



# TECHNICAL GUIDE FOR IMPLEMENTATION

Version 1.0

## Disclaimer

THIS VERSION IS A SIMPLIFIED VERSION OF THE BROADER TECHNICAL GUIDE. THE GOAL IS TO AVOID NEW USERS TO GO THROUGH A 300 PAGES DOCUMENT BUT IT IS THEREFORE LESS COMPLETE.

Limited details are likely to change in the future. The overall design & architecture should not.

This is a technical document drafted to facilitate the IT implementations needed in the framework of the VSP Contract. As such the mutual rights and obligations of Elia and the VSP specified in the regulated VSP Contract, prevail over the provided technical documents in case of inconsistencies.

**On a manual workaround**

Note that for all the incoming data (schedules, unavailabilities, bids) Elia will create a webclient in which Market Parties can introduce their data manually. The webclient will offer a user interface where Market Parties can have a view on the status of (automatically or manually) sent data. The documentation concerning this solution will be sent in a later phase.

**Table of Contents**

1	Document Version and contact persons .....	5
1.1	Document version.....	5
1.2	Contact person.....	5
2	Introduction .....	6
2.1	Background .....	6
2.2	Scope.....	6
3	Overview of communication requirements .....	7
3.1	General Defintions .....	7
3.2	Requirements for Voltage Service Providers .....	8
4	External Communication Layer.....	10
4.1	General details .....	10
4.2	Connection Information.....	11
4.3	Queues and exchanges naming convention .....	13
4.4	Message operations.....	17
5	Generic message specifications.....	21
5.1	JSON format and date format.....	21
5.2	Market document structure .....	24
5.3	Identification and versioning .....	25
5.4	Message granularity .....	26
6	Acknowledgement and answer messages.....	27
6.1	Acknowledgement .....	27
6.2	Answer .....	28
7	Notification messages.....	30
7.1	Description.....	30
7.2	Notification submitted message.....	30
7.3	Notification acknowledged message .....	32
8	Voltage Service Provider Guide .....	33

8.1	Role overview .....	33
8.2	Submitting MVAR events.....	33
8.3	Retrieving MVAR availability plan details .....	42
8.4	Retrieving MVAR availability plan overview .....	47
8.5	Retrieving mvar events.....	52
8.6	Receiving a Market Party Notification.....	56
8.7	Receiving a mvar activation request.....	57
8.8	Receiving a mvar communication test.....	62
9	Validation rules description.....	66
9.1	Generic.....	66
9.2	Reactive Power Outage Planning.....	73

## 1 Document Version and contact persons

### 1.1 Document version

Version	Date	Changes
0.1	27/11/2024	Working version of the document still under review
0.2	03/02/2025	New working version
0.3	19/02/2025	Version started after submission to VSP
1.0	03/10/2025	Version containing description of both Outages and Activations

### 1.2 Contact person

For any question, please contact your KAM Energy:

Name	E-mail
Bram Deckers	Bram.Deckers@elia.be

For specific IT-related question, please contact the following email address:

E-mail
<a href="mailto:IT-ECL@elia.be">IT-ECL@elia.be</a>

## **2 Introduction**

### **2.1 Background**

Market Parties are invited to use the Elia External Communication Layer (ECL) for the new market interactions defined in the T&C VSP applicable from 1<sup>st</sup> Jan 2027 and for which the ECL Go-Live is scheduled for 1<sup>st</sup> Feb 2027. A manual method will also be developed and detailed in another document.

The format of all communications through this platform is based on the Common Information Model (CIM) standards, with some modifications defined by Elia when local needs require it.

This guide contains general technical information about the External Communication Layer and specific information on communication required by each market role in the context of the T&Cs.

### **2.2 Scope**

This implementation guide provides all the information that you need to adapt your systems in order to communicate with Elia.

This document is directed to VSPs.

The main topics covered by this document are:

- Technical description of the External Communication Layer.
- Overview of communication flows with the Voltage Service Provider.
- Description of all messages exchanges.
- Message format definition (MarketDocuments)

Note that for all the incoming data (unavailabilities and service control type events) Elia will create a webclient in which Market Parties can introduce their data manually. The webclient will offer a user interface where Market Parties can have a view on the status of (automatically or manually) sent data. The documentation concerning this solution will be sent in a later phase.

### **3 Overview of communication requirements**

#### **3.1 General Definitions**

The present section describes the communication channels used by Elia to communicate and receive information to and from the different market parties in the context of Mvar Evolution Project.

The communication channels are separated in:

- “Main communication channel(s)” that must be used by the market parties. If several channels are available, the VSP can choose to use only one of them or all of them.
- “Back-up communication channel(s)” that must be used by the market parties if the main communication channels fail. If several channels are available, the VSP can choose to use only one of them or all of them. The VSP must have a back-up communication channel. Working in back-up mode does not imply working in a degraded mode and should ensure the same quality level of services from Elia and the Market Party.

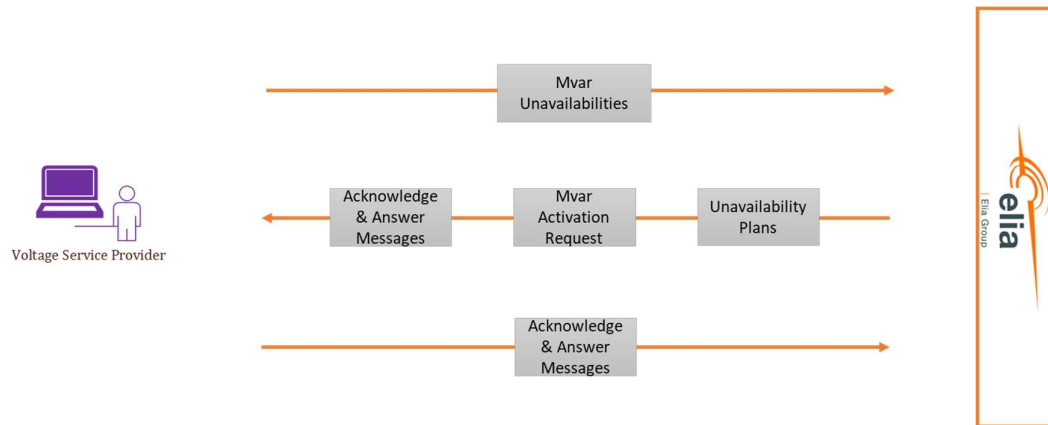
The different communication channels will be:

- The External Communication Layer (ECL) described in the present document.
- WebClient (B2C) that is yet to be described but that will ultimately be detailed in Optiflex’s user manuals. It is likely to be very similar to the outage planning tool webclient.
- The e-mail.
- Phone
- ...

### 3.2 Requirements for Voltage Service Providers

Each VSP is required to implement at least, one main communication channel and one back-up communication channel.

A Voltage Service Provider is involved in the following communications:



VSP can either submit **unavailability events** via:

#### Main communication channel:

- The External Communication Layer put in place by Elia and to be used for the exchange of asynchronous messages between Elia and Market Parties.
- The Webclient available for the Market Parties to upload their data manually using an Excel template. The webclient offers a user interface where Market Parties can have a view on the status of (automatically and manually) sent data.

#### Back-up communication channel:

- Each of the main communication channel is the back-up for the other one.
- In case of issues in both channels, phone and email could also be used.

Elia can communicate Mvar Activation Requests using

#### Main communication channel:

- The External Communication Layer put in place by Elia and to be used for the exchange of asynchronous messages between Elia and Market Parties.

#### Back-up communication channel:



- Phone calls
- Emails

*AS IS, the backup communication channel is the activations by phone. It is expected that this channel will remain the back up for sending activations.*

- VSP must be able to handle **Market Party notifications** that are sent by Elia via:
  - JSON messages via ECL
  - Automated emails
- VSP must be able to handle **Answer Messages** that are sent by Elia via:
  - JSON messages via ECL
  - Automated emails

## 4 External Communication Layer

The External Communication Layer is put in place by Elia and must be used for the exchange of asynchronous messages between Elia and Market Parties.

### Attention Point:

The Go-Live of the CIM-based ECL will occur on 1<sup>st</sup> of February 2027.

Before that date, the MVAR Activation messages and their acknowledgment messages will be exchanged using the legacy messaging system Revolt, with phone calls and emails as backup systems. This corresponds to the system in use in 2025 and 2026. Messages related to the entrance or exit of Power Saving Mode will not be received by Elia before the 1<sup>st</sup> of February 2027. Other availability messages will be either sent through the webclient or through phone calls or emails.

This section describes how to use the Elia External Communication Layer to send and receive messages.

### 4.1 General details

#### 4.1.1 Dedicated queues/exchanges

There are dedicated queues/exchanges for each type of message:

- For messages sent by Elia to a Market Party, there is one **queue** for each type of message. Queues are specific to only one Market Party. This means that only one Market Party can read messages from each queue.
- For messages sent by a Market Party to Elia, there is one **exchange** for each type of message. An exchange is not specific to each Market Party. It can be used by multiple Market Parties to send messages.

#### 4.1.2 Sending messages

To send a message to the External Communication Layer, the Market Party has to write the message to the corresponding Exchange.

The *'Write pattern'* must include a retry logic: the sender is responsible to ensure that the message has been acknowledged by the queuing platform before considering it as delivered.

#### 4.1.3 Receiving messages

To receive a message out of the External Communication Layer, the Market Party has to read the message from a dedicated queue.

The *'Read pattern'* must include a transaction approach: it is the responsibility of the application that is reading a message to first persist (in storage) the information *before* committing the read of the message on the queue (resulting in its permanent deletion).

The application that is reading a message must foresee re-ordering of messages (when applicable): the messaging system is by design multi-threading/multi-server/multi-site and the order of messages cannot be guaranteed.

#### 4.1.4 Virtual Host

A Virtual Host is a virtual separation of External Communication Layer elements. It contains all elements related to the same domain/concepts.

In the scope of the External Communication Layer, the messages exchanged will be always under the scope of the *'AncillaryServices'* Virtual Host.

**Note:** Currently, it is assumed that the same queue system will be used as for Icaros. Therefore "mvar messages" will be in the same system as other messages (for example: power outage planning).

**Note II: Upcoming RabbitMQ version upgrade consideration**

In the near future, we will need to upgrade our RabbitMQ version, necessitating a transition from classic queues to quorum queues. The impact should be only on ELIA side and therefore transparent for VSPs.

## 4.2 Connection Information

### 4.2.1 Authentication and Authorization

Every Market Party will receive a dedicated User/Password, per environment (Demo and Prod), that Elia will use for authentication.

A Market Party will only have access to its own queues, with a strictly "read" rights. It means that any *'QueueDeclare'* operation will fail. Only *'BasicConsume'* type operation are authorized.

### 4.2.2 Protocols

AMQP(S) 0.9.1 or 1.0 are supported.

### 4.2.3 URLs and ports

The following URLs are available for Production and Demo environments:

- Production: messaging.elia.be
- Demo: messaging-demo.elia.be

The available ports to connect are:

- 5672 for AMQP (Please do not use)
- 5671 for AMQPS

For the Production environment it is strongly advised to use the encrypted port 5671 to avoid any IT security risk.

For AMQPS: the only supported TLS version is 1.2 for market parties already connected to ELIA for the icaros project.

New market parties are requested to use the latest TLS 1.3 protocol given less secure protocols will be phased out in due time aligned with the current market parties.

#### 4.2.4 Virtual Hosts

This Virtual Host has to be specified in the connection parameters when accessing to queues/exchanges:

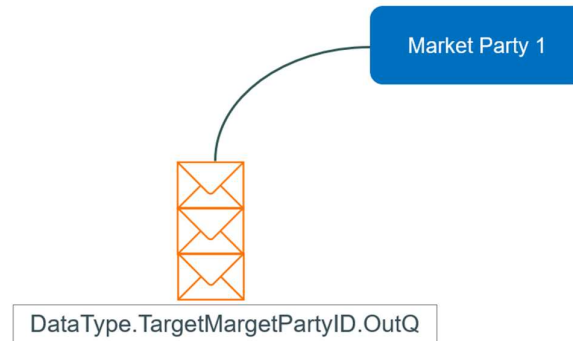
- *'AncillaryServices'* for both Production and Demo environment

### 4.3 Queues and exchanges naming convention

#### 4.3.1 Generic Reading Queue Pattern (Elia to Market Party Communication)

The naming convention used for the Queues to read from (*Elia to Market Party communication*) is:

**[DataType].[TargetMarketPartyID].OutQ**



Where:

- **DataType** describes the messages content (i.e. UnavailabilityAnswered, ActivationRequested, ...)
- **TargetMarketPartyID** is the EIC code of the Market Party to which the message is sent (e.g. 10X1001A1001A094)

Some example of queue names:

- **MvarEventAnswered.10X1001A1001A094.OutQ**
- **ScheduleAnswered.10X1001A1001A094.OutQ**
- ....

Note: As it can be seen, messages in the out-queue come from different services. It is likely that the unavailabilities related to reactive power would have a common prefix. Maybe Mvar.

#### 4.3.2 Generic Writing Exchange Pattern

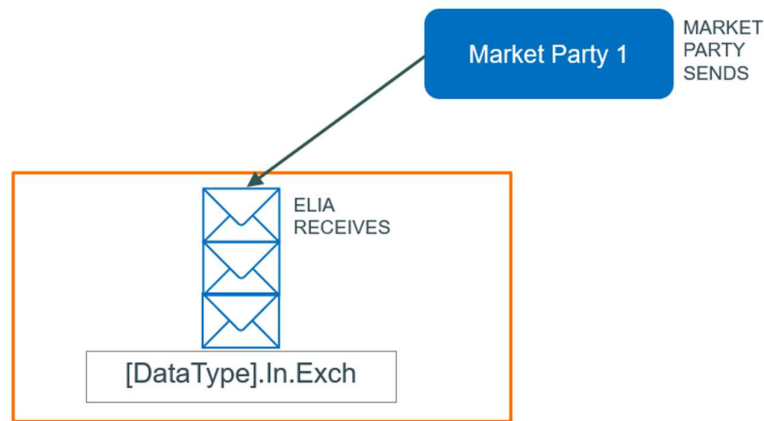
The naming convention used for the Exchange to write to (*Market Party to Elia communication*) is:

**[DataType].In.Exch**

Where:

- **DataType** describes the messages content (i.e. UnavailabilitySubmitted, ActivationAcknowledged, ...)

**Important:** The header properties 'user-id' is needed in order to pass the security check on incoming messages.



Some example of the exchange names:

- `MvarEventSubmitted.In.Exch`
- `ScheduleSubmitted.In.Exch`
- `ReturnToScheduleAcknowledged.In.Exch`

**Note:** As it can be seen, messages in the in-queues come from different services. It is likely that the queues related to reactive power would have a common prefix. Maybe Mvar.

### 4.3.3 Generic Error Patterns

This section details the handling of problematic messages (error handling).

A message is considered as problematic (or *wrong*) when it doesn't respect the agreed format specified in this document. It is therefore important to inform the sender that this message was not understandable. That's the purpose of the "ErrorQueue" and "ErrorExchange".

Note: the Error queues and exchanges are not to be used in case of any other processing error on either side.

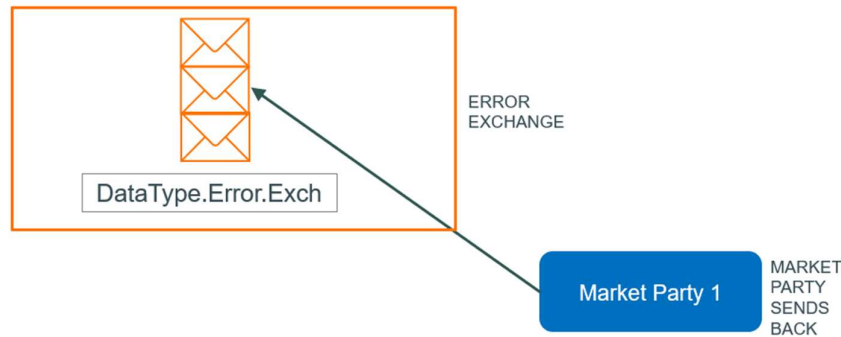
### 4.3.4 Generic Error Exchange Pattern

When the (Market Party) reading process is not able to treat a specific message (read on a \*.OutQ) due to a technical validation problem (i.e. format of the message is not as expected/agreed), the message is sent back to the source using the *Error Exchange*.

It is always the original message that is moved to the 'Error Exchange', and it is the responsibility of the originator of the message (Elia in this case) to handle this error.

The naming convention used for the Error Exchange is:

**[DataType].Error.Exch**



Where :

- `DataType` describes the messages content (i.e. `MvarUnavailabilitySubmitted`, `MvarActivationAcknowledged`, ...)

**Reminder:** The header properties 'user-id' is needed in order to pass the security check on incoming messages.

Some examples of Error Exchange names:

- `MvarEvent.Error.Exch`
- `ScheduleAnswered.Error.Exch`
- `ReturnToScheduleRequested.Error.Exch`

**Note:** As it can be seen, messages in the exchanges come from different services. It is likely that the exchanges related to reactive power would have a common prefix. Maybe `Mvar`.

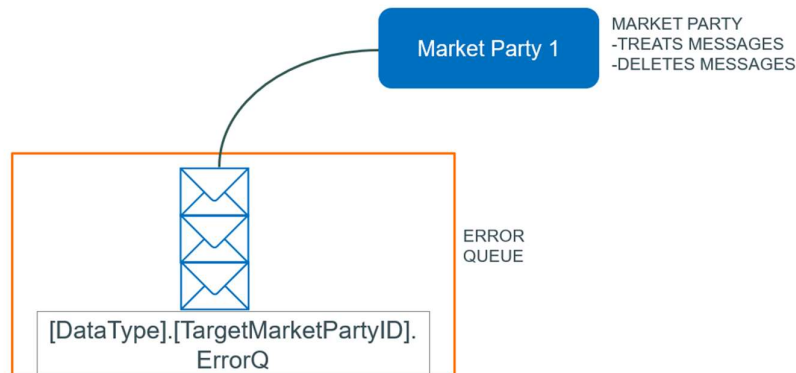
#### 4.3.5 Generic Error Queue Pattern

When the (Elia) reading flow is not able to treat a specific message due to a technical problem (e.g. format is not as expected/agreed), the message is sent back to the source (the Market Party) to inform about the problem.

It is always the original message that is moved to the Error queue, and it is the responsibility of the originator of the message (the Market Party in this case) to handle this error.

When Elia doesn't understand a message sent by a Market Party, the message is sent back on an 'Error Queue' dedicated for the specified Market Party. The naming convention used for the Error Queue to is:

**[DataType].[TargetMarketPartyID].ErrorQ**



Where:

`DataType` describes the messages content (i.e. `MvarEventAnswered`, `MvarActivationRequested`, ...)

- `TargetMarketPartyID` is the EIC code of the Market Party to which the message is sent (e.g.. 10X1001A1001A094)

Some example of queue names:

- `MvarEventSubmitted.10X1001A1001A094.ErrorQ`
- `ScheduleSubmitted.10X1001A1001A094.ErrorQ`
- `ReturnToScheduleAcknowledged.10X1001A1001A094.ErrorQ`

Note: the Market Party is responsible to treat (and delete) the messages available on the Error Queues.

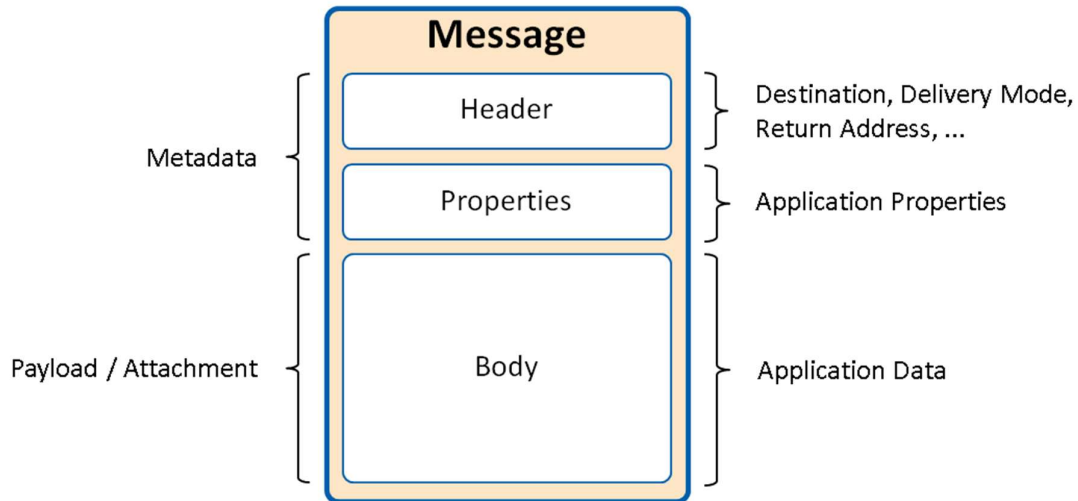
**Note:** As it can be seen, messages in the error queues come from different services. It is likely that the error queues related to reactive power would have a common prefix. Maybe Mvar.



## 4.4 Message operations

### 4.4.1 Message structure

A message is composed of 3 parts:



- The **header** (metadata): contains technical metadata information on the message.
- The **properties** (metadata): contains some technical fields required for the security and tracking of the communication. The description is provided hereafter and must be applied to every message exchanged.
- The **body** (payload): contains a MarketDocument in JSON format, based on the CIM specifications. Every type of message will be described in future sections of this document.

### 4.4.2 Properties section

The properties section is used to transfer structured exchange metadata on the message.

Three properties are mandatory to facilitate the tracing and the security checks of the messages:

- **message\_id**: contains a unique identifier of the message. Every message created on the system must have a unique ID. UUID based ID is recommended.  
I.e.: 5eb8aec9-6f58-4b6d-a318-ad050007bfa4
- **correlation\_id**: contains the unique identifier of the communication. This ID is used to correlate a message with its acknowledgement or answer message.  
If you are the originator of the communication, this ID must be filled in with a unique ID.  
If you are sending a message for a communication that is already initiated (answer or acknowledgement message), the correlation\_id must be copied from the message that is being answered/acknowledged. UUID based ID is recommended.  
I.e.: b30c7c03-eaa9-4c96-97cf-ad050007bfa4

- **user\_id**: contains the VSP's Username used to connect to the messaging platform. This field will be used to apply security validation on the origin of messages sent to Elia.  
I.e.: ISOEXT\DEMOMARKETPARTY1

Some other properties can be set to facilitate the interoperability of the platform, hereafter a non-exhaustive list:

- **content\_type**: describes the content of the Body section of the message. It must be used with a value of: "application/json"
- **timestamp**: date/time of the creation of the message (UTC based) in iso format ISO 8601. For example 2029-04-01T00:00:00.000Z.

The following properties should not be used:

- **expiration**: *do not setup* this property as messages are not supposed to have an automatic expiration date/time.

#### 4.4.3 Header section

The header section can be used to store any information needed. The following header property is required:

- **conversation\_id**: contains the unique identifier of the conversation. It allows the tracking of messages flows spans on more than one message exchange (= multiple correlation-id). The first message sent in a new conversation must be assigned with a unique conversation\_id. This property must be propagated to every message part of the same exchange: this is the conversation. UUID based ID is recommended.  
I.e.: a20c5c13-aaa9-4c96-97cf-ad050007bfa4

As a summary, a message conversation is:

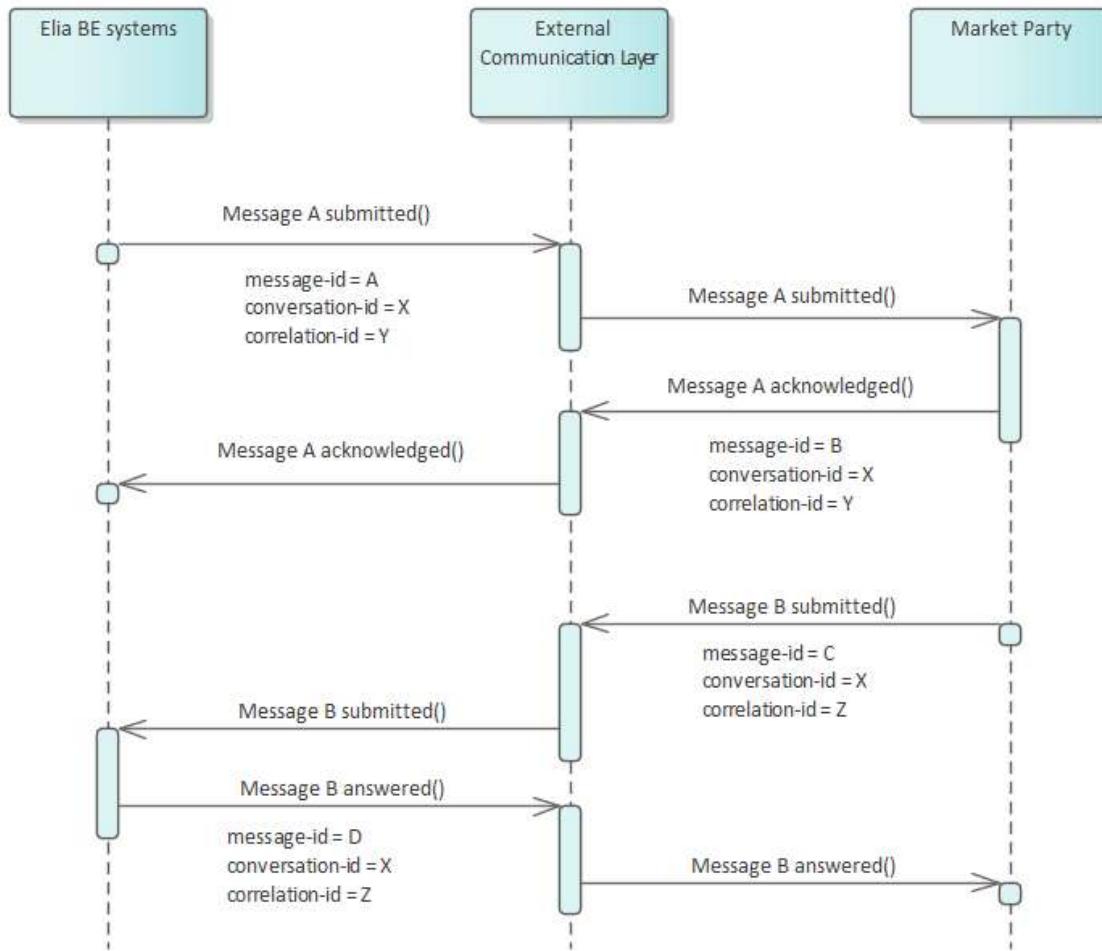
- one unique conversation\_id,
- one or more correlation\_id, depending on the number of "request-reply" scenario involved in the conversation,
- one or more message\_id, depending on the number of messages needed for the exchange.

#### 4.4.4 Example of message\_id, correlation\_id and conversation\_id properties

The description of each message exchange contains a sequence diagram with all messages involved. The following rules must be applied to each sequence:

- Each message of the sequence must have a unique message\_id
- All messages of the sequence must have the same conversation\_id
- Every new message submitted must have a unique correlation\_id
- Every answer or acknowledge must use the same correlation\_id of the message that is being answered or acknowledged

The following example shows a sequence diagram and the use of message-id, conversation-id and correlation-id:



#### 4.4.5 Body

The body of each message contains a MarketDocument in JSON format.

Section 5 details the general JSON structure expected in the body of the message.

The specific details of each MarketDocument are described later in this guide.

#### 4.4.6 Sending a message

Sending a message to Elia is done by writing a specific message (and its metadata: header and properties) to the specified Exchange. The message is sent as soon as the write operation is committed.

#### **4.4.7 Receiving a message**

Receiving a message from Elia is performed by reading the message from the queue and by confirming the operation. The message will be removed from the queue only after the read operation is confirmed ('Ack'). There is no mechanism foreseen for the Market Parties to request sending again the messages that have already been read and confirmed.

## 5 Generic message specifications

### 5.1 JSON format and date format

#### 5.1.1 JSON format

The body of all messages created in the framework of a data exchange must be written in Javascript Object Notation (JSON). This section contains a basic introduction to JSON documents and their structure.

##### 5.1.1.1 JSON Overview

JSON (JavaScript Object Notation) is a text-based and human-readable syntax for storing and exchanging data between applications.

The Media Type is: **application/json**

##### 5.1.1.2 A well formatted message

All elements in the JSON message must be correctly delimited. The use of properly nested start character “{” and end character “}” (in simple values, arrays and objects) is essential if the JSON message is to be read and interpreted correctly.

For example, the following element is correctly delimited:

```
{ "position": 20 }
```

The name of the root element must be present in the JSON message.

For example, for a message of type Acknowledgement\_MarketDocument:

```
{
  "Acknowledgement_MarketDocument": {
    ....
    ....
    ....
  }
}
```

In this example, the block “Acknowledgement\_MarketDocument” is the root of the message and contains all fields as described in the message specification.

**Note!** Each element must respect the indicated lower case or upper case letters.

##### 5.1.1.3 Data types

The following table describes all datatypes allowed in JSON structure specifications that are used in the External Communication Layer:

Data type	Description	Lexical pattern
Number (int)	Represented in base 10 number system. No decimals allowed. No separators are allowed.	<code>[~+]?[0-9]+</code>

Data type	Description	Lexical pattern
	It can support negative values. Example: { "position": 20 } { "volume": -5 }	
Number (decimal)	Represented in base 10 number system. Decimal characters allowed, separated by a point. No other separators allowed. It can support negative values. Example: { "volume": 25.23 }	$[-+]?[0-9]+(\.[0-9]+)?$
String	Sequence of zero or more characters. Must be written in double quotes. Example: { "company": "Elia" }	<code>.*</code>
Boolean	It can be either true or false Example: { "result": true } { "result": false }	<code>true false</code>
Null	Indicates that there is no value Example: "value": null	<code>null</code>
Object	It is an unordered set of fields and values. Surrounded by curly braces {} Example: { "Person":{ "name": "Peter", "age": 20} }	
Array	It is an ordered collection of values. Surrounded by brackets [] Example: { "Point" : [ {"id": 123}, {"id": 124}, {"id": 125} ] }	
White space	It can be inserted between two different tokens in order to facilitate readability. Example: {"name": "Peter"} {"name": "Peter"}	

#### 5.1.1.4 A valid message

In order to be understood, a JSON file must follow a predefined structure. The structure of the message is set out in a "Schema".

A JSON schema specifies the structure of JSON data. It is based on the concepts from XML Schema (XSD), but is JSON-based.

For example, if one of the mandatory elements in the schema is missing then the message is termed "invalid": it does not comply with the Schema. An invalid message will always be rejected by Elia systems.

#### 5.1.1.5 A correct message

The fact that your JSON message is “valid”, does not necessarily mean that it is “correct” in terms of specifying your intended message. Not all the requirements of the JSON message can be defined simply in terms of the elements it contains.

For this reason, every message is subject to a number of “business” validation rules that must be applied and that are not controlled by the Schema.

In some cases Elia applies specific constraints on the messages that it will accept. For example, if Elia only accepts a certain set of values in a string field.

Details on these rules are given in the definition of each message.

#### 5.1.2 Dates and times

##### 5.1.2.1 Format

All dates and datetimes must be expressed in Coordinated Universal Time (UTC) with the format: “YYYY-MM-DDThh:mm:ssZ” (ISO-8601 extended format)

Where

- YYYY refers to the year,
- MM refers to the month
- DD refers to the day
- T is a fixed entry and indicates the start of the time definition
- hh refers to the hour
- mm refers to the minutes
- ss refers to the seconds
- Z is a fixed entry indicating that the Time Coordinate is UTC.

Example 1: 2024-05-10T13:00:00Z means then 10th of May 2024 at 15h in Belgian Summer local time

**Note!** DateTimes values are written in the JSON message as string data type.

##### 5.1.2.2 Daylight saving time

The daylight saving times (DST) issue is solved by the use of UTC time. However, the sender of a message must include the correct amount of time intervals for each case.

Example 1: summer time to winter time in Belgium in 2024.

ISO	Local time	UTC
2024-10-25 00:00+02	0h	2024-10-24T22:00Z
2024-10-25 01:00+02	1h	2024-10-24T23:00Z
2024-10-25 02:00+02	2h	2024-10-25T00:00Z
2024-10-25 02:00+01	at 3h it is 2h	2024-10-25T01:00Z
2024-10-25 03:00+01	3h	2024-10-25T02:00Z

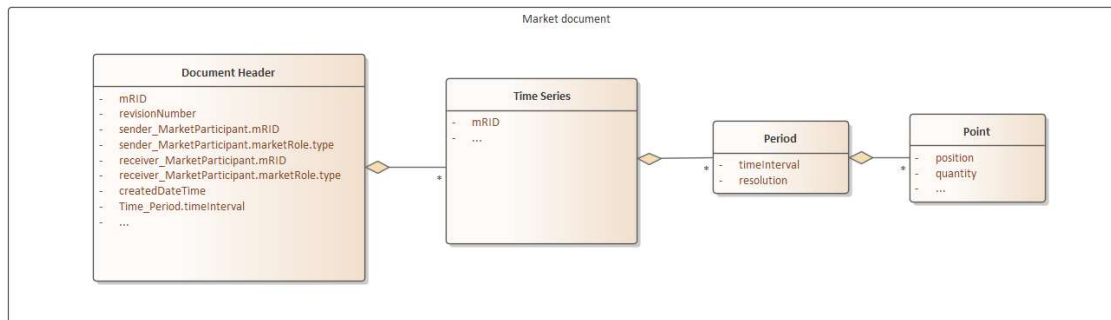
Example 2: winter time to summer time in Belgium in 2020.

ISO	Local time	UTC
2024-03-29 00:00+01	0h	2024-03-28T23:00Z
2024-03-29 01:00+01	1h	2024-03-29T00:00Z
2024-03-29 03:00+02	at 2h it is 3h	2024-03-29T01:00Z
2024-03-29 04:00+02	4h	2024-03-29T02:00Z

## 5.2 Market document structure

The format of all messages described in this document is specified with a MarketDocument.

A MarketDocument is a definition of a message body, which contains the following main elements: header, timeseries, periods and points.



### 5.2.1 Header

It corresponds to a MarketDocument and contains an identification of the document (mRID = master Resource Identifier), a revision number, information about sender and receiver, and a list of timeseries.

A document is defined for a certain time interval with a start and end datetime.

It contains other fields that may be specific for each MarketDocument.

### 5.2.2 Timeseries

Timeseries can contain periods with a start and end date.

It contains other fields that may be specific for each timeseries.

The identification of the timeseries (mRID) must be unique within the message.

### 5.2.3 Period

It is a period of a timeseries with a start and end datetime. This period must be included in the time interval defined in the header.

It contains a list of Points.



The field “resolution” indicates the length (in minutes) of the points.

#### 5.2.4 Point

The number of points must be in line with the duration of the Period.

Field “position” is a sequential field (integer) that indicates the position of the point in the period. It starts from 1.

It contains other fields that may be specific for each MarketDocument.

### 5.3 Identification and versioning

#### 5.3.1 Identification

The header of each MarketDocument used for communication contains an identifier “mRID” and a version number “revisionNumber”.

The mRID of the market document must be unique. UUID based ID is recommended.

#### 5.3.2 Updates

For any update, the same “mRID” must be used, with a higher “revisionNumber”. This message will **completely replace** any existing information of the previous version of this MarketDocument. Note however that Elia will **only take into account current and future values** of sent information, unless explicitly stated otherwise. This includes the ongoing time interval. When updates are sent with only the latter of the day, not sending the current time interval will be considered as an update (removal) of the current time interval information.

For example an initial submission of a document will have an identifier and a revision number equal to 1:

```
{
  "Example_MarketDocument": {
    "mRID": "4e7791aa-df87-4cac-9ee7-3d6c218a0579",
    "revisionNumber": 1

    ...
    <other contents of the message>
    ...
  }
}
```

Any update of the elements contained in the previous message will require a same identifier “mRID”, and a higher revision number:

```
{
  "Example_MarketDocument": {
    "mRID": "4e7791aa-df87-4cac-9ee7-3d6c218a0579",
    "revisionNumber": 2

    ...
    <other contents of the message>
    ...
  }
}
```

```
}

```

After the update, all elements contained in revision 1 of the documents are no longer valid so next to the changed element, elements that are not changed must be sent in again.

Depending on the information flow, it is necessary to explicitly **cancel** either a MarketDocument or a specific timeseries. Same as for updates, this must be communicated using the same MarketDocument mRID but accompanied by a new revision number. As this replaces the last received MarketDocument, other (un)changed timeseries must be sent in as well.

The following example shows what has to be (re)sent in the market document

1. Initial version
2. Deletion of timeseries C
3. Update of timeseries B
4. Update of timeseries A

What needs to be sent:

	Market Document X v1	Market Document X v2	Market Document X v3	Market Document X v4
<b>Timeseries mRID</b>	A B C	A B C (delete)	A B (update) C (delete)	A (update) B (update) C (delete)

In all cases, the latest MarketDocument holds the entire set of correct information.

## 5.4 Message granularity

The content of each message is defined by the granularity of its respective object. The object will thus determine the split of information into different messages.

Granularity is specified in the description of each message.

## 6 Acknowledgement and answer messages

All messages submitted by Elia or by a Market Party will require at least one response message. The need and the format of each response message is described for each message flow.

There are two different kind of responses: acknowledgements and answers.

Acknowledgement and answer messages cannot be responded with any other acknowledgement or answer message. In case an acknowledgement or answer does not respect the rules defined, they will be placed in the corresponding error queue.

### 6.1 Acknowledgement

For all messages sent by Elia to a Market Party, a proof of reception of the message is required.

This is a message with a name "Acknowledged" (for example, ActivationAcknowledged). This type of message is specified with an Acknowledgement MarketDocument.

Fields "received\_MarketDocument.mRID" and "received\_MarketDocument.revisionNumber" must correspond to the mRID and revisionNumber of the MarketDocument that is being acknowledged. If this information is not correct, the acknowledgement is invalid.

Once an acknowledgement message is sent with no errors, no more acknowledges to the same MarketDocument are expected.

**Note:** Elia will not send acknowledgement messages. A Market Party can consider a message as delivered once the write operation has succeeded (more information about sending a message can be found in the description of the External Communication Layer).

#### 6.1.1 Message description

Acknowledgement_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identifier for the MarketDocument
type	Y	Code for type of the MarketDocument <b>A17</b> = Acknowledgement Document
createdDateTime	Y	The timestamp on which the message was sent in iso format ISO 8601 (for example: 2029-04-01T00:00:00.000Z )
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code)
sender_MarketParticipant.marketRole.type	Y	The role code associated with the sender <b>A27</b> = Ressource Provider
receiver_MarketParticipant.mRID	Y	The identification of the receiver (EIC code)
receiver_MarketParticipant.marketRole.type	Y	The role code associated with the receiver <b>A04</b> = System Operator
received_MarketDocument.mRID	Y	The MarketDocument identification (mRID) to which is acknowledged
received_MarketDocument.revisionNumber	Y	The MarketDocument revision number to which is acknowledged. If the Market Document being acknowledged does not have a revision number, 1 should be used here.
Reason	Y	Indicates a status for the acknowledgement.

		This list that can only contain one element.
--	--	--

Reason		
Field	Mandatory	Description
code	Y	The code that represents the acknowledgement <b>A01</b> = Accepted <b>999</b> = Rejected (only allowed in case of technical error)

## 6.2 Answer

For all messages sent by a Market Party to Elia, Elia will send a functional answer after the validation and process of an incoming message.

These are messages with a name “Answered” (for example, MvarEventAnswered. This type of message is specified with a [Confirmation MarketDocument](#).

Fields “confirmed\_MarketDocument.mRID” and “confirmed\_MarketDocument.revisionNumber” must correspond to the mRID and revisionNumber of the message that is being answered. If this information is not correct, the confirmation message is invalid.

Elia will perform a technical validation and a functional validation of each message received and add the validation result details (validation status and possible errors or warnings) in the answer message.

The technical and functional validation rules are described in this document.

**Note:** the answer message is only sent once the original request (message that is being answered) has been fully processed.

### 6.2.1 Message description

Confirmation_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identifier for the market document.
type	Y	Code for type of the MarketDocument. <b>A18</b> = Confirmation report
sender_MarketParticipant.mRID	Y	The identification number of the sender (EIC code). The value must be <b>10X1001A1001A094</b> = Elia
sender_MarketParticipant.marketRole.type	Y	The role code associated with the sender <b>A04</b> = System Operator
receiver_MarketParticipant.mRID	Y	The identification ID of the receiver (EIC code).
receiver_MarketParticipant.marketRole.type	Y	The role code associated with the receiver <b>A27</b> = Resource Provider
createdDateTime	Y	The timestamp on which the confirmation message was sent
confirmed_MarketDocument.mRID	Y	mRID of the MarketDocument that is being replied
confirmed_MarketDocument.revisionNumber	Y	revisionNumber of the MarketDocument that is being replied
Reason	Y	List with reasons associated to the MarketDocument It indicates the response status of the message

Confirmed_TimeSeries	N	<p>The timeseries replied to. It contains a status for each timeseries of the MarketDocument that is being replied.</p> <p>In case the answer contains error on MarketDocument level, this list will be empty</p>
----------------------	---	---

Confirmed_TimeSeries		
Field	Mandatory	Description
mRID	Y	mRID of the timeseries replied to
Reason	Y	This list that can only contain one element with the reason associated to the timeseries

Reason		
Field	Mandatory	Description
code	Y	<p>For Confirmation_MarketDocument:  <b>A01</b> = Message fully accepted  <b>Y99</b> = Message is waiting for confirmation (currently not foreseen to be used)  <b>Y98</b> = Message accepted with warnings (currently not foreseen to be used)  <b>A02</b> = Message fully rejected</p> <p>In case of error (<b>A02</b>) MarketDocument level, a second reason code will be included to detail the rejection.</p> <p>For accepted Confirmed_TimeSeries:  <b>B06</b> = Accepted</p> <p>For accepted with warning Confirmed_TimeSeries:  This is the warning reason code described in the validation rules</p> <p>For rejected Confirmed_TimeSeries:  This is the error reason code described in the validation rules</p> <p>For waiting for confirmation Confirmed_TimeSeries:  This is the reason code described in the validation rules</p>
text	Y	The text associated with the status code ("Message fully accepted" or "Message fully rejected")
details	N	(if any) details to help pinpoint the reason for the issue.

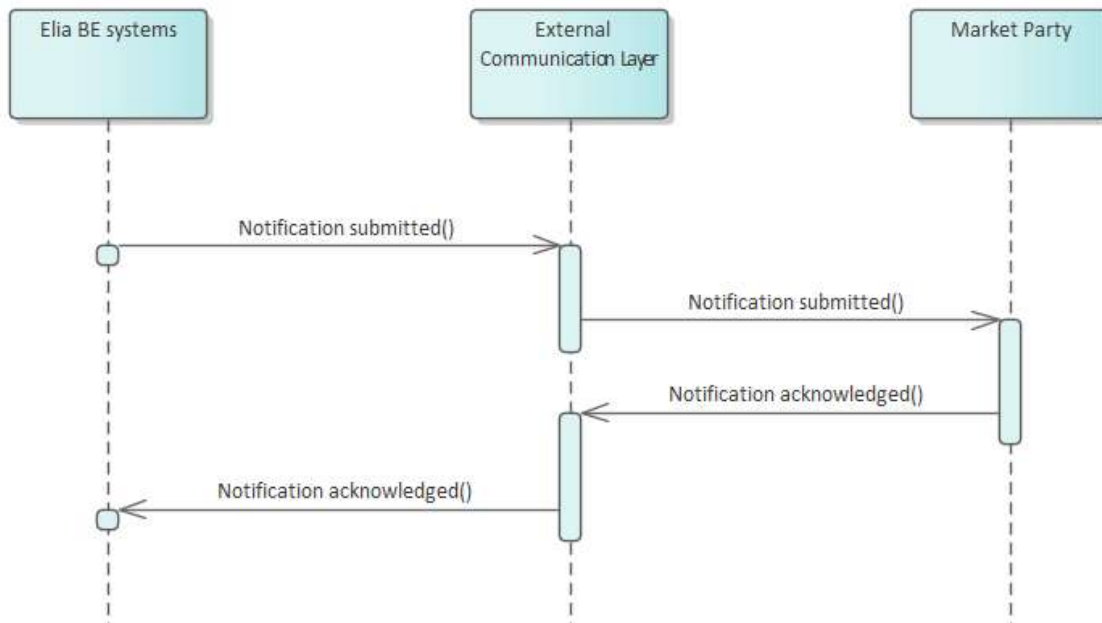
## 7 Notification messages

### 7.1 Description

Notification messages allow Elia to notify Market Parties of certain events. The message itself is generic and the reason indicates the event that happened. This message type can be used for a range of different notifications and can be contextualized via associated objects. Even though the notifications can be of a very different nature, this standardized message allows automatic processing. Note that the same notifications can be offered via email depending on the preference of communication of the concerned Market Party.

The catalog of the notifications and the associated objects types that will give them context will be exposed via a webclient where they can be consulted.

An acknowledgement message will be expected after the reception of the notification message.



### 7.2 Notification submitted message

#### 7.2.1 Message granularity

The message granularity will depend on the notification. It will include one or more objects for a specific notification.

#### 7.2.2 Message timeframe

Dependent on type of notification.

#### 7.2.3 Message description

A document [Notification MarketDocument](#) is used for the notification.

Optional fields of the market document that are not described in this chapter cannot be used.

Notification_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identifier for the message
revisionNumber	Y	Version number for the MarketDocument
type	Y	<b>Z05</b> = Notification report
sender_MarketParticipant.mRID	Y	The identification ID of the sender (EIC code). Fixed value: <b>10X1001A1001A094</b> = Elia
sender_MarketParticipant.marketRole.type	Y	The role code associated with sender. Fixed value: <b>A04</b> = System Operator
receiver_MarketParticipant.mRID	Y	The identification number of the receiver (EIC code)
receiver_MarketParticipant.marketRole.type	Y	The role code associated with receiver <b>Z02</b> = Scheduling Agent <b>A46</b> = Balancing Service Provider <b>Z03</b> = Outage Planning Agent <b>A27</b> = Ressource Provider (used for Voltage Service Provider)
createdDateTime	Y	The timestamp on which the message was sent
Reason	Y	This list can only contain one element
PrimaryObject	N	List of a Primary Object Type & Object instances related to the notification reason

PrimaryObject		
Field	Mandatory	Description
type	Y	Type of object. The potential objects types that can be used will be communicated at a later time
value	Y	Id or value of the object for a specific object type. Can be a date, an EAN code, Bid Group Id, ...
SecondaryObject	N	List of Secondary Object types & Object instances related to an existing primary object. Only used for notifications that use nested objects.

SecondaryObject		
Field	Mandatory	Description
type	Y	Type of object. The potential objects types that can be used will be communicated at a later time
value	Y	Id or value of the object for a specific object type. Can be a date, an EAN code, Bid Group Id, ...

Reason		
Field	Mandatory	Description
code	Y	The code that represents the notification. The potential codes that can be used will be communicated at a later time
text	Y	The notification text. This can be missing information or the detection of a forced outage for example. The exhaustive list will be published at a later time

### **7.3 Notification acknowledged message**

#### **7.3.1 Message granularity**

An acknowledgement must be sent for each notification message received.

#### **7.3.2 Message timeframe**

The acknowledgement message must be sent at the moment of the reception of the notification message

#### **7.3.3 Message description**

As described in the definition of acknowledgement message ([see Acknowledgement and answer messages](#)).



## 8 Voltage Service Provider Guide

### 8.1 Role overview

A Voltage Service Provider is involved in the following communications:

1. VSP can submit MVAR events (availabilities changes, ...)

Main communication channel:

- The External Communication Layer put in place by Elia and to be used for the exchange of asynchronous messages between Elia and Market Parties.
- The Webclient available for the Market Parties to upload their data manually. The webclient offers a user interface where Market Parties can have a view on the status of (automatically or manually) sent data.

Back-up communication channel:

- Each of the main communication channel is the main back-up for the other one.
- In case of issues in both channels, phone and email could also be used.

2. Elia can communicate Mvar Activation Requests

Main communication channel:

- The External Communication Layer put in place by Elia and to be used for the exchange of asynchronous messages between Elia and Market Parties.

Back-up communication channel:

- Phone calls
- Emails

*AS IS, the backup communication channel is the activations by phone. It is expected that this channel will remain the back up for sending activations.*

3. **Notifications** are sent by the Notification Service via:

- JSON messages via ECL
- Automated emails

### 8.2 Submitting MVAR events

#### 8.2.1 Description

Once the MVAR Evolutions project is live: at the beginning of each year an availability plan will automatically be constructed for all the years ahead up to year + 3.

- First, all units of a VSP will be assumed to be systematically available at the band comprised between the contractual Qtech min and Qtech max (as defined in Annex 1 of the VSP terms & conditions)
- Depending on the reactive power availability logic declared by the VSP in Annex 1 of the VSP contract, Elia will use OPA and/or SA data to determine a preliminary Availability Status for each Technical Unit of the VSP for all the years up to year+3. It is then the responsibility of the VSP to make corrections of reactive power availability that could not be concluded from OPA and/or SA data.
- The service control type will also be derived from the contractual values.

In case of **band deviation** to the reactive power availability plan, the VSP responsible for the deviating unit is requested to submit an **unavailability event**. There are three types of unavailability events:

- **Forced Outage:** Are for units suffering from an unexpected event.
- **Testing:** Are events happening just after a maintenance or before the final commissioning for which the goal is to verify the correct operation of an unit. These events could have an impact on the power grid and must therefore be announced 1 month in advance at least.
- **Planned unavailability:** in case of future events that lead to a reduction of the available band.

Notes:

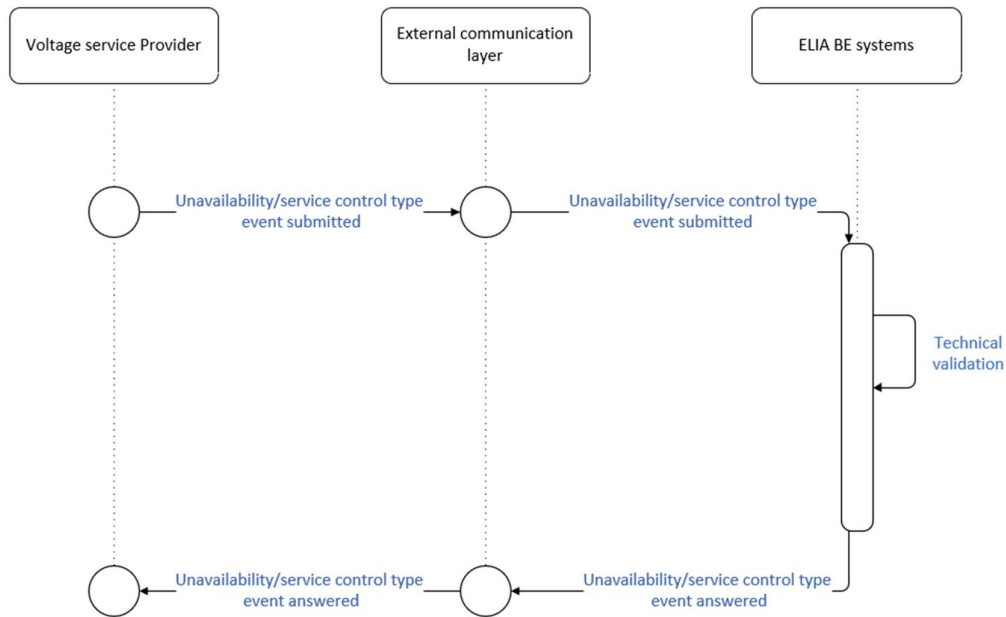
- The band could be set to 0 for any of those events or any other fixed value if setting  $Q_{min}=Q_{max}$ .
- Some events can be submitted more than 3 years in advance even though the availability plan is not yet generated.

In case of a **variation of the service control type** of an unit: the responsible VSP needs to submit a **service control type event**. The VSP can submit either of the four following events.

- **Manual mode unavailable:** Controlling units can declare that they can only do automatic control. Therefore they must declare they can not use their manual mode.
- **Automatic control unavailable:** Controlling units can declare that they can only do manual control as their automatic control mode is unavailable.
- **Power saving mode:** Specific mode in which automatic control is not readily available but could be in a short timeframe (see T&C for more details). Manual control is considered available. A power saving mode can either have an end date or none. In case no end date is specified, the exiting power saving mode message is expected in order to end the event.
- **Exiting power saving mode:** Message to be sent when exiting power saving mode with no end date.
- In case an end date was not specified for a delivery point, the exiting power saving mode indicates the mode has ended.
  - o In case of power saving mode with end date, exiting power saving mode will have no impact. The correct way would be to amend the end of the power saving mode. If the exiting power saving mode has no start date, it is assumed the exit happens at the time of reception.
  - o If the exiting power saving mode has a start date, the power saving mode will be assumed to end at the start date time.

- In any case, the end date for a exiting power saving mode message will not be considered.

All events types are always automatically accepted, but they could be asked to be changed manually at a later phase if some issues would be spotted. Historically it is not something that happened very often. All cancellations will therefore be motivated and not mandatory. Forced outage and power saving mode events are not foreseen to be cancellable.



### 8.2.2 Queue information (section still in development)

The below table contains the queues and exchange to send and receive message in a normal scenario.

Message Type	Description	Sender	Receiver	Queue/Exchange
<b>Mvar event Submitted</b>	Submission of new mvar event or mvar event update (mvar events are both service control type and unavailability events)	VSP	Elia	MvarEventSubmitted.In.Exch
<b>Mvar event Answered</b>	Answer to a mvar event message	Elia	VSP	MvarEventAnswered.[TargetMarketPartyID].OutQ

### Error queues (section still in development)

In case of errors, the following queues and exchanges are used.

Message Type	Sender	Receiver	Queue/Exchange
<b>Mvar event Submitted</b>	Elia	VSP	MvarEventSubmitted.[TargetMarketPartyID].ErrorQ

Message Type	Sender	Receiver	Queue/Exchange
Mvar event Answered	VSP	Elia	MvarEventAnswered.Error.Exch

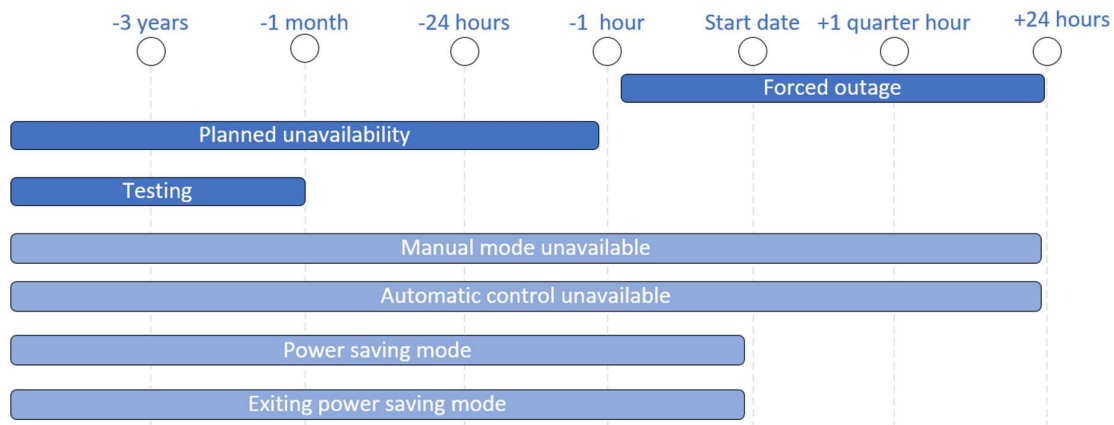
### 8.2.3 MVAR event message submission

#### 8.2.3.1 Message granularity

For both mvar and service control type events, the granularity is set on **event** and **delivery point** level. Therefore for each combination of those two objects, only one message is expected per sent version.

#### 8.2.3.2 Message timeframe

The below image shows the time windows in which messages can be sent.



Control service type events should be sent before or at the moment the switch happens but a reasonable technical delay will be allowed.

As it can be seen on the here above schema, events can be submitted at a very long time in advance even before an unavailability plan is generated (a [performance limit](#) is set, though, which de facto will be limiting the possibility to submit messages in advance, depending on the format of submission).

The **exiting power saving mode** message can only be sent after an entering power saving mode message was sent to indicate the previous message is ended.

#### 8.2.3.3 Update and withdrawals

##### Update of unavailabilities and service control type events

The rules governing the update of already submitted events can be found in the generic section of this document. But also the rules of validation for their respective events (except timeframe, see here below).

An update must have the same market document with a revision number higher than the last version of that same market document previously submitted.

The market document needs to include a timeseries and available\_period block (see the message description section here below).

### Withdrawing unavailability and service control type events

To withdraw unavailabilities, a Market Document including the TimeSeries block but without the Available\_Period block must be used and the docStatus should be set to “Withdrawn” (**A13**).

#### 8.2.3.4 Unavailability event message description

An mvar unavailability market document is the message that must be used in order to submit the band amendments to an availability plan.

Optional fields of the MarketDocument that are not described in this chapter cannot be used.

MVAR_Unavailability_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identifier for the MarketDocument
revisionNumber	Y	Version number for the MarketDocument
type	Y	Code for type of MarketDocument. <b>Z17</b> = Mvar Unavailability Document
process.processType	Y	Code for type of process. <b>Z19</b> = Voltage Unavailability Information
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code)
sender_MarketParticipant.marketRole.type	Y	The role code associated with the sender: <b>A27</b> = Ressource Provider
receiver_MarketParticipant.mRID	Y	The identification of the receiver (EIC code): <b>10X1001A1001A094</b> = Elia TSO
receiver_MarketParticipant.marketRole.type	Y	The role code associated with the receiver: <b>A04</b> = System Operator
createdDateTime	Y	The timestamp on which the message was sent. In isoformat ISO 8601 .
unavailability_Time_Period.timeInterval	Y	The start and end date and time of the period to which the unavailability refers to
docStatus	N	Status only to be used to identify an mvar unavailability document that has been cancelled. <b>A13</b> = Withdrawn
TimeSeries	Y	This list can contain only one element.

TimeSeries		
Field	Mandatory	Description
mRID	Y	Sender's identification of the timeseries
businessType	Y	Identifies the nature of the unavailability event. The following CIM codes are used <b>A53</b> = Planned Maintenance (Planned Unavailability)

TimeSeries		
Field	Mandatory	Description
		<b>A54</b> = Unplanned Outage (Forced Outage) <b>B83</b> = Testing
registeredResource.mRID	Y	The delivery point EAN representing the point for which the unavailability is sent
start_DateAndOrTime.date	Y	The date at which the unavailability event starts
start_DateAndOrTime.time	Y	The time at which the unavailability event starts
end_DateAndOrTime.date	Y	The date at which the unavailability event ends
end_DateAndOrTime.time	Y	The time at which the unavailability event ends
curveType	Y	<b>A01</b> = Sequential fixed size block (default if no availability). Used when the available period is constructed using same resolution. <b>A03</b> = Variable sized block. Used when the available period is constructed using different resolution.
quantity_Measure_Unit.name	Y	The measure of the qmin_submitted and qmax_submitted. Fixed value: <b>MAR</b>
reason_code	Y	Three different codes are allowed Y231="Human" Y232="Technical" Y233= "Other"
reason_text	Y	Free text box (Minimum 10 characters & 1 blank space)
Available_Period	N	This list allows 1 or more elements to compose periods with different intervals. Only optional in case of withdrawal.

Available_Period		
Field	Mandatory	Description
timeInterval	Y	The start and end date and time to which the available period refer to
resolution	Y	Amount of time for each interval in which a data value is defined. For example: <b>PT1M</b> = per minute <b>PT15M</b> = 15 minutes <b>PT1H</b> = 1 hour <b>PT1D</b> = 1 day <b>PT1MO</b> = 1 month
Point	Y	List of points associated to the period.  It should contain as many points as needed to complete the period. If only 1 point is given, it is assumed that the same qmin and qmax are applicable for the entire period.

Point		
Field	Mandatory	Description
position	Y	The interval number defining which position in the period is indicated. It must start at 1.

Point		
Field	Mandatory	Description
Qmin_submitted	Y	The lower limit of the new available band (can be negative)
Qmax_submitted	Y	The higher limit of the new available band (can be negative)

timeInterval		
Field	Mandatory	Description
start	Y	The start date and time of the interval
end	Y	The end date and time of the interval

### 8.2.3.5 Service control type event message description

An mvar service control type market document is the message that must be used in order to submit the service control type amendments to an availability plan.

**To be noted:** Contrary to all other events, power saving mode is not expected to have a known end date. Therefore two messages are foreseen. One when entering the mode and one when exiting the mode. It is also possible to use the end date if desired.

Optional fields of the MarketDocument that are not described in this chapter cannot be used.

MVAR_ServiceControlType_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identifier for the MarketDocument
revisionNumber	Y	Version number for the MarketDocument
type	Y	Code for type of MarketDocument. <b>Z18</b> = Mvar Service Control Type Document
process.processType	Y	Code for type of process. <b>Z19</b> = Voltage Service Control Type Information
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code)
sender_MarketParticipant.marketRole.type	Y	The role code associated with the sender: <b>A27</b> = Resource Provider
receiver_MarketParticipant.mRID	Y	The identification of the receiver (EIC code): <b>10X1001A1001A094</b> = Elia TSO
receiver_MarketParticipant.marketRole.type	Y	The role code associated with the receiver: <b>A04</b> = System Operator
createdDateTime	Y	The timestamp on which the message was sent in iso format ISO 8601.
serviceControlType_Time_Period.timeInterval	Y	The start and end date and time of the period to which the service control type event refers to
docStatus	N	Status only to be used to identify a service control type document that has been cancelled. <b>A13</b> = Withdrawn
TimeSeries	Y	This list can contain only one element.

TimeSeries		
Field	Mandatory	Description
mRID	Y	Sender's identification of the timeseries
businessType	N	Not used at the moment, field can nevertheless be sent empty or with anything.
registeredResource.mRID	Y	The delivery point EAN representing the point for which the service control type is sent
start_DateAndOrTime.date	Y	The date at which the service control type event starts
start_DateAndOrTime.time	Y	The time at which the service control type event starts
end_DateAndOrTime.date	N	The date at which the service control type event ends
end_DateAndOrTime.time	N	The time at which the service control type event ends
curveType	Y	<b>A01</b> = Sequential fixed size block (default if no availability). Used when the available period is constructed using same resolution. <b>A03</b> = Variable sized block. Used when the available period is constructed using different resolution.
reason	Y	Three different values are allowed "Technical", "Human" or "Other"
reason_details	Y	Free text box (Minimum 20 characters & 4 blank spaces)
available_Period	N	This list allows 1 or more elements to compose periods with different intervals. Only optional in case of withdrawal.

Available_Period		
Field	Mandatory	Description
timeInterval	Y	The start and end date and time to which the available period refer to
resolution	N	Amount of time for each interval in which a data value is defined. For example: <b>PT1M</b> = per minute <b>PT15M</b> = 15 minutes <b>PT1H</b> = 1 hour <b>PT1D</b> = 1 day <b>PT1MO</b> = 1 month
Point	Y	Lists of points associated to the period.  It should contain as many points as needed to complete the period. If only 1 point is given, it is assumed that the same service control type is applicable for the entire period.

Point		
Field	Mandatory	Description
position	Y	The interval number defining which position in the period is indicated. It must start at 1.
serviceControlType	Y	Accepts the following values: "Manual mode unavailable", "automatic mode unavailable", "Power saving mode", "Exiting power saving mode",

timeInterval



Field	Mandatory	Description
start	Y	The start date and time of the interval
end	N/Y	The end date and time of the interval. The field is optional only for power saving mode entry or exit.

#### 8.2.3.6 Message resolution and performance

Considering the potential duration of unavailability events and to preserve the best performances of the system, a maximum of 120 time intervals are allowed for each resolution submitted ([see validation rules](#)).

#### 8.2.4 Validation of a MVAR event message submitted

All messages will be subjected to the [generic](#) and [specific](#) rules. Those rules are mostly to avoid internal logic issues. No rules are present to check the “business” reasoning behind unavailabilities. The good faith of the VSP is expected and only post-check will be performed to see if there are any misuses of the tool.

If no issues are seen in the generic or specific rules, the submission will be automatically accepted.

For example:

- An unavailability event claiming that a unit will be able provide/absorb more reactive power than it is, contractually, capable of providing/absorbing will be considered erroneous.
- If two unavailabilities are submitted for the same unit, on the same period, with same identification variables but with different values, the system is unable to understand which one makes the most sense. Therefore the latest submission will be refused.
- If a VSP is submitting forced outage with the only purpose to avoid penalties: no pre-check will limit the submissions but post-check might lead to some consequences for the VSP “cheating”.
- All other messages not breaking any generic or specific rules will automatically be accepted.

#### 8.2.5 MVAR answer messages

The answers to unavailability events are submitted by Elia and received by the VSP.

All unavailability event messages should be answered by Elia.

**Note:** Acknowledgement to answers sent by Elia are not supported.

##### 8.2.5.1 Message granularity

One confirmation message will be sent for each event submitted.

##### 8.2.5.2 Message timeframe

Answers will be sent as soon as the submitted event has been processed.

### 8.2.5.3 Message description

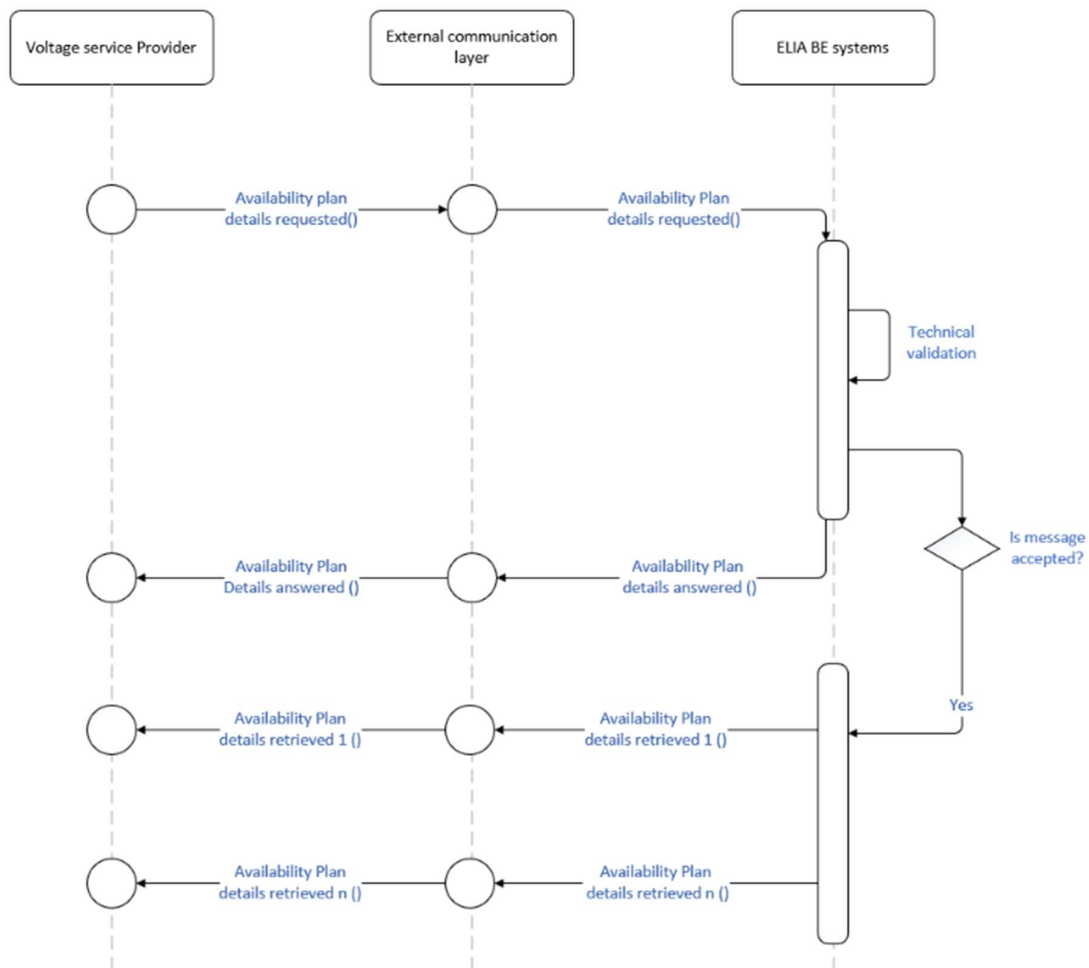
[See acknowledgement and answer messages](#)

## 8.3 Retrieving MVAR availability plan details

### 8.3.1 Description

Voltage service provider are able to retrieve their own **daily** availability plan sent via the External Communication Layer. Only the most recent MVAR availability plans, comprised of the most recent market documents received at the moment of the request, will be returned. Elia will send back each Markert Document in a separate message.

The VSP should therefore expect an answer on the request itself + 1 message per Market Document.



**Note:** in the next section, the availability plan overview allowing to retrieve availabilities for 1 year is described.

### 8.3.2 Queue information (section still in development)

The here below table contains the queues and exchanges to send and receive messages in a normal scenario. The MVAR Availability Plan Details Answered queue will be used to indicate whether the request was valid and whether or not results were found. If the request was valid and results were found, they will be published on the MVAR Availability Plan Retrieved queue.

Message Type	Description	Sender	Receiver	Queue/Exchange
<b>Mvar Availability Plan Details Requested</b>	Requesting Availability Plan to be retrieved	VSP	Elia	MvarAvailabilityPlanDetailsRequested.In.Exch
<b>Mvar Availability Plan Details Retrieved</b>	The result of the requested Availability Plan	Elia	VSP	MvarAvailabilityPlanDetailsRetrieved.[TargetMarketPartyID].OutQ
<b>Mvar Availability Plan Details Answered</b>	The answer to the request message	Elia	VSP	MvarAvailabilityPlanDetailsAnswered.[TargetMarketPartyID].OutQ

### Error Queues (section still in development)

Only in case of errors, the here below queues and exchanges should be used.

Message Type	Description	Sender	Receiver	Queue/Exchange
<b>Mvar Availability Plan Details Requested</b>	Requesting Availability Plan to be retrieved	ELIA	VSP	MvarAvailabilityPlanDetailsRequested.[TargetMarketPartyID].ErrorQ
<b>Mvar Availability Plan Details Retrieved</b>	The result of the requested Availability Plan	VSP	ELIA	MvarAvailabilityPlanDetailsRetrieved.Error.Exch
<b>Mvar Availability Plan Details Answered</b>	The answer to the request message	VSP	ELIA	MvarAvailabilityPlanDetailsAnswered.Error.Exch

### 8.3.3 MVAR availability plan details: request message

#### 8.3.3.1 Message granularity

Requesting Availability Plan messages are limited to **one day** of availability plan details per delivery point.

Input parameters can be used for filtering. If the input parameters are not used, all relevant Market Documents for the requested period will be returned.

Elia will send back each Market Document in a separate message.

#### 8.3.3.2 Message timeframe

Requests will be limited in frequency and period for performance reasons as described in the validation rules.

#### 8.3.3.3 Message description

A MVAR request Marketdocument is needed to request an availability plan

MVAR_Request_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identification of the Request Market Document
type	Y	Type of market document. <b>Z25</b> = MVAR Availability Plan Request
process.processType	Y	Code for type of process: <b>Z26</b> = MVAR Availability plan day
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code)
sender_MarketParticipant.marketRole.type	Y	The role code associated with the sender: <b>A27</b> = <b>Resource Provider</b>
receiver_MarketParticipant.mRID	Y	The identification number of the Receiver(EIC code). Fixed value: <b>10X1001A1001A094</b> = Elia
receiver_MarketParticipant.marketRole.type	Y	The role code associated with sender. Fixed value: <b>A04</b> = System Operator
createdDateTime	Y	The date and time of the creation of the Request Market Document
request_Period.timeInterval	Y	The beginning and ending date and time of the period covered by the document. This cannot exceed an execution day.
Request_TimeSeries	Y	Request_TimeSeries contained in the message

Request_TimeSeries		
Field	Mandatory	Description
mRID	Y	Unique identification of the Request_TimeSeries within the market document
Requested_RegisteredResource.mRID	N	The delivery point EAN representing the point for which the availability plan is sent. If not used, this filter is not applied

		and availability plans will be returned for all Delivery Points within VSP's portfolio.
--	--	---

timeInterval		
Field	Mandatory	Description
start	Y	The start date and time of the interval
end	Y	The end date and time of the interval

### 8.3.4 Validation of MVAR availability plan details request message

Requests will be subjected to all [generic validation rules](#) & [specific validation rules](#).

- When valid, the MVAR Availability Plan Retrieval message described in the next section will be returned.
- When invalid, the request will be answered with an answer message as described in [acknowledgement and answer messages](#). The following specific reason codes will be sent out

Validation Rule	Reply Status	Reason Code	Level
The request did not generate any results	Accepted with warnings	Y11	MarketDocument
The number of requests per 15 minutes cannot exceed the threshold limit	Reject message	Y10	MarketDocument
The number of days requested can not exceed the threshold limit	Reject message	Y09	MarketDocument

### 8.3.5 Retrieved message: Availability Plan details

- The details of the the availability plan will contain per quarter hour
  - o A pair of Qmin and Qmax.
  - o A service control type flag (Contractual values, Manual Mode Only, Power saving mode).
- Only the latest version of a Market Document will be sent.

#### 8.3.5.1 Message granularity

- For each combinaison of delivery point & day, a Market Document will be generated.
- Each Market document is sent in a separate message.

#### 8.3.5.2 Message timeframe

Messages will be returned as soon as possible after a valid request.

#### 8.3.5.3 Message description

MVAR_AvailabilityPlan_MarketDocument (Exactly one element per message)		
Field	Mandatory	Value(s)

mRID	Y	Unique identification of the market document (UUID)
type	Y	Type of market document. Fixed value: <b>Z27</b> = MVAR Availability Plan Document
process.processType	Y	Code for type of process: <b>Z28</b> = MVAR Availability Plan day
process.classificationType	Y	Defines whether the mvar availability plan is an aggregation or a classification. Fixed value: <b>A01</b> = detail type
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code). Fixed value: <b>10X1001A1001A094</b> = Elia
sender_MarketParticipant.marketRole.type	Y	The role code associated with sender. Fixed value: <b>A04</b> = System Operator
receiver_MarketParticipant.mRID	Y	The identification of the receiver (EIC code)
receiver_MarketParticipant.marketRole.type	Y	The role code associated with receiver: <b>XXX</b> = Voltage service provider
createdDateTime	Y	The date and time of the Market Document generation by Elia
AvailabilityPlan_Time_Period.timeInterval	Y	The date and time of the day to which the availability plan refers to (execution date)
domain.mRID	Y	<b>10YBE-----2</b> = Belgian bidding zone
TimeSeries	Y	This list only allows 1 element

TimeSeries		
Field	Mandatory	Value(s)
mRID	Y	Sender's identification of the timeseries.
Version	Y	Version of the availability plan
businessType	Y	Identifies the trading nature of the timeseries: <b>Z29</b> =Mvar Availability status
objectAggregation	Y	Identifies how the object is aggregated. Fixed value: <b>Z01</b> = Delivery Point
registeredResource.mRID	Y	The delivery point EAN representing the point for which the availability plan is sent
ExecutionDate	Y	The target date to which the availability plan refers to
Period	Y	This list allows up to 100 elements (1 per quarter hour) that composes the availability plan of an execution day

Period		
Field	Mandatory	Value(s)
timeInterval	Y	The start and end date and time to which the availability plan timeseries refer to
resolution	Y	Amount of time for each interval in which a data value is defined. Fixed value: <b>PT15M</b> = 15 minutes
Point	Y	List of points associated to the period.  It should contain as many point as needed to complete the period.

Point		
Field	Mandatory	Value(s)

position	Y	The interval number defining which position in the timeseries is indicated. It must start at 1.
AvailabilityStatus	Y	The availability status of the interval: Unknown = 0 PlannedUnavailable = 1 Testing = 2 ForcedOutage = 3 Available = 4
AvailabilityStatusDetail	N	Free text allowing to determine how the status was determined (via active power outage planning for example)
Qmin	Y	The lower limit of the available band
Qmax	Y	The higher limit of the available band
ServiceControlTypeStatus	Y	The service control type status of the interval: ExitingPowerSavingMode = 0 PowerSavingMode = 1 ManualModeUnavailable = 2 AutomaticModeUnavailable= 3

#### 8.3.5.4 Answered message : Mvar availability plan details

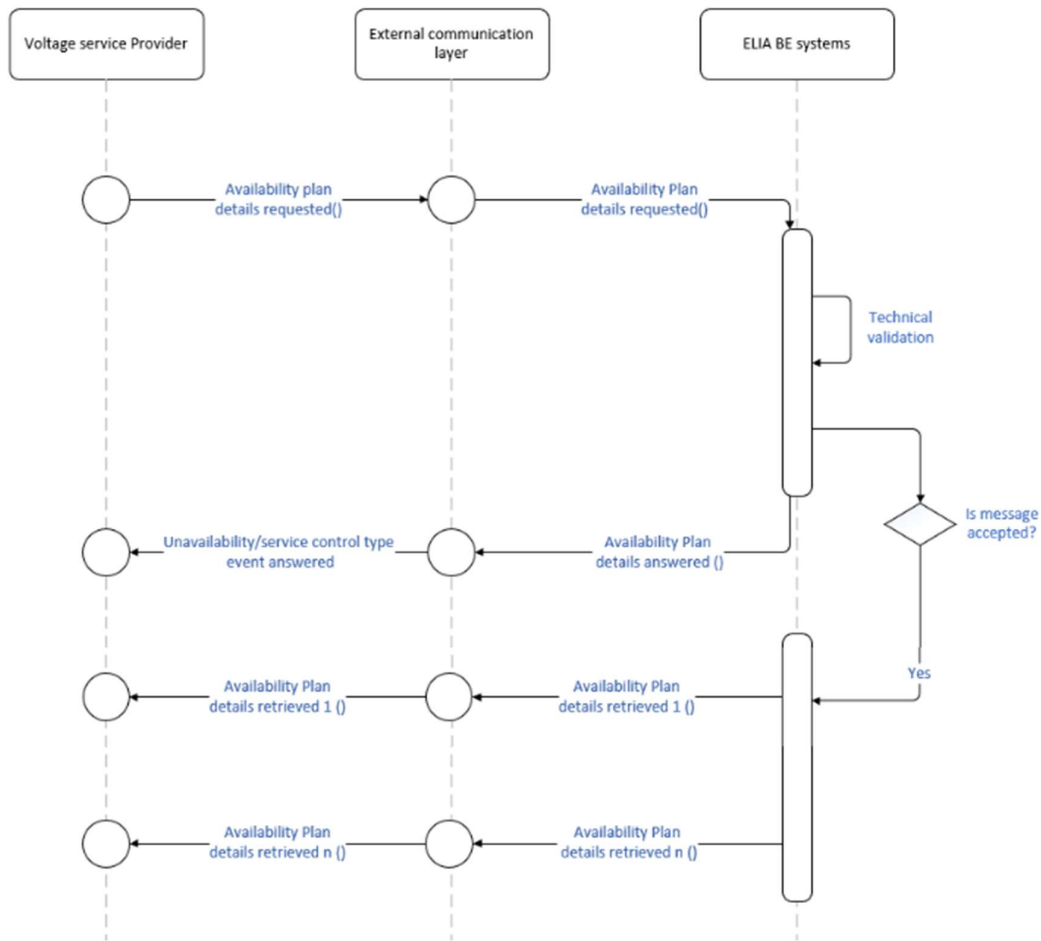
The answer message is identical to the ones found in [acknowledgement and answer messages](#).

## 8.4 Retrieving MVAR availability plan overview

### 8.4.1 Description

This message exchange allows Voltage Service Providers to retrieve their yearly availability plan overview. As for the MVAR availability plan, Elia will return the latest accepted Market Documents at the time of the request.

The below flow describes the process of requesting the availability plan overview via the the external communication layer (it is the same as for the MVAR availability plan).



#### 8.4.2 Queue information (section still in development)

The here below table contains the queues and exchanges to send and receive messages in a normal scenario. The MVAR availability plan overview answered queue will be used to indicate

- whether the request was valid
- whether results were found

If the request was valid and results were found, results will be published on the MVAR availability plan retrieved queue (as for the mvar availability plan details).

Message Type	Description	Sender	Receiver	Queue/Exchange
<b>MVAR Availability Plan overview Requested</b>	Requesting Availability Plan overview to be retrieved	VSP	Elia	MvarAvailabilityPlanOverviewRequested.In.Exch
<b>MVAR Availability Plan Overview Retrieved</b>	The result of the requested Availability Plan overview	Elia	VSP	MvarAvailabilityPlanOverviewRetrieved.[TargetMarketPartyID].OutQ



<b>MVAR Availability Plan Details Answered</b>	The answer to the request message	Elia	VSP	MvarAvailabilityPlanOverviewAnswered.[TargetMarketPartyID].OutQ
--	-----------------------------------	------	-----	---

### Error queues (section still in development)

Only in case of errors: the following queues and exchanges are to be used

Message Type	Sender	Receiver	Queue/Exchange
<b>Availability Plan Overview Requested</b>	Elia	VSP	MvarAvailabilityPlanOverviewRequested.[TargetMarketPartyID].ErrorQ
<b>Availability Plan Overview Retrieved</b>	VSP	Elia	MvarAvailabilityPlanOverviewRetrieved.Error.Exch
<b>Availability Plan Overview Answered</b>	VSP	Elia	MVarAvailabilityPlanOverviewAnswered.Error.Exch

## 8.4.3 Mvar Availability Plan overview: request message

### 8.4.3.1 Message granularity

Mvar Requesting Availability Plan Overview will be limited to **one year** of availability plan per delivery point and can be filtered using input parameters.

- If these are not used, all relevant Market Documents for the requested period will be returned.
- In the retrieval, Elia will send back each Market Document in a separate message.

### 8.4.3.2 Message timeframe

The number of requests will be limited in terms of frequency and period for performance reasons. Those are not set-up yet.

### 8.4.3.3 Message description

For the availability plan overview request: the following MarketDocument should be used (same as for Mvar availability plan details, only with different values for the same fields).

MVAR_Request_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identification of the Request Market Document
type	Y	Type of market document. <b>Z25</b> = Mvar Availability Plan Request
process.processType	Y	Code for type of process: <b>Z30</b> = Mvar Availability plan overview
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code)
sender_MarketParticipant.marketRole.type	Y	The role code associated with the sender: <b>A27</b> = Resource Provider

receiver_MarketParticipant.mRID	Y	The identification number of the Receiver(EIC code). Fixed value: <b>10X1001A1001A094</b> = Elia
receiver_MarketParticipant.marketRole.type	Y	The role code associated with sender. Fixed value: <b>A04</b> = System Operator
createdDateTime	Y	The date and time of the creation of the Request Market Document
request_Period.timeInterval	Y	The beginning and ending date and time of the period covered by the document. This cannot exceed an execution day.
Request_TimeSeries	Y	Request_TimeSeries contained in the message

Request_TimeSeries		
Field	Mandatory	Description
mRID	Y	Unique identification of the Request_TimeSeries within the market document
Requested_RegisteredResource.mRID	N	The delivery point EAN representing the point for which the MVAR availability plan is sent. If not used, this filter is not applied and MVAR availability plans will be returned for all Delivery Points within VSP's portfolio.

timeInterval		
Field	Mandatory	Description
start	Y	The start date and time of the interval
end	Y	The end date and time of the interval

#### 8.4.4 Validation of an Mvar Availability Plan overview requested message

Same as for the [Mvar Availability Plan Details](#)

#### 8.4.5 Retrieved message: Availability Plan overview

##### 8.4.5.1 Message granularity

The granularity is per delivery point and **timeperiod (will be specified later on)**. Per each combinaison of delivery point and timeframe, one message will be sent. So if a request contains multiple delivery points, they will be split into several messages back.

Only the latest version of market documents will be sent.

##### 8.4.5.2 Message timeframe

The message(s) will be sent as soon as possible after a valid request is received.

##### 8.4.5.3 Message description

MVAR_AvailabilityPlan_MarketDocument (Exactly one element per message)		
Field	Mandatory	Value(s)
mRID	Y	Unique identification of the market document (UUID)
type	Y	Type of market document. Fixed value: <b>Z27</b> = Mvar Availability Plan Document
process.processType	Y	Code for type of process: <b>Z30</b> = Availability Plan overview
process.classificationType	Y	Defines whether the mvar availability plan is an aggregation or a classification. Fixed value: <b>A01</b> = detail type
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code). Fixed value: <b>10X1001A1001A094</b> = Elia
sender_MarketParticipant.marketRole.type	Y	The role code associated with sender. Fixed value: <b>A04</b> = System Operator
receiver_MarketParticipant.mRID	Y	The identification of the receiver (EIC code)
receiver_MarketParticipant.marketRole.type	Y	The role code associated with receiver: <b>A27</b> = Resource provider
createdDateTime	Y	The date and time of the Market Document generation by Elia
AvailabilityPlan_Time_Period.timeInterval	Y	The date and time of the day to which the availability plan refers to (execution date)
domain.mRID	Y	<b>10YBE-----2</b> = Belgian bidding zone
TimeSeries	Y	This list only allows 1 element

TimeSeries		
Field	Mandatory	Value(s)
mRID	Y	Sender's identification of the timeseries.
Version	Y	Version of the availability plan
businessType	Y	Identifies the trading nature of the timeseries: <b>Z29</b> =Mvar Availability status
objectAggregation	Y	Identifies how the object is aggregated. Fixed value: <b>Z01</b> = Delivery Point
registeredResource.mRID	Y	The delivery point EAN representing the point for which the availability plan is sent
ExecutionDate	Y	The target date to which the availability plan refers to
Period	Y	This list allows up to 366 elements (1 per day)

Period		
Field	Mandatory	Value(s)
timeInterval	Y	The start and end date and time to which the availability plan timeseries refer to
resolution	Y	Amount of time for each interval in which a data value is defined. Fixed value: <b>PT1D</b> = 1 day
Point	Y	List of points associated to the period.  It should contain as many point as needed to complete the period.

Point		
Field	Mandatory	Value(s)
position	Y	The interval number defining which position in the timeseries is indicated. It must start at 1.
AvailabilityStatus	Y	The availability status of the interval: Unknown = 0 PlannedUnavailable = 1 Testing = 2 ForcedOutage = 3 Available = 4
Qmin	Y	The lower limit of the available band
Qmax	Y	The higher limit of the available band
ServiceControlTypeStatus	Y	The service control type status of the interval: ExitingPowerSavingMode = 0 PowerSavingMode = 1 ManualModeUnavailable = 2 AutomaticModeUnavailable= 3

#### 8.4.5.4 Answered message : Mvar availability plan overview

The answer message is identical to the ones found in [acknowledgement and answer messages](#).

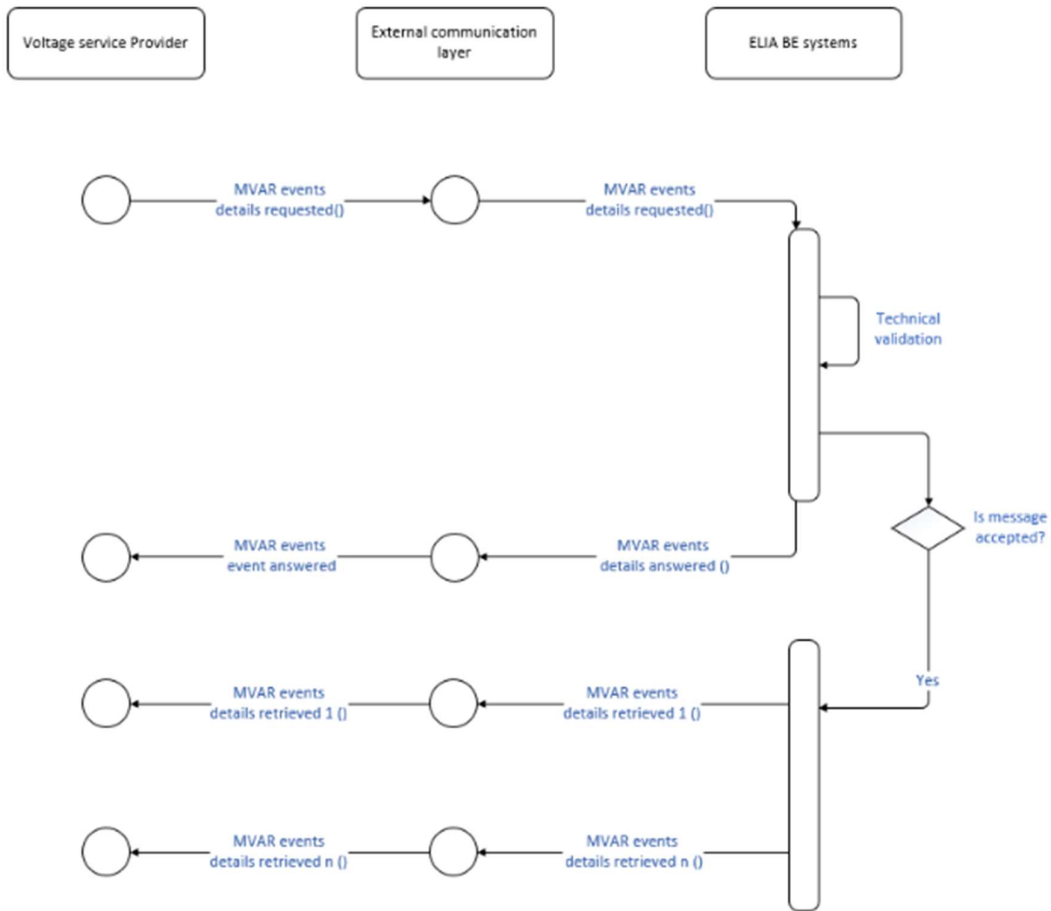
### 8.5 Retrieving mvar events

#### 8.5.1 Description

This message exchange allows Voltage Service Providers to retrieve their submitted mvar events.

As for the others retrieval messages, Elia will only return the latest accepted Market Documents at the time of the request.

The below flow describes the process of requesting the mvar events (it is the same as for the MVAR availability plan).



### 8.5.2 Queue information (section still in development)

The here below table contains the exchanges and queues to send and receive messages in a normal scenario.

- The MVAR events details answered queue will be used to indicate
  - if the request was valid
  - wheter results were found or not

If the request was valid and results were found: the results will be published on the mvar events retrieved queue.

Message Type	Description	Sender	Receiver	Queue/Exchange
<b>Mvar Events Details Requested</b>	Requesting Mvar Events to be retrieved	VSP	Elia	MvarEventsDetailsRequested.In.Exch
<b>Mvar Events Details Retrieved</b>	The result of the request Mvar Events message	Elia	VSP	MvarEventsDetailsRetrieved.[TargetMarketPartyID].OutQ
<b>Mvar Events Details Answered</b>	The answer to the request message	Elia	VSP	MvarEventsDetailsAnswered.[TargetMarketPartyID].OutQ

### Error queues (section still in development)

Queues and exchanges to use in case of error.

Message Type	Sender	Receiver	Queue/Exchange
<b>Mvar Events Details Requested</b>	Elia	VSP	MvarEventsDetailsRequested.[TargetMarketPartyID].ErrorQ
<b>Mvar Events Details Retrieved</b>	VSP	Elia	MvarEventsDetailsRetrieved.Error.Exch
<b>Mvar Events Details Answered</b>	VSP	Elia	MvarEventsDetailsAnswered.Error.Exch

## 8.5.3 Mvar Events details: request message

### 8.5.3.1 Message granularity

Requesting mvar events will be limited to X events per requests. Each Market Document will be sent back by Elia in a separate message.

### 8.5.3.2 Message timeframe

As described in the [validation rules](#), for performance reasons, there will be limits to the number and sizes of request made.

### 8.5.3.3 Message description

A document [Request\\_MarketDocument](#) is used for the Mvar event Request, it is the same as for the other retrieval documents (with different values).

Optional fields of the MarketDocument that are not described in this chapter cannot be used.

MVAR_Request_MarketDocument (exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identification of the Request Market Document
type	Y	Type of market document. <b>Z30</b> = Mvar unavailability events details request <b>Z31</b> = Mvar service control type event details request <b>Z32</b> = Mvar all event types request

process.processType	Y	Code for type of process: <b>Z33</b> = Mvar events information
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code)
sender_MarketParticipant.marketRole.type	Y	The role code associated with the sender: <b>A27</b> = Resource provider
receiver_MarketParticipant.mRID	Y	The identification number of the Receiver(EIC code). Fixed value: <b>10X1001A1001A094</b> = Elia
receiver_MarketParticipant.marketRole.type	Y	The role code associated with sender. Fixed value: <b>A04</b> = System Operator
createdDateTime	Y	The date and time of the creation of the Request Market Document
Request_TimeSeries	Y	Request_TimeSeries contained in the message

Request_TimeSeries		
Field	Mandatory	Description
mRID	Y	Unique identification of the Request_TimeSeries within the market document
requested_MarketDocument.mRID	Y	The mRID of the requested Market Document

#### 8.5.4 Validation of an Mvar event details request message

Same as for the [Mvar Availability Plan Details](#)

#### 8.5.5 Retrieved message: Mvar events details

##### 8.5.5.1 Message granularity

For unavailability events, the granularity is set at the **delivery point** and **mvar event**. Meaning that for each combination of these objects, one single message will be sent.

##### 8.5.5.2 Message timeframe

The message(s) will be returned as soon as possible after a valid request was made.

##### 8.5.5.3 Message description: Mvar unavailability events retrieved

The mvar unavailability market documents can be sent back depending on the request made (see type in Mvar\_Request\_MarketDocument). In case of a request for all market documents, it can be sent alongside service control type events.

The message sent back will be the exact same as the message originally sent by the VSP. It has therefore the same structure and description as the [mvar unavailability messages submitted](#).

##### 8.5.5.4 Message description: service control type events retrieved

The mvar service control type event market documents can be sent back depending on the request made (see type in Mvar\_Request\_MarketDocument). In case of a request for all market documents, it can be sent mvar unavailability events.

The message sent back will be the exact same as the message originally sent by the VSP. It has therefore the same structure and description as the [service control type events submitted](#).

#### 8.5.5.5 Answered message : Retrieving mvar events

The answer message is identical to the ones found in [acknowledgement and answer messages](#).

## 8.6 Receiving a Market Party Notification

### 8.6.1 Description

The exchange message is the same as described in the [Notification messages section](#).

For the Voltage Service Provider, the foreseen use of the notification queues is mainly foreseen for settlement purposes (and the same information could also be sent via email depending on the market party preference).

### 8.6.2 Queue information (section still in development)

This table contains the queues and exchanges to send and receive messages in a normal scenario:

Message Type	Description	Sender	Receiver	Queue/Exchanges
<b>Voltage Service Provider Market Notification Submitted</b>	Submission of a Market Party notification	Elia	VSP	VoltageServiceProviderNotificationSubmitted.[TargetMarketPartyID].OutQ
<b>Voltage Service Provider Market Notification Acknowledged</b>	Reception confirmation of a Market Party notification	VSP	Elia	VoltageServiceProviderNotificationAcknowledged.In.Exch

### Error queues (section still in development)

This table contains the queues and exchanges to send and receive messages in case of error:



Message Type	Sender	Receiver	Queue/Exchange
Voltage Service Provider Market Notification Submitted	VSP	Elia	VoltageServiceProviderNotificationSubmitted.Error.Exch
Voltage Service Provider Market Notification Acknowledged	Elia	VSP	VoltageServiceProviderNotificationAcknowledged.[TargetMarketPartyID].ErrorQ

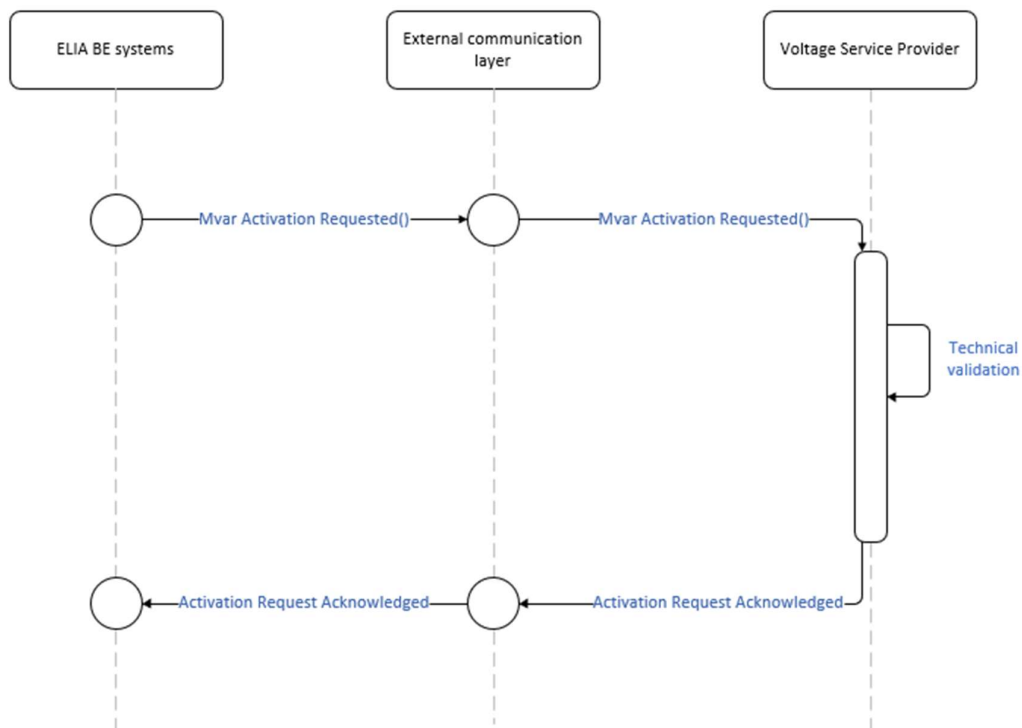
## 8.7 Receiving a mvar activation request

### 8.7.1 Description

Elia will send asynchronous mvar activation requests messages to the VSP.

VSP will generate one acknowledgement:

- *Acknowledgement indicates the good reception of the activation request message*



All communications in this flow are done asynchronously.

### 8.7.2 Queue information (section still in development)

This table contains the queues and exchanges to send and receive messages in a normal scenario:

Message Type	Description	Sender	Receiver	Queue/Exchange
<b>MvarActivation Requested</b>	Activation request	Elia	VSP	MvarActivationRequested.[TargetMarketPartyID].OutQ
<b>MvarActivation Request Acknowledged</b>	Reception confirmation of an activation request message	VSP	Elia	MvarActivationAcknowledged.In.Exch

### Error queues (section still in development)

This table contains the queues and exchanges to send and receive message only in case of error:

Message Type	Sender	Receiver	Queue/Exchange
<b>Mvar Activation Requested</b>	VSP	Elia	MvarActivationRequested.Error.Exch
<b>MvarActivation Request Acknowledged</b>	Elia	VSP	MvarActivationAcknowledged.[TargetMarketPartyID].ErrorQ

### 8.7.3 Activation requested message

#### 8.7.3.1 Message granularity

A mvar activation request message groups all selected DPs from a respective VSP.

#### 8.7.3.2 Message timeframe

The mvar activation request message can be sent at any time. The ramping starts immediately or at least starts at the timing mentioned in the message.

The start and end times communicated in the activation request message will be used as the reference for the setpoint target and remuneration.

#### 8.7.3.3 Message description

An Activation Market Document is the message that is used to request a mvar activation.

Activation_MarketDocument (Exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identifier for the MarketDocument
revisionNumber	Y	Version number for the MarketDocument.
Type	Y	Code for type of MarketDocument. <b>A40</b> = Mvar Activation Document
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code) <b>10X1001A1001A094</b> = Elia

sender_MarketParticipant.marketRole.type	Y	The role code associated with receiver. Fixed value: <b>A04</b> = System Operator
receiver_MarketParticipant.mRID	Y	The identification of the receiver (EIC code)
receiver_MarketParticipant.marketRole.type	Y	The role code associated with the receiver: <b>A27</b> = Ressource Provider
createdDateTime	Y	The timestamp on which the message was sent
rampingStartTime	Y	The start of the ramping of the DP
activation_Time_Period.timeInterval	Y	This information provides the start and end date and time of the activation time interval
TimeSeries	Y	Timeseries associated to the market document.  It must contain at least one element.

TimeSeries		
Field	Mandatory	Description
mRID	Y	Unique identifier for the TimeSeries
businessType	Y	Identifies the reason why an activation is requested. <b>Z21</b> = Normal <b>Z23</b> = Start-Up
process.processType	Y	Code for type of process. <b>Z24</b> = Process with End Time <b>Z22</b> = Process without End time
measurement_Unit.name	Y	<b>MAR</b> (MegaVar)
RegisteredResource	Y	DPs associated to timeseries. It must contain at least one element.

RegisteredResource		
Field	Mandatory	Description
mRID	Y	External Unique identifier of the DP to be activated
timeInterval	Y	The start and end date and time of the mvar activation.
flowDirection.direction	Y	The coded identification of the direction of energy flow. <b>A01</b> = UP <b>A02</b> = DOWN
Point	Y	List of points associated to the period.
resolution	Y	Amount of time for each interval in which a data value is defined. <b>PT1M</b> = 1 minutes

timeInterval		
Field	Mandatory	Description
start	Y	The start date and time of the interval
end	Y	The end date and time of the interval

Point		
Field	Mandatory	Description
position	Y	The interval number defining which position in the timeseries is indicated. It must start at 1.
setPoint	Y	The Mvar Setpoint that is requested. We require an accuracy of 0.1 Mvar.

#### 8.7.4 Activation acknowledged message

##### 8.7.4.1 Message granularity

One mvar activation acknowledgement must be sent for each mvar activation request submitted by Elia.

##### 8.7.4.2 Message timeframe

This acknowledgement must be sent immediately after the reception of the mvar activation request message.

##### 8.7.4.3 Message description

As described in the definition of acknowledgement message (see [Acknowledgement and answer messages](#)).

#### 8.7.5 Back-up activation request message

As mentioned in chapter 0, a back-up communication channel is put in place for the sending of activation request messages.

If Elia does not receive an activation acknowledged message within a reasonable timeframe after the sending of the activation requested message, Elia will automatically trigger the back-up process.

##### Activation requested message

The mvar activation requested message will be sent via e-mail to the contractual 24/7 email address provided by the VSP.

Title of the message: Elia – Activation Request – Mvar – Start Date and Time – End Date and Time

*Example: Elia – Activation Request – Mvar – 05/10/2023 12:45 – 05/10/2023 17:00*

Body of the message:

Hello – ELIA has not received a technical acknowledgement after an activation was requested.

This message serves as a back-up communication for the below mvar activation:

<b>Start time</b>	<b>DP</b>
-------------------	-----------

Requested setpoint	List of DP EAN included in the activation request
--------------------	---

Example:

<b>12:45</b>	<b>DP</b>
25,0	541453137445795650;541453127036684861

Attachement:

The Mvar activation requested message described in Chapter 8.7.3 initially sent to the VSP (JSON) will be included as attachement to the email.

#### **Activation acknowledged message**

Activation acknowledged message will be accepted via e-mail but are not compulsory for the good execution of the process. Such message has to be sent to [activation@elia.be](mailto:activation@elia.be) ([ActivationDEMO@Elia.be](mailto:ActivationDEMO@Elia.be) for the DEMO environment).

The validation rules applied to the Activation acknowledged message will be similarly applied to the Activation acknowledged message sent via email.

Title of the message: VSP Name – Activation acknowledged message – Mvar – Start Date and Time – End Date and Time

*Example:* VSP1- Activation acknowledged message – Mvar – 05/10/2023 12:45 – 05/10/2023 13:00

Body of the message:

Nothing is expected in the body of the message. Any content added in the body of the message will be accepted but disregarded by Elia.

Attachement:

The Activation acknowledged message described in Chapter 8.7.4 that should have been sent via ECL by the VSP(JSON) has to be included as attachement to the email.

## 8.8 Receiving a mvar communication test

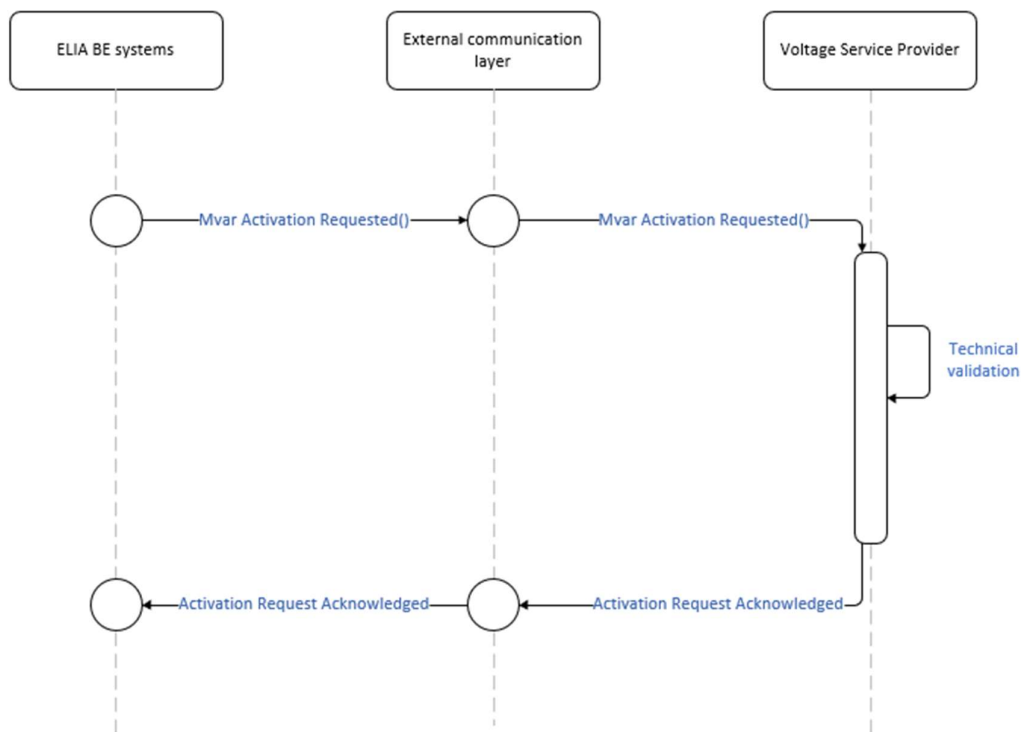
### 8.8.1 Description

The goal of the test is to test the communication between Elia and voltage service provider. Therefore, the communication test will go through the same communication layers as an activation.

A communication test has an activation message with a specific DP EAN: "999999999999999999".

After reception of the activation message in the communication test, the voltage service provider generates one acknowledgement :

- *Acknowledgement indicates the good reception of the communication test request message*



All communications in this flow are done asynchronously.

### 8.8.2 Queue information (section still in development)

This table contains the queues and exchanges to send and receive messages in a normal scenario:

Message Type	Description	Sender	Receiver	Queue/Exchange
<b>MvarActivation Requested</b>	Activation request	Elia	VSP	MvarActivationRequested.[TargetMarketPartyID].OutQ

<b>MvarActivation Request Acknowledged</b>	Reception confirmation of an activation request message	VSP	Elia	MvarActivationAcknowledged.In.Exch
--	---	-----	------	------------------------------------

### Error queues (section still in development)

This table contains the queues and exchanges to send and receive message only in case of error:

Message Type	Sender	Receiver	Queue/Exchange
<b>Mvar Activation Requested</b>	VSP	Elia	MvarActivationRequested.Error.Exch
<b>MvarActivation Request Acknowledged</b>	Elia	VSP	MvarActivationAcknowledged.[TargetMarketPartyID].ErrorQ

## 8.8.3 Activation requested message

### 8.8.3.1 Message granularity

Similarly to a mvar activation request, a mvar communication test message groups all selected DPs from a respective VSP.

### 8.8.3.2 Message timeframe

The mvar communication test message can be sent at any time. The ramping starts immediately or at least starts at the timing mentioned in the message.

The start and end times communicated in the communication test message will be used as the reference for the setpoint target and remuneration.

### 8.8.3.3 Message description

An Activation Market Document is the message that is used to request a mvar communication test.

<b>Activation_MarketDocument</b> (Exactly one element per message)		
Field	Mandatory	Description
mRID	Y	Unique identifier for the MarketDocument
revisionNumber	Y	Version number for the MarketDocument.
Type	Y	Code for type of MarketDocument. <b>A40</b> = Mvar Activation Document
sender_MarketParticipant.mRID	Y	The identification of the sender (EIC code) <b>10X1001A1001A094</b> = Elia
sender_MarketParticipant.marketRole.type	Y	The role code associated with receiver. Fixed value: <b>A04</b> = System Operator
receiver_MarketParticipant.mRID	Y	The identification of the receiver (EIC code)
receiver_MarketParticipant.marketRole.type	Y	The role code associated with the receiver: <b>A27</b> = Ressource Provider
createdDateTime	Y	The timestamp on which the message was sent

TimeSeries		
Field	Mandatory	Description
mRID	Y	Unique identifier for the TimeSeries
businessType	Y	Identifies the reason why an activation is requested. <b>Z21</b> = Normal <b>Z23</b> = Start-Up
process.processType	Y	Code for type of process. <b>Z24</b> = Process with End Time <b>Z22</b> = Process without End time
measurement_Unit.name	Y	<b>MAR</b> (MegaVar)
RegisteredResource	Y	DPs associated to timeseries. It must contain at least one element.

timeInterval		
Field	Mandatory	Description
start	Y	The start date and time of the interval
end	Y	The end date and time of the interval

Page 64 of 80



position	Y	The interval number defining which position in the timeseries is indicated. It must start at 1.
setPoint	Y	The Mvar Setpoint that is requested. We require an accuracy of 0.1 Mvar.

#### **8.8.4 Activation acknowledged message**

##### **8.8.4.1 Message granularity**

One mvar communication test acknowledgement must be sent for each mvar communication test submitted by Elia.

##### **8.8.4.2 Message timeframe**

This acknowledgement must be sent immediately after the reception of the mvar communication test message.

##### **8.8.4.3 Message description**

As described in the definition of acknowledgement message (see [Acknowledgement and answer messages](#)).

## 9 Validation rules description

This chapter includes the generic validation rules that will be applied to all messages as well as the descriptions of the validation rules that are referenced in the validation rule sections of the respective guides.

### 9.1 Generic

All MarketDocuments submitted to Elia are subject to a technical validation of the message.

In case one of the rules for technical validation fails, the message will be rejected and the answer message will indicate the code of the error and a descriptive text inside a Reason object.

The values for the Reason code are fixed and described in this guide. However, the descriptive text can be changed at any moment by Elia.

Example:

```
"Reason": [
  {
    "code": "A51",
    "text": "A higher version of the message already exists"
  }
]
```

This chapter describes the technical validation rules that apply to all MarketDocuments described in the subsequent sections, and their corresponding codes.

The Reason object can either be associated to the MarketDocument or to a specific timeseries.

Note: Level indicates if the rule applies to the whole MarketDocument or to the timeseries individually.

ID	Validation Rule	Reply Status	Reason Code	Level
GEN_001	Message format must be correct	Reject message	Not applicable. Message will be transferred to error queue	Not applicable
GEN_002	Mandatory fields must be present	Reject message	A69	MarketDocument
GEN_003	Data formats must be respected	Reject message	Y29	MarketDocument
GEN_004	Value of fields must be known	Reject message	Y28	MarketDocument
GEN_005	Time interval start date and time must be smaller than the end date and time	Reject message	Y97	MarketDocument Timeseries
GEN_006	The timeseries mRID must be unique within the MarketDocument	Reject message	A55	Timeseries
GEN_007	Timeseries period must fall within the MarketDocument period	Reject message	A81	Timeseries
GEN_008	No overlap of periods allowed for the same timeseries within the message	Reject message	Y96	Timeseries

ID	Validation Rule	Reply Status	Reason Code	Level
GEN_009	The revisionNumber of the incoming message must be greater than an existing revisionNumber for the same MarketDocument mRID	Reject message	A51	MarketDocument
GEN_010	The number of points must match with the time interval of the period	Reject message	A49	Timeseries
GEN_011	The position of a point within a period must be correct	Reject message	Y95	Timeseries
GEN_012	Business key must be known by Elia - <i>Delivery point</i> - <i>Sender company</i>	Reject message	A05	MarketDocument
GEN_013	sender_MarketParticipant.mRID must contain the EIC code of the Market Party linked to the user-id of the message	Reject message	A78	MarketDocument
GEN_014	A MarketDocument with revisionNumber n must contain all timeseries contained in revision n-1	Reject message	A52	MarketDocument
GEN_015	MarketDocument mRID must be unique	Reject message	Y94	MarketDocument
GEN_016	Field cannot be used	Reject message	Y93	MarketDocument
GEN_017	MarketDocuments provided on message queues must be processed within a specific time window	Reject message	Y06	MarketDocument
GEN_018	Market Document waiting for confirmation becomes obsolete due to valid intermediary update	Reject message	X97	MarketDocument

#### GEN\_001 – Message format must be correct

The file format of the message provided on the message queue must be correct. If the message does not have the correct file format, the message will be rejected and transferred to the corresponding error queue.

#### GEN\_002 – Mandatory fields must be present

The message must contain a value for all mandatory fields. The mandatory fields per message can be found in the message specifications. If a value is missing for a mandatory field in the message, the message will be rejected.

#### GEN\_003 – Data formats must be respected

The data format of the fields in the message must be respected. The data format per field can be found in the message specifications. If a message contains a value of an incorrect data format for a field, the message will be rejected.

#### GEN\_004 – Value of fields must be known

The value of the fields in the message must be known by the receiver. The allowed values per field can be found in the message specifications. If a message contains a value which is not known to the receiver, the message will be rejected.

Example:

For a field that has only two possible values: "A60" or "A61".

This rule will check that the value received for the field is either "A60" or "A61".

#### **GEN\_005 – Time interval start date and time must be smaller than the end date and time**

In the message a time interval can be specified:

- On the MarketDocument level
- For every period within the message

For both of these time intervals the start date and time must be smaller than the end date and time. If the start date and time is greater than or equal to the end date and time, the message will be rejected.

#### **GEN\_006 – The timeseries mRID must be unique within the MarketDocument**

For every timeseries in a message a unique mRID must be provided. This mRID must be unique within the message. If a message contains two or more timeseries with the same mRID, the message will be rejected.

#### **GEN\_007 – Timeseries period interval must fall within the MarketDocument period interval**

In the message a time interval can be specified:

- On the MarketDocument level
- For every period within the message

The time interval specified on a timeseries period must fall within the time interval specified on the MarketDocument. This means that for every period:

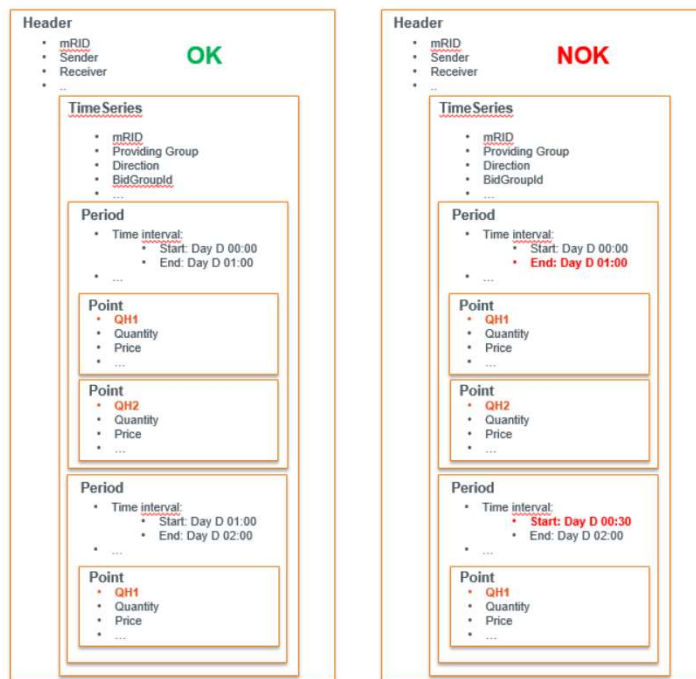
- the start date and time of the period time interval must be greater than or equal to the start date and time of the MarketDocument time interval  
AND that
- The end date and time of the period time interval must be smaller than or equal to the end date and time of the MarketDocument time interval

If the time interval of a period does not fall within the time interval of the MarketDocument, the message will be rejected.

#### **GEN\_008 – No overlap of periods allowed for the same timeseries within the message**

A timeseries within the message contains one or multiple periods. For every period a time interval is specified. The time intervals of periods within the same timeseries cannot overlap. If there are periods with overlapping time intervals within the same timeseries, the message is rejected.

Example:



#### GEN\_009 – The revisionNumber of the incoming message must be greater than an existing revisionNumber for the same MarketDocument mRID

For messages updates can be submitted. When submitting an update of a message the same MarketDocument mRID must be used and the revisionNumber must be increased. The message with the greatest revisionNumber is assumed to be latest and most up to date version of the message.

If a message is received where a greater revisionNumber for the same MarketDocument mRID already exists, the message will be rejected.

#### GEN\_010 – The number of points must match with the time interval of the period

A time interval is specified for every period in the message. The period must contain a point for every quarter hour in the period, if not the message will be rejected.

Example:

- A period has a time interval of 20:15 to 21:30 on day D
- This means that the duration of this time interval is 75 minutes or 5 quarters
  - o This means that there must be 5 positions in the period

If message contains too few or too many points for the concerned period, the message will be rejected

A full day must account for 96 quarter hours. There is an exception for the days when the change to summer or winter time takes place.

- The last Sunday of October must account for 100 quarter hours for a full day

⋮ *The last Sunday of March must account for 92 quarter hours for a full day*

Example 2: summer time to winter time in Belgium in 2020.

Since all dates must be in UTC, a period from midnight until 3am (local time) for the day 2020-10-25, corresponds to the following period in UTC: 2020-10-24T22:00Z until 2020-10-25T02:00Z.

This means that 16 positions are expected for that period.

Example 3: winter time to summer time in Belgium in 2020.

Since all dates must be in UTC, a period from midnight until 4 am (local time) for the day 2020-03-29, corresponds to the following period in UTC: 2020-03-28T23:00Z until 2020-03-29T02:00Z.

This means that 12 positions are expected for that period.

#### **GEN\_011 – The position of a point within a period must be correct**

A position indicates the position of a period related to the time interval of the period. Every position is numbered starting at one for the first point and is incremented by one for every subsequent point within the period.

For example:

- A period has time interval of 20:15 to 21:30 on date x
- As indicated above, 5 positions are expected for this period
- This means that position values between 1 and 5 are allowed.

Position	Description	Time interval
1	First quarter	20:15 – 20:30
2	Second quarter	20:30 – 20:45
3	Third quarter	20:45 – 21:00
4	Fourth quarter	21:00 – 21:15
5	Fifth quarter	21:15 – 21:30

If the message contains incorrect position values relative to the period time interval, the message will be rejected.

#### **GEN\_012 – Business key must be known by Elia**

The value of eventual business keys in the message must be known by Elia. The message specifications indicate for which fields in the message business keys must be used. If a message contains a business key which is not known by Elia, the message will be rejected.

The following business keys are identified:

- Delivery point
- Sender market participant

**GEN\_013 – sender\_MarketParticipant.mRID must contain the EIC code of the Market Party linked to the user-id of the message**

Every Market Party will have a dedicated user to connect to the External Communication Layer. Every message sent by a Market Party will include a user-id in the header of the message.

Elia will control that the user-id corresponds to the Market Party EIC code specified in the MarketDocument, in field sender\_MarketParticipant.mRID.

**GEN\_014 – A MarketDocument with revisionNumber n must contain all timeseries contained in revision n-1**

A MarketDocument with revisionNumber “n” must contain all timeseries contained in the previous accepted revisionNumber, unless the entire period of a timeseries block falls in the past.

Example 1:

MarketDocument A with revisionNumber = 1 contains time series A, B and C.

MarketDocument A with revisionNumber = 2 must contain A, B, C and any other additional timeseries.

Example 2:

MarketDocument A with revisionNumber = 1 is created at 10:47 and contains a timeseries B, with a Period that ends at 12:15.

MarketDocument A with revisionNumber = 2 is created at 11:32, it must contain timeseries B.

MarketDocument A with revisionNumber = 3 is created at 12:23, it does not require the resent of timeSeries B, because the period of the timeseries has ended.

**GEN\_015 – MarketDocument mRID must be unique**

MarketDocument mRID must be unique.

The same mRID can only be used (and must be used) when submitting a new revisionNumber of the same MarketDocument.

**GEN\_016 – Field cannot be used**

Fields of the MarketDocument that are not applicable for a specific message cannot be used or should include a null value.

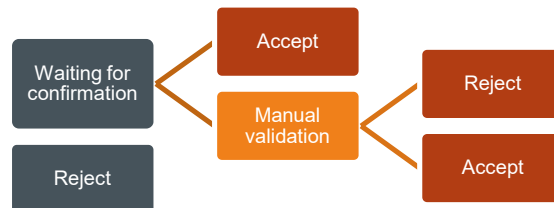
**GEN\_017 – MarketDocuments provided on message queues must be processed within a specific time window**

When MarketDocuments are provided to Elia, the time of posting the MarketDocument on the message queue will be compared with the time of processing the message in the back-end application. If the processing of the message cannot be executed within a specific time window, the MarketDocument will be rejected.

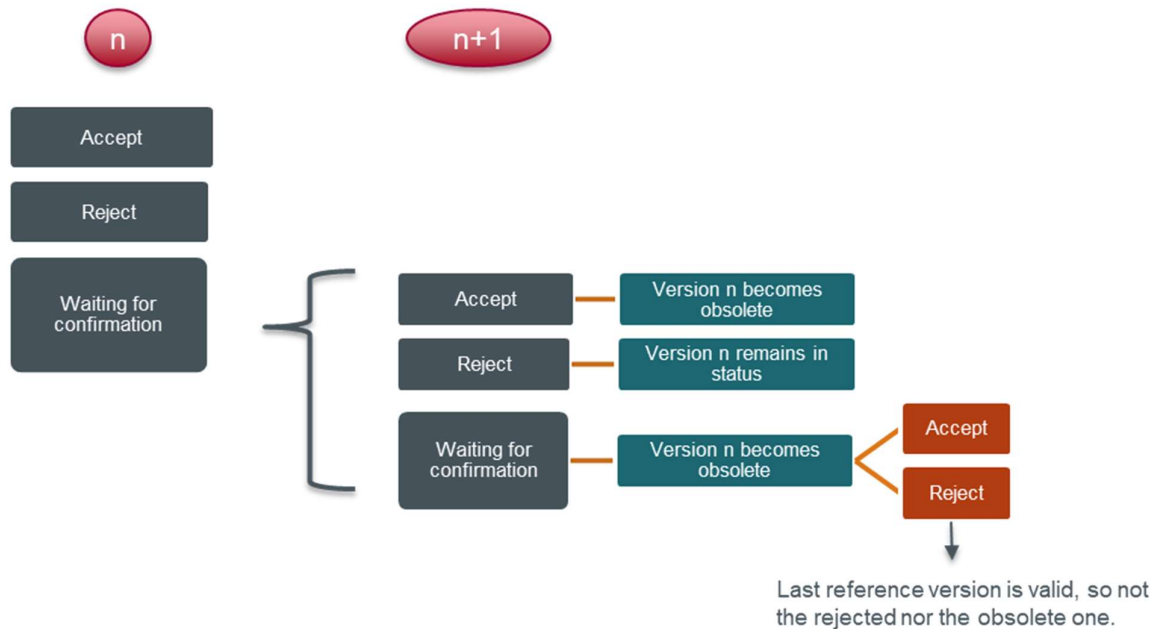
### GEN\_018 – Market Document waiting for confirmation becomes obsolete due to valid intermediary update

Some validation rules in the different guides require an extended confirmation period between reception and validation or rejection of a Market Document for specific reasons (ex. Need for manual validation by Elia).

Following states in confirmation messages can be expected for these cases:



When a message is technically or functionally rejected, this will immediately be confirmed to the Market Party. When it is functionally correct, it will come into the waiting for confirmation state. Subsequently it gets either accepted by a certain trigger (ex. End of the standstill period) or needs to undergo a manual validation process which can still end up in a Reject or an Accept as a result.



When an update (n+1) of that Market Document is sent while the latest known valid version (n) was still in a “Waiting for confirmation” status, the following scenarios are possible

- Update n+1 is accepted, meaning that version n becomes obsolete. A confirmation message will be sent for
  - Version n : a reject with the code of the here described rule
  - Version n+1 : an accept



- Update n+1 is rejected, meaning that version n remains the last valid reference version as if no update was sent. A confirmation message will be sent for
  - Version n+1 : a reject with the code of the rule triggering the reject
- Update n+1 is in waiting for confirmation, meaning that version n becomes obsolete. A confirmation message will be sent for
  - Version n : a reject with the code of the here described rule
  - Version n+1 : a waiting for confirmation with the code of the rule triggering a waiting for confirmation reason

Note that in the last scenario, if a reject of version n+1 would subsequently take place, this would mean that version n and version n+1 are both rejected and the last accepted version becomes the last reference version of the Market Document.

#### Internal error

In exceptional cases, Elia could reject a message due to an internal error.

Reply Status	Reason Code	Level
Reject message	999	MarketDocument

## 9.2 Reactive Power Outage Planning

### 9.2.1 Introduction

**Disclaimer:** This current documentation in this next section is a draft. It is intended to give an idea only to VSPs about the possible rules.

**What can be found in this section?** The acceptance rules when a VSP submits unavailability or control service type events are detailed in this section

### 9.2.2 Error codes

Each validation rule will send back a specific reason code (in addition to the error text).

The error code can be found in each error section.

When EIC codes were not existing for the specific error, custom codes were created. Those codes start with the letter “Y” and will be discussed with ENTSO-E in the future. Nevertheless, they are not expected to change in the short term.

### 9.2.3 Definitions

#### 9.2.3.1 Short & common definitions

**Qmin/max:** the min/max reactive power a delivery point can deliver (in mvar) as defined in the VSP terms & conditions.

**Qmin/max submitted:** The Qmin and Qmax submitted in an unavailability event

**Band:** The band included between a Qmin\_submitted and a Qmax\_submitted

**Reference setpoint:** The default Q for a given delivery point (most cases reference setpoint=0).

#### **9.2.3.2 Unavailability event**

An event that can be submitted by market parties. It can be one of 3 types: Planned unavailability, Testing or Forced Outage. It is accompanied by a band.

Those events always lead to a reduction of the available band.

The event is always linked to 1 or more delivery points.

#### **9.2.3.3 Planned unavailability**

Planned unavailability events are at the discretion of the market party submitting it: rules and rejections are therefore slightly more strict.

#### **9.2.3.4 Testing unavailability**

Testing unavailability are events occurring after a maintenance or before the final commission or for any other reason leading to a potential impact on the grid operation.

#### **9.2.3.5 Forced outage**

Forced outages are events happening without the knowledge or consent from a market party. Rules are therefore less strict.

#### **9.2.3.6 Control service type event**

Control service type event are a different type of event than the unavailability events. They are therefore submitted to different rules. They are not accompanied by a band.

#### **9.2.3.7 Availability plan overview/details request message**

An availability plan request message is a message sent by Voltage Service Providers destined to get the availability plan overview/details generated by Elia for their assets.

#### **9.2.3.8 Mvar event requests**

Mvar event requests are messages sent by market parties in order to get returned a specific event sent by them earlier.

### **9.2.4 Conditions for all Mvar events**

#### **9.2.4.1 A VSP can not submit events for delivery points belonging to another VSP**

A VSP can only submit events for delivery points on which he has VSP contractual rights/obligations (as defined in its VSP contract).

Error code: Y200

#### **9.2.4.2 Versioning & ids**

Each event must have a unique ID & version number (per VSP).

Error code: Y201

#### **9.2.4.3 Each unavailability event must have a valid reason**

In TimeSeries > reason, only three values are allowed: "Technical", "Human", "Other". Any other value will lead to the rejection of the message (capitalization does not matter).

Error code: Y202

#### **9.2.4.4 Each unavailability event must be accompanied by reason details**

In TimeSeries > reason\_details, the values must at least be 10 characters long and contain 1 blank space.

Error code: Y203

#### **9.2.4.5 Technical start and end date limitations**

The start\_DateAndOrTime.date and end\_DateAndOrTime.date are limited in order to avoid unlogical or overloading submissions.

The start\_DateAndOrTime.date can not be more than 10 years in the future from the submission date.

The end\_DateAndOrTime.date (if specified) can not be more than 10 years in the future from the submission date.

Error code: Y211

### **9.2.5 Conditions for all unavailability events**

#### **9.2.5.1 All unavailability events must be submitted alongside a valid band**

An unavailability event will be rejected if it is not accompanied by a valid band.

- The band must have a Qmin submitted and a Qmax submitted, with Qmin lower than or equal to Qmax
- Qmin and Qmax are numbers (float)
- The minimum and maximum values can be equal to each other.

Error code: Y204

#### **9.2.5.2 Qmin submitted can not be lower than the contractual Qmin**

Unavailabilities event having Qminsubmitted < Qmin will be rejected.

Error code: Y205

#### **9.2.5.3 Qmax submitted can not be higher than the contractual Qmax**

Unavailabilities events having Qmaxsubmitted > Qmax will be rejected.

Error code: Y206

#### **9.2.5.4 Submitted band must include the contractual reference setpoint**

The unavailability event will be rejected if the submitted band does not include the contractual reference setpoint. Or in other words it will be rejected if

- $Q_{min\_submitted} > \text{Reference Setpoint}$
- Or  $Q_{max\_submitted} < \text{Reference Setpoint}$

Error code: Y207

#### **9.2.5.5 Only three types of unavailability events are accepted**

Only the three following values will be accepted as businessType in the Timeseries of an unavailability event.

**A53** = Planned Maintenance (Planned Unavailability)

**A54** = Unplanned Outage (Forced Outage)

**B83** = Testing

Error code: A62

#### **9.2.5.6 Unavailability events can not overlap with others unavailability events**

The start and end date of any unavailability event can not be in between the start and end date of another unavailability event.

They can however overlap with control service type events.

Error code: Y38

#### **9.2.5.7 Non-forced outages unavailability events should not lead to a non-manageable reduction in reactive power control.**

In such cases, Elia will contact the market party to find an arrangement.

#### **9.2.5.8 Updates made after the normal timeframe of their events must be “less restrictive”**

Updates of already submitted events can happen after the normal deadline of their events only if they are leading to a less restrictive outcome than initially foreseen.

For example: A planned unavailability event can be updated less than 1 hour before its start date only if it leads to a broader band than in the last update.

Error code: None is this rule is an exception to the normal timeframe rules.

#### 9.2.5.9 There can be maximum 120 time intervals per resolution submitted

For performance reasons, a maximum of 120 time intervals per resolution are allowed.

Resolution	-	Period Maximum
Minute	PT1M	2 hours
15 minutes	PT15M	30 hours
Hourly	PT1H	5 Days
Daily	PT1D	120 Days
Monthly	PT1Mo	5 years (cfr validation rules)

*Valid examples that are not rejected*

Event ID	Event duration	Message timeintervals per periods in the message				
		PT1MO	PT1D	PT1H	PT15M	PT1M
UE01	1 year, 20 days, 6 hours, 34 minutes	12	20	6	2	4
UE02	3 months, 2 hours	3		2		
			90			120

Error code: Y209

#### 9.2.5.10 The unit of measure should be MAR

In TimeSeries > quantity\_Measure\_Unit.name : value should always be "MAR" (=MVAR in CIM model)

Error code: Y210

#### 9.2.6 Conditions for Planned unavailability events

##### 9.2.6.1 Planned unavailabilities start date can not be in less than 1 hour from reception time

The planned unavailability will be rejected if the beginning of the first unavailability period is in less than 1 hour from its reception time.

Error code: Y212

#### 9.2.7 Conditions for Forced Outage unavailability event

##### 9.2.7.1 Forced outage start date must be maximum 24 hours in the past or 24 hours in the future

The start date of a forced outage can not be more than 24 hours in the future nor more than 96 quarter-hour (24 hours) in the past. Note that in the terms and conditions, 1 hour for past

submission is mentioned. In case of misuse it is possible that the delay would be reduced to the contractual value.

Error code: Y213

### **9.2.8 Conditions for Testing unavailability event**

#### **9.2.8.1 Testing unavailabilities must be submitted more than 1 month in advance**

The testing unavailability must be submitted more than 1 month (=30 days=30\*24hours) before the start of the 'testing' status.

Error code: Y214

### **9.2.9 Conditions for service control type events**

#### **9.2.9.1 Only 4 event types can be submitted**

Namely: "Manual mode unavailable", "Automatic control unavailable", "Power saving mode", "Exiting power saving mode" and "Return to contractual mode", other messages will be rejected.

Error code: Y215

#### **9.2.9.2 Automatic mode unavailable can only be sent by units having an automatic mode available**

Units having only a manual mode can not send a message saying that their automatic mode is unavailable as they are not having one.

Error code: Y216

#### **9.2.9.3 Entering power saving mode can only be sent for units which are defined as having one contractually.**

Units not defined as having a power saving mode can not enter power saving mode. Hence, the message will be rejected.

Error code: Y217

#### **9.2.9.4 Exiting power saving can only be sent by unit having sent a power saving mode with no end date.**

Exiting power saving mode is a message sent to signal the power saving mode with no end date has finished. In case such underlying event does not exist, the message makes no sense and will therefore be rejected. In case of a power saving mode with an end date, an update of the end date is expected.

Error code: Y218

**9.2.9.5 Control service type events can not be more than 24 hours after it happened.**

Controls service type events will be rejected if they are submitted with a start date more than 24 hours in the past.

Error code: Y219

**9.2.9.6 Power saving mode events can not be submitted for the past**

Power saving mode events will be rejected if they are submitted with a start date in the past.

**Note:** A reasonable technical delay of 1 minute will be accepted.

Error code: Y220

**9.2.10 Conditions for Availability plan details requests****9.2.10.1 The number of requests per 15 min cannot exceed the threshold limit**

There is a maximum of X requests per 15 min for messages requesting availability plan details.

**9.2.10.2 The number of days requested can not exceed the threshold limit**

One message can not request more than X days of availability plans.

**9.2.11 Conditions for Availability plan overview requests****9.2.11.1 The number of requests per 15 min cannot exceed the threshold limit**

There is a maximum of X requests per 15 min for messages requesting availability plan overview.

**9.2.11.2 The number of years requested can not exceed the threshold limit**

One message can not request more than X years of availability plan overview.

**9.2.12 Conditions for Mvar events requests****9.2.12.1 The number of requests per 15 min cannot exceed the threshold limit**

There is a maximum of X requests per 15 min for messages requesting mvar events.

**9.2.12.2 The number of events requested can not exceed the threshold limit**

One message can not request more than X Mvar event.

