European Commission

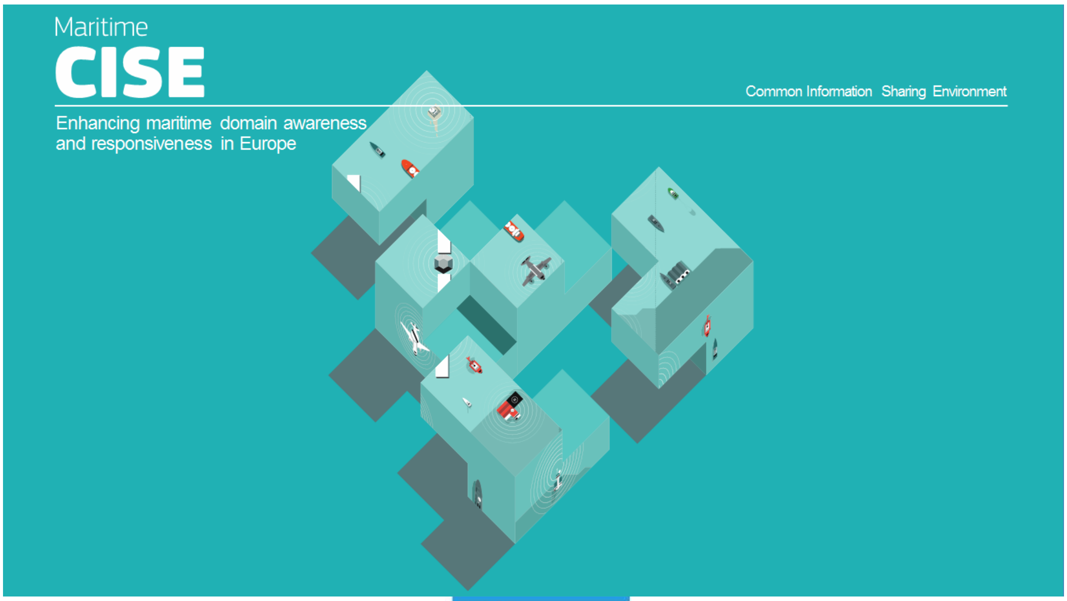
JRC TECHNICAL REPORTS

The Entity Service Model for CISE

Service Model Specifications

David Berger
Jesus Hermida
Franco Oliveri
Gian Carlo Pace

2017


Maritime
CISE
Common Information Sharing Environment
Enhancing maritime domain awareness and responsiveness in Europe

Joint Research Centre

European Commission
Joint Research Centre
Institute for the Protection and Security of the Citizen

Contact information
Franco Oliveri
Address: Joint Research Centre, Via Enrico Fermi 2749, TP 670, 21027 Ispra (VA), Italy
E-mail: Franco.Oliveri@ec.europa.eu
Tel.: +39 0332 78 9320
Fax: +39 0332 78 9156

http://www.jrc.ec.europa.eu/

This publication is a Technical Report by the Joint Research Centre of the European Commission.

Legal Notice
This publication is a Technical Report by the Joint Research Centre, the European Commission's in-house science service.
It aims to provide evidence-based scientific support to the European policy-making process. The scientific output expressed does not imply a policy position of the European Commission.Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use which might be made of this publication.

Printed in Italy

Document Metadata

Property	Value
Release date	28 Feb 2017
Status	Draft
Version	1.5
Authors	David Berger, Jesus Hermida, Franco Oliveri, Gian Carlo Pace
Reviewed by	Franco Oliveri
Approved by	Franco Oliveri
Contact	jrc-cise-dev@jrc.ec.europa.eu

Document History

Version	Date	Author	Description
V0.1	07.05.2015	JH	First draft ToC
V0.5	19.05.2015	JH	First draft
V0.9	03.06.2015	JH, DB	Second draft
V0.99	07.09.2015	JH, DB	Third draft with the Message Objects
V1.0	30.10.2015	JH, DB	Simplification of the service model, support of acknowledgement.
V1.1	05.11.2015	DB, GCP, MM	Updates with the Feedback message in conformance with EUCISE2020 requirements
V1.2	06.11.2015		Added the XML schemas
V1.3	26.11.2015		Improved the explanation of the asynchronous message exchange, including feedback messages.
V1.4	17.08.2016	JH, DB	Updates related to the granularity of the services (to avoid unecessay multiple exchanges)
V1.4.2	16.02.2017	JH, GCP, DB	Alignment with EUCISE 2020
V1.5	28.02.2017	JH, GCP, DB	Recovered the concept of Service Profile for service discovery. Fixed description of the service types in Annex B. Enumeration using long codes.

Table of Contents

1 Introduction

The CISE service model is the specification of the communication protocol, services and message data structures agreed to exchange information between CISE participants. In CISE, in order to ensure the interoperability of the maritime surveillance systems, all the information providers will implement the common service model.

The Cooperation Project (CoopP) defined the first Common Service Model based on the use cases identified and the data entities defined in the Common Data Model for CISE. After the end of the project, the Incubator project performed a first test on the functionalities of the service model and obtained some preliminary conclusions. At present, the EUCISE2020 project, which started in December 2014, has refined the proposed model based on the results of the previous projects and their own experience and requirements. The present version benefits from the first experimental implementations aside EUCISE 2020 project and proposes some improvements to optimize the exchanges.

The current technical report aims at describing the entity service model for CISE, updated with the first feed backs from the EUCISE2020 project and the first implementations.

2 Background

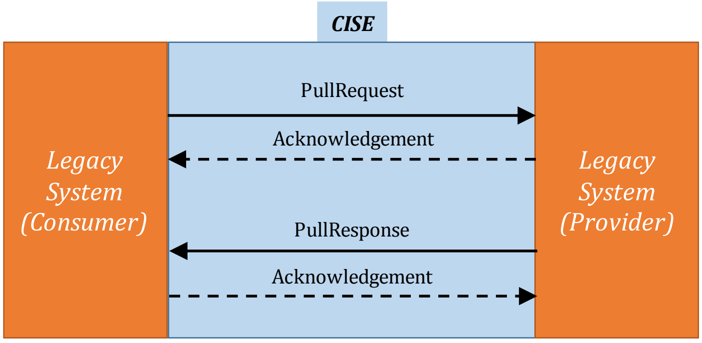
2.1 The CISE Communication Patterns for Information Exchange

The following subsections describe how to implement each communication pattern using service descriptions and message objects defined in the previous sections. All the patterns are asynchronous, i.e., the request of information and the response from the provider are separate processes, only loosely linked by correlation numbers.

Detailed sequence diagrams and messages examples can be found in Annex C.

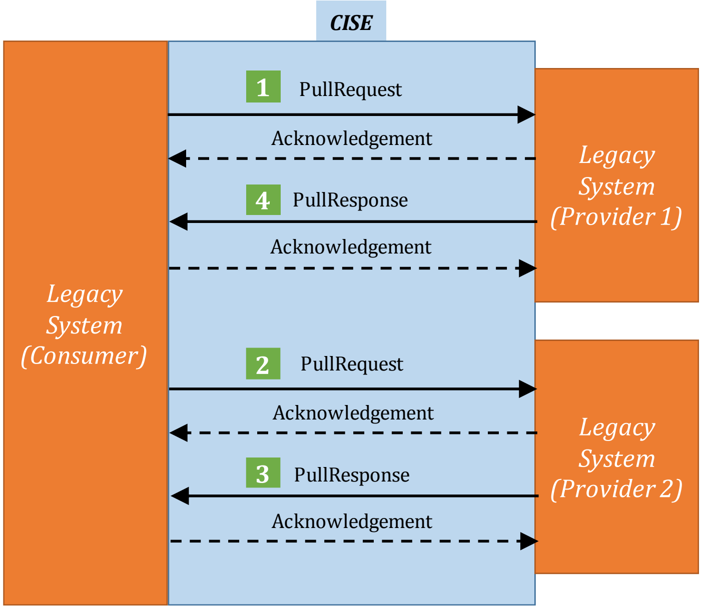
2.1.1 Pull

The Pull pattern is based on the *need-to-know* principle. In this pattern, the CISE consumer requests a piece of information to the CISE provider through the Pull operation using the PullRequest message. The CISE provider replies using the PullResponse of the CISE consumer.



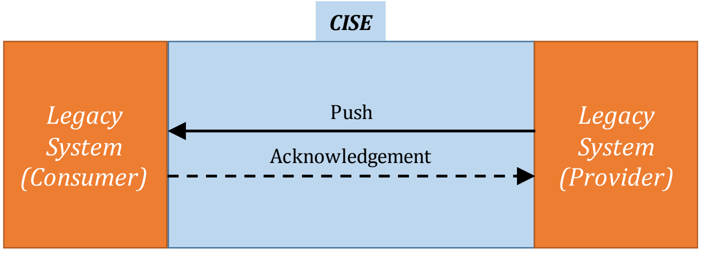
2.1.2 Multicast Pull

In the multicast pull pattern, the CISE consumer requests a piece of information to a group of CISE providers using the PullRequest operation. The CISE providers might be known previously or the CISE consumer could look for providers using the CISE Authority/Service Registry.



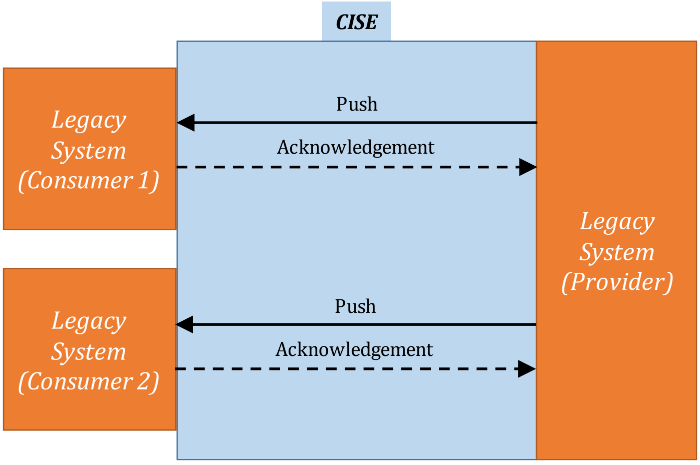
2.1.3 Push

The Push pattern is based on the *responsibility-to-share* principle. In this pattern, the CISE provider sends a piece of information to the CISE consumer (which might be of interest) using the Push operation and message.



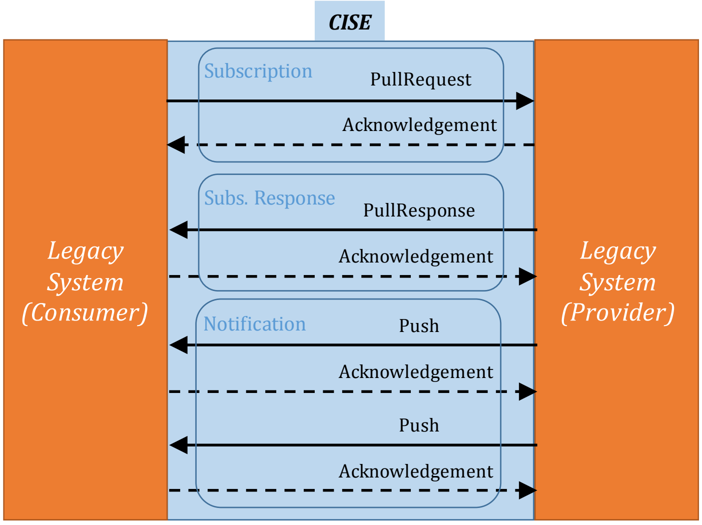
2.1.4 Multicast Push

In the Multicast Push pattern, the CISE provider sends a piece of information to a group of CISE consumers (who might be interested in it) using the Push operation. The CISE consumer might be known previously or the CISE provider could look for consumers using the CISE Authority Registry.



2.1.5 Publish/Subscribe

The Publish/Subscribe pattern results from the combination of the *need-to-know* and *responsibility-to-share* principles. In this communication pattern, the CISE consumer subscribes to a piece of information from the CISE provider using the PullRequest operation with the *subscribe PullType*. When the piece of information is available in the CISE provider, the provider notifies all the subscribers.



2.1.6 Discovery mechanism

In the different communication patterns described above, we assume that the consumer knows who the providers are (in the Pull pattern) or that the provider knows who the potential consumers are (in the Push pattern). In reality, the organisation and the missions of each authority vary a lot from one Member State to the other. Hence it is difficult to know who can offer what service in advance.

CISE offers a discovery mechanism, using a registry of service (see section 2.2). This mechanism allows to define two additional communication pattern:

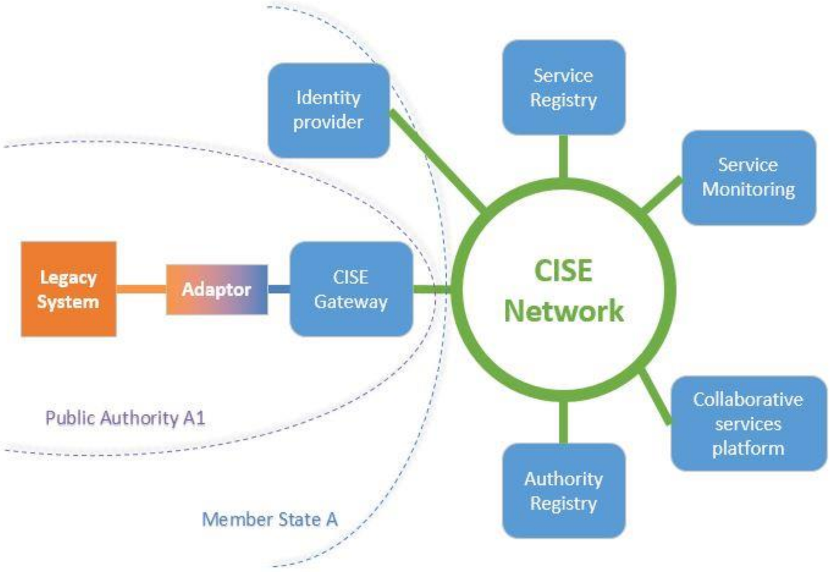
- The **PULL UNKNOWN**: allowing for a consumer to request information to one or several providers without knowing in advance which one can answer his request.
- The **PUSH UNKNOWN**: allowing a provider to push information to one or several potential interested consumers without knowing in advance who may be interested in receiving this information.

After going through the discovery mechanism (described in section 3.2.3) the PULL UNKNOWN may lead to a PULL pattern or a MULTICAST PULL pattern. The same way, the PUSH UNKNOWN may lead to a PUSH or MULTICAST PUSH pattern.

2.2 The CISE Building Blocks

2.2.1 Definition

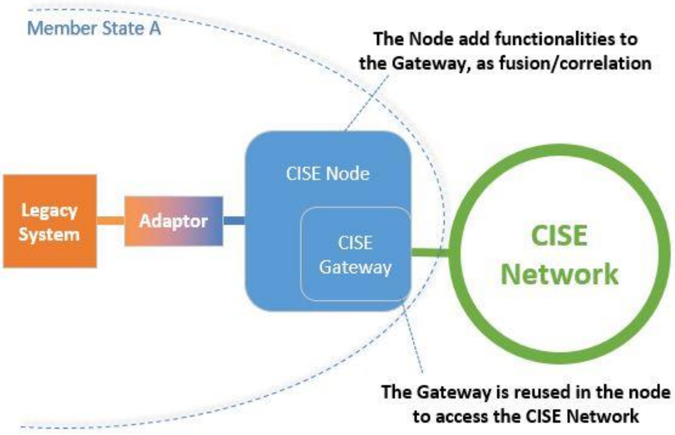
The CISE Hybrid Architecture Vision defined a set of building blocks that should be specified and implemented in CISE to enable the information exchange between partners:



- **CISE Gateway**: The CISE Gateway takes care of the information exchange protocol between participants, including the security, access control and reliability aspects.
- **Adaptor**. The CISE Adaptor takes care about the mapping of the data model and exchange protocol between the CISE data and service model and the Legacy System. This component should be developed specifically for each Legacy System.
- **Authority Registry.** A directory containing the contact details and information of the CISE participants.
- **Service Registry.** A directory of all the services offered by the CISE participants. It should contain metadata defining the service contracts and be accessible in an automatic way by the CISE Gateways.
- **Collaborative Service Platform.** A set of tools for virtual collaboration between public authorities. These tools facilitate audio and video communication, instant messaging, etc.
- **Service Monitoring.** This building block monitors the performance and availability of the services in the CISE gateways and can provide statistics, which after an agreement among the Member States, should be consistent and comparable.
- **Identity Provider.** This building block manages all aspects of identification and authentication at National level. CISE participants will rely on the same authentication mechanism, independent from the CISE gateway which they request data from.

In the CISE hybrid architecture, the following system need also to be defined:

- The **Legacy System** (LS) represents an existing system owned by a participant and used by operational users. LS can hold information that could be shared through CISE. LS can receive and use information from CISE. A LS could also be a National Node already gathering information from different other Legacy Systems (see section 2.3.2).
- The **CISE Node**: A CISE node is a CISE Gateway that implements additional specifications for data correlation and fusion rules. A Node can hold information from several cross-sectorial information sources of different authorities. The Node pre-processes this information (e.g. through correlation, fusion, aggregation) with the help of integrated intelligence capabilities.


Member State A
The Node add functionalities to the Gateway, as fusion/correlation
Legacy System
Adaptor
CISE Node
CISE Gateway
CISE Network
The Gateway is reused in the node to access the CISE Network

2.2.2 Functionalities covered by each building blocks

This section defines the roles and responsibilities of each building block described in the previous sections. The non trivial functionalities are further described in section **Error! Reference source not found.**.

The CISE Gateway:

- Accepts incoming messages from other Gateways and from the adaptors
- Authenticates and authorises the incoming messages
- Validates messages (xml correctness, message correctness, message signatures…)
- Routes messages using the Service Registry
- Delivers messages to the Adaptors connected
- Transfers messages to another Gateway and ensure the proper delivery

The Adaptor:

- Communicates with a LS (custom protocol)

- Translates the LS data model into the CISE Data Model (and vice versa)
- Adapts the specific LS Protocol to the CISE Messaging Protocol (and vice versa)
- Maps the LS credentials to the CISE Function (and vice versa). The Function of the person who does a request is used to assess the access rights.
- Sign/Encrypt messages on behalf of the LS (if needed)
- Delivers a message to the Gateway
- Delivers a message to the LS

The Service Registry:

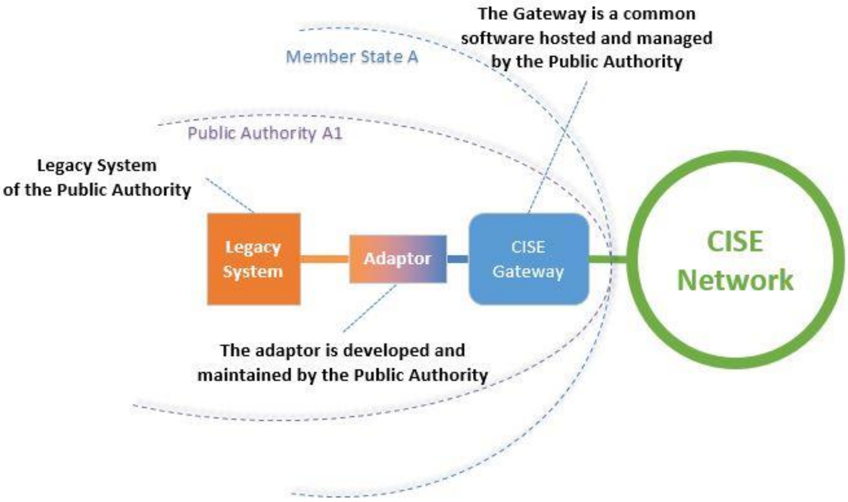
- Registers a service
- Unregisters a service
- Finds a service by id
- Finds services by queries
- Adds authorities and contact points
- Updates authorities and contact points
- Deletes authorities and contact points

2.3 Hybrid connection to CISE

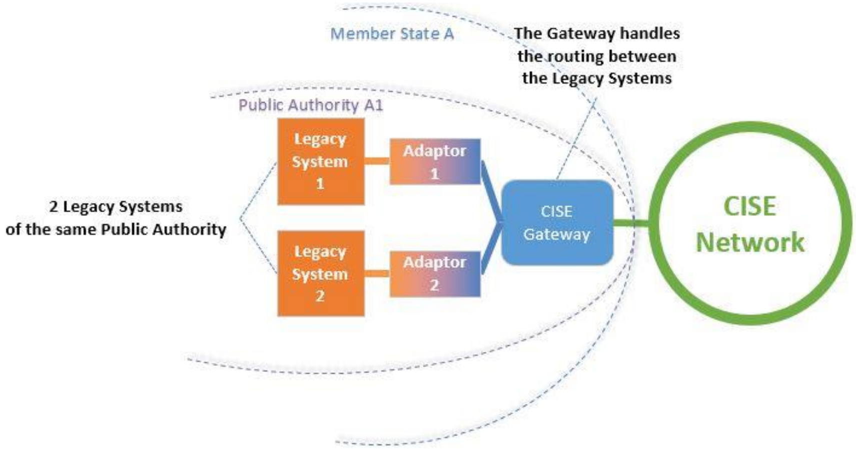
In the CISE Hybrid Architecture, each participant can choose to share or have access to information through a dedicated CISE access point or through a National, Regional or sectorial system. This decision will depend on the internal organisation of the authority (or Member State) and the authority's maritime surveillance systems and IT infrastructure.

2.3.1 A Public Authority directly connected to CISE

A Public Authority has the possibility to provide and consume information from the CISE Network, directly connected to it. In this case, the CISE Gateway may be directly hosted and managed by the Public Authority.

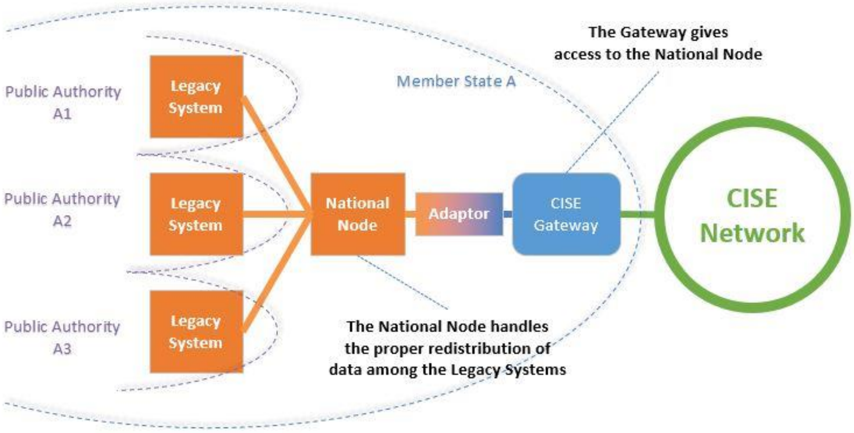

The Gateway is a common software hosted and managed by the Public Authority
Member State A
Public Authority A1
Legacy System of the Public Authority
Legacy System
Adaptor
CISE Gateway
CISE Network
The adaptor is developed and maintained by the Public Authority

If this Public Authority handles different Legacy Systems, for instance linked to different business processes, it is possible to connect all of them to the same Gateway. The Gateway will handle the proper routing and access right of information between the different Legacy Systems.



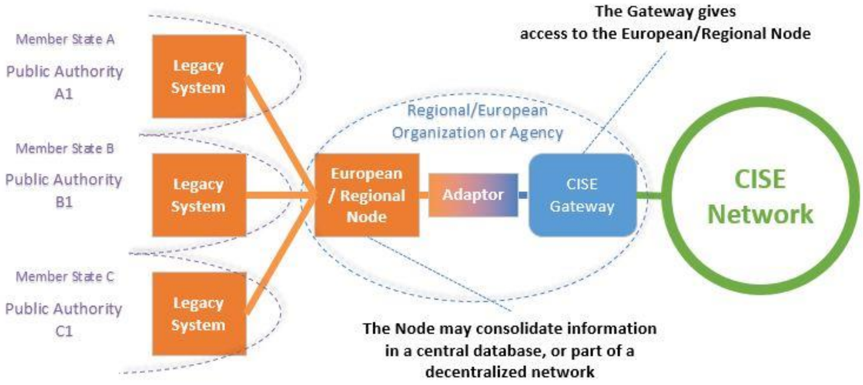
2.3.2 A Public Authority connected through a National Node

At National level, the decision could be to connect to CISE Network through an existing National Node. This node may only redirect messages or may consolidate the information in its own database. The functions handle by the different Legacy Systems and the end users of the Node should be identified to handle a proper access right management.



2.3.3 A Public Authority connected through a Regional or an European Node

Information exchange already exist cross-border at sectorial level. To benefit from the existing infrastructures, it is possible to enrich the existing European or Regional Nodes with information from the CISE Network, or to share more widely, cross sector, the information consolidated at sectorial level. The European or Regional node will have to propagate the access rules and redistribute properly the information to the Legacy Systems to respect the access rights defined by the providers.



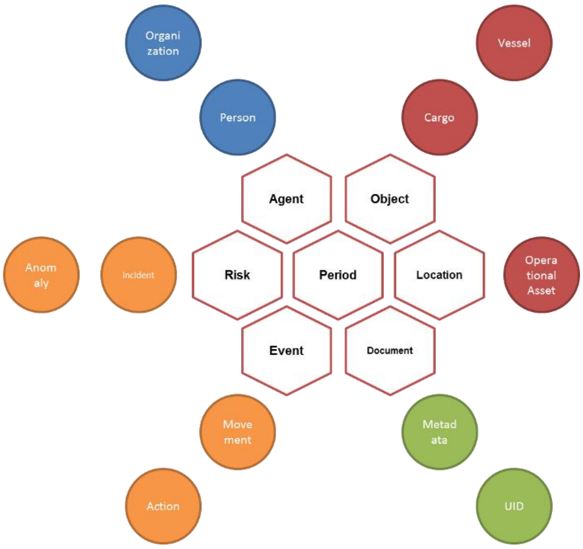
In this option, the Public Authority will depend on the choice of implementation of the European or Regional organization.

2.4 The CISE Data Model for Information Exchange

The CISE data model represents the common language used to exchange information across sectors and borders. The CISE Data Model has been developed by the Member States during the CISE pilot projects. The model takes into account the existing data standards used in the systems for maritime surveillance in Europe in order to facilitate the adaptation of these systems to CISE.

Design principles: sector neutral (no specific business riles embedded), flexible (it should adapt to any context), extensible (minimize the impact in case of extension), simple and understandable (for domain experts).

It focuses on information needs related to maritime surveillance to be exchanged between sectors and between borders, the specificity of each sector is not present if it has not being identified of interesting for the other sectors. The CISE data model represents the following main data entitities and the relationships between them: Vessel, Operational assets, Cargo, Movement, Location, Action, Incident, Anomaly, Risk, Person, Organization and Document.



3 The CISE Service Model

The Service Model describes the CISE services for information exchange. **A service is a self-describing, high-level abstraction of coarse-grained business capability**. In CISE, services hide the complexity of the LS's infrastructure and functionalities and the heterogeneity of platforms behind standards-based interfaces.

The Service model implements the five CISE communication patterns: pull/multicast pull, push/multicast push and publish-subscribe. The Service Model was designed to be flexible and adaptable to several use cases. Using the CISE services, the CISE participants can exchange information independently from its source. They could share:

- *Raw* information: information from the participants' database, collected from any source (e.g., sensors or mandatory reporting) and stored in the participants' systems.
- *Interpreted* information: information to which providers added some value from the interpretation of a piece of "raw" information.

More specifically, the model supports the following aspects:

- **Service definition**: description of the services provided by the CISE participants. It is use to declare a service in the Service Registry and to search for available service in a dynamic way.
- **Addressing**: methods for discovering and invoking services in the CISE network
- **Messaging**: message types, and their properties, exchanged through the CISE services.

3.1 Service Definition

In CISE, services are defined by the following information:

- **Service ID**: unique identifier of a service in CISE following an agreed scheme (URN), e.g., *eu.cise.authority.vesselService123*.
Providers can freely define an ID within the namespace assigned to them.

- **Service Type**: the type of a service indicates the main data entity exchanged using this service, e.g., VesselService. Annex A contains a list of the possible service types in the CISE service model (from the CISE Data Model, see Section 2.4).
Service providers can offer several services of the same type handling different subsets of data. For instance, providers could define one service (type VesselService) to exchange information from a vessel database and a second one (type VesselService) to exchange vessel information with their location obtained from a sensor.
Providers will decide which attributes and related entities of the main entity will be exchanged using the service. For instance, a service of type VesselService will enable the exchange of Vessel data entities and could also handle information of the Cargo, Incident, Location data entities (and the corresponding relationships), depending on the service provider and the capabilities of the legacy systems.

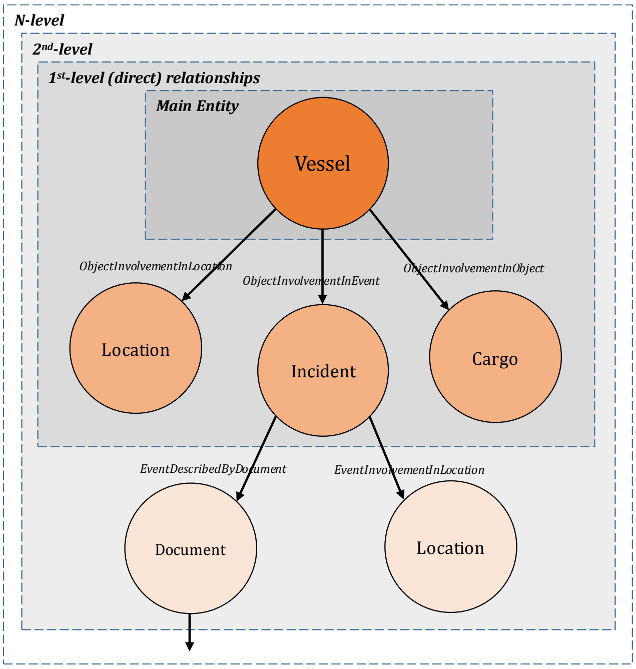


Figure 1: Main entity and related entities

- **Operation(s) supported:** Pull, Push, Subscribe, Feedback.
Each service can offer a set of operations to support the exchange of the data entities as defined in the CISE communication patterns (see section **Error! Reference source not found.**). Service providers can choose which operations should be implemented according to their needs.
The Feedback operation allows authorities to communicate issues in the information exchanged previously.

The parameters and the return values for each operation are described in **Error! Reference source not found.**.

Table 1. Parameters and return values of the service operations.

Operation	Description	Parameters	Return Value
Pull	This operation is used to request information using a query-by-example mechanism.	Data template (including the main entity of the service) Capabilities requested	A list of [Main Entity] + [Entities directly related to the main one] Capabilities offered

Push	This operation is used to push information to a CISE consumer. The origin of the notification might be a previous subscription.	A list of [Main Entity] + [Entities directly related to the main one] Capabilities offered	Acknowledgement
Subscribe	This operation is used to subscribe or unsubscribe to a series of notifications on specific information.	Data template (including the main entity of the service) Capabilities requested	Acknowledgement Capabilities offered
Feedback	This operation is used to provide feedback on information already exchanged.	Reference to previous exchange Nature of the issue	Acknowledgement

- Service Gateway: description of the gateway offering the access to the service, including its physical address. This information allows the proper routing of the message.
- **Service Profile**:
  - Metadata describing the profile of the service provider.
  - Characteristics of the data provided (e.g., freshness) and the data entities supported by the service (including the main entity).
  - Entity template: Providers should define which information will be available through the service, i.e., which attributes and relationships of the main entity will be provided.
- **Service Capabilities:** metadata describing the capabilities of the service (e.g., max. connection number, max. delay time, etc.)

Providers should register the service (along with the characteristics described above, including a template of the available information), in the ***CISE Service registry***. The registration will help other CISE participants to understand what can be expected from the service

Error! Reference source not found. shows the description of two services of type VesselService used to exchange different entities.

Table 2. Examples of service definitions in the CISE service model.

Service Name	Entity Template	Service Type	Operation	Parameters	Return values
eu.cise. authorityA1. vesselService 123	Main: Vessel Related: Cargo, ObjectInvolvementInObject (e.g., Vessel with Cargo, to find out the list of cargo related to a vessel)	VesselService	Pull	Vessel, Cargo (optional), ObjectInvolvementInObject (optional) Capabilities requested	Vessel, Cargo (optional), ObjectInvolvement InObject (optional) Capabilities offered
			Push	Vessel, Cargo (optional), ObjectInvolvementInObject (optional)	Acknowledgement
eu.cise. authorityB. vesselService 456	Main: Vessel Related: Incident, ObjectInvolvementInEvent (e.g., Vessel with Incident, to find out the list of vessels involved in a given Incident)	VesselService	Pull	Vessel, Incident (optional), ObjectInvolvementInEvent (optional) Capabilities requested	Vessel, Incident (optional), ObjectInvolvement InEvent (optional) Capabilities offered

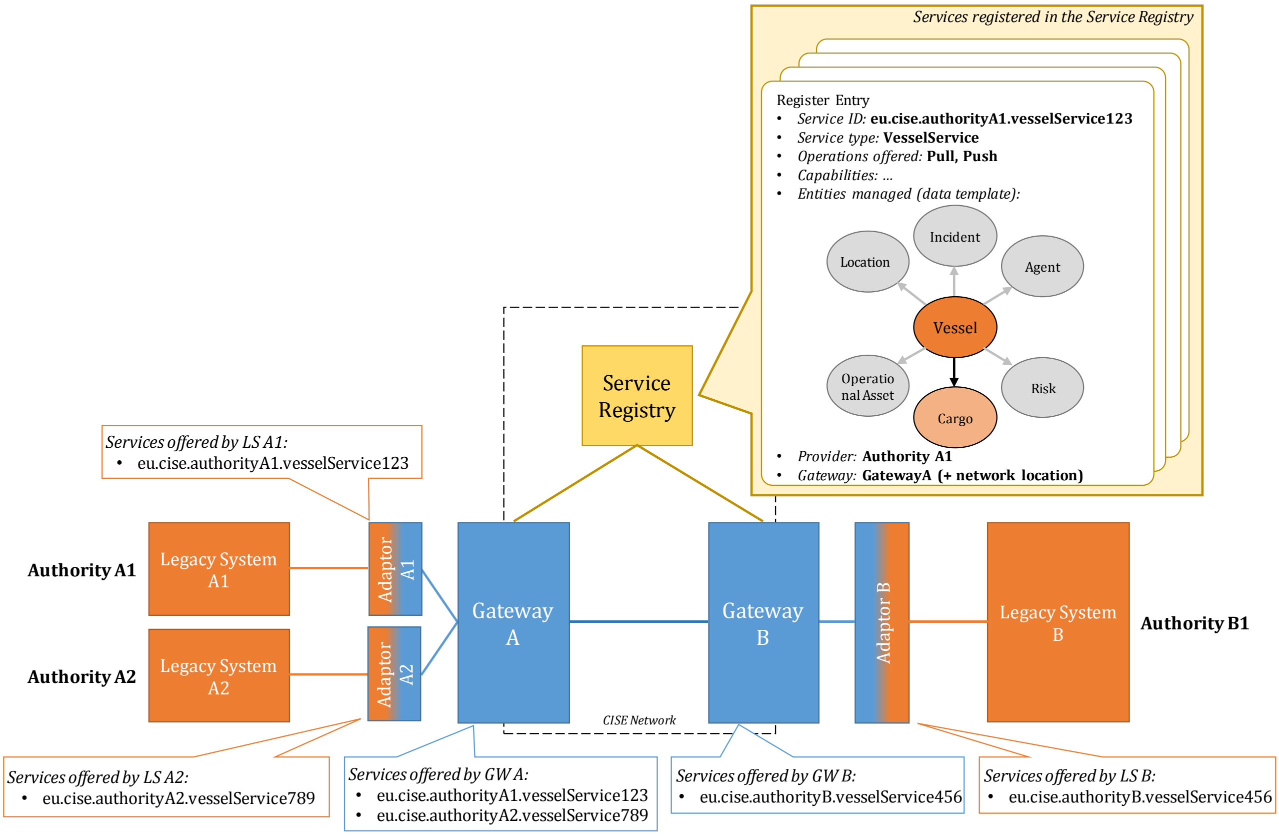

Services registered in the Service Registry
Register Entry
• Service ID: eu.cise.authorityA1.vesselService123
• Service type: VesselService
• Operations offered: Pull, Push
• Capabilities: …
• Entities managed (data template):
Incident
Location
Agent
Vessel
Operational Asset
Risk
Cargo
• Provider: Authority A1
• Gateway: GatewayA (+ network location)
Service Registry
Services offered by LS A1:
• eu.cise.authorityA1.vesselService123
Authority A1
Legacy System A1
Adaptor A1
Authority A2
Legacy System A2
Adaptor A2
Gateway A
Gateway B
Adaptor B
Legacy System B
Authority B1
CISE Network
Services offered by LS A2:
• eu.cise.authorityA2.vesselService789
Services offered by GW A:
• eu.cise.authorityA1.vesselService123
• eu.cise.authorityA2.vesselService789
Services offered by GW B:
• eu.cise.authorityB.vesselService456
Services offered by LS B:
• eu.cise.authorityB.vesselService456

Figure 2: Service declaration in the Service Registry

3.2 Business Logic

3.2.1 Using the CISE Services to Exchange Information

3.2.1.1 Requesting information from a provider

Use case: "Which vessels have been seen in a specific zone in a given period of time?"

Prerequisites:

- LS consumer and provider are in the CISE network and have been authenticated.
- The LS consumer knows that the LS provider offers the service *eu.cise.authorityA1.service123* and the declaration of the service.
- The LS consumer is allowed to invoke the service *eu.cise.authorityA1.service123*.

Process:

1. The CISE consumer requests the provider for the information needed using the Vessel service and the PullRequest message. In this case, the payload of the message contains the relationship needed.

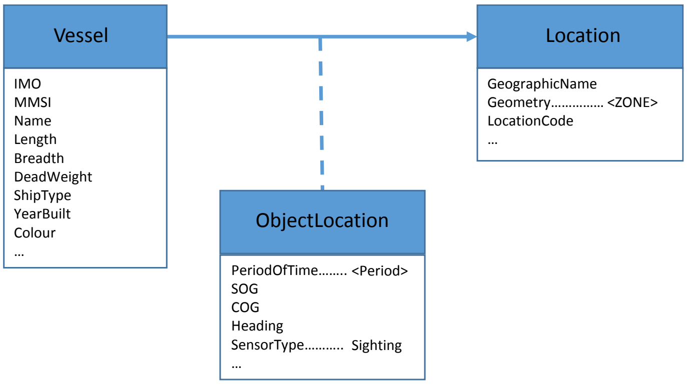


Figure 3: information sent in a PullRequest

2. The provider analyses the entity given as example and checks if there are similar entities in its legacy systems. It also checks whether the consumer can access the information according to the access rights rules.
3. The provider sends to the consumer, in the payload of a PullResponse message, those entities that are similar to the one requested. In the reply, the provider should also include some metadata indicating which type of reply it is providing: approximate or exact.

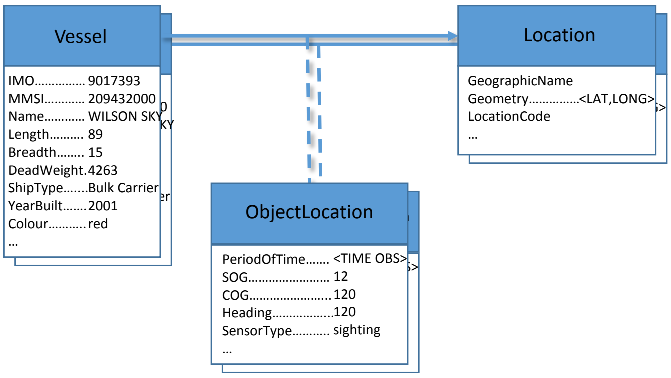


Figure 4: information received in a PullResponse

3.2.1.2 Requesting information from an unknown provider

Use case: "Which vessels have been seen in a specific zone in a given period of time?"

Prerequisites:

- Both actors have been authenticated in CISE.
- The consumer doesn't know the providers available in the CISE network offering the Vessel Service with the Location relashionship.

Process:

1. The CISE consumer requests its CISE Gateway for the information needed using the Vessel service and the PullRequest message. In this case, the payload of the message contains the relationship needed.

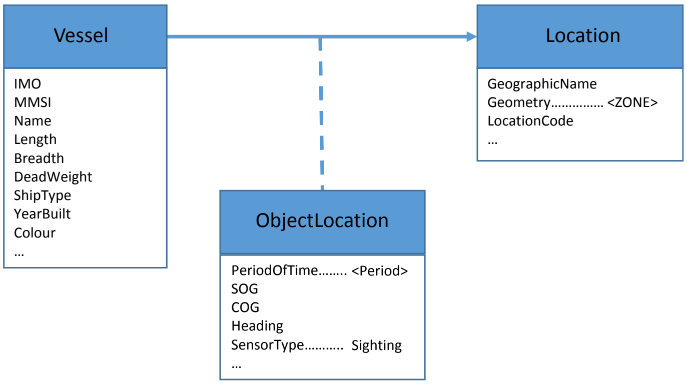


Figure 5: information sent in a PullRequest

2. The CISE Gateway of the consumer looks into the Service Registry for providers offering:
    a. A Vessel Service with the Location relationship available
    b. The Pull pattern Operation available for this service
    c. Providers allowing (a priori) the present consumer to access this service
3. The CISE Gateway sends to all providers retrieved on the previous step a PullRequest message with the payload containing the previous relationship.
4. Each provider analyses the entity given as example and checks if there are similar entities in its legacy systems. It also checks whether the consumer can access the information according to the access rights rules.
5. Each provider sends to the consumer, in the payload of a PullResponse message, those entities that are similar to the one requested. In the reply, the provider should also include some metadata indicating which type of reply it is providing: approximate or exact.

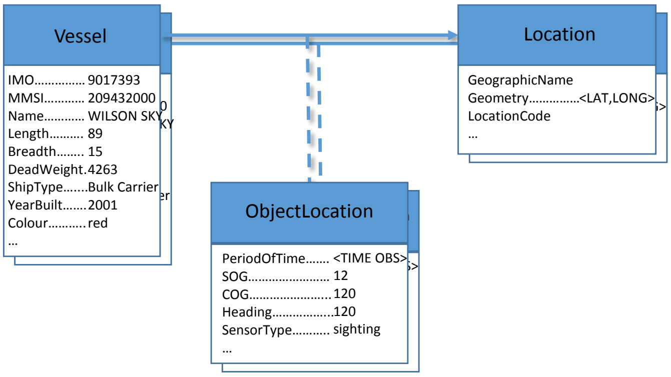


Figure 6: information received in a PullResponse

3.2.1.3 Requesting additional information

When we are looking for all the Incidents related to a Vessel, we will have to use the IncidentService, not the VesselService. In this case, the information passed in the *PullRequest* is a Vessel with its identification, an empty Incident and an ObjectEvent entity, specifying, if needed the type of relation between the Vessel and the Incident (e.g. is the vessel directly involved in the incident or participating to the rescue operation). This service, according to the query by example mechanism (see more details in section 3.2.2.1), should return a list of Incident linked to the Vessel (using the entities Incident, Vessel and ObjectEvent).

In the case several entities need to be exchanged to describe a more complex situation, this can be done if the "one level relashionship rule" is respected. For instance, to query detailed information about an incident, it is possible to send a query with the following entities in the payload:

- Incident (and its identification),
- Vessel (and ObjectEvent) to know the vessels involved,
- Organization (and AgentEvent) to know who is in charge,
- Location (and EventLocation) to know where it happened,
- Document to get possibly reports related to the incident…

The entities sent by the requester can be empty. If a few attributes are filled, they should be used as a filter by the provider. The Service provider should reply with at least the Incident entity and as much information from the other entities as possible.

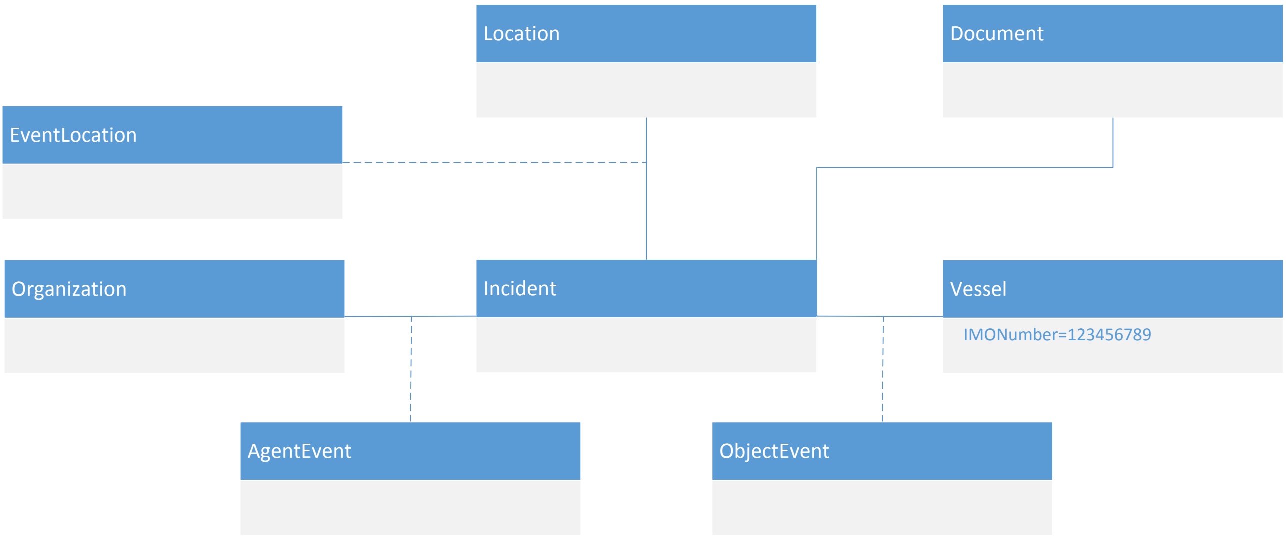


Figure 7: Example of payload sent in an IncidentService request

3.2.1.4 Providing Added Value through CISE Entity Services

This section illustrates two particular cases in which the providers can share information they previously processed.

3.2.1.4.1 Finding a Vessel from a Provider of Satellite Images

Use Case: "I need to know which vessels were in a zone at a specific time."

From a CISE perspective, this use case is equivalent to the one introduced in Section **Error! Reference source not found.** since the process of information exchange is the same. The main difference resides in the process of acquisition and processing of the information done by the provider before the exchange.

If the consumer asks for the information about the vessels in a zone, the provider (in this case a satellite provider) should detect the vessels from the satellite images available. Once detected, the provider can represent them using the CISE data model and transmit the information to the consumer. In this case, the information will be represented with the Vessel, Location and ObjectLocation entities.

The initial CISE requirements do not foresee the need of exchanging satellite images within the network. However, it could be also possible using the Document entity or the Metadata entity attached to any of the other entities.

3.2.1.4.2 Sharing Anomaly Detected at the Sea

Use Case: "I need to know which anomalies occurred in a zone at a specific time."

From a CISE perspective, this use case is equivalent to the one introduced in Section **Error! Reference source not found.** since the process of information exchange is the same. The main difference resides in the process of acquisition and processing of the information done by the provider before the exchange.

In this case, before the information exchange, the provider gathers the information about the vessel tracks and detects the anomalies with the voyage patterns. This information can be structured using the CISE data model, and more specifically, the Anomaly or the EventLocation entities. Thus, the provider can make the anomalies available by means of the Anomaly or the EventLocation services.

The provider does not need to exchange the information related to the analysis or the input data for the analysis, just the results.

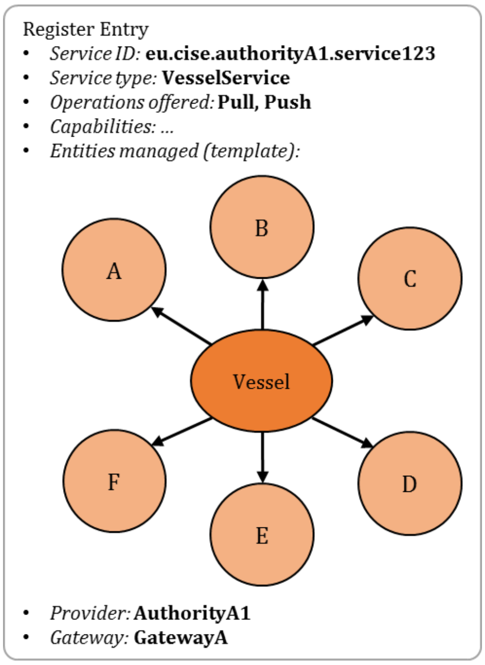
3.2.1.5 Giving Feedback on the Content

CISE participants can provide feedback on the content they shared or received. With this mechanism providers can notify an error in the information already sent (and request to delete it) and consumers can inform on an error or inaccuracy in the information received.

3.2.2 Service Behaviour

This section describes what entities are expected to be exchanged depending of the way the service has been declared in the Service Registry. This rules should be implemented in the adaptor of the Legacy System providing the information.

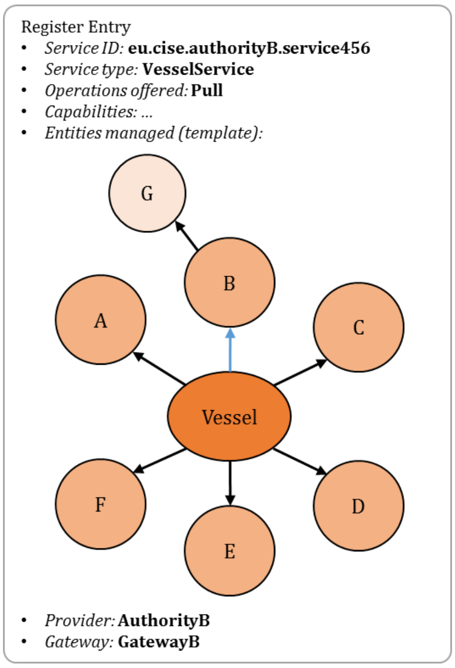
In the following sub-sections, the behaviour is illustrated by a set of examples based on the following declaration in the service registry:



Register Entry

- *Service ID:* **eu.cise.authorityA1.service123**
- *Service type:* **VesselService**
- *Operations offered:* **Pull, Push**
- *Capabilities: …*
- *Entities managed (template):*

- *Provider:* **AuthorityA1**
- *Gateway:* **GatewayA**



Register Entry

- *Service ID:* **eu.cise.authorityB.service456**
- *Service type:* **VesselService**
- *Operations offered:* **Pull**
- *Capabilities: …*
- *Entities managed (template):*

- *Provider:* **AuthorityB**
- *Gateway:* **GatewayB**

3.2.2.1 Pull

The specification of the Pull operation for the Pull and Multicast Pull patterns foresaw the implementation of a query-by-example mechanism. Using this mechanism, CISE consumers can request to a CISE provider all the data entities that are similar to a given example.

The figure below shows an example in which the CISE consumer sends to a CISE provider a Vessel entity whose name is “Queen Mary” as the input of the PullRequest operation. The CISE provider should reply with all the Vessel entities whose name is the given one.

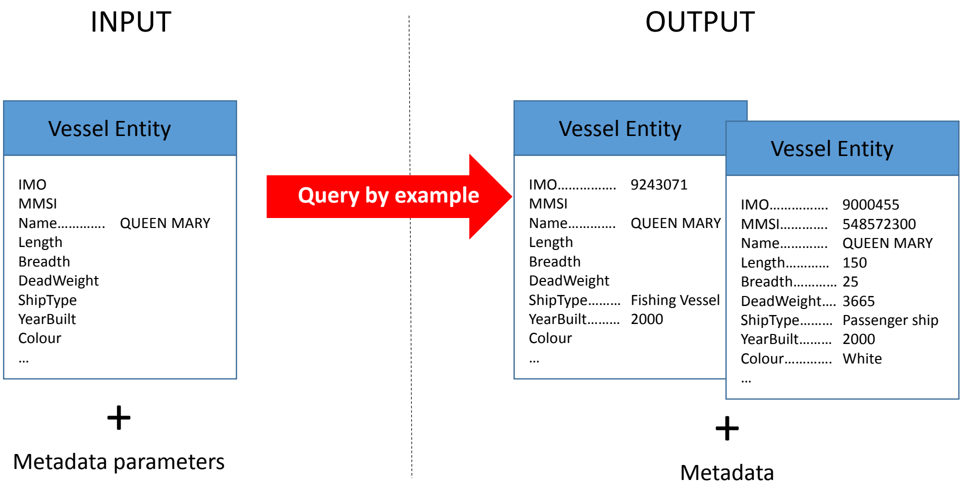
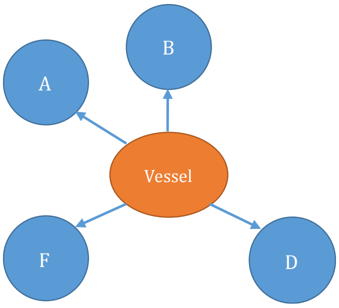
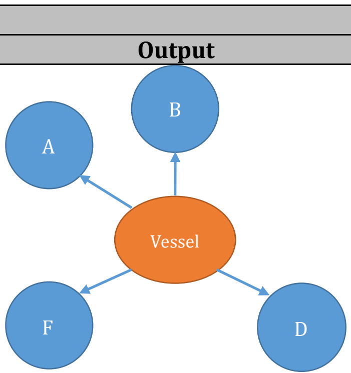
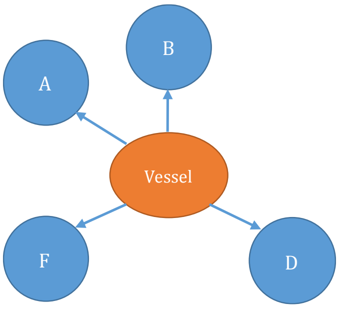
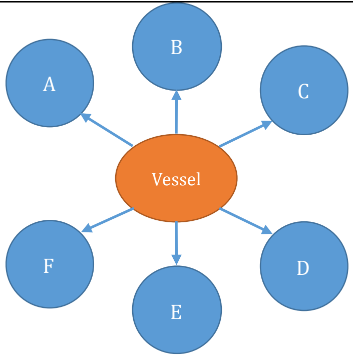
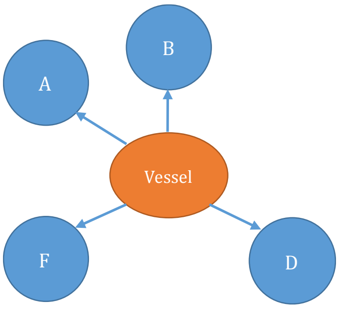
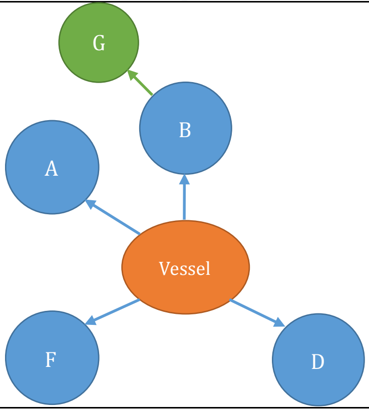
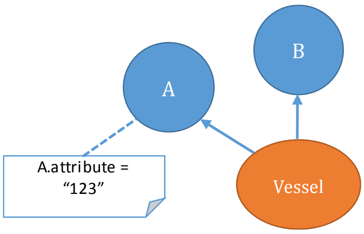
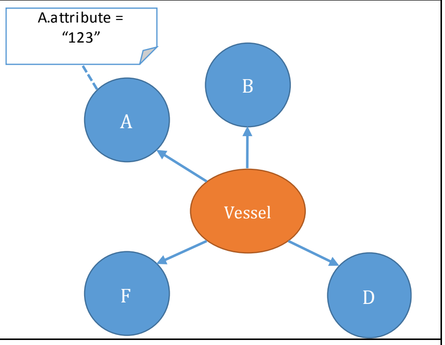
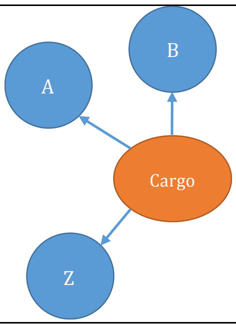


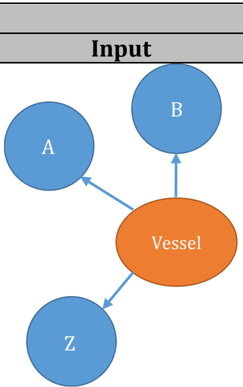
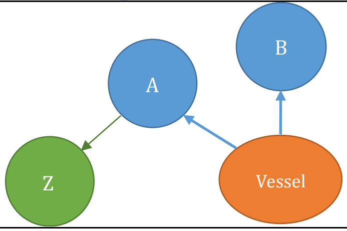
Figure 8: Query by example

In order to limit the complexity of the service operation and facilitate its implementation in CISE, some constraints were established:

- **Only one level of entity relationships.** The entity payload of any message can only contain one type of data entity and the data entities directly related to it. For instance, according to the CISE data model, a service could transmit the Vessel involved in an Incident (Incident-Vessel relationship) but it would not be possible to transmit the Vessel with its cargo involved in an incident (Incident-Vessel-Cargo relationships) with a single service. The Cargo related to the Vessel should be exchanged in a second step. Nevertheless, several entity relationships can be transmitted at the same time, for instance: the Vessels involved in an Incident along with the MRCC in charge of the incident and the location of the incident.
- **Best match.** The CISE providers should deliver a PullResponse message containing data entities that are the most similar to the requested query-by-example entity contained in the corresponding PullRequest message. ServiceCapability parameters are required to indicate which kind of response (QueryByExampleType) is required/delivered, i.e., exact or best match.
- **Full vs. partial response.** Open queries can result in a high number of responses, or in large responses, which could slow down (or even block) the provider's systems or the CISE network. In order to avoid this problem, some metadata within the service capabilities is also required to indicate the maximum number of responses (MaxEntitiesPerMsg). The provider may also want to restrict the number of responses transmitted for each exchange. This property is declared in the Service Registry (see 5.3.8).

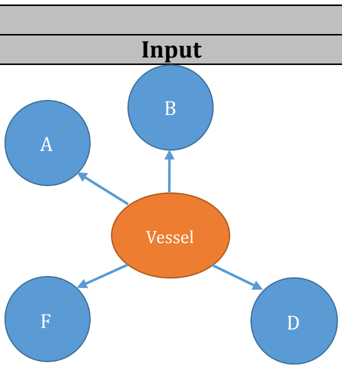
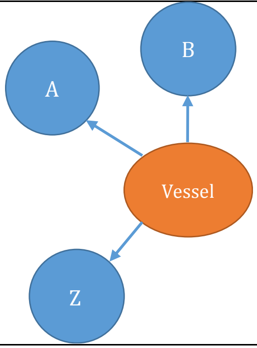
Case		Input	Output
Basic pull request	***C1***	A list of entities that match the input template (exact match).	A list of entities that match the input template (exact match).
	C2		A list of entities that match the supported template according to the service definition.
Pull request with attribute filter	**C3**	An entity template supported by the service provider (with a subset of the relationships) with a filter in one of the attributes.	A list of entities that match the input template (exact match).
	C4		A list of entities that match the supported template according to the service definition.
Not supported templates (error cases)	**C5**	Wrong main entity	Error
	C6	Template with a relationship not supported in the service definition	Error
	C7	Template with a 2nd level relationship	Error

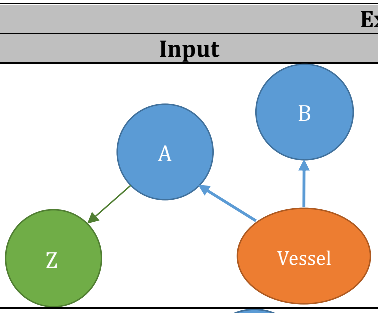
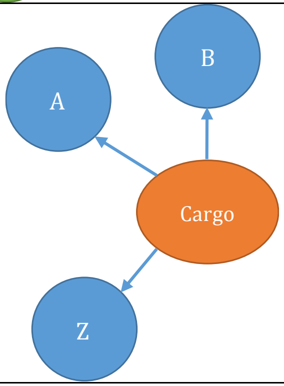
Service	Case	Example	
		Input	Output
eu.cise.authorityA1.service123 pull	C1		
eu.cise.authorityA1.service123 pull	C2		
eu.cise.authorityB.service456 pull	C1		
eu.cise.authorityA1.service123 pull	C3 C4		
eu.cise.authorityA1.service123 pull	C5		**Error: Main entity not found.**

Service	Case	Example	
		Input	Output
eu.cise.authorityA1. **service123** **pull**	**C6**		**Error: Z not supported.**
eu.cise.authorityA1. **service123** **pull**	**C7**		**Error: Only direct relationships are supported.**

3.2.2.2 Push

Case		Input	Output
Normal notification	***C1***	List of entities according to the template supported in the service definition.	Acknowledgement
Entities not according to the service template	**C2**	List of entities. Some entities contain relationships not supported in the service definition.	Acknowledgement
	C3	List of entities. Some entities contain second level relationships.	Acknowledgement
Not supported templates (error cases)	**C4**	List of main entities not supported by the service definition.	Error

Service	Case	Example	
		Input	Output
eu.cise.authorityA1. **service123** **push**	**C1**		**Acknowledgement**
eu.cise.authorityA1. **service123** **push**	**C2**		**Acknowledgement** **Z could be discarded by the consumer.**

Service	Case	Example	
		Input	Output
eu.cise.authorityA1. **service123** **push**	**C3**		**Acknowledgement.** **Z could be discarded by the service provider**
eu.cise.authorityA1. **service123** **push**	**C4**		**Error: Main entity not found.**

3.2.2.3 Subscribe

The Publish/Subscribe pattern consists of three independent communication processes: one for subscription (handled with the Pull operation) and another for publication (handled with the Push operation).

- **Subscribe**: the CISE consumer subscribes to a piece of information of the CISE provider using the Pull operation. In this case, the PullRequest message should indicate that it is a subscription process (PullType = subscribe). The CISE provider will register the CISE consumer and will send back a correlation ID (correlationID) in the PullResponse message.
- **Publish**: When the CISE provider wants to make available a piece of information (e.g., a list of data objects), the provider gateway needs to check the list of subscribers (CISE consumers). Subsequently, to each subscriber, the provider sends a Push message with the correlation ID provided before.
- **Unsubscribe.** The subscription can last for a time period given in the ServiceCapability parameter (SubscriptionDuration) or until the CISE consumer unsubscribes using the PullRequest message (PullType = unsubscribe).

The sequence diagram of this mechanism can be found in Section C.5.

3.2.2.4 The Multicast Patterns

The multicast patterns enable the CISE participants to request or send information to more than one participant. There are two main aspects in the management of the multicast communications:

- **Identification of the target participants.** In order to start the information exchange, the CISE participants should identify the target participants at the other end. For this, the DiscoveryProfile must describe the criteria for selecting the target participants according to the services they offer. The CISE gateway will search the identity of the target participants in the Authority and Service registries.
- **Independent communication between participants**. The exchange between two participants (e.g., A – B) is independent from the other exchanges (e.g., A –C, A – D, etc.) For instance, the retry mechanism, the acknowledgement protocol or the errors (among others) should be managed independently.

The sequence diagram of this mechanism can be found in sections C.2. and C.4.

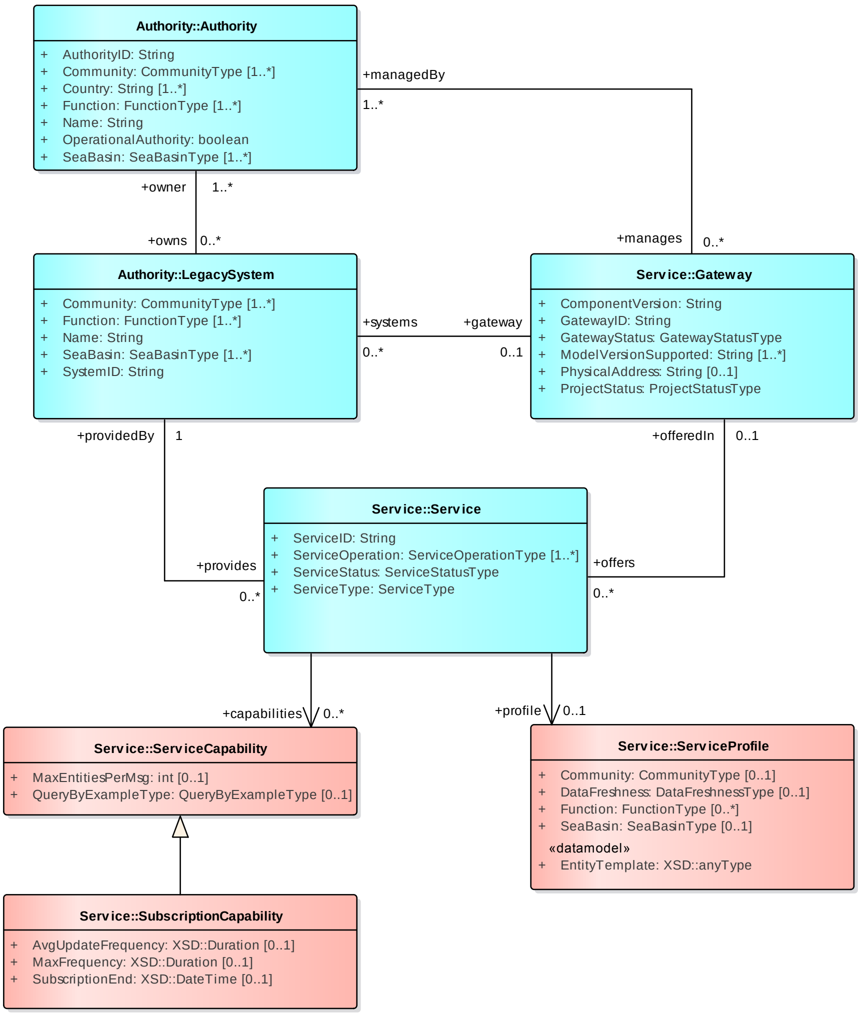
3.2.3 The Service Discovery Mechanism

The Legacy System may notify or request information to a sub-group of CISE participants. This can be done using some properties of the participants, such as the maritime function or the sea basin covered. In this case, the CISE GW will request to the CISE Service Registry the definition of all the services compliant with these criteria.

The CISE Service Registry offers two services:

Service Name	Description	Input	Output
FindService	This service is invoked by the Gateways to find services provided by participants.	Optional attributes: - Sea basin (ServiceProfile.SeaBasin) - Member State (ServiceProfile.Counrty) - Community (ServiceProfile.Community) - Function (ServiceProfile.Function) - Service type (Service.ServiceType) - Entities to exchange with optionallyspecific attributes (ServiceProfile.EntityTemplate) - Type of communication pattern (Service.ServiceOperation)	- List of services with their characteristics using the following classes:Service - Node - ServiceProfile - ServiceCapability - SubscriptionCapability (if the pattern is publish/subscribe)
GetServiceDetails	This service is invoked by the Gateways if the entity service is already known to find out the service details, for instance the network parameters to address properly the message	Service ID –(Service.ServiceID)	The service described by the following classes: - Service - Node - ServiceProfile - ServiceCapability - SubscriptionCapability (if the pattern is publish/subscribe)

The CISE Service Registry will manage at least the following information:


Authority::Authority
+ AuthorityID: String
+ Community: CommunityType [1..*]
+ Country: String [1..*]
+ Function: FunctionType [1..*]
+ Name: String
+ OperationalAuthority: boolean
+ SeaBasin: SeaBasinType [1..*]
+managedBy
1..*
+owner
1..*
+owns
0..*
+manages
0..*
Authority::LegacySystem
+ Community: CommunityType [1..*]
+ Function: FunctionType [1..*]
+ Name: String
+ SeaBasin: SeaBasinType [1..*]
+ SystemID: String
+systems
0..*
+gateway
0..1
Service::Gateway
+ ComponentVersion: String
+ GatewayID: String
+ GatewayStatus: GatewayStatusType
+ ModelVersionSupported: String [1..*]
+ PhysicalAddress: String [0..1]
+ ProjectStatus: ProjectStatusType
+providedBy
1
+offeredIn
0..1
Service::Service
+ ServiceID: String
+ ServiceOperation: ServiceOperationType [1..*]
+ ServiceStatus: ServiceStatusType
+ ServiceType: ServiceType
+provides
0..*
+offers
0..*
+capabilities
0..*
+profile
0..1
Service::ServiceCapability
+ MaxEntitiesPerMsg: int [0..1]
+ QueryByExampleType: QueryByExampleType [0..1]
Service::ServiceProfile
+ Community: CommunityType [0..1]
+ DataFreshness: DataFreshnessType [0..1]
+ Function: FunctionType [0..*]
+ SeaBasin: SeaBasinType [0..1]
«datamodel»
+ EntityTemplate: XSD::anyType
Service::SubscriptionCapability
+ AvgUpdateFrequency: XSD::Duration [0..1]
+ MaxFrequency: XSD::Duration [0..1]
+ SubscriptionEnd: XSD::DateTime [0..1]

The attribute ServiceProfile.EntityTemplate should be filled with a XSD message extracted from the CISE XSD Data Model. This XSD message should give only the attributes needed or provided and the restrictions on some of the attributes (e.g. in case of list, the value supported).

4 Technical Specifications

4.1 Technical Interface for Information Exchange

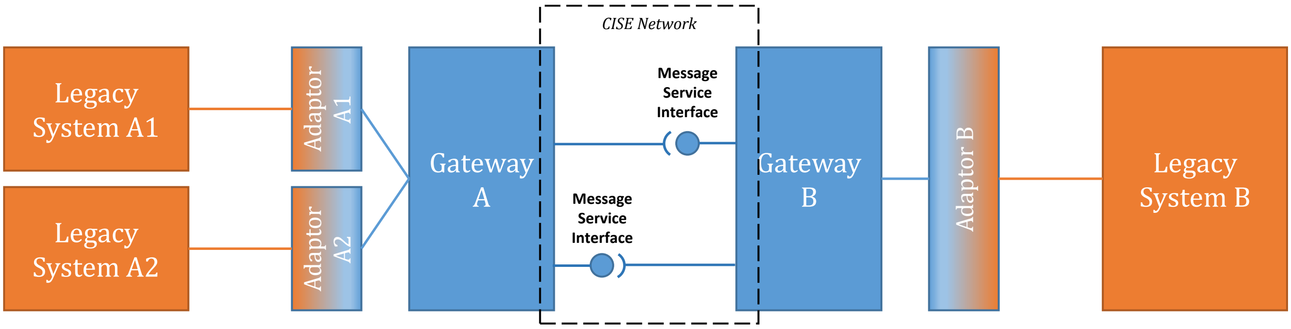
At technical level, the communication between the CISE partners is based on the exchange of messages, through which gateways can request/push information following the information exchange patterns.

- *Message metadata:* basic information of the message, e.g., message types, correlation with other messages, etc.
- *Service discovery:* information to support service definition, publication and discovery activities.
- *Addressing:* information describing the sender and the destination(s) of a specific message. This information is mainly used for the routing, acknowledgements and access right management;
- *Reliability profile:* information defining the retry strategy in exchanging a message between CISE participants.
- *Payload:* the exchanged information, structured according to using the CISE Data model. The payload can also describe metadata related to the information exchanged, such as security aspects.

In order to invoke a service offered by any legacy system connected to a gateway, the messages will contain a service code (or identifier). This code uniquely refers to an operation of an entity service provided by a specific legacy system at CISE level. The registration and the management of the identifiers will be performed in the CISE Service Registry.

4.1.1 Gateway-to-Gateway Interface

Every gateway will implement technical interface offers a single service, called MessageService, that manages the flow of incoming messages and process them to invoke the services (and operations) offered by the legacy systems.



Technical Service Interface	Description	Input	Output
MessageService	This operation receives and accepts a CISE message. After reception, the service can route the message to the final legacy system, it can invoke an entity service operation or it can performed any other action required by the CISE communication patterns. It is implemented by the CISE gateways.	Message (available message types: PullRequest, PullResponse, Push, Acknowledgement, Feedback)	Receipt (e.g.., if the message was accepted, and the reason for rejection, if any)

After receiving a message, the CISE gateways should analyse the message information and check whether the message is addressed to any of its legacy systems. If so, the message should be interpret and the corresponding actions launched.

The behaviour of the service operations is determined by the messages exchanged in the communication process (see table below).

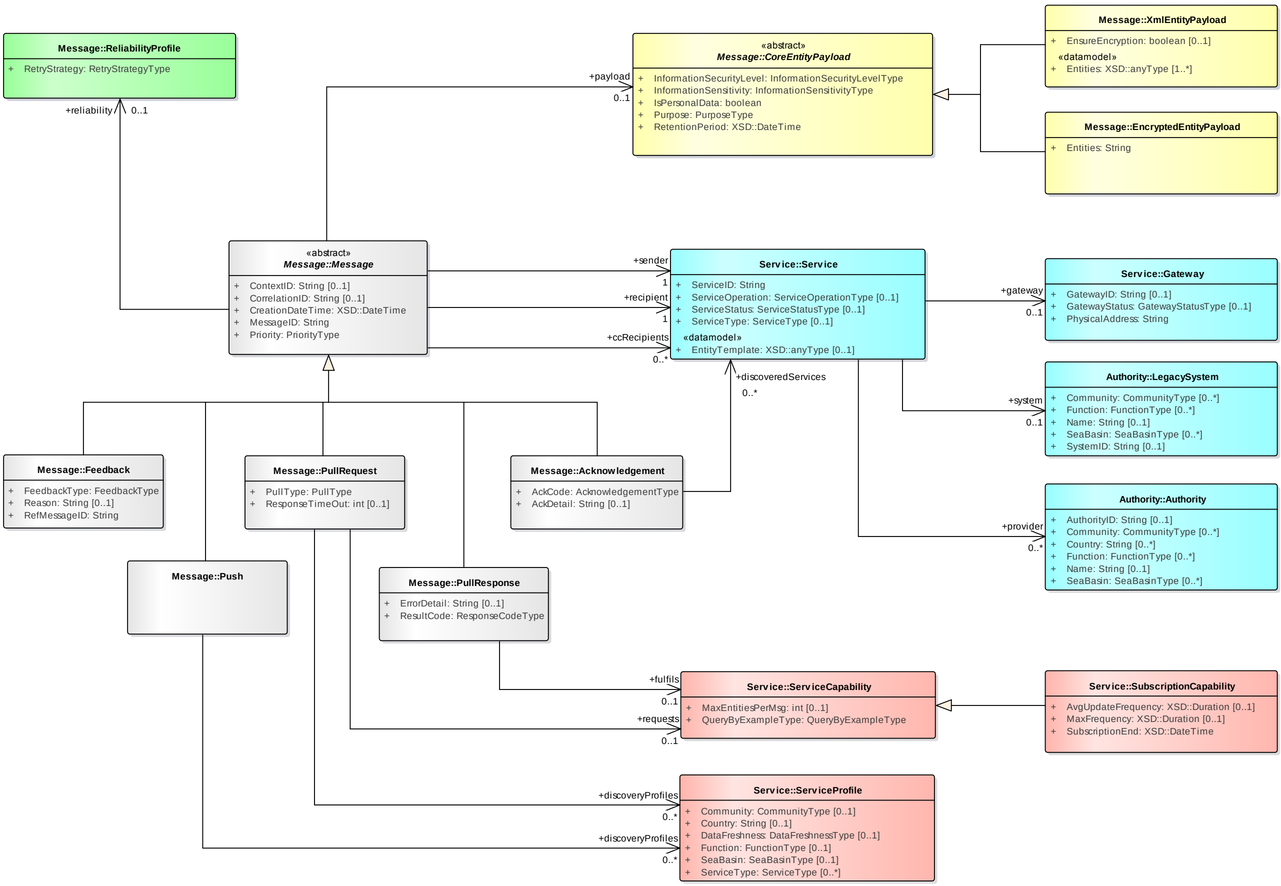
Service Type	Operation	Accepted messages
[Entity]Service	Pull	PullRequest, PullResponse, Acknowledgement
	Push	Push, Acknowledgement
	Subscribe	PullRequest, PullResponse, Acknowledgement

4.2 *Message Structures*

This section describes the messages used by the entity services. The CISE entity service model supports the representation and handling of the following information:

- *Message metadata:* basic information of the message, e.g., message types, correlation with other messages, etc.
- *Service discovery:* information to support service definition, publication and discovery activities.
- *Addressing:* information describing the sender and the destination(s) of a specific message. This information is mainly used for the routing, acknowledgements and access right management;
- *Payload:* the exchanged information, structured according to using the CISE Data model. The payload can also describe metadata related to the information exchanged, such as security aspects.
- *Reliability profile:* information defining the retry strategy in exchanging a message between CISE participants.

The following diagram describes the message structures for the CISE service model. The complete diagram of the message model can be found in Annex D.


Message::ReliabilityProfile
+ RetryStrategy: RetryStrategyType
+reliability
0..1
+payload
0..1
«abstract»
Message::CoreEntityPayload
+ InformationSecurityLevel: InformationSecurityLevelType
+ InformationSensitivity: InformationSensitivityType
+ IsPersonalData: boolean
+ Purpose: PurposeType
+ RetentionPeriod: XSD::DateTime
Message::XmlEntityPayload
+ EnsureEncryption: boolean [0..1]
«datamodel»
+ Entities: XSD::anyType [1..*]
Message::EncryptedEntityPayload
+ Entities: String
«abstract»
Message::Message
+ ContextID: String [0..1]
+ CorrelationID: String [0..1]
+ CreationDateTime: XSD::DateTime
+ MessageID: String
+ Priority: PriorityType
+sender
1
+recipient
1
+ccRecipients
0..*
Service::Service
+ ServiceID: String
+ ServiceOperation: ServiceOperationType [0..1]
+ ServiceStatus: ServiceStatusType [0..1]
+ ServiceType: ServiceType [0..1]
«datamodel»
+ EntityTemplate: XSD::anyType [0..1]
+gateway
0..1
Service::Gateway
+ GatewayID: String [0..1]
+ GatewayStatus: GatewayStatusType [0..1]
+ PhysicalAddress: String
+discoveredServices
0..*
+system
0..1
Authority::LegacySystem
+ Community: CommunityType [0..*]
+ Function: FunctionType [0..*]
+ Name: String [0..1]
+ SeaBasin: SeaBasinType [0..*]
+ SystemID: String [0..1]
Message::Feedback
+ FeedbackType: FeedbackType
+ Reason: String [0..1]
+ RefMessageID: String
Message::PullRequest
+ PullType: PullType
+ ResponseTimeOut: int [0..1]
Message::Acknowledgement
+ AckCode: AcknowledgementType
+ AckDetail: String [0..1]
+provider
0..*
Authority::Authority
+ AuthorityID: String [0..1]
+ Community: CommunityType [0..*]
+ Country: String [0..*]
+ Function: FunctionType [0..*]
+ Name: String [0..1]
+ SeaBasin: SeaBasinType [0..*]
Message::Push
Message::PullResponse
+ ErrorDetail: String [0..1]
+ ResultCode: ResponseCodeType
+fulfils
0..1
+requests
0..1
Service::ServiceCapability
+ MaxEntitiesPerMsg: int [0..1]
+ QueryByExampleType: QueryByExampleType
Service::SubscriptionCapability
+ AvgUpdateFrequency: XSD::Duration [0..1]
+ MaxFrequency: XSD::Duration [0..1]
+ SubscriptionEnd: XSD::DateTime
+discoveryProfiles
0..*
+discoveryProfiles
0..*
Service::ServiceProfile
+ Community: CommunityType [0..1]
+ Country: String [0..1]
+ DataFreshness: DataFreshnessType [0..1]
+ Function: FunctionType [0..1]
+ SeaBasin: SeaBasinType [0..1]
+ ServiceType: ServiceType [0..*]

In each step of the communication process (e.g., message creation, routing, service discovery, authorization, etc.), messages will only contain some part of the information of the model. For example, during the service discovery process or the acknowledgment process, messages do not need a payload. Other properties, such as those related to addressing, are always used.

4.3 Business Rules

4.3.1 The Use of Message Identifiers

Message objects can contain three types of identifiers:

- *MessageID*: Identifier of the message. It is unique for the CISE participant who created the message.
- *CorrelationID*: This identifier correlates the request and response messages of/to a service (for the Pull or the Publish/Subscribe communication patterns)
- *ContextID*: This identifier correlates the messages that share an operational need. For instance, in order to update the information of an incident, several CISE entity services need to be invoked (e.g., IncidentService, EventLocationService, etc.) Thus, the messages exchanged with these CISE entity services are related by this ID type.

Annex D describes more in detail how the ContextID can be used to update the Maritime Situation.

4.3.2 The Acknowledgement Mechanism

The acknowledgement mechanism between CISE participants is asynchronous.

An Acknowledgement message is sent back to the sender every time a message PullRequest, PullResponse or Push is delivered to the final destination. It notifies the sender if the message was successfully delivered or not after the CISE GW applied the error handling mechanisms specified by the retry strategy.

A detailed sequence diagram of the acknowledgement mechanism can be found in Section C.1.

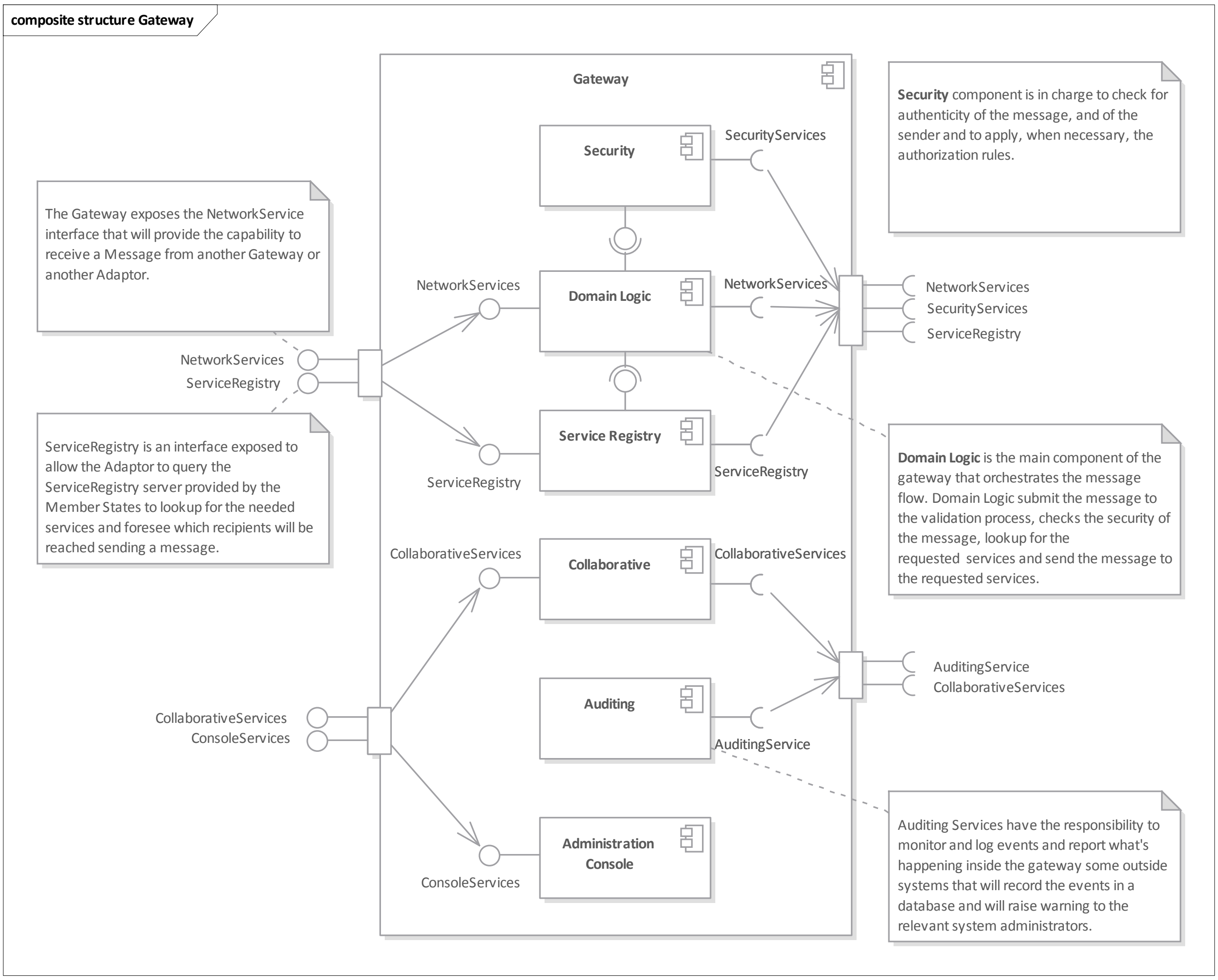
4.3.3 Giving Feedback on the Content

To do this, CISE participants should use the Feedback message.

For more details about the feedback mechanism, please check Section C.6.

4.4 Software Architecture

4.4.1 Gateway



The Gateway exposes the NetworkService interface that will provide the capability to receive a Message from another Gateway or another Adaptor.

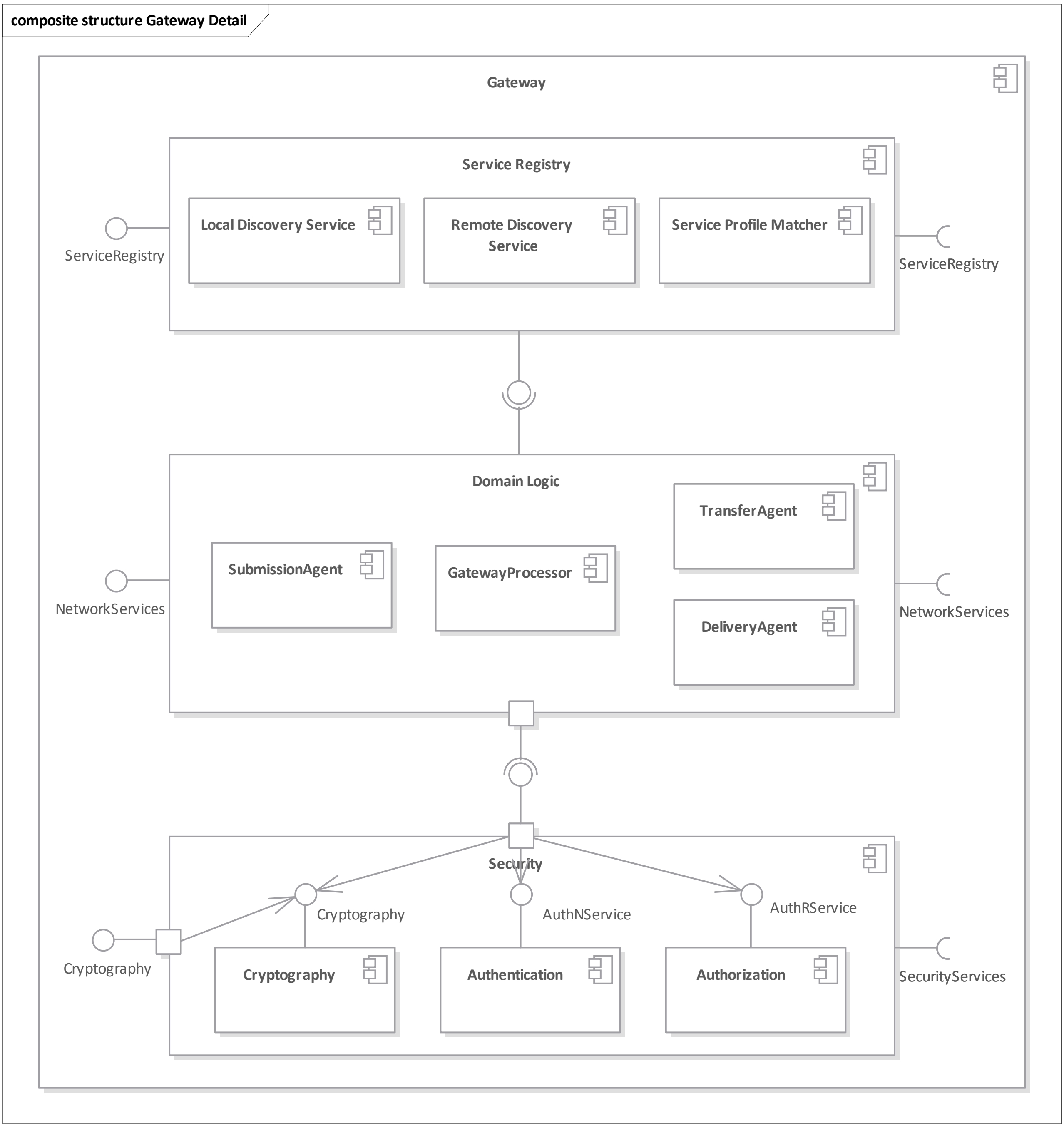
ServiceRegistry is an interface exposed to allow the Adaptor to query the ServiceRegistry server provided by the Member States to lookup for the needed services and foresee which recipients will be reached sending a message.

Security component is in charge to check for authenticity of the message, and of the sender and to apply, when necessary, the authorization rules.

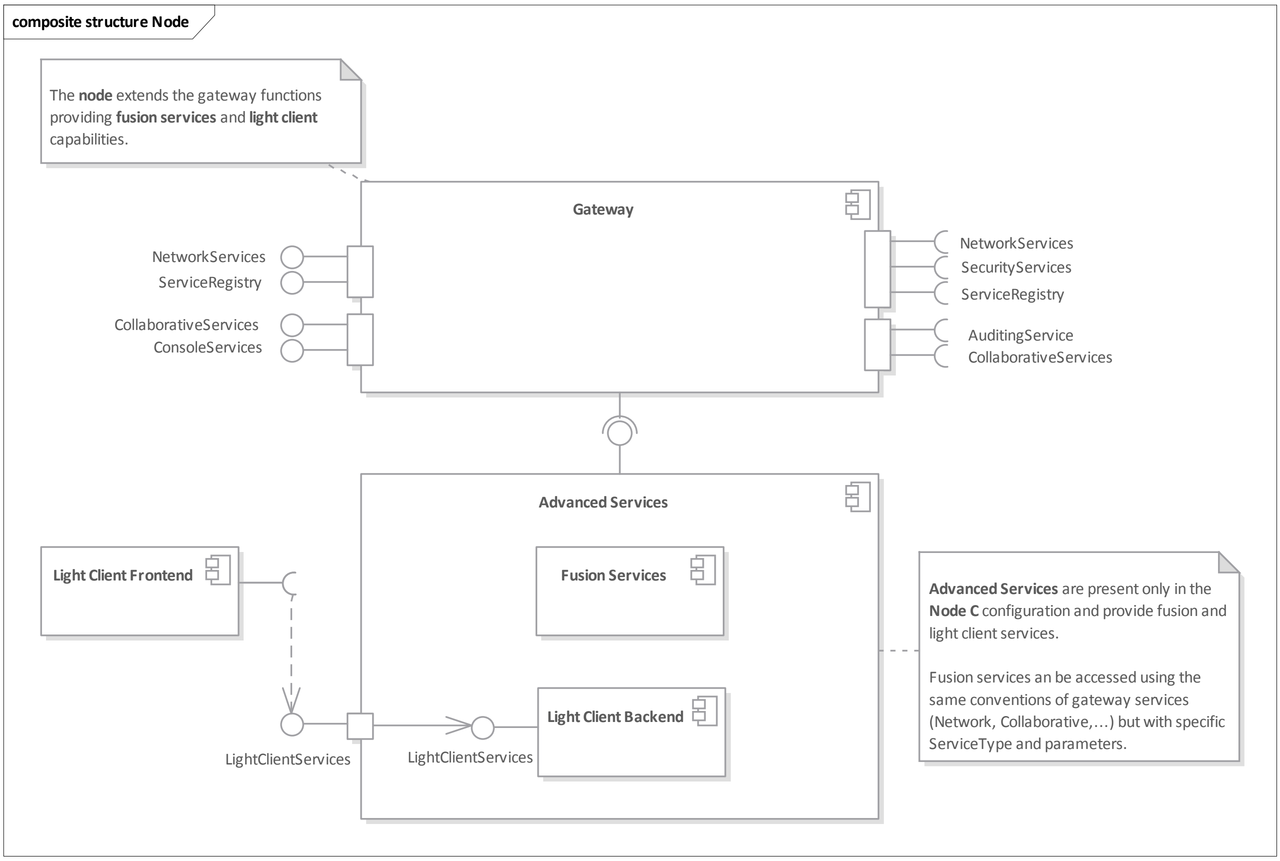
Domain Logic is the main component of the gateway that orchestrates the message flow. Domain Logic submit the message to the validation process, checks the security of the message, lookup for the requested services and send the message to the requested services.

Auditing Services have the responsibility to monitor and log events and report what's happening inside the gateway some outside systems that will record the events in a database and will raise warning to the relevant system administrators.

4.4.1.1 Gateway Detail



4.4.2 Node

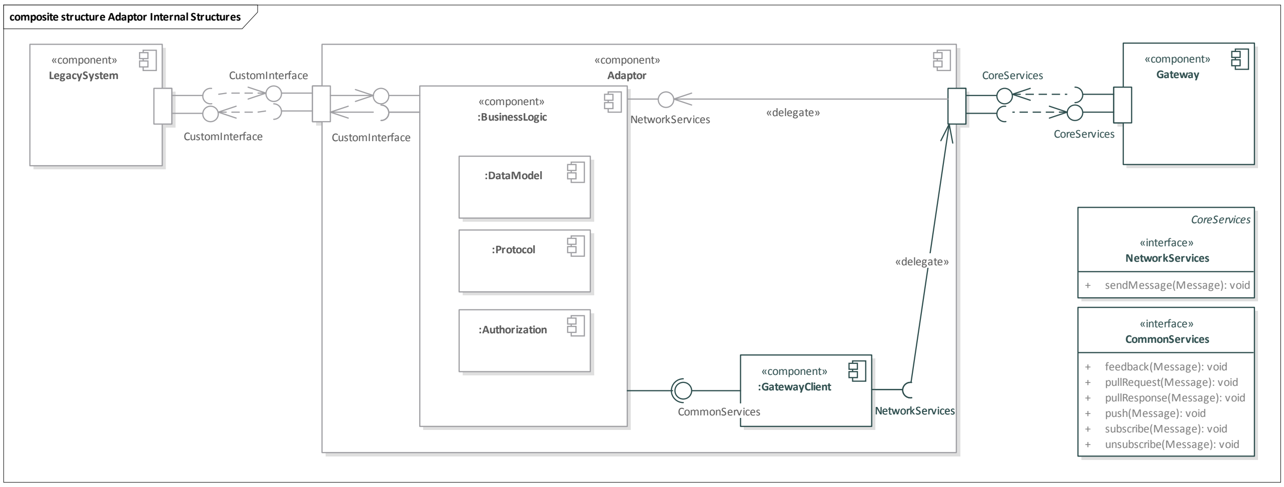


The **node** extends the gateway functions providing **fusion services** and **light client** capabilities.

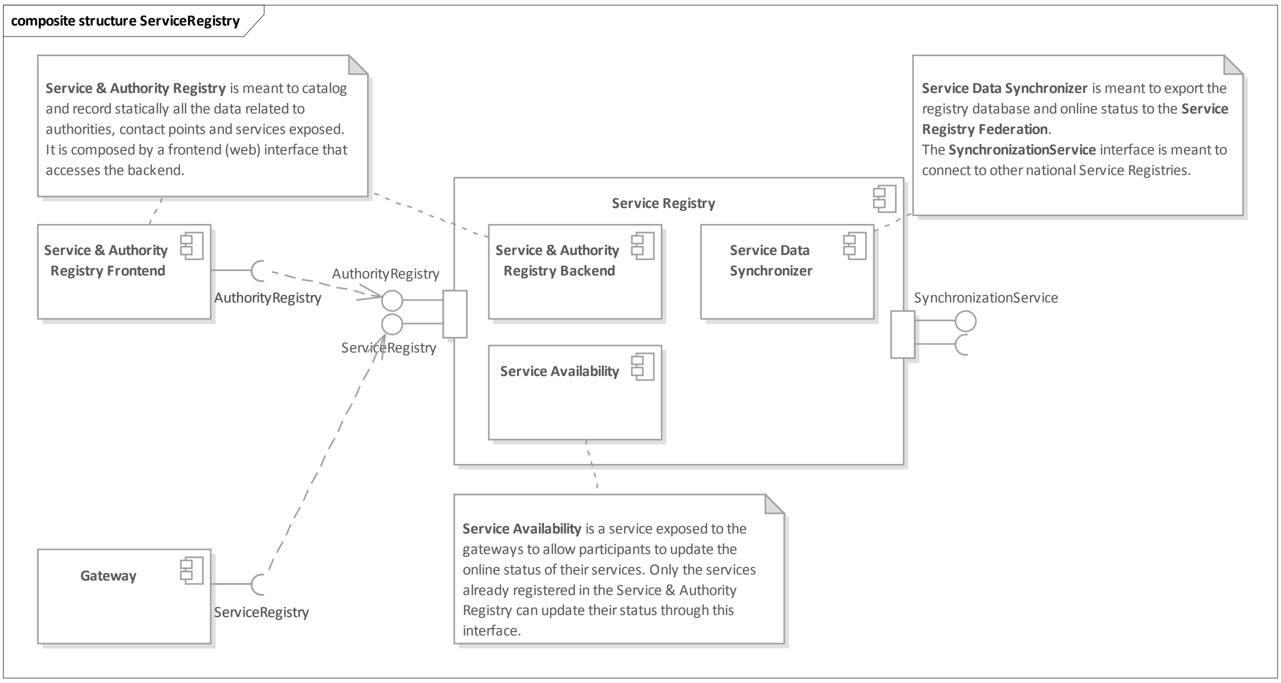
Advanced Services are present only in the **Node C** configuration and provide fusion and light client services.

Fusion services an be accessed using the same conventions of gateway services (Network, Collaborative,…) but with specific ServiceType and parameters.

4.4.3 Adaptor



4.4.4 Service Registry


composite structure ServiceRegistry
Service & Authority Registry is meant to catalog and record statically all the data related to authorities, contact points and services exposed. It is composed by a frontend (web) interface that accesses the backend.
Service Data Synchronizer is meant to export the registry database and online status to the **Service Registry Federation**. The **SynchronizationService** interface is meant to connect to other national Service Registries.
Service Registry
Service & Authority Registry Frontend
AuthorityRegistry
AuthorityRegistry
Service & Authority Registry Backend
Service Data Synchronizer
SynchronizationService
ServiceRegistry
Service Availability
Gateway
ServiceRegistry
Service Availability is a service exposed to the gateways to allow participants to update the online status of their services. Only the services already registered in the Service & Authority Registry can update their status through this interface.

Bibliography

[1] Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions, Gregor Hohpe, Bobby Woolf, Addison-Wesley, 2004

[2] Patterns for Service-Oriented Information Exchange Requirements, Ayman Mahfouz, Leonor Barroca, Robin Laney, Bashar Nuseibeh, ACM Conference Proceeding, 2006

[3] Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma… [et al.], Addison-Wesley Professional Computing Series, 2000

Annex A. Further improvement required to this document

- The reliability profile (including the retry mechanism) should be further detailed
- Error management in the service model
- Example of message exchanged should be added
- In section 4.1, add a table summarizing the Gateway-to-Adaptor Interface and the Gateway-to-Registry Interface
- Annex C: add XML examples
- Annex C4, C5 and C6 to update

Annex B. Service Types in the CISE Service Model

Service Type	Description
ActionService	This service exchanges data objects of the Action data entity[1].
AgentService	This service exchanges the data objects of the Agent data entity[1].
AircraftService	This service exchanges the data objects of the Aircraft data entity[1].
AnomalyService	This service exchanges the data objects of the Anomaly data entity[1].
CargoDocumentService	This service exchanges the data objects of the CargoDocument data entity[1].
CargoService	This service exchanges the data objects of the CargoUnit data entity[1].
CertificateDocumentService	This service exchanges the data objects of the CertificateDocument data entity[1].
CrisisIncidentService	This service exchanges the data objects of the CrisisIncident data entity[1].
DocumentService	This service exchanges the data objects of the Document data entity[1].
EventDocumentService	This service exchanges the data objects of the EventDocument data entity[1].
IncidentService	This service exchanges the data objects of the Incident data entity[1].
IrregularMigrationIncidentService	This service exchanges the data objects of the IrregularMigrationIncident data entity[1].
LandVehicleService	This service exchanges the data objects of the LandVehicle data entity[1].
LawInfringementIncidentService	This service exchanges the data objects of the LawInfringementIncident data entity[1].
LocationService	This service exchanges the data objects of the Location data entity[1].
LocationDocumentService	This service exchanges the data objects of the LocationDocument data entity[1].
MaritimeSafetyIncidentService	This service exchanges the data objects of the MaritimeSafetyIncident data entity[1].
MeteoOceanographicObservationService	This service exchanges the data objects of the MeteoOceanographicObservation data entity[1].
MovementService	This service exchanges the data objects of the Movement data entity[1].
OperationalAssetService	This service exchanges the data objects of the OperationalAsset data entity[1].
OrganizationService	This service exchanges the data objects of the Organization data entity[1].
OrganizationDocumentService	This service exchanges the data objects of the OrganizationDocument data entity[1].
PersonService	This service exchanges the data objects of the Person data entity[1].
PersonDocumentService	This service exchanges the data objects of the PersonDocument data entity[1].
RiskDocumentService	This service exchanges the data objects of the RiskDocument data entity[1].
RiskService	This service exchanges the data objects of the Risk data entity[1].
VesselDocumentService	This service exchanges the data objects of the VesselDocument data entity[1].

[1] The service could also exchange data objects of an associated data entity if supported by the information provider.

Service Type	Description
VesselService	This service exchanges the data objects of the Vessel data entity[1].

Annex C. Information Exchange Patterns

C.1. Pull Pattern

The following UML sequence schema describes the message exchange between the CISE participants using the Pull Communication Pattern.

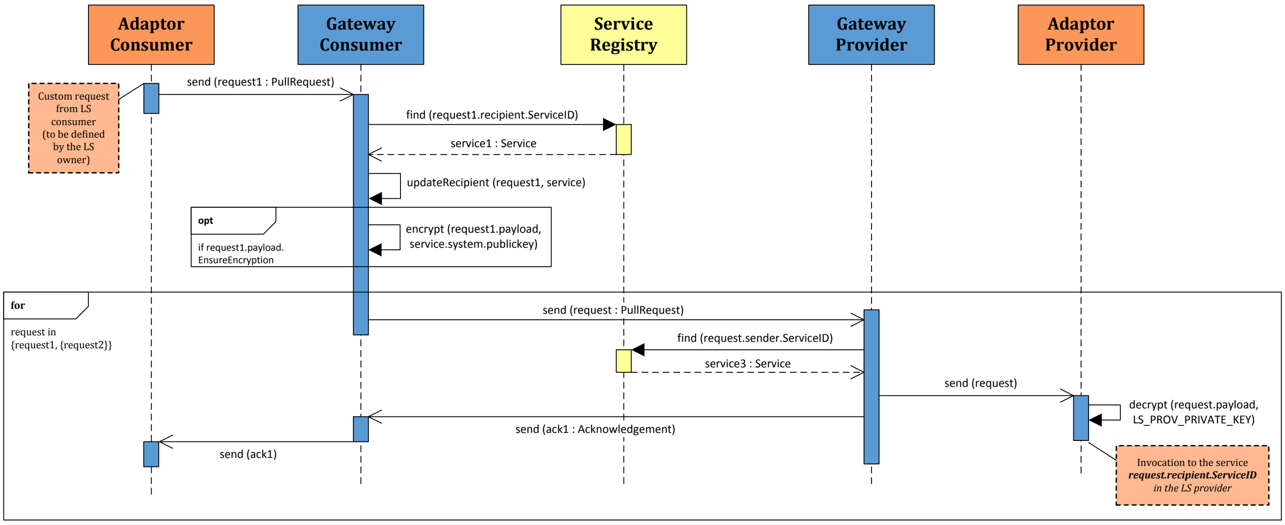

Adaptor Consumer
Gateway Consumer
Service Registry
Gateway Provider
Adaptor Provider
Custom request from LS consumer (to be defined by the LS owner)
send (request1 : PullRequest)
find (request1.recipient.ServiceID)
service1 : Service
updateRecipient (request1, service)
opt
if request1.payload. EnsureEncryption
encrypt (request1.payload, service.system.publickey)
for
request in {request1, {request2}}
send (request : PullRequest)
find (request.sender.ServiceID)
service3 : Service
send (request)
decrypt (request.payload, LS_PROV_PRIVATE_KEY)
send (ack1 : Acknowledgement)
send (ack1)
Invocation to the service request.recipient.ServiceID in the LS provider

Figure 9: PullRequest (to a known provider)

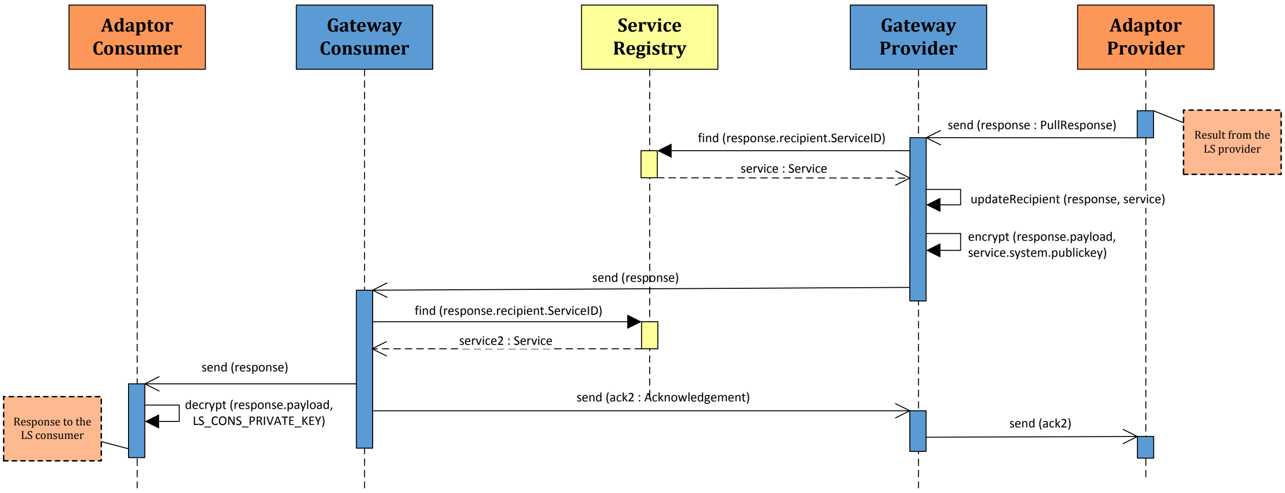

Adaptor Consumer
Gateway Consumer
Service Registry
Gateway Provider
Adaptor Provider
send (response : PullResponse)
Result from the LS provider
find (response.recipient.ServiceID)
service : Service
updateRecipient (response, service)
encrypt (response.payload, service.system.publickey)
send (response)
find (response.recipient.ServiceID)
service2 : Service
send (response)
decrypt (response.payload, LS_CONS_PRIVATE_KEY)
Response to the LS consumer
send (ack2 : Acknowledgement)
send (ack2)

Figure 10: PullResponse from the provider

Use Case:

Guardia Civil (Spain, **LS Consumer**) needs to request information about a vessel named *QUEEN MARY* to **Marina Militare Italiana** (Italy, **LS Provider**).

Process:

Step	Description	Messages Exchanged
1	Guardia Civil sends a PullRequest message to Marina Militare Italiana that specifies the name of the vessel of interest.	request1, request2, request3: PullRequest

2	Guardia Civil receives an Acknowledgement message indicating that the message was delivered to Marina Militare Italiana.	ack1, ack2: Acknowledgement
3	Marina Militare Italiana sends a **PullResponse** message to Guardia Civil with the information contained in the system.	response1, response2, response3: PullResponse
4	Marina Militare Italiana receives an Acknowledgement message when the PullResponse message is delivered to Guardia Civil.	ack3, ack4: Acknowledgement

Messages:

1. request1 PullRequest Message

The LS Consumer (Guardia Civil) creates the request1 message and sends it to the GW Consumer.

Attributes:

- **MessageID**: a unique identifier, issued by the LS Consumer, i.e., Guardia Civil.
- **CorrelationID**: an identifier that correlates the request to the response(s) and the acknowledgements, issued by the LS Consumer, i.e., Guardia Civil.
- **sender/service/ServiceID**: Identifier of the callback service (for responses and acknowledgements), e.g., the ID of the Guardia Civil Service
- **recipients/service/ServiceID**: Identifier of the provider service, e.g., the ID of the Marina Militare Service.
- **payload/Entities**: it contains the entities given as example for the request, specified as described [ref QueryByExample]

Note: The *LS Consumer* only specifies the sender's and receiver's **serviceIDs**. The *GW Consumer* will subsequently discover the information of the *GW Provider* offering the *LS Provider Service*.

2. request2 PullRequest Message

The GW Consumer discovers the GW Provider address/id through the Discovery Service and delivers the request2 PullRequest message to the final destination.

Attributes:

- Same as request1
- **sender/node/NodeID**: Identification of the GW Consumer
- **sender/node/NodeAddress**: Network address of the GW Consumer
- **recipients/node/NodeID**: Identification of the GW Provider
- **recipients/node/NodeAddress**: Network address of the GW Provider

3. request3 PullRequest Message

Same as the request2 message.

4. *ack1 Acknowledgement Message*

The delivery of the request3 *PullRequest message* to the *LS Provider* triggers an acknowledgement process in the GW Provider to inform the *LS Consumer*.

Attributes:

- **AckCode**: a code to detail the result of the operation.
- **AckDetail**: additional information about the result (success or error details).
- **sender/service/ServiceID**: the service identifier that is sending the acknowledgement.
- **recipients/service/ServiceID**: the service identifier that will receive the acknowledgment. This corresponds to the senderID specified in the PullRequest messages (request1, request2 or request3).
- **MessageID**: is a unique identifier for the acknowledgment message.
- **CorrelationID**: CorrelationID specified in the PullRequest messages (request1, request2 or request3).

5. *ack2 Acknowledgement Message*

Same as the ack1 message.

6. *response1 PullResponse Message*

The *LS Provider* creates a *PullResponse message* with some relevant information answering the request from the LS Consumer. The transmission of the *PullResponse message* follows the same process that the PullRequest Message from the LS Consumer.

Attributes:

- **MessageID**: a unique identifier issued by *Marina Militare Service*(the originator LS Provider)
- **CorrelationID**: CorrelationID specified in the PullRequest messages (request1, request2 or request3).
- **sender/service/ServiceID**: the *Marina Militare Service* specific identifier
- **recipients/service/ServiceID**: the *Guardia Civil Service* specific identifier
- **payload/Entities**: it contains the result of the query

7. *response2 PullResponse Message*

Attributes:

- Same as response1
- **sender/node/NodeID**: Identification of the GW Provider
- **sender/node/NodeAddress**: Network address of the GW Provider
- **recipients/node/NodeID**: Identification of the GW Consumer
- **recipients/node/NodeAddress**: Network address of the GW Consumer

8. *response3 PullResponse Message*

Same as the response2 message.

9. *ack3 Acknowledgement Message*

The delivery of the response3 PullResponse message to the LS Consumer triggers an acknowledgement process in the GW Consumer to inform the LS Provider.

10. *ack4 Acknowledgement Message*

Same as the ack3 message.

C.2. Multicast Pull Pattern

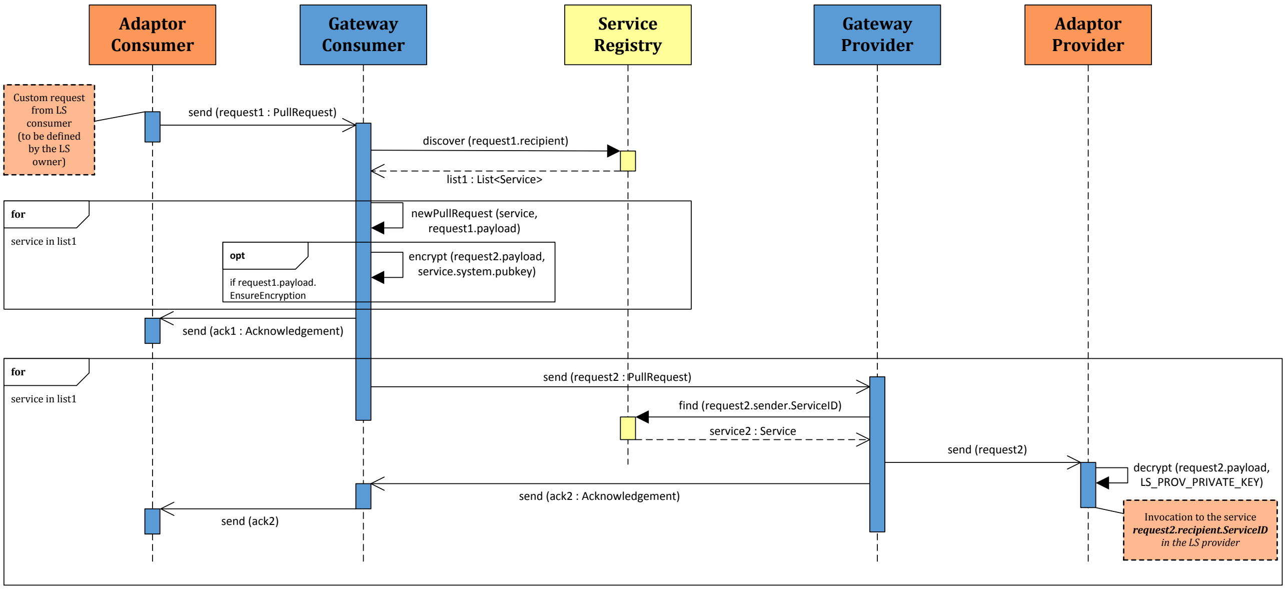

Adaptor Consumer
Gateway Consumer
Service Registry
Gateway Provider
Adaptor Provider
Custom request from LS consumer (to be defined by the LS owner)
send (request1 : PullRequest)
discover (request1.recipient)
list1 : List<Service>
for
service in list1
newPullRequest (service, request1.payload)
opt
if request1.payload. EnsureEncryption
encrypt (request2.payload, service.system.pubkey)
send (ack1 : Acknowledgement)
for
service in list1
send (request2 : PullRequest)
find (request2.sender.ServiceID)
service2 : Service
send (request2)
decrypt (request2.payload, LS_PROV_PRIVATE_KEY)
send (ack2 : Acknowledgement)
send (ack2)
Invocation to the service request2.recipient.ServiceID in the LS provider

Figure 11: PullRequest (to unknown provider)

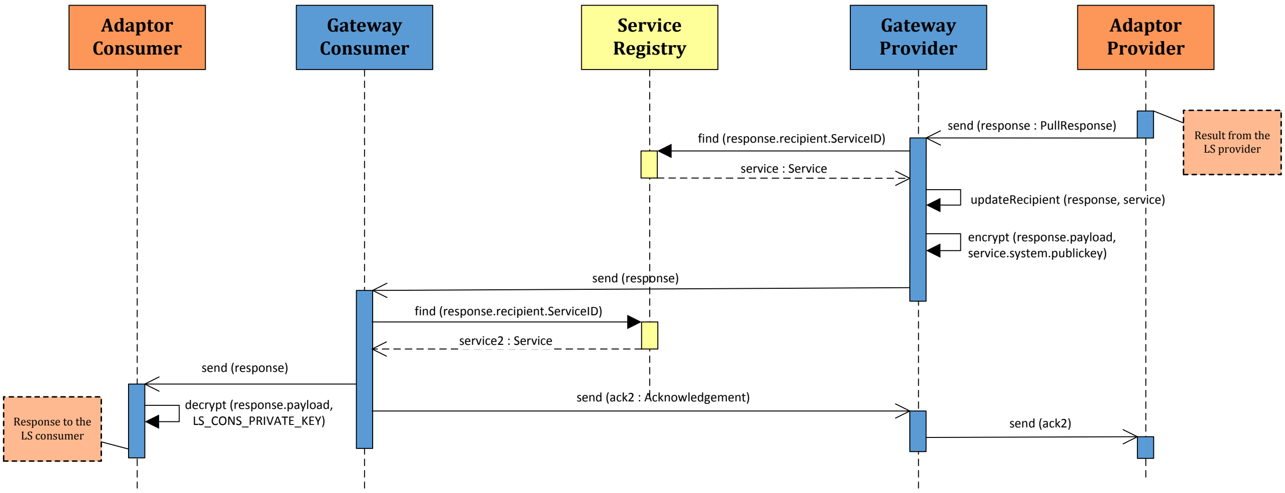

Adaptor Consumer
Gateway Consumer
Service Registry
Gateway Provider
Adaptor Provider
send (response : PullResponse)
Result from the LS provider
find (response.recipient.ServiceID)
service : Service
updateRecipient (response, service)
encrypt (response.payload, service.system.publickey)
send (response)
find (response.recipient.ServiceID)
service2 : Service
send (response)
decrypt (response.payload, LS_CONS_PRIVATE_KEY)
Response to the LS consumer
send (ack2 : Acknowledgement)
send (ack2)

Figure 12: PullResponse (for each provider)

C.3. Push Pattern

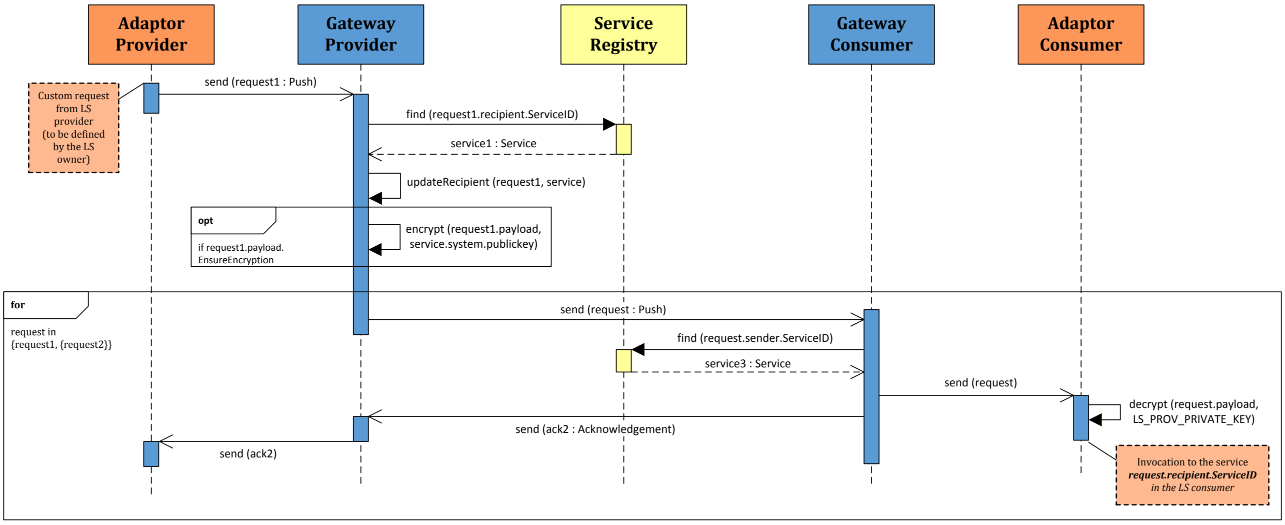

Adaptor Provider
Gateway Provider
Service Registry
Gateway Consumer
Adaptor Consumer
Custom request from LS provider (to be defined by the LS owner)
send (request1 : Push)
find (request1.recipient.ServiceID)
service1 : Service
updateRecipient (request1, service)
opt
if request1.payload. EnsureEncryption
encrypt (request1.payload, service.system.publickey)
for
request in {request1, {request2}}
send (request : Push)
find (request.sender.ServiceID)
service3 : Service
send (request)
decrypt (request.payload, LS_PROV_PRIVATE_KEY)
send (ack2 : Acknowledgement)
send (ack2)
Invocation to the service
request.recipient.ServiceID
in the LS consumer

Figure 13: Push (to a known consumer)

Use Case:

Marina Militare Italiana (Italy, **LS Provider**) needs to provide an event information to **Guardia Civil** (Spain, **LS Consumer**) about the vessel named *QUEEN MARY* involved into an incident.

Process:

Step	Description	Messages Exchanged
1	**Marina Militare Italiana** send to **Guardia Civil** a *push* message containing a payload with an event entity that contains information about the incident and one vessel involved in.	notification1: Push
2	**Marina Militare Italiana** receives an *Acknowledgement* message indicating that the message was delivered to **Guardia Civil**.	ack1, ack2: Acknowledgement
3	**Guardia Civil** receive a *push* message containing the event information	notification3: Push

Messages:

1. notification1: Push Message

The **LS Provider** creates the notification1 message and sends it to the **GW Provider.**

Attributes:

- **MessageID**: a unique identifier, issued by the **LS Provider**;
- **CorrelationID**: an identifier that correlates the message and the acknowledgements / feedback, issued by the **LS Provider**;
- **sender/service/ServiceID**: Identifier of the callback service (for acknowledgements);
- **recipients/service/ServiceID**: Identifier of the consumer service;
- **payload/Entities**: it contains a specific entity the LS consumer service will understand (i.e. *ObjectEventRel*).

2. notification2: Push Message

The **GW Provider** discovers the **GW Consumer** address/id through the Discovery Service and delivers the notification2 Push message to the final gateway (**GW Consumer**).

Attributes: see Annex D.

3. notification3: Push Message

The **GW Consumer** delivers the notification3 Push message to the final destination (**LS Consumer**).

Attributes: same as ***notification2: Push Message****.*

4. ack1: Acknowledgement Message

The delivery of the notification3 *Push message* to the **LS Consumer** triggers an acknowledgement process in the **GW Consumer** to inform the **LS Provider** about the delivery status of the message.

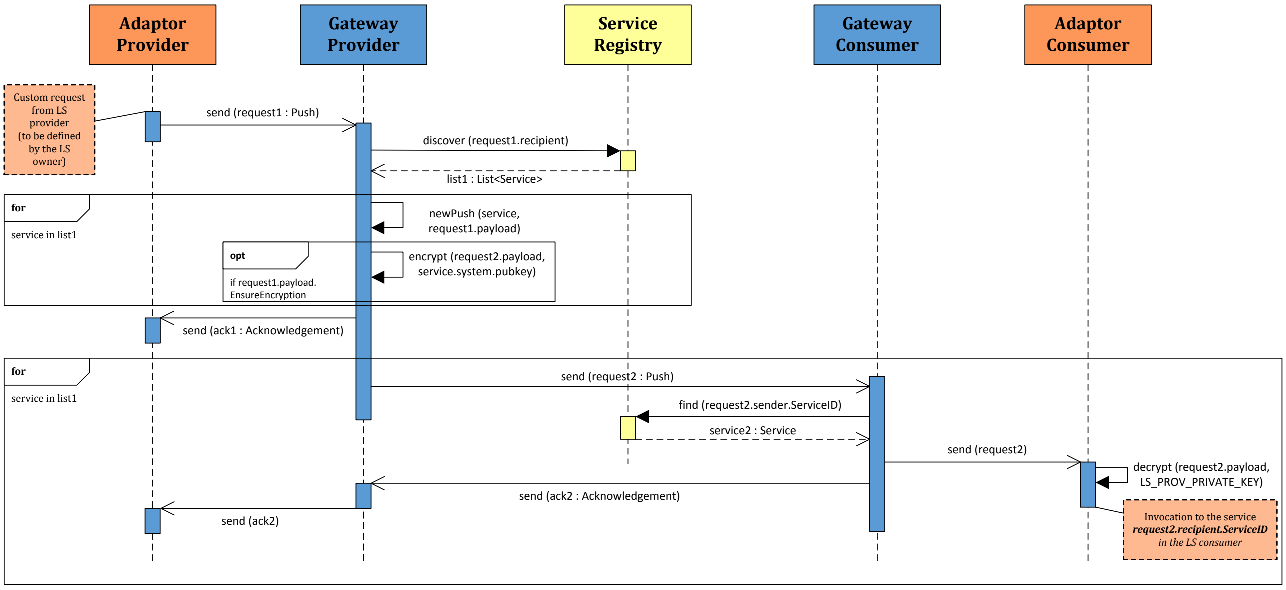
Attributes: for the main structure and fields of the ack message, please consult Annex D.

5. ack2: Acknowledgement Message

The delivery of the notification3 *Push message* to the **LS Consumer** is delivered to the **LS Provider** by **GW Consumer.**

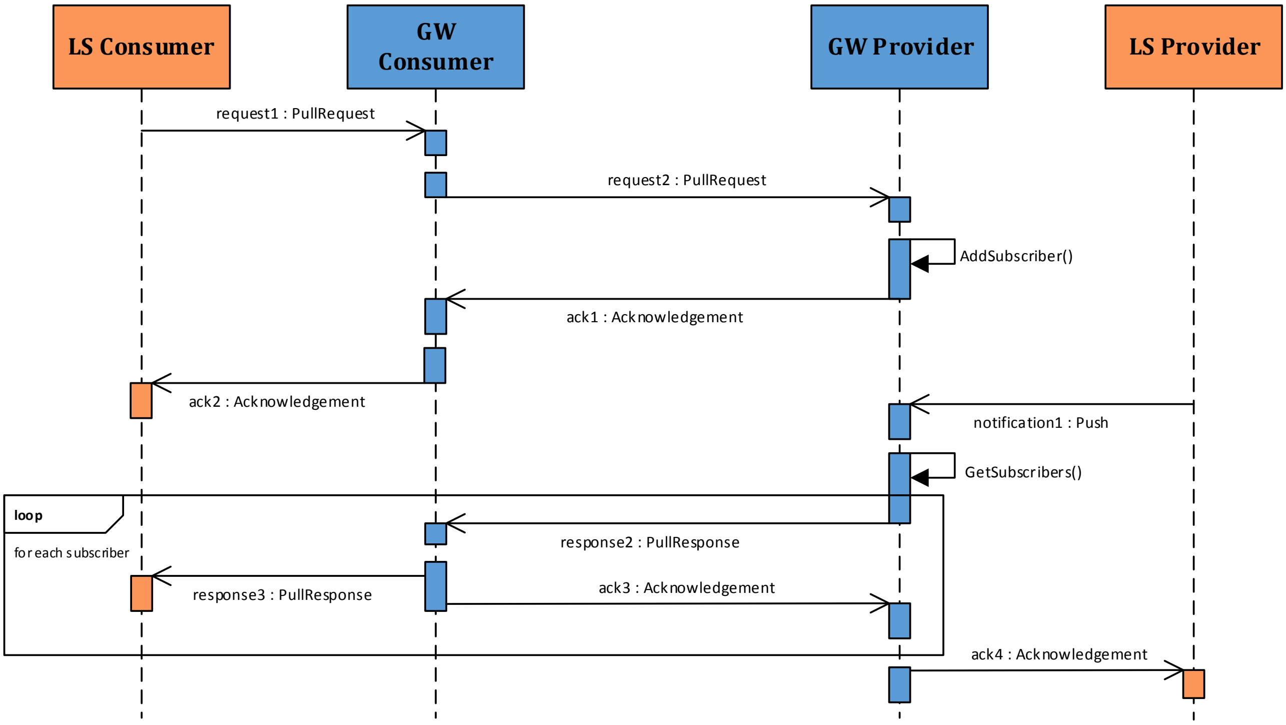
Attributes: for the main structure and fields of the ack message, please consult see Annex D.

C.4. Multicast Push Pattern

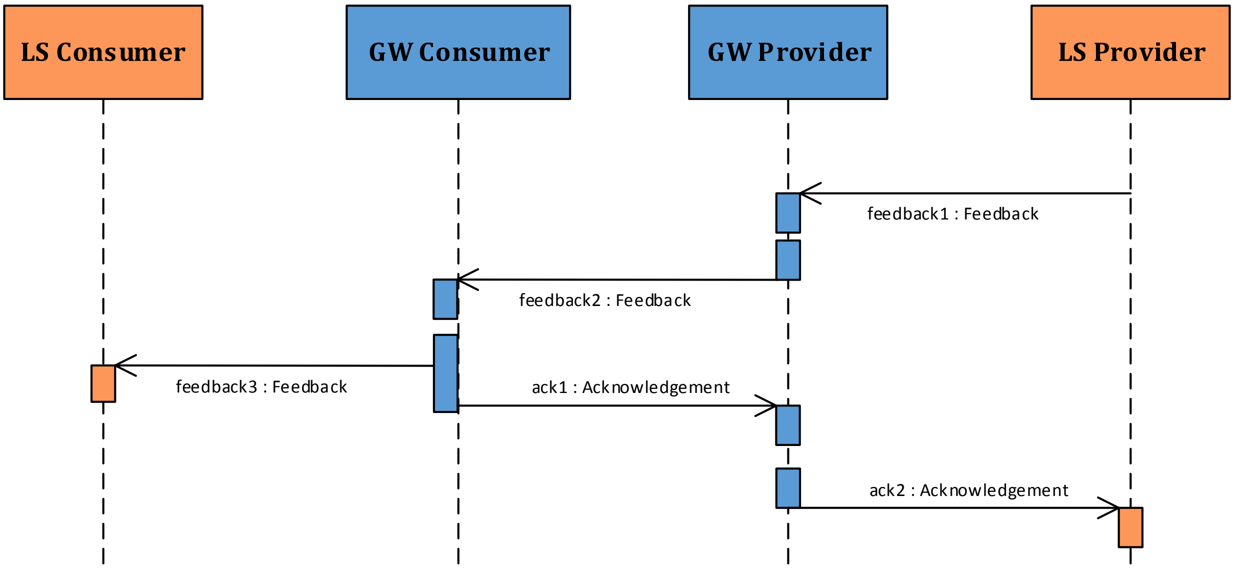


Adaptor Provider
Gateway Provider
Service Registry
Gateway Consumer
Adaptor Consumer
Custom request from LS provider (to be defined by the LS owner)
send (request1 : Push)
discover (request1.recipient)
list1 : List<Service>
for
service in list1
newPush (service, request1.payload)
opt
if request1.payload. EnsureEncryption
encrypt (request2.payload, service.system.pubkey)
send (ack1 : Acknowledgement)
for
service in list1
send (request2 : Push)
find (request2.sender.ServiceID)
service2 : Service
send (request2)
decrypt (request2.payload, LS_PROV_PRIVATE_KEY)
send (ack2 : Acknowledgement)
send (ack2)
Invocation to the service
request2.recipient.ServiceID
in the LS consumer

C.5. Publish/Subscribe Pattern



C.6. Feedback Message Flow (based on PUSH scenario)



Use Case:

Marina Militare Italiana (Italy, **LS Provider**) needs to provide additional feedback (e.g **delete)** information to **Guardia Civil** (Spain, **LS Consumer**) about the vessel named QUEEN MARY that was involved into the incident *incident_001* as described in the use case presented in section **Error! Reference source not found.Error! Reference source not found.**.

The reason of sending the feedback was that Marina Militare acknowledge the incident as not a real one but part of a planned exercise.

The information about the vessel was sent before using a **Push** message having (as described in **Error! Reference source not found.**) is uniquely identified by the:

- **MessageID=** `sdfsd72d-1400-4f28-943a-d77129afzzdd`

Process:

Step	Description	Messages Exchanged
1	**Marina Militare Italiana** send to **Guardia Civil** a *feedback* message related to a previously sent *Push* message (uniquely identified by MessageID).	feedback1, feedback2: Feedback
2	**Marina Militare Italiana** receives an *Acknowledgement* message indicating that the message was delivered to **Guardia Civil**.	ack1, ack2: Acknowledgement
3	**Guardia Civil** receive a feedback message about a previously received message (uniquely identified by MessageID).	feedback3: Feedback

Messages:

1. ***feedback1: Feedback Message***

The **LS Provider** creates the feedback1 message and sends it to the **GW Provider**.

Attributes:

- **MessageID**: a unique identifier, issued by the **LS Provider**;
- **CorrelationID**: an identifier that correlates the message and the acknowledgements / feedback, issued by the **LS Provider**;
- **sender/service/ServiceID**: Identifier of the call-back service (for acknowledgements);
- **recipients/service/ServiceID**: Identifier of the consumer service;
- **FeedbackType:** it describe the scope of the feedback from a list of values: *info* or *delete*;
- **RefMessageID:** specify an already message (MessageID) to what the feedback is provided
- **Reason:** a short user readable reason of the feedback context or reason

2. ***feedback2: Feedback Message***

The **GW Provider** discovers the **GW Consumer** address/id through the Discovery Service and delivers the feedback2 Feedback message to the final gateway (**GW Consumer**).

Attributes: see paragraph **Annex D.**

3. ***feedback3: Feedback Message***

The **GW Consumer** delivers the feedback3 Feedback message to the final destination (**LS Consumer**).

*Attributes: same as **notification2: Push Message**.*

4. ***ack1: Acknowledgement Message***

The delivery of the notification3 *Push message* to the **LS Consumer** triggers an acknowledgement process in the **GW Consumer** to inform the **LS Provider** about the delivery status of the message.

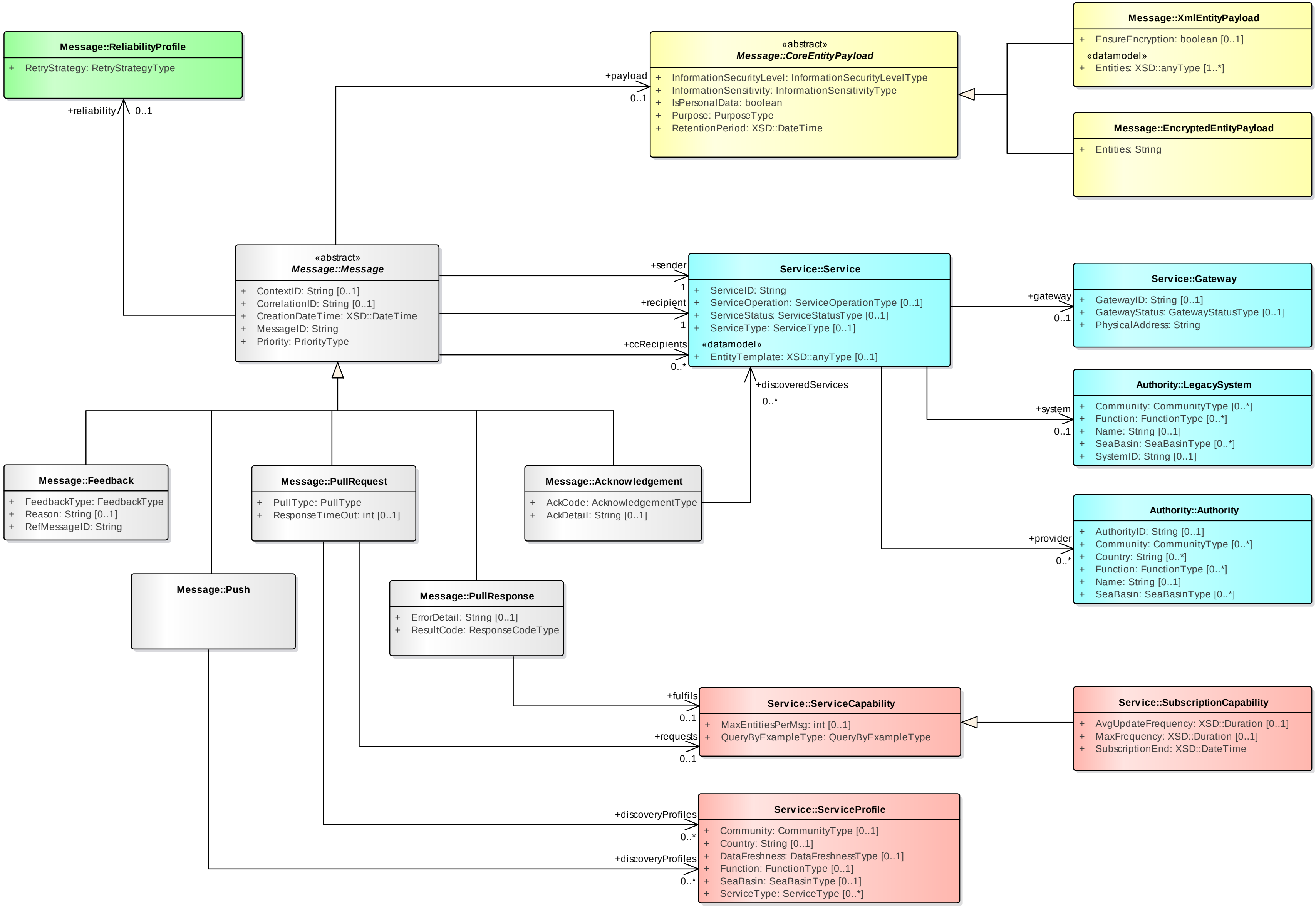
Attributes: for the main structure and fields of the ack message, please consult Annex D.

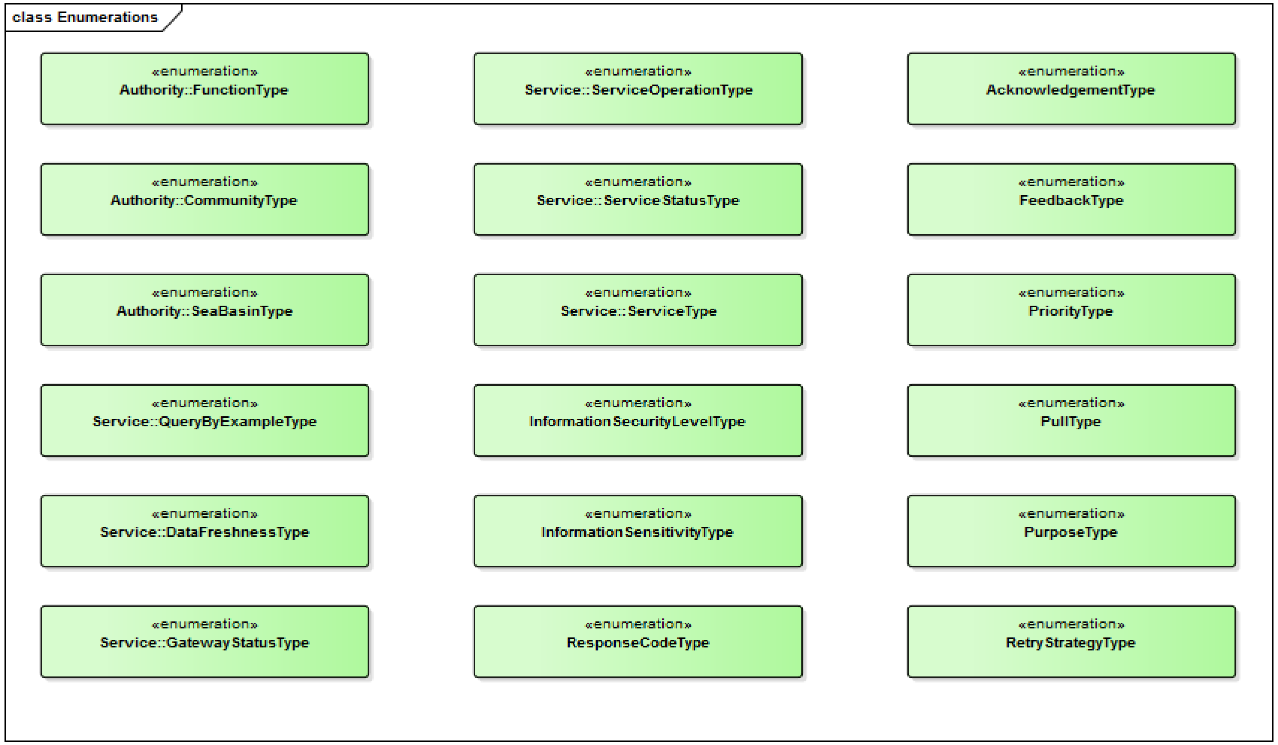
5. ***ack2: Acknowledgement Message***

The delivery of the notification3 *Push message* to the **LS Consumer** is delivered to the **LS Provider** by **GW Consumer**.

Attributes: for the main structure and fields of the ack message, please consult Annex D.

Annex D. Message Objects


Message::ReliabilityProfile
+ RetryStrategy: RetryStrategyType
+reliability
0..1
+payload
0..1
«abstract»
Message::CoreEntityPayload
+ InformationSecurityLevel: InformationSecurityLevelType
+ InformationSensitivity: InformationSensitivityType
+ IsPersonalData: boolean
+ Purpose: PurposeType
+ RetentionPeriod: XSD::DateTime
Message::XmlEntityPayload
+ EnsureEncryption: boolean [0..1]
«datamodel»
+ Entities: XSD::anyType [1..*]
Message::EncryptedEntityPayload
+ Entities: String
«abstract»
Message::Message
+ ContextID: String [0..1]
+ CorrelationID: String [0..1]
+ CreationDateTime: XSD::DateTime
+ MessageID: String
+ Priority: PriorityType
+sender
1
+recipient
1
+ccRecipients
0..*
Service::Service
+ ServiceID: String
+ ServiceOperation: ServiceOperationType [0..1]
+ ServiceStatus: ServiceStatusType [0..1]
+ ServiceType: ServiceType [0..1]
«datamodel»
+ EntityTemplate: XSD::anyType [0..1]
+gateway
0..1
Service::Gateway
+ GatewayID: String [0..1]
+ GatewayStatus: GatewayStatusType [0..1]
+ PhysicalAddress: String
+system
0..1
Authority::LegacySystem
+ Community: CommunityType [0..*]
+ Function: FunctionType [0..*]
+ Name: String [0..1]
+ SeaBasin: SeaBasinType [0..*]
+ SystemID: String [0..1]
+provider
0..*
Authority::Authority
+ AuthorityID: String [0..1]
+ Community: CommunityType [0..*]
+ Country: String [0..*]
+ Function: FunctionType [0..*]
+ Name: String [0..1]
+ SeaBasin: SeaBasinType [0..*]
+discoveredServices
0..*
Message::Feedback
+ FeedbackType: FeedbackType
+ Reason: String [0..1]
+ RefMessageID: String
Message::PullRequest
+ PullType: PullType
+ ResponseTimeOut: int [0..1]
Message::Acknowledgement
+ AckCode: AcknowledgementType
+ AckDetail: String [0..1]
Message::Push
Message::PullResponse
+ ErrorDetail: String [0..1]
+ ResultCode: ResponseCodeType
+fulfils
0..1
+requests
0..1
Service::ServiceCapability
+ MaxEntitiesPerMsg: int [0..1]
+ QueryByExampleType: QueryByExampleType
Service::SubscriptionCapability
+ AvgUpdateFrequency: XSD::Duration [0..1]
+ MaxFrequency: XSD::Duration [0..1]
+ SubscriptionEnd: XSD::DateTime
+discoveryProfiles
0..*
+discoveryProfiles
0..*
Service::ServiceProfile
+ Community: CommunityType [0..1]
+ Country: String [0..1]
+ DataFreshness: DataFreshnessType [0..1]
+ Function: FunctionType [0..1]
+ SeaBasin: SeaBasinType [0..1]
+ ServiceType: ServiceType [0..*]


class Enumerations
«enumeration»
Authority::FunctionType
«enumeration»
Service::ServiceOperationType
«enumeration»
AcknowledgementType
«enumeration»
Authority::CommunityType
«enumeration»
Service::ServiceStatusType
«enumeration»
FeedbackType
«enumeration»
Authority::SeaBasinType
«enumeration»
Service::ServiceType
«enumeration»
PriorityType
«enumeration»
Service::QueryByExampleType
«enumeration»
InformationSecurityLevelType
«enumeration»
PullType
«enumeration»
Service::DataFreshnessType
«enumeration»
InformationSensitivityType
«enumeration»
PurposeType
«enumeration»
Service::GatewayStatusType
«enumeration»
ResponseCodeType
«enumeration»
RetryStrategyType

Annex E. CISE Service Model Vocabulary Specification

E.1. CISE Core Data Specifications

E.1.1. Document Metadata

Document title	CISE Core Data Specifications
Publisher	DG JRC
Modified	2017-02-28 18:39:12
Version	1.5
Status	Under development

E.2. Authority Core Entity

E.2.1. UML models

E.2.2. Elements defined in the Core Vocabulary

E.2.2.1. Authority Class

Authority (-ies) providing the service.

E.2.2.1.1. Attributes

UML Name	Data type	Description	Example	Source
AuthorityID	String	Identification of the authority.	eu.cise.authority1	
Community	CommunityType	One of the 7 user communities participating to CISE. This field is related to the community of the provider of the service.	"02" for Customs	
Country	String	The Country of the provider of the service. The code ISO-3166-1 alpha-2 is used: 2 character country code.	"FR" for France.	
Function	FunctionType	The functions covered by the authority providing the service.	"Monitoring of security of ships"	
Name	String	Official name of the authority.		
SeaBasin	SeaBasinType	The sea basin covered by the authority.	"05" for Mediterranean	

E.2.2.2. LegacySystem Class

This entity represents information about the provider legacy system.

E.2.2.2.2. Attributes

UML Name	Data type	Description	Example	Source
Community	CommunityType	One of the 7 user communities participating to CISE. This field is related to the community of the provider of the service.	"02" for Customs	
Function	FunctionType	The functions covered by the system providing the service.	"Monitoring of security of ships"	
Name	String	Name of the legacy system.		
SeaBasin	SeaBasinType	The sea basin covered by the legacy system.	"05" for Mediterranean	
SystemID	String	Identification of the system.	eu.cise.authority1.system1	

E.2.2.3. CommunityType Enumeration

Enumeration of the different communities that can belong a CISE participant

E.2.2.3.3. Enumeration Values

Value	Label	Description	Source
GeneralLawEnforcement		General Law Enforcement	
Customs		Customs	
MarineEnvironment		Marine Environment	
MaritimeSafetySecurity		Maritime Safety and Security	
DefenceMonitoring		Defence	
FisheriesControl		Fisheries control	
BorderControl		Border control	
Other		Other	
NonSpecified		Non-specified	

E.2.2.3.4. Enumeration Usage

The following attributes use this enumeration as data type:

- Community (Authority)
- Community (LegacySystem)
- Community (ServiceProfile)

E.2.2.4. FunctionType Enumeration

Enumeration of the different functions covered by a CISE participant.

E.2.2.4.5. Enumeration Values

Value	Label	Description	Source
VTM		Vessel traffic management	
Safety		Vessel Traffic Safety	

Value	Label	Description	Source
Security		Monitoring of security of ships	
SAR		Search and Rescue	
Operation		Support of response and enforcement operations (anti-piracy, SAR, salvage)	
FisheriesWarning		Early warning of illegal fisheries or fish landings,	
FisheriesMonitoring		Monitoring of compliance with regulations on fisheries	
FisheriesOperation		Support of response and enforcement operations	
EnvironmentMonitoring		Monitoring of compliance with regulations	
EnvironmentWarning		Early warning of environmental accidents and incidents	
EnvironmentResponse		Support of pollution response operations	
CustomsMonitoring		Monitoring of compliance with customs regulation on import, export and movement of goods	
CustomsOperation		Support of enforcement operations	
BorderMonitoring		Monitoring of compliance with regulations on immigration and border control crossings	
BorderOperation		Support of enforcement operations	
LawEnforcementMonitoring		Monitoring of compliance with applicable legislation in sea areas where police competence is required	
LawEnforcementOperation		Support to enforcement and response operations	
DefenceMonitoring		Monitoring in support of defence tasks such as national sovereignty at sea	
CounterTerrorism		Combatting terrorism and other hostile activities outside the EU	
CSDPTask		Other CSDP tasks as defined in Articles 42 and 43 TEU	

E.2.2.4.6. Enumeration Usage

The following attributes use this enumeration as data type:

- Function (LegacySystem)
- Function (ServiceProfile)
- Function (Authority)

E.2.2.5. SeaBasinType Enumeration

Enumeration of the different European Sea Basins.||This enumeration comes from http://ec.europa.eu/maritimeaffairs/atlas/seabasins/index_en.htm

E.2.2.5.7. Enumeration Values

Value	Label	Description	Source
BalticSea		Baltic Sea	
NorthSea		North Sea	
CelticSea		Celtic Sea	

Value	Label	Description	Source
BiscayBay		Bay of Biscay and the Iberian coast	
Mediterranean		Mediterranean	
BlackSea		Black Sea	
OutermostRegions		Outermost regions	
ArcticOcean		Arctic Ocean	

E.2.2.5.8. Enumeration Usage

The following attributes use this enumeration as data type:

- SeaBasin (Authority)
- SeaBasin (ServiceProfile)
- SeaBasin (LegacySystem)

E.3. Message Core Entity

E.3.1. UML models

E.3.2. Elements defined in the Core Vocabulary

E.3.2.1. Acknowledgement Class (subclass of Message)

This message is to acknowledge the reception of the other types of messages. It is sent asynchronously to allow CISE gateways to handle the retry mechanism.

E.3.2.1.9. Attributes

UML Name	Data type	Description	Example	Source
AckCode	AcknowledgementType	This code represents the type of fault that occurred	"02" for security error	
AckDetail	String	Additional text to clarify details about the fault.	"The request did not specify a valid Entity Template"	
ContextID	String	This identifier correlates the messages that share an operational need. For instance, in order to update the information of an incident, several CISE entity services need to be invoked (e.g., IncidentService, EventLocationService, etc.) Thus, the messages exchanged with these CISE entity services are related by this ID type.	"d1a50126-172f-4081-831d-d6d449b2687d"	
CorrelationID	String	This identifier correlates the request and response messages of/to a service (for the Pull or the	"d1a50126-172f-4081-831d-d6d449b2687d"	

UML Name	Data type	Description	Example	Source
		Publish/Subscribe communication patterns)		
CreationDateTime	XSD::DateTime	The date and time when this messaging object was created	2013-09-24T13:05:23+00:00	
MessageID	String	Identifier of the message. It is unique for the CISE participant who created the message.	"d1a50126-172f-4081-831d-d6d449b2687d"	
Priority	PriorityType	Priority of the message, to help the receiver of the message to deal with prioritizing the messages received.	"High"	

(*) Inherited attributes are coloured in grey.

E.3.2.1.10. Association Roles

UML Name	Data type	Description	Multiplicity
ccRecipients	Service		0..*
discoveredServices	Service		0..*
payload	CoreEntityPayload	This field contains the payload (business content) and its associated metadata.	0..1
recipient	Service	This field describes the recipient of the message.	1
reliability	ReliabilityProfile	This field describes the reliability profile requested by the sender of the message.	0..1
sender	Service	This field describes the sender of the message.	1

(*) Inherited association roles are coloured in grey.

E.3.2.2. CoreEntityPayload Class

This entity contains the business payload of the message and the metadata related to this payload. This is used by all messages types expect the Acknowledgement type.

E.3.2.2.11. Attributes

UML Name	Data type	Description	Example	Source
InformationSecurityLevel	InformationSecurityLevelType	Level of security associated with the payload.	"NonClassified"	
InformationSensitivity	InformationSensitivityType	Level of sensitivity related to the payload.	"Green"	
IsPersonalData	boolean	If the payload contains personal data.	"false"	

UML Name	Data type	Description	Example	Source
Purpose	PurposeType	The purpose of the message. It can be used to handle access rights on the provider side.	"01 - Vessel Traffic Management"	
RetentionPeriod	XSD::DateTime	Date and Time until when the payload can be kept. This information can be used for the legal constraints associated with the management of personal data.	2013-09-24T13:05:23+00:00	

E.3.2.3. EncryptedEntityPayload Class (subclass of CoreEntityPayload)

E.3.2.3.12. Attributes

UML Name	Data type	Description	Example	Source
Entities	String			
InformationSecurityLevel	InformationSecurityLevelType	Level of security associated with the payload.	"NonClassified"	
InformationSensitivity	InformationSensitivityType	Level of sensitivity related to the payload.	"Green"	
IsPersonalData	boolean	If the payload contains personal data.	"false"	
Purpose	PurposeType	The purpose of the message. It can be used to handle access rights on the provider side.	"01 - Vessel Traffic Management"	
RetentionPeriod	XSD::DateTime	Date and Time until when the payload can be kept. This information can be used for the legal constraints associated with the	2013-09-24T13:05:23+00:00	

UML Name	Data type	Description	Example	Source
		management of personal data.		

(*) Inherited attributes are coloured in grey.

E.3.2.4. Feedback Class (subclass of Message)

This message type allows to provide feedback on a message already sent (for example when a message was sent by error) or on a message received.

E.3.2.4.13. Attributes

UML Name	Data type	Description	Example	Source
FeedbackType	FeedbackType	The type of feedback to provide	"delete"	
Reason	String	The description of the reason for feedback. This field is a free text.	"The information sent was related to a test and forwarded by error"	
RefMessageID	String	The Message ID that this feedback message refers to.	d1a50126-172f-4081-831d-d6d449b2687d	
ContextID	String	This identifier correlates the messages that share an operational need. For instance, in order to update the information of an incident, several CISE entity services need to be invoked (e.g., IncidentService, EventLocationService, etc.) Thus, the messages exchanged with these CISE entity services are related by this ID type.	"d1a50126-172f-4081-831d-d6d449b2687d"	
CorrelationID	String	This identifier correlates the request and response messages of/to a service (for the Pull or the Publish/Subscribe communication patterns)	"d1a50126-172f-4081-831d-d6d449b2687d"	
CreationDateTime	XSD::DateTime	The date and time when this messaging object was created	2013-09-24T13:05:23+00:00	
MessageID	String	Identifier of the message. It is unique for the CISE participant who created the message.	"d1a50126-172f-4081-831d-d6d449b2687d"	
Priority	PriorityType	Priority of the message, to help the receiver of the message to deal with prioritizing the messages received.	"High"	

(*) Inherited attributes are coloured in grey.

E.3.2.4.14. Association Roles

UML Name	Data type	Description	Multiplicity
ccRecipients	Service		0..*
payload	CoreEntityPayload	This field contains the payload (business content) and its associated metadata.	0..1

UML Name	Data type	Description	Multiplicity
recipient	Service	This field describes the recipient of the message.	1
reliability	ReliabilityProfile	This field describes the reliability profile requested by the sender of the message.	0..1
sender	Service	This field describes the sender of the message.	1

(*) Inherited association roles are coloured in grey.

E.3.2.5. Message Class

This abstract entity describes the message metadata, it is used to identify the message type linked to a communication pattern and the correlation with other message.

E.3.2.5.15. Attributes

UML Name	Data type	Description	Example	Source
ContextID	String	This identifier correlates the messages that share an operational need. For instance, in order to update the information of an incident, several CISE entity services need to be invoked (e.g., IncidentService, EventLocationService, etc.) Thus, the messages exchanged with these CISE entity services are related by this ID type.	"d1a50126-172f-4081-831d-d6d449b2687d"	
CorrelationID	String	This identifier correlates the request and response messages of/to a service (for the Pull or the Publish/Subscribe communication patterns)	"d1a50126-172f-4081-831d-d6d449b2687d"	
CreationDateTime	XSD::DateTime	The date and time when this messaging object was created	2013-09-24T13:05:23+00:00	
MessageID	String	Identifier of the message. It is unique for the CISE participant who created the message.	"d1a50126-172f-4081-831d-d6d449b2687d"	
Priority	PriorityType	Priority of the message, to help the receiver of the message to deal with prioritizing the messages received.	"High"	

E.3.2.5.16. Association Roles

UML Name	Data type	Description	Multiplicity
ccRecipients	Service		0..*
payload	CoreEntityPayload	This field contains the payload (business content) and its associated metadata.	0..1
recipient	Service	This field describes the recipient of the message.	1
reliability	ReliabilityProfile	This field describes the reliability profile requested by the sender of the message.	0..1
sender	Service	This field describes the sender of the message.	1

E.3.2.6. PullRequest Class (subclass of Message)

The message Pull Request is used in the Pull communication pattern to request information

E.3.2.6.17. Attributes

UML Name	Data type	Description	Example	Source
PullType	PullType	The Pull Type is to distinguish between the simple request and the subscription mechanism. It can also be use to unsubscribe to a flow.	"request"	
ResponseTimeOut	int	Time in seconds. The request should be answered within this time limit. After this time, the response may not be considered by the requesting system.	86400	
ContextID	String	This identifier correlates the messages that share an operational need. For instance, in order to update the information of an incident, several CISE entity services need to be invoked (e.g., IncidentService, EventLocationService, etc.) Thus, the messages exchanged with these CISE entity services are related by this ID type.	"d1a50126-172f-4081-831d-d6d449b2687d"	
CorrelationID	String	This identifier correlates the request and response messages of/to a service (for the Pull or the Publish/Subscribe communication patterns)	"d1a50126-172f-4081-831d-d6d449b2687d"	
CreationDateTime	XSD::DateTime	The date and time when this messaging object was created	2013-09-24T13:05:23+00:00	
MessageID	String	Identifier of the message. It is unique for the CISE participant who created the message.	"d1a50126-172f-4081-831d-d6d449b2687d"	
Priority	PriorityType	Priority of the message, to help the receiver of the message to deal with prioritizing the messages received.	"High"	

(*) Inherited attributes are coloured in grey.

E.3.2.6.18. Association Roles

UML Name	Data type	Description	Multiplicity
ccRecipients	Service		0..*
discoveryProfiles	ServiceProfile	This field is used for the Legacy System to request the CISE Gateway to look for services of a specific type and/or from a specific type of provider (using the community, country, sea basin...)	0..*
payload	CoreEntityPayload	This field contains the payload (business content) and its associated metadata.	0..1
recipient	Service	This field describes the recipient of the message.	1

UML Name	Data type	Description	Multiplicity
reliability	ReliabilityProfile	This field describes the reliability profile requested by the sender of the message.	0..1
requests	ServiceCapability	Service Capability required by the system requesting the information. This will indicate for instance the maximum entities expected in return.	0..1
sender	Service	This field describes the sender of the message.	1

(*) Inherited association roles are coloured in grey.

E.3.2.7. PullResponse Class (subclass of Message)

The message Pull Response is used in the Pull communication pattern to respond to a request. This response is sent asynchronously.

E.3.2.7.19. Attributes

UML Name	Data type	Description	Example	Source
ErrorDetail	String	This field may give a textual description of an error that could have happened during the process of the pull request message. This can be used to communicate an error that happened after sending the acknowledgement message.		
ResultCode	ResponseCodeType	This field provides an OK code if the response is sent along with the pull response. It can also provide an error code if an error occurred after sending the acknowledgement message.		
ContextID	String	This identifier correlates the messages that share an operational need. For instance, in order to update the information of an incident, several CISE entity services need to be invoked (e.g., IncidentService, EventLocationService, etc.) Thus, the messages exchanged with these CISE entity services are related by this ID type.	"d1a50126-172f-4081-831d-d6d449b2687d"	
CorrelationID	String	This identifier correlates the request and response messages of/to a service (for the Pull or the Publish/Subscribe communication patterns)	"d1a50126-172f-4081-831d-d6d449b2687d"	
CreationDateTime	XSD::DateTime	The date and time when this messaging object was created	2013-09-24T13:05:23+00:00	
MessageID	String	Identifier of the message. It is unique for the CISE participant who created the message.	"d1a50126-172f-4081-831d-d6d449b2687d"	

UML Name	Data type	Description	Example	Source
Priority	PriorityType	Priority of the message, to help the receiver of the message to deal with prioritizing the messages received.	"High"	

(*) Inherited attributes are coloured in grey.

E.3.2.7.20. Association Roles

UML Name	Data type	Description	Multiplicity
ccRecipients	Service		0..*
fulfils	ServiceCapability	Describes the characteristics used to respond to the request. For instance, the type of query performed (exact or best effort).	0..1
payload	CoreEntityPayload	This field contains the payload (business content) and its associated metadata.	0..1
recipient	Service	This field describes the recipient of the message.	1
reliability	ReliabilityProfile	This field describes the reliability profile requested by the sender of the message.	0..1
sender	Service	This field describes the sender of the message.	1

(*) Inherited association roles are coloured in grey.

E.3.2.8. Push Class (subclass of Message)

The message Push is used in the Push communication pattern to transmit information to other CISE participants.

E.3.2.8.21. Attributes

UML Name	Data type	Description	Example	Source
ContextID	String	This identifier correlates the messages that share an operational need. For instance, in order to update the information of an incident, several CISE entity services need to be invoked (e.g., IncidentService, EventLocationService, etc.) Thus, the messages exchanged with these CISE entity services are related by this ID type.	"d1a50126-172f-4081-831d-d6d449b2687d"	
CorrelationID	String	This identifier correlates the request and response messages of/to a service (for the Pull or the Publish/Subscribe communication patterns)	"d1a50126-172f-4081-831d-d6d449b2687d"	
CreationDateTime	XSD::DateTime	The date and time when this messaging object was created	2013-09-24T13:05:23+00:00	
MessageID	String	Identifier of the message. It is unique for the CISE participant who created the message.	"d1a50126-172f-4081-831d-d6d449b2687d"	

UML Name	Data type	Description	Example	Source
Priority	PriorityType	Priority of the message, to help the receiver of the message to deal with prioritizing the messages received.	"High"	

(*) Inherited attributes are coloured in grey.

E.3.2.8.22. Association Roles

UML Name	Data type	Description	Multiplicity
ccRecipients	Service		0..*
discoveryProfiles	ServiceProfile	This field is used for the Legacy System to request the CISE Gateway to look for services of a specific type and/or from a specific type of provider (using the community, country, sea basin...)	0..*
payload	CoreEntityPayload	This field contains the payload (business content) and its associated metadata.	0..1
recipient	Service	This field describes the recipient of the message.	1
reliability	ReliabilityProfile	This field describes the reliability profile requested by the sender of the message.	0..1
sender	Service	This field describes the sender of the message.	1

(*) Inherited association roles are coloured in grey.

E.3.2.9. ReliabilityProfile Class

This entity contains information about the retry strategy in case of error during the transmission of the message.

E.3.2.9.23. Attributes

UML Name	Data type	Description	Example	Source
RetryStrategy	RetryStrategyType	The type of retry strategy required by this message. For each type, a retry mechanism has been agreed at EU level (e.g. number of retry, time between each try...). This mechanism is implemented by the CISE Gateways.	"high reliablility"	

E.3.2.10. XmlEntityPayload Class (subclass of CoreEntityPayload)

E.3.2.10.24. Attributes

UML Name	Data type	Description	Example	Source
EnsureEncryption	boolean			
Entities	XSD::anyType	The list of entities transmitted in the message. This refers to the CISE data model. The		

UML Name	Data type	Description	Example	Source
		content of this field will be checked with the data model XML Schema.		
InformationSecurityLevel	InformationSecurityLevelType	Level of security associated with the payload.	"NonClassified"	
InformationSensitivity	InformationSensitivityType	Level of sensitivity related to the payload.	"Green"	
IsPersonalData	boolean	If the payload contains personal data.	"false"	
Purpose	PurposeType	The purpose of the message. It can be used to handle access rights on the provider side.	"01 - Vessel Traffic Management"	
RetentionPeriod	XSD::DateTime	Date and Time until when the payload can be kept. This information can be used for the legal constraints associated with the management of personal data.	2013-09-24T13:05:23+00:00	

(*) Inherited attributes are coloured in grey.

E.3.2.11. AcknowledgementType Enumeration

Enumeration of the type of errors handled in the Acknowledgement message.

E.3.2.11.25. Enumeration Values

Value	Label	Description	Source
Success		Success	
EndPointNotFound		End point not found	
SecurityError		Security error	
InternalGatewayFault		Internal gateway fault	
InvalidRequestObject		Invalid request object	
Unauthorized		Unauthorized error	
BadRequest		Bad request	

Value	Label	Description	Source
EntityTypeNotAccepted		Entity type not accepted	
ObjectTypeNotAccepted		Object type not accepted	
ServerError		Server error	
TimestampError		Timestamp error	
AuthenticationError		Authentication error	

E.3.2.11.26. Enumeration Usage

The following attributes use this enumeration as data type:

- AckCode (Acknowledgement)

E.3.2.12. FeedbackType Enumeration

Enumeration of the type of feedback for the feedback message.

E.3.2.12.27. Enumeration Values

Value	Label	Description	Source
info		Information on a previous message	
delete		Request for deleting a previous message	

E.3.2.12.28. Enumeration Usage

The following attributes use this enumeration as data type:

- FeedbackType (Feedback)

E.3.2.13. InformationSecurityLevelType Enumeration

This enumeration presents the possible values for information security classification. The enumeration is based in the security rules for protecting EU classified information (<a href="http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:141:0017:0065:EN:PDF"><font color="#000080"><u>http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:141:0017:0065:EN:PDF</u></font></a>).

E.3.2.13.29. Enumeration Values

Value	Label	Description	Source
EUTopSecret	EU top secret	Information and material the unauthorized disclosure of which could cause exceptionally grave prejudice to the essential interests of the European Union or of one or more of the Member States	
EUSecret	EU secret	Information and material the unauthorized disclosure of which could seriously harm the essential interests of the European Union or of one or more of the Member States	
EUConfidential	EU confidential	Information and material the unauthorized disclosure of which could harm the essential interests of the European Union or of one or more of the Member States	

Value	Label	Description	Source
EURestricted	EU restricted	Information and material the unauthorized disclosure of which could be disadvantageous to the interests of the European Union or of one or more of the Member States	
NonClassified	non-classified	It can be used for information and material whose classification level is still pending	
NonSpecified	non-specified	It can be used for information and material whose classification level is not specified	

E.3.2.13.30. Enumeration Usage

The following attributes use this enumeration as data type:

- InformationSecurityLevel (CoreEntityPayload)

E.3.2.14. InformationSensitivityType Enumeration

This enumeration presents the possible values for information sensitivity degree. The Traffic Light Protocol (TLP) of US-CERT is applied (<a href="http://www.us-cert.gov/tlp"><font color="#000080"><u>http://www.us-cert.gov/tlp</u></font></a>).

Source: US-CERT

E.3.2.14.31. Enumeration Values

Value	Label	Description	Source
Red	red	TLP: RED when information cannot be effectively acted upon by additional parties, and could lead to impacts on a party's privacy, reputation, or operations if misused.	
Amber	amber	TLP: AMBER when information requires support to be effectively acted upon, but carries risks to privacy, reputation, or operations if shared outside of the organizations involved.	
Green	green	TLP: GREEN when information is useful for the awareness of all participating organizations as well as with peers within the broader community or sector.	
White	white	TLP: WHITE when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release.	
NonSpecified	non-specified	It can be used for information and material whose classification level is not specified	

E.3.2.14.32. Enumeration Usage

The following attributes use this enumeration as data type:

- InformationSensitivity (CoreEntityPayload)

E.3.2.15. PriorityType Enumeration

Enumeration of the different priority of a message.

E.3.2.15.33. Enumeration Values

Value	Label	Description	Source
Low	Low	Low priority message	
Medium	Medium	Medium priority message	
High	High	High priority message	

E.3.2.15.34. Enumeration Usage

The following attributes use this enumeration as data type:

- Priority (Message)

E.3.2.16. PullType Enumeration

Enumeration to handle the different type of Pull Request message.

E.3.2.16.35. Enumeration Values

Value	Label	Description	Source
Request	request	request	
Subscribe	subscribe	subscribe	
Unsubscribe	unsubscribe	unsubscribe	

E.3.2.16.36. Enumeration Usage

The following attributes use this enumeration as data type:

- PullType (PullRequest)

E.3.2.17. PurposeType Enumeration

Enumeration of the different purpose linked to a request.

E.3.2.17.37. Enumeration Values

Value	Label	Description	Source
VTM		Vessel Traffic Management	
Safety		Vessel Traffic Safety	
Security		Monitoring of security of ships	
SAR		Search and Rescue	
Operation		Support of response and enforcement operations (anti-piracy, SAR, salvage)	
FisheriesWarning		Early warning of illegal fisheries or fish landings	
FisheriesMonitoring		Monitoring of compliance with regulations on fisheries	
FisheriesOperation		Support of response and enforcement operations	
EnvironmentMonitoring		Monitoring of compliance with regulations	
EnvironmentWarning		Early warning of environmental accidents and incidents	
EnvironmentResponse		Support of pollution response operations	

Value	Label	Description	Source
CustomsMonitoring		Monitoring of compliance with customs regulation on import, export and movement of goods	
CustomsOperation		Support of enforcement operations	
BorderMonitoring		Monitoring of compliance with regulations on immigration and border control crossings	
BorderOperation		Support of enforcement operations	
LawEnforcementMonitoring		Monitoring of compliance with applicable legislation in sea areas where police competence is required	
LawEnforcementOperation		Support to enforcement and response operations	
DefenceMonitoring		Monitoring in support of defence tasks such as national sovereignty at sea	
CounterTerrorism		Combatting terrorism and other hostile activities outside the EU	
CSDPTask		Other CSDP tasks as defined in Articles and TEU	

E.3.2.17.38. *Enumeration Usage*

The following attributes use this enumeration as data type:

- Purpose (CoreEntityPayload)

E.3.2.18. *ResponseCodeType Enumeration*

Enumeration to describe the type of result of a Pull Response.

E.3.2.18.39. *Enumeration Values*

Value	Label	Description	Source
Success		Success	
EndPointNotFound		End point not found	
SecurityError		Security error	
InternalGatewayFault		Internal gateway fault	
InvalidRequestObject		Invalid request object	
Unauthorized		Unauthorized error	
BadRequest		Bad request	
EntityTypeNotAccepted		Entity type not accepted	
ObjectTypeNotAccepted		Object type not accepted	
ServerError		Server error	
TimestampError		Timestamp error	
AuthenticationError		Authentication error	

E.3.2.18.40. *Enumeration Usage*

The following attributes use this enumeration as data type:

- ResultCode (PullResponse)

E.3.2.19. RetryStrategyType Enumeration

Enumeration of the different type of retry strategy.

E.3.2.19.41. Enumeration Values

Value	Label	Description	Source
NoRetry		no retry	
LowReliabilty		low reliabilty	
HighReliability		high reliability	

E.3.2.19.42. Enumeration Usage

The following attributes use this enumeration as data type:

- RetryStrategy (ReliabilityProfile)

E.4. Service Core Entity

E.4.1. UML models

E.4.2. Elements defined in the Core Vocabulary

E.4.2.1. Gateway Class

Gateway providing the service. This entity provides the information required for routing the messages between gateways.

E.4.2.1.43. Attributes

UML Name	Data type	Description	Example	Source
GatewayID	String	The unique identifier of the Gateway.	eu.cise.authority1.gatewayA	
GatewayStatus	GatewayStatusType	The current status of the node.	"online"	
PhysicalAddress	String	The URL where the service can be reached.	http://10.10.2.75:7845/CISE	

E.4.2.2. Service Class

This entity contains the description of a service.

E.4.2.2.44. Attributes

UML Name	Data type	Description	Example	Source
EntityTemplate	XSD::anyType	The restriction linked to a service. For instance, if a		

UML Name	Data type	Description	Example	Source
		VesselService cover only fishing ship, this field provides a Vessel entity with the VesselType=02 (fishing vessel).		
ServiceID	String	This is the unique ID of the service.	eu.cise.authority1.vesselservices.service123	
ServiceOperation	ServiceOperationType	The type of communication pattern supported by the service (Pull/Push/Subscribe)	"pull"	
ServiceStatus	ServiceStatusType	The status of the service. Draft = the service is not yet available for use but can be seen in the Service Registry Online = the service is available for use Maintenance = the service is temporally not available for use Deprecated = the service is available for use but will be soon offline, either replaced by a new version of the service or discontinued Offline = the service is not available anymore. For historic purpose, it can still be seen in the Service Registry with an offline status.	"Online"	
ServiceType	ServiceType	The service type gives the type of entities exchanged (based on the CISE data model).	"IncidentService" can be used to inform partners of an incident (push operation), request about incident (pull operation) or ask to be inform when an incident happen or to get updated about a specific incident (subscribe operation)	

E.4.2.2.45. Association Roles

UML Name	Data type	Description	Multiplicity
gateway	Gateway	Gateway offering the service. This relationship provides the identification and the physical address of the gateway.	0..1
provider	Authority		0..*
system	LegacySystem		0..1

E.4.2.3. ServiceCapability Class

The different capabilities of a service. This can depends of the operation supported.

E.4.2.3.46. Attributes

UML Name	Data type	Description	Example	Source
MaxEntitiesPerMsg	int	Maximum number of entities returned in a pull response.	1000	
QueryByExampleType	QueryByExampleType	Type used for the query by example. This mechanism supports either exact answers or approximate answers.	"01" for best effort	

E.4.2.4. ServiceProfile Class

The characteristics associated to the service and to the provider of the service.

E.4.2.4.47. Attributes

UML Name	Data type	Description	Example	Source
Community	CommunityType	One of the 7 user communities participating to CISE. This field is related to the community of the provider of the service.	"02" for Customs	
Country	String	The Country of the provider of the service. The code ISO-3166-1 alpha-2 is used: 2 character country code.	"FR" for France	
DataFreshness	DataFreshnessType	This field specify what type of data is provided by the service. This is to distinguish real time data from historic data.	"02" for real time	
Function	FunctionType	The function covered by the provider of the service.	"Monitoring of security of ships"	

UML Name	Data type	Description	Example	Source
SeaBasin	SeaBasinType	The sea basin covered by the service.	"05" for Mediterranean	
ServiceType	ServiceType	The service type for the discovered services.	"IncidentService" can be used to inform partners of an incident (push operation), request about incident (pull operation) or ask to be inform when an incident happen or to get updated about a specific incident (subscribe operation)	

E.4.2.5. SubscriptionCapability Class (subclass of ServiceCapability)

Capabilities specific for the subscribe pattern

E.4.2.5.48. Attributes

UML Name	Data type	Description	Example	Source
AvgUpdateFrequency	XSD::Duration	In the subscription pattern, the average time between two updates. This information can be requested by the subscriber or given for information by the service provider.		
MaxFrequency	XSD::Duration	In the subscription pattern, the maximum frequency of update available. This information can given for information by the service provider.		
SubscriptionEnd	XSD::DateTime	The date and time when the subscription should end. This information can be requested by the subscriber.		
MaxEntitiesPerMsg	int	Maximum number of entities returned in a pull response.	1000	
QueryByExampleType	QueryByExampleType	Type used for the query by example. This mechanism supports either exact answers or approximate answers.	"01" for best effort	

(*) Inherited attributes are coloured in grey.

E.4.2.6. DataFreshnessType Enumeration

Enumeration of the different type of data related to the time (real time or historic information).

E.4.2.6.49. Enumeration Values

Value	Label	Description	Source
Historic		historic	
RealTime		real time	
NearlyRealTime		nearly real time	

Value	Label	Description	Source
Unknown		unknown	

E.4.2.6.50. Enumeration Usage

The following attributes use this enumeration as data type:

- DataFreshness (ServiceProfile)

E.4.2.7. GatewayStatusType Enumeration

Enumeration of the different status of availability of a Node.

E.4.2.7.51. Enumeration Values

Value	Label	Description	Source
Online	online	online	
Offline	offline	offline	
Maintenance	maintenance	maintenance	

E.4.2.7.52. Enumeration Usage

The following attributes use this enumeration as data type:

- GatewayStatus (Gateway)

E.4.2.8. QueryByExampleType Enumeration

Enumeration of the different type of query.

E.4.2.8.53. Enumeration Values

Value	Label	Description	Source
BestEffort		best effort	
ExactSearch		exact search	

E.4.2.8.54. Enumeration Usage

The following attributes use this enumeration as data type:

- QueryByExampleType (ServiceCapability)

E.4.2.9. ServiceOperationType Enumeration

Enumeration of the different type of communication pattern available for each service type.

E.4.2.9.55. Enumeration Values

Value	Label	Description	Source
Pull		Pull	

Value	Label	Description	Source
Push		Push	
Subscribe		Subscribe	
Feedback		Feedback	

E.4.2.9.56. Enumeration Usage

The following attributes use this enumeration as data type:

- ServiceOperation (Service)

E.4.2.10. ServiceStatusType Enumeration

Enumeration of the different status of a service.

E.4.2.10.57. Enumeration Values

Value	Label	Description	Source
Draft	draft	The service is available only to the provider	
Online	online	The service is available to any CISE participant according to the access rights policy	
Maintenance	maintenance	The service is under maintenance, not available	
Deprecated	deprecated	This service is online and available for compatibility with previous versions of the specifications, but it can be removed in the next versions.	
Offline	offline	The service is registered but not available	

E.4.2.10.58. Enumeration Usage

The following attributes use this enumeration as data type:

- ServiceStatus (Service)

E.4.2.11. ServiceType Enumeration

Enumeration of the different type of service related to the entity Data Model of CISE.

E.4.2.11.59. Enumeration Values

Value	Label	Description	Source
ActionService	Action Service	This service exchanges data objects of the Action data entity.	
AgentService	Agent Service	This service exchanges the data objects of the Agent data entity.	
AircraftService	Aircraft Service	This service exchanges the data objects of the Aircraft data entity.	
AnomalyService	Anomaly Service	This service exchanges the data objects of the Anomaly data entity.	

Value	Label	Description	Source
CargoDocumentService	Cargo Document Service	This service exchanges the data objects of the CargoDocument data entity.	
CargoService	Cargo Service	This service exchanges the data objects of the CargoUnit data entity.	
CertificateDocumentService	Certificate Document Service	This service exchanges the data objects of the CertificateDocument data entity	
CrisisIncidentService	Crisis Incident Service	This service exchanges the data objects of the CrisisIncident data entity.	
DocumentService	Document Service	This service exchanges the data objects of the Document data entity.	
EventDocumentService	Event Document Service	This service exchanges the data objects of the EventDocument data entity.	
IncidentService	Incident Service	This service exchanges the data objects of the Incident data entity.	
IrregularMigrationIncidentService	Irregular Migration Incident Service	This service exchanges the data objects of the IrregularMigrationIncident data entity.	
LandVehicleService	Land Vehicle Service	This service exchanges the data objects of the LandVehicle data entity.	
LawInfringementIncidentService	Law Infringement Incident Service	This service exchanges the data objects of the LawInfringementIncident data entity.	
LocationService	Location Service	This service exchanges the data objects of the Location data entity.	
LocationDocumentService	Location Document Service	This service exchanges the data objects of the LocationDocument data entity.	
MaritimeSafetyIncidentService	Maritime Safety Incident Service	This service exchanges the data objects of the MaritimeSafetyIncident data entity.	
MeteoOceanographicObservationService	Meteo-Oceanographic Observation Service	This service exchanges the data objects of the MeteoOceanographicObservation data entity.	
MovementService	Movement Service	This service exchanges the data objects of the Movement data entity.	
OperationalAssetService	Operational Asset Service	This service exchanges the data objects of the OperationalAsset data entity.	
OrganizationService	Organization Service	This service exchanges the data objects of the Organization data entity.	
OrganizationDocumentService	Organization Document Service	This service exchanges the data objects of the OrganizationDocument data entity.	

Value	Label	Description	Source
PersonService	Person Service	This service exchanges the data objects of the Person data entity.	
PersonDocumentService	Person Document Service	This service exchanges the data objects of the PersonDocument data entity.	
RiskDocumentService	Risk Document Service	This service exchanges the data objects of the RiskDocument data entity.	
RiskService	Risk Service	This service exchanges the data objects of the Risk data entity.	
VesselDocumentService	Vessel Document Service	This service exchanges the data objects of the VesselDocument data entity.	
VesselService	Vessel Service	This service exchanges the data objects of the Vessel data entity.	

E.4.2.11.60. Enumeration Usage

The following attributes use this enumeration as data type:

- ServiceType (Service)
- ServiceType (ServiceProfile)

JRC Mission

As the Commission's in-house science service, the Joint Research Centre's mission is to provide EU policies with independent, evidence-based scientific and technical support throughout the whole policy cycle.

Working in close cooperation with policy Directorates-General, the JRC addresses key societal challenges while stimulating innovation through developing new methods, tools and standards, and sharing its know-how with the Member States, the scientific community and international partners.

Serving society
Stimulating innovation
Supporting legislation

Publications Office