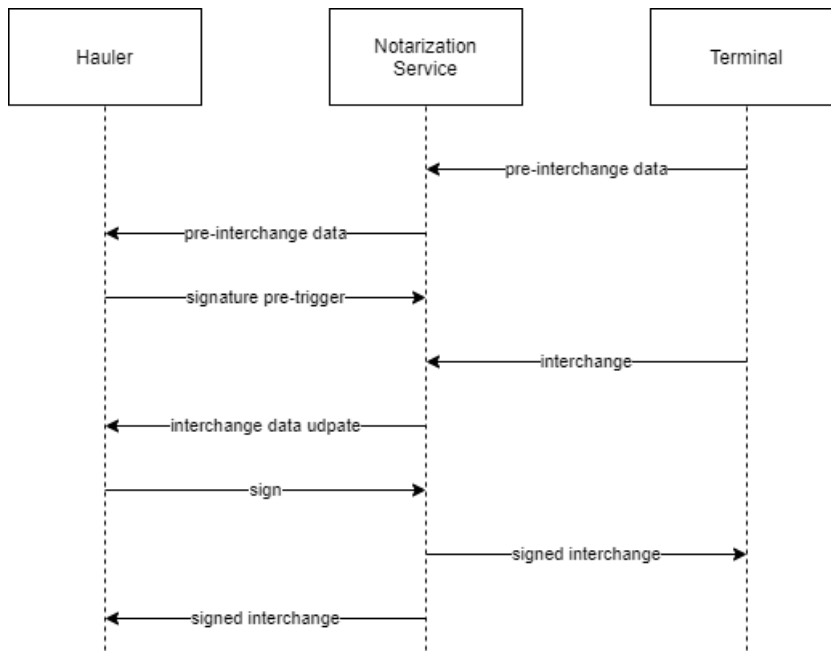


Figure 3 - Case C

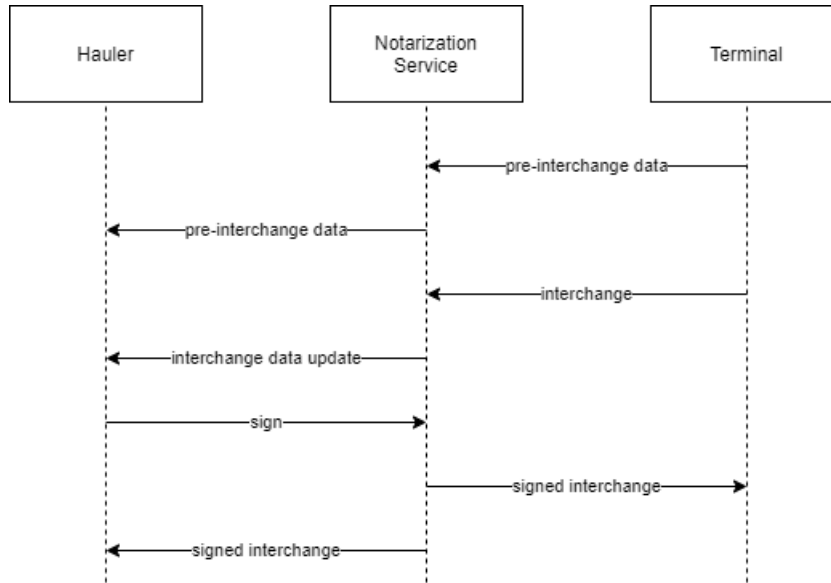


Figure 4 - Case D

2.3 DEFINIZIONE ENDPOINT

I seguenti endpoint sono esposti dal servizio notarile al terminal:

Pre-interchange data

HTTP Method	POST
URL	https://TBD/api/notarization-manager/terminal/preinterchange
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Interchange data object
Response Body	Signature status object

Interchange

HTTP Method	POST
URL	https://TBD/api/notarization-manager/terminal/interchange
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Interchange object
Response Body	Signature status object

I seguenti endpoint sono esposti dal servizio notarile al trasportatore:

Signature pre-trigger

HTTP Method	POST
URL	https:// TBD/api/notarization-manager/hauler/pretrigger
Required Headers	<i>Content-Type:</i> must be set to application/json <i>Authorization:</i> must contain the authentication token (See the authentication guide)
Request Body	Signature reference object
Response Body	Signature status object

Sign

HTTP Method	POST
URL	https:// TBD/api/notarization-manager/hauler/countersign
Required Headers	<i>Content-Type:</i> must be set to application/json <i>Authorization:</i> must contain the authentication token (See the authentication guide)
Request Body	Signature reference object
Response Body	Signature status object

Following endpoints are exposed by the Terminal System to the Notarization Service:

Signed interchange

HTTP Method	POST
URL	TBD
Required Headers	TBD
Request Body	Signed interchange object
Response Body	Empty

I seguenti endpoint sono esposti dal sistema del trasportatore al servizio notarile:

Pre-interchange data

HTTP Method	POST
URL	TBD
Required Headers	TBD
Request Body	Interchange data object
Response Body	Empty

Signed interchange

HTTP Method	POST
URL	TBD

Required Headers	TBD
Request Body	Signed interchange object
Response Body	Empty

Ready-to-sign notification

HTTP Method	POST
URL	TBD
Required Headers	TBD
Request Body	Signature reference object
Response Body	Empty

Interchange data update

HTTP Method	POST
URL	TBD
Required Headers	TBD
Request Body	Interchange data object
Response Body	Empty

2.4 DEFINIZIONE OGGETTI

Interchange data

Schema:

```
{
  driver          driver object
  vehicle         vehicle object
  operations      operation object array
  signatureReference signature reference object
}
```

Example:

```
{
  "driver": {
    "surname": "Rossi",
    "name": "Mario",
    "personalDocuments": [
      {
        "typeCode": "APGE_BADGE",
        "number": "12345",
        "countryCode": "IT"
      },
      {
        "typeCode": "DRIVING_LICENCE",
        "number": "111111111",
        "countryCode": "IT"
      }
    ]
  }
}
```



```

    ],
  },
  "vehicle": {
    "plate": "AA000AA",
    "countryCode": "IT"
  },
  "operations": [
    {
      "typeCode": "releas",
      "goods": {
        "typeCode": "22G1",
        "unitNumber": "AAAU7654321",
        "unitEmpty": TRUE,
        "description": ""
      }
    },
    {
      "typeCode": "PICKUP",
      "goods": {
        "typeCode": "22G1",
        "unitNumber": "AAAU1234567",
        "unitEmpty": false,
        "description": "Fornitures"
      }
    }
  ],
  "signatureReference": {
    "reference": "IC_A_0102",
    "referenceDate": "2021-03-15",
    "signer": "IT111111111111",
    "countersigner": "BE222222222222"
  }
}

```

Interchange

```

Schema:
{
  "interchangeData": interchange data object
  "fileContent": string
  "fileContentType": string
}

```

```

Example:
{
  "interchangeData": {
    "driver": {
      "surname": "Rossi",
      "name": "Mario",
      "personalDocuments": [

```



```

    "typeCode": "APGE_BADGE",
    "number": "12345",
    "countryCode": "IT"
  },
  {
    "typeCode": "DRIVING_LICENCE",
    "number": "11111111",
    "countryCode": "IT"
  }
],
},
"vehicle": {
  "plate": "AA000AA",
  "countryCode": "IT"
},
"operations": [
  {
    "typeCode": "releas",
    "goods": {
      "typeCode": "22G1",
      "unitNumber": "AAAU7654321",
      "unitEmpty": TRUE,
      "description": ""
    }
  },
  {
    "typeCode": "PICKUP",
    "goods": {
      "typeCode": "22G1",
      "unitNumber": "AAAU1234567",
      "unitEmpty": false,
      "description": "Fornitures"
    }
  }
],
"signatureReference": {
  "reference": "IC_A_0102",
  "referenceDate": "2021-03-15",
  "signer": "IT1111111111",
  "countersigner": "BE2222222222"
},
"fileContent": "JVBERi0xLjcKCjQgMC...Rgo=",
"fileContentType": "application/pdf"
}

```

Signature reference

```

Schema:
{
  reference      string
}

```

```
referenceDate      string (format: yyyy-mm-dd)
signer             string
countersigner     string
}
```

Example:

```
{
  "reference": "IC_A_0102",
  "referenceDate": "2021-03-15",
  "signer": "IT11111111111",
  "countersigner": "BE22222222222"
}
```

Signature status

Schema:

```
{
  signatureReference  signature reference object
  processStatus       string
  statusChangeTime   string (format: yyyy-mm-dd hh:MM:ss)
  signatureType       string
}
```

Example:

```
{
  "signatureReference": {
    "reference": "IC_A_0102",
    "referenceDate": "2021-03-15",
    "signer": "IT11111111111",
    "countersigner": "BE22222222222"
  }
  "processStatus": "CREATED",
  "statusChangeTime": "2021-03-15 18:23:54",
  "signatureType": "PRE_ARMED_DOUBLE"
}
```

Driver

Schema:

```
{
  surname           string
  name              string
  personalDocuments personal document object array
}
```

Example:

```
{
  "surname": "Rossi",
  "name": "Mario",
  "personalDocuments": [
    {
      "typeCode": "APGE_BADGE",
```

```
"number": "12345",  
"countryCode": "IT"  
},  
{  
  "typeCode": "DRIVING_LICENCE",  
  "number": "111111111",  
  "countryCode": "IT"  
}  
]  
}
```

Personal document

Schema:

```
{  
  typeCode      string  
  number        string  
  countryCode    string  
}
```

Example:

```
{  
  "typeCode": "APGE_BADGE",  
  "number": "12345",  
  "countryCode": "IT"  
}
```

Vehicle

Schema:

```
{  
  plate          string  
  countryCode    string  
}
```

Example:

```
{  
  "plate": "AA000AA",  
  "countryCode": "IT"  
}
```

Operation

Schema:

```
{  
  typeCode      string  
  goods         good object array  
}
```

Example:

```
{
  "typeCode": "PICKUP",
  "goods": {
    "typeCode": "22G1",
    "unitNumber": "AAAU1234567",
    "unitEmpty": false,
    "description": "Fornitures"
  }
}
```

Good

Schema:

```
{
  typeCode      string
  unitNumber    string
  unitEmpty     boolean
  description    string
}
```

Example:

```
{
  "typeCode": "22G1",
  "unitNumber": "AAAU1234567",
  "unitEmpty": false,
  "description": "Fornitures"
}
```

Signed interchange

Schema:

```
{
  signatureReference  signature reference object
  fileContent         string
  fileContentType     string
}
```

Example:

```
{
  "signatureReference": {
    "reference": "IC_A_0102",
    "referenceDate": "2021-03-15",
    "signer": "IT11111111111",
    "countersigner": "BE22222222222"
  },
  "fileContent": "JVBERi0xLjcKCjQgMC...Rgo=",
  "fileContentType": "application/pdf"
}
```


3 PROCESSO APPLICAZIONE DI PESATURA

3.1 DESCRIZIONE DEL PROCESSO

Lo scopo dell'applicazione di pesatura è raccogliere le richieste di pesatura, comporle con i dati operativi, attivare dispositivi di pesatura esterni, produrre documenti di pesatura e notificare i dati acquisiti agli stakeholder.

La configurazione del servizio sarà formata dai seguenti moduli:

- Weighing Service
- Weighing Dashboard
- Users Manager
- Edifact Service
- File Service

Il sistema TOS invia una prenotazione di pesata al Servizio di pesatura. La prenotazione si riferisce ad un container e può contenere la targa del camion o il codice attrezzatura del veicolo utilizzato per movimentare il container. Nella prenotazione sono elencati anche i riferimenti degli stakeholder che richiedono una copia del documento di peso. La prenotazione può essere aggiornata o cancellata se non è associata a un processo.

Quando il camion arriva alla stazione di pesatura, l'autista o l'operatore inserisce il codice del container nella pagina "nuova pesatura" del cruscotto. Il servizio ricerca un processo con stato APERTO o, se non ne viene trovata, una prenotazione non correlata ad alcun processo per il codice contenitore richiesto. Se non viene trovato alcun processo o prenotazione, l'operatore può inserire una nuova richiesta di prenotazione nella dashboard.

A seconda del completamento del processo, l'operatore può quindi selezionare di inserire una delle quattro operazioni disponibili:

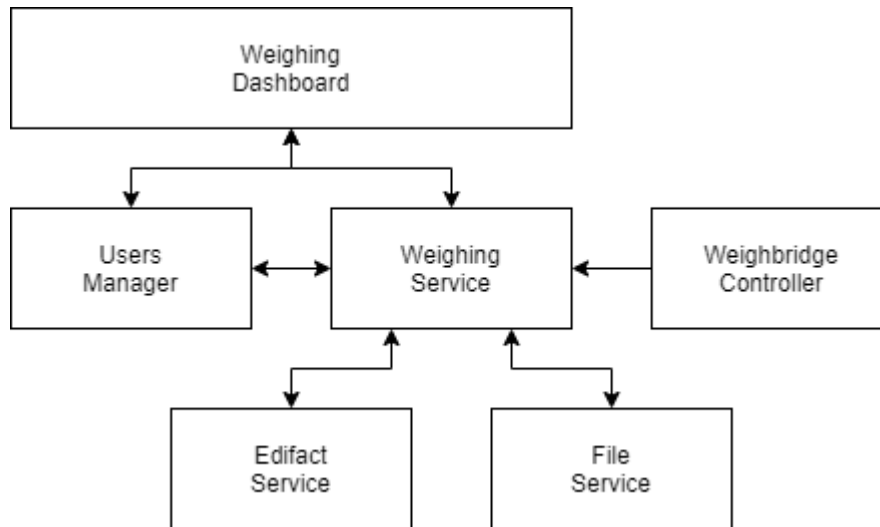
- Gross weight measurement
- Tare measurement
- Net measurement
- Tare manual entry

Se si seleziona la misurazione lorda, tara o netta, l'operatore può utilizzare l'interfaccia per attivare il controller della pesa a ponte e ottenere il peso misurato.

Se si seleziona l'inserimento manuale della tara l'operatore può inserire direttamente il valore della tara. Quindi l'operatore conferma il salvataggio del valore.

Se il sistema dispone di dati sufficienti per calcolare il peso effettivo del contenitore (misurazione netta, misurazione lorda + misurazione tara o misurazione lorda + inserimento manuale tara completato), all'operatore viene chiesto di chiudere il processo. Il Servizio di Pesatura richiede al Servizio Edifact la creazione del documento VERMAS, lo salva con il File Service e ne invia una copia a tutti gli stakeholder.

3.2 MODULI



Weighing Service

This is the core module of the application, which will receive and manage weighing booking, create the process, interact with the weighbridge system, notify the weigh values, and orchestrate the activities of the other modules.

Weighing Dashboard

This module is the front end of the application allowing the interaction with the users through a kiosk or a mobile device in proximity of the weighbridge.

Weighbridge Controller

This module manages the direct communication with the weighbridge control interface allowing the system to get the weigh value.

Users Manager

This module is responsible for the authentication of users that can interact with the application through the dashboard or through an API.

Edifact Service

This module is responsible for the creation of the VERMAS file.

File Service

This module is responsible for the storage of VERMAS files that needs to be forwarded to the stakeholders.

3.3 SERVIZIO DI PESATURA

Public APIs

Bookings

Create Booking

HTTP Method	POST
URL	TBD /bookings
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Booking object
Response Body	Booking object

Update Booking

HTTP Method	PUT
URL	TBD /bookings/{id}
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Booking object
Response Body	Booking object

Delete Booking

HTTP Method	DELETE
URL	TBD /bookings/{id}
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	-
Response Body	-

Retrieve Booking

HTTP Method	GET
URL	TBD /bookings/{id}
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	-
Response Body	Booking object

Browse Bookings

HTTP Method	POST
URL	TBD /bookings/browse
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	BrowseRequest
Response Body	BrowseResponse<Booking> object

Processes

Create Process

HTTP Method	POST
URL	TBD /processes

Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Process object
Response Body	Process object

Update Process

HTTP Method	PUT
URL	TBD /processes/{id}
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Process object
Response Body	Process object

Delete Process

HTTP Method	DELETE
URL	TBD /processes/{id}
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	-
Response Body	-

Retrieve Process

HTTP Method	GET
URL	TBD /processes/{id}
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	-
Response Body	Process object

Browse Processes

HTTP Method	POST
URL	TBD /processes/browse
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	BrowseRequest
Response Body	BrowseResponse< Process> object

Operations

Create Operation

HTTP Method	POST
URL	TBD /operations/{processId}/operations/
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Operation object
Response Body	Operation object

Objects

A booking represents the request for a container weighing.

Booking	
id	Long
container	String
vehiclePlate	String
equipmentCode	String
requestedTime	LocalDateTime
stakeholdersReferences	StakeholderReference list
deleted	Boolean

A process is used to collect all data related to a weighing and manage the status of the whole operation.

Process	
id	Long
bookingId	Long
container	String
containerType	String
status	ProcessStatus enum

The process status indicates if the operations are completed or if the process has been canceled.

ProcessStatus
OPEN
CLOSED
CANCELED

An operation represents a single measurement in the process of obtaining the actual container weigh.

Operation	
id	Long
processId	Long
component	Component enum
weight	BigDecimal
measurement unit	MeasurementUnit enum
time	LocalDateTime
number	String
weighbridge code	String
manual	Boolean

A component is a part of the weight.

Component
GROSS
NET
TARE

A measurement unit specify the unit used to measure.

MesaurementUnit
KG

A weighbridge is the device used to perform the measurement.

Weighbridge	
id	Long
code	String
description	String

A stakeholder is a subject interested in receiving the container weigh data.

Stakeholder	
id	Long
description	String
vatNumber	String
vatNumberCountryCode	String
configuration	String
enabled	Boolean

A stakeholder reference is the minimal data to identify a stakeholder.

StakeholderReference	
id	Long
vatNumber	String
vatNumberCountryCode	String

3.4 CONTROLLORE PESA

Public APIs

Weigh

Start

HTTP Method	GET
URL	TBD /weigh/start
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	-
Response Body	WeighResponse object

Objects

WeighResponse	
weighStatus	WeighStatus
weigh	BigDecimal
weighNumber	String
weighbridgeCode	String
errorMessage	String

WeighStatus
VALID
UNSTABLE
OVERLOAD
UNDERLOAD
OFFRANGE
OVERFLOW
UNDERFLOW

3.5 GESTORE UTENTI

Public APIs

Auth

Authenticate

HTTP Method	POST
URL	TBD /auth/
Required Headers	<i>Content-Type</i> : must be set to application/json
Request Body	Credentials object
Response Body	LoginResponse object

Objects

Credentials	
username	String
password	String

LoginResponse	
token	String
user	User

User	
id	Long
username	String
lastName	String
firstName	String
email	String
companyId	Long
groups	UsersGroup[]
company	Company

UsersGroup	
id	Integer
code	String
description	String

Company	
id	Long
code	String
name	String
vatNumber	String
vatCountryCode	String
companyRoles	CompanyRole[]

CompanyRole	
id	Integer
code	String
description	String

Edifact Service

Public APIs

Vermas

Compose D17B

HTTP Method	POST
URL	TBD /vermas/d17b/compose
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Interchange object
Response Body	ParseResponse object

Objects

Interchange	
senderCode	String
recipientCode	String
preparationDateTime	LocalDateTime
referenceNumber	String
messages	Message[]

Message <abstract>	
referenceNumber	String
messageTypeIdentifier	String
messageNameCoded	String
messageFunction	MessageFunction

MessageFunction
CANCELLATION
ADDITION
DELETION
CHANGE
REPLACEMENT
ORIGINAL
FINAL
COPY
CHANGE_HEADER
CHANGE_DETAIL
DEFINITIVE

Vermas <extends Message>

equipment	Equipment
bookingNumber	String
locationCode	String

Equipment

equipmentQualifier	EquipmentQualifier
unitNumber	String
sizeTypeCode	String
equipmentCycle	EquipmentCycle
fullEmptyStatus	FullEmptyStatus
netWeight	BigDecimal
grossWeight	BigDecimal
maxWeight	BigDecimal
tare	BigDecimal
vgm	BigDecimal
seals	String[]

EquipmentQualifier

CH
CN
RG
RR
SW
TE

EquipmentCycle

CONTINENTAL
EXPORT
IMPORT
TRANSHIPMENT

FullEmptyStatus

EMPTY
FULL
UNKNOWN

3.6 FILE SERVICE

Public APIs

Files

CreateFile

HTTP Method	POST
URL	TBD /files/
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Container object
Response Body	Reference object

Objects

Container	
reference	Reference
content	Byte[]
base64	Boolean

Reference	
id	Integer
contentId	Integer
fileName	String
virtualPath	String
type	String
compressionType	CompressionType
storageType	StorageType
archived	Boolean
processed	Boolean
ownerCompanyId	Integer
createTimestamp	LocalDateTime
createUserId	Integer
updateTimestamp	LocalDateTime
updateUserId	Integer
tags	Tag[]

CompressionType
NONE
ZIP

StorageType
FS
DB

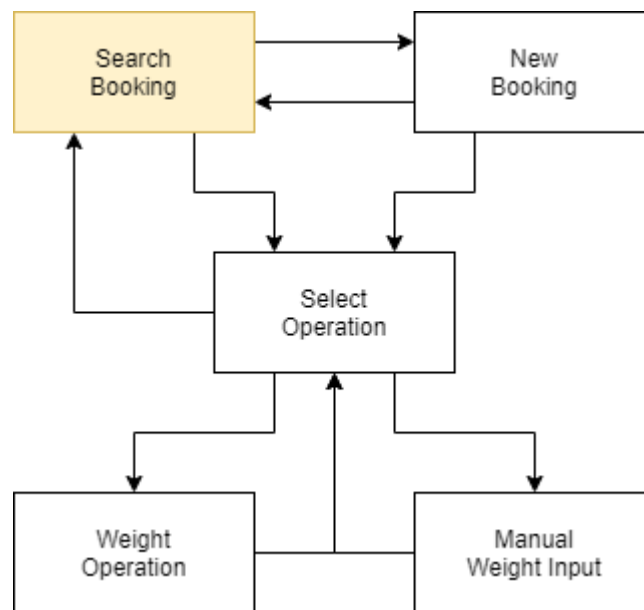
Tag	
key	String
value	String

4 INTERFACCIA UTENTE APPLICAZIONE DI PESATURA

4.1 INTRODUZIONE

L'interfaccia utente è pensata per essere utilizzata su dispositivi da campo come chioschi e tablet. Tutto il testo e gli elementi grafici dovrebbero essere grandi. I colori dovrebbero avere contrasti elevati

4.2 WORKFLOW



Una volta che l'operatore ha effettuato l'accesso, viene portato alla vista Ricerca prenotazione.

4.3 VISTA RICERCA PRENOTAZIONE

Search Booking

Container Number

- Quando l'operatore entra per la prima volta nella pagina, il pulsante Nuova prenotazione è disabilitato.
- L'operatore inserisce il numero del contenitore e quindi seleziona il pulsante Cerca.
- Se la ricerca ha esito positivo, l'operatore viene portato alla visualizzazione Seleziona operazione.
- Se la ricerca non produce alcun risultato:
 - o viene visualizzata una finestra di dialogo di avviso con il testo "Prenotazione non trovata per il contenitore richiesto".
 - o il pulsante Nuova prenotazione è abilitato.
- Quando l'operatore seleziona il pulsante Nuova prenotazione, viene portato alla visualizzazione Seleziona operazione.

4.4 VISTA NUOVA PRENOTAZIONE

New Booking

Container Number

Vehicle Plate

Equipment Code

- Quando l'operatore seleziona il pulsante Salva, viene eseguita la convalida dell'input.
- Se il numero del contenitore è vuoto, viene visualizzata una finestra di dialogo con il testo "È richiesto il numero del contenitore".
- Se sia la targa del veicolo che il codice dell'apparecchiatura sono vuoti, viene visualizzata una finestra di dialogo con il testo "Sono richiesti la targa o il codice dell'apparecchiatura".
- Se il numero del container non è vuoto e la targa del veicolo o il codice dell'apparecchiatura non sono vuoti, viene visualizzata una finestra di dialogo di conferma. Alla conferma, la prenotazione viene creata e l'operatore viene portato alla visualizzazione dell'operazione di selezione.

4.5 VISTA SELEZIONE OPERAZIONE

Container AAAU1234567

Measure
Gross Weight

Measure
Tare

Measure
Net Weight

Manual
Tare

Cancel

- Quando la pagina viene caricata, viene cercato il processo relativo alla prenotazione. Se non ne viene trovato nessuno, ne viene creato uno nuovo.
- A seconda dei componenti operativi già associati al processo, i pulsanti possono essere abilitati o disabilitati.

Operation component already associated	Measure Gross Weight button	Measure Net Weight button	Measure Tare button	Manual Tare button
No operation	Enabled	Enabled	Enabled	Enabled
Gross	Disabled	Disabled	Enabled	Enabled
Tare	Enabled	Disabled	Disabled	Disabled
Net	Disabled	Disabled	Disabled	Disabled
Gross + Tare	Disabled	Disabled	Disabled	Disabled

- Se è selezionato il pulsante Misura peso lordo, Misura peso netto o Misura tara, l'operatore viene portato alla visualizzazione Operazione di pesatura.
- Se si seleziona il pulsante Tara manuale, l'operatore viene portato alla vista Inserimento peso manuale.
- Se si seleziona il pulsante Annulla, l'operatore viene portato alla vista Cerca prenotazione.
- Se è presente Componente netto o Componente lordo e Componente tara, viene visualizzata una finestra di dialogo per l'operatore che notifica la fine del processo e il peso netto finale. Quindi il processo viene chiuso e l'operatore viene portato alla vista Cerca prenotazione.

4.6 VISTA OPERAZIONE DI PESA

Container AAAU1234567 - GROSS WEIGHT

Start Weighing

15054 Kg

Save

Cancel

- Quando l'operatore entra per la prima volta nella pagina, il pulsante Salva è disabilitato.
- Se è selezionato il pulsante Annulla, l'operatore viene portato alla visualizzazione Seleziona operazione.
- Quando l'operatore seleziona il pulsante Avvia pesatura, viene inviata una richiesta di pesatura alla pesa a ponte. Quindi il risultato viene visualizzato e il pulsante Salva è abilitato.
- Quando l'operatore seleziona il pulsante Salva, viene visualizzata una finestra di dialogo di conferma. Alla conferma, l'operazione viene registrata e l'operatore viene portato alla visualizzazione Seleziona operazione.

4.7 VISTA INSERIMENTO MANUALE PESO

Container AAAU1234567 - MANUAL TARE

Weight (Kg)

- Quando l'operatore entra per la prima volta nella pagina, il pulsante Salva è disabilitato.
- Se è selezionato il pulsante Annulla, l'operatore viene portato alla visualizzazione Seleziona operazione.
- Quando l'operatore inserisce il peso nel campo di immissione, il pulsante di salvataggio è abilitato.
- Quando l'operatore seleziona il pulsante Salva, viene visualizzata una finestra di dialogo di conferma. Alla conferma, l'operazione viene registrata e l'operatore viene portato alla visualizzazione Seleziona operazione.

4.8 CONTROLLORE PESA

Public APIs

Weigh

Start

HTTP Method	GET
URL	TBD /weigh/start
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	-
Response Body	WeighResponse object

Objects

WeighResponse	
weighStatus	WeighStatus
weigh	BigDecimal
weighNumber	String
weighbridgeCode	String
errorMessage	String

WeighStatus
VALID
UNSTABLE
OVERLOAD
UNDERLOAD
OFFRANGE
OVERFLOW
UNDERFLOW

4.9 GESTIONE UTENTI

Public APIs

Auth

Authenticate

HTTP Method	POST
URL	TBD /auth/
Required Headers	<i>Content-Type</i> : must be set to application/json
Request Body	Credentials object
Response Body	LoginResponse object

Objects

Credentials	
username	String
password	String

LoginResponse	
token	String
user	User

User	
id	Long
username	String
lastName	String
firstName	String
email	String
companyId	Long
groups	UsersGroup[]
company	Company

UsersGroup	
id	Integer
code	String
description	String

Company	
id	Long
code	String
name	String
vatNumber	String
vatCountryCode	String
companyRoles	CompanyRole[]

CompanyRole	
id	Integer
code	String
description	String

Edifact Service

Public APIs

Vermas

Compose D17B

HTTP Method	POST
URL	TBD /vermas/d17b/compose
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Interchange object
Response Body	ParseResponse object

Objects

Interchange	
senderCode	String
recipientCode	String
preparationDateTime	LocalDateTime
referenceNumber	String
messages	Message[]

Message <abstract>	
referenceNumber	String
messageTypeIdentifier	String
messageNameCoded	String
messageFunction	MessageFunction

MessageFunction
CANCELLATION
ADDITION
DELETION
CHANGE
REPLACEMENT
ORIGINAL
FINAL
COPY
CHANGE_HEADER
CHANGE_DETAIL
DEFINITIVE

Vermas <extends Message>	
equipment	Equipment
bookingNumber	String
locationCode	String

Equipment	
equipmentQualifier	EquipmentQualifier
unitNumber	String
sizeTypeCode	String
equipmentCycle	EquipmentCycle
fullEmptyStatus	FullEmptyStatus
netWeight	BigDecimal
grossWeight	BigDecimal
maxWeight	BigDecimal
tare	BigDecimal
vgm	BigDecimal
seals	String[]

EquipmentQualifier
CH
CN
RG
RR
SW
TE

EquipmentCycle
CONTINENTAL
EXPORT
IMPORT
TRANSHIPMENT

FullEmptyStatus
EMPTY
FULL
UNKNOWN

4.10 FILE SERVICE

Public APIs

Files

CreateFile

HTTP Method	POST
URL	TBD /files/
Required Headers	<i>Content-Type</i> : must be set to application/json <i>Authorization</i> : must contain the authentication token (See the authentication guide)
Request Body	Container object
Response Body	Reference object

Objects

Container	
reference	Reference
content	Byte[]
base64	Boolean

Reference	
id	Integer
contentId	Integer
fileName	String
virtualPath	String
type	String
compressionType	CompressionType
storageType	StorageType
archived	Boolean
processed	Boolean
ownerCompanyId	Integer
createTimestamp	LocalDateTime
createUserId	Integer
updateTimestamp	LocalDateTime
updateUserId	Integer
tags	Tag[]

CompressionType
NONE
ZIP

StorageType
FS
DB

Tag

key	String
value	String

5 ANTICIPO DOCUMENTALE

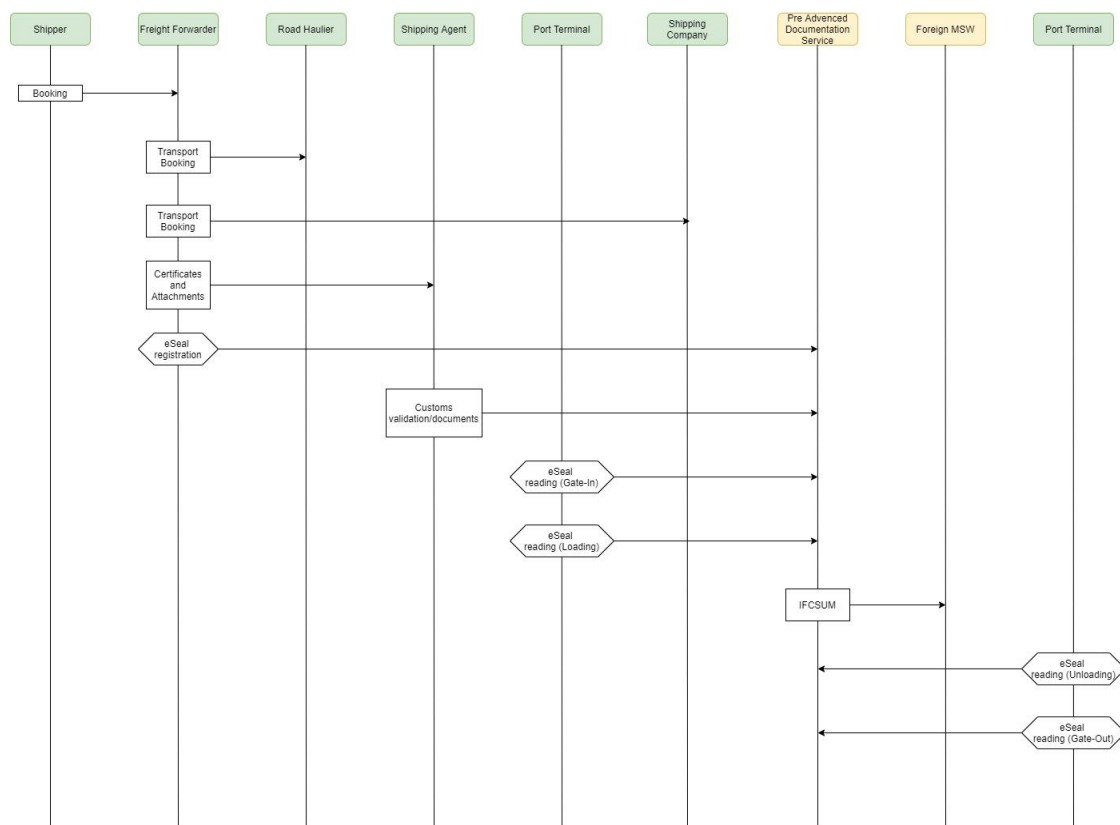
5.1 INTRODUZIONE

Questo capitolo ha lo scopo di descrivere il design della soluzione e le specifiche di integrazione per il servizio Pre Advanced Documentation, un sistema di microservizi che consente agli operatori portuali e logistici di condividere la documentazione logistica e personalizzata non appena prodotta.

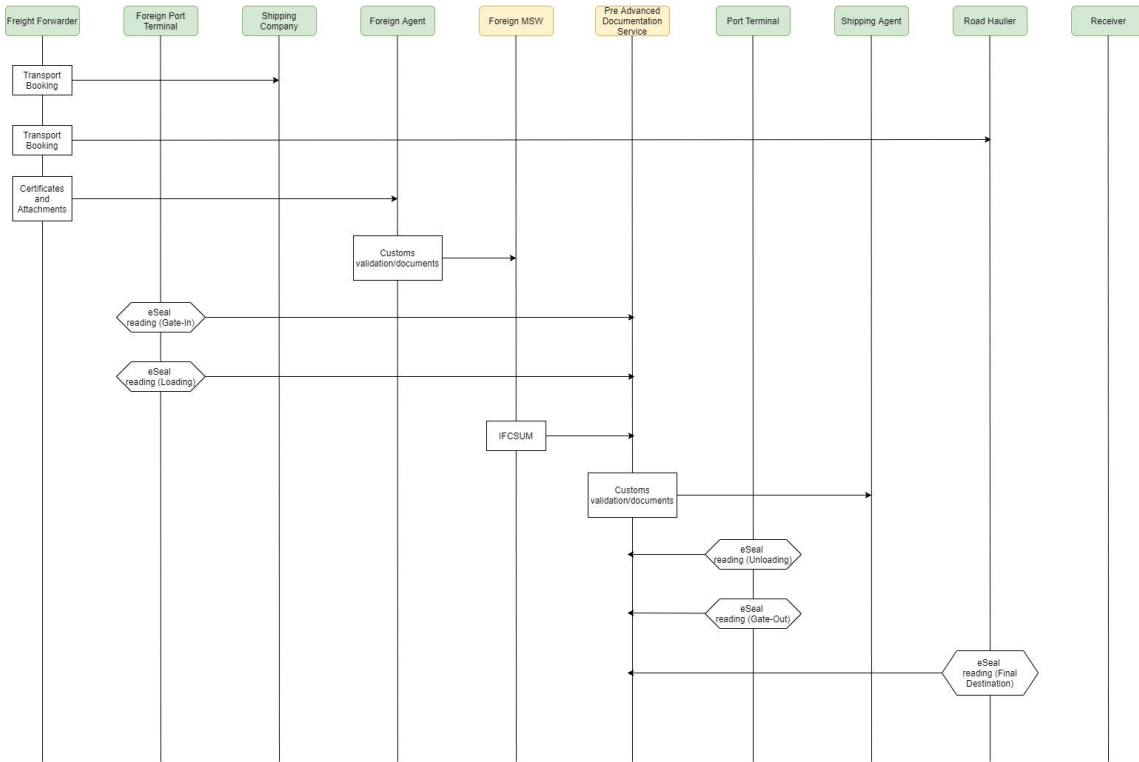
5.2 SERVIZI

Community Connector

The Community Connector is responsible for the communication with terminals, ship agents and freight forwarders IT systems. It allows the sharing of documents and events related to the import/export process.



Export Flows



Import Flows

Public APIs

Store File

HTTP Method	POST
URL	/files/store
Required Headers	<i>Content-Type:</i> must be set to application/json <i>Authorization:</i> must contain the authentication token (See the authentication guide)
Request Body	Container object
Response Body	Reference object

Register Event

HTTP Method	POST
URL	/events/
Required Headers	<i>Content-Type:</i> must be set to application/json <i>Authorization:</i> must contain the authentication token (See the authentication guide)
Request Body	Event object
Response Body	void

Objects

Container

reference	Reference object
content	Byte array
base64	Boolean

Reference

id	Integer
contentId	Integer
fileName	String
virtualPath	String
type	String
compressionType	CompressionType enum
storageType	StorageType enum
archived	Boolean
processed	Boolean
ownerCompanyId	Integer
createTimestamp	LocalDateTime
createUserId	Integer
updateTimestamp	LocalDateTime
updateUserId	Integer
tags	List of Tag object

CompressionType

NONE
ZIP

StorageType

FS
DB

Tag

key	String
value	String

Event

type	String
dateTime	LocalDateTime
tags	List of Tag object

Discovery Service

The Discovery Service is responsible for automatic services registration and discovery within the solution setup.

Every service deployed in the cluster must implement the service registration client and will be addressable by other services without any need of manual configuration.

Edifact Service

The Edifact Service is responsible for parsing and composition of documents in EDIFACT format. The EDIFACT type managed are:

- IFCSUM d97a for data related to the cargo manifests.
- CODECO d95b for data related to gate in and gate out events.

Files Service

The Files Service is responsible for the management of persistent files storage. It allows to store also metadata related to each file.

Files Service implements two way to handle the physical storage of the files:

- DB for storage in database tables through a virtual file system.
- FS for storage in the host filesystem.

IFSTL Connector

The IFSTL Connector orchestrate the acquisition and delivery of IFCSUM EDIFACT documents between the parties, the file conversions and notifications dispatching.

IFSTL Manager

The IFSTL Manager is the backend service providing services to the frontend for entities management and visualization.

Users Manager

The Users Manager service is responsible for the management of user's accounts and authentication.

Public APIs

Authenticate

HTTP Method	POST
URL	/auth/
Required Headers	<i>Content-Type</i> : must be set to application/json
Request Body	Credentials object
Response Body	LoginResponse object

Objects

Credentials

username	String
password	String

LoginResponse

token	String
user	User

User

id	Long
username	String
lastName	String
firstName	String
email	String
companyId	Long
groups	UsersGroup[]
company	Company

UserGroup

id	Integer
code	String
description	String

Company

id	Long
code	String
name	String
vatNumber	String
vatCountryCode	String
companyRoles	CompanyRole[]

CompanyRole

id	Integer
code	String
description	String

Frontend

The Frontend is the service UI allowing the management users to monitor the information handled by the system.