EUDAMED user guide

Machine-to-machine data exchange

Production v 2.14.1
2024

Table of Contents

1 Glossary

Term	Description
Access Point	The Access Point (AP) of CEF eDelivery implements the AS4 message exchange protocol according to the e-SENS profile. This ensures standardised, interoperable, secure and reliable data exchange. For more information, please refer to the CEF Digital Portal.
AS4	The AS4 profile of eDelivery is the AS4 Usage Profile/ implementation guidelines defined by e-SENS based on the AS4 specification of OASIS, itself a profile of OASIS ebXML Messaging Services Version 3.0, which in turn is based on various Web Services specifications of OASIS.
Bulk Upload (BLK)	The GUI-to-machine upload feature of the restricted website in EUDAMED
Criteria	A pull request will contain search conditions; called criteria
DTX	Data Exchange, a term used in EUDAMED that covers the Bulk Upload (GUI) and Machine-to-Machine (M2M) services
Domibus	eDelivery AS4 solution by DIGIT. Implementation of eDelivery (Specification)
eDelivery	AS4 Standard
Entity Model	The Entity Data Model (EDM) is an extended version of the Entity-Relationship model which specifies the conceptual model of the data using various modelling technique. It also refers to a set of concepts that describe data structure, regardless of its stored form
EU SEND	Allows for the use of a persistent encryption mechanism that can be applied to some parts of the message and to payloads within a message as well. Persistent encryption can be leveraged as an additional layer of security for Internet based messaging. A SaaS deployment of Domibus.
Message Type	Pull, Push
M2M	Machine-to-Machine
Operation	PUT, GET, POST, PATCH
Page	Responses are split into pages based on the PageSize provided with the request
Pagination	The EUDAMED services offer the possibility to split responses into different pages to manage the message size
PageNumber	In case of paginated response, the requester can orchestrate multiple page response and ask for a specific page on page to be provided in the response. The first page is page number ‘0’.
PageSize	Required maximum number of entities on a specific response page
PartyID	Party Identifier part of the sender/node/nodeID, identifies the AP that sent the payload
Payload	It may contain the main service entity in case of a query by example
Recipient	EUDAMED service
Service	Operations made available on the M2M environment
SOAP	Simple Object Access Protocol
XML	Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable
XSD	An XML Schema Definition describes the structure of an XML document

2 Introduction

2.1 Purpose

The current user guide aims at providing guidelines to the IT/Software development team of the public/private organisations, in order to successfully perform M2M data exchange. Moreover, this document assumes that the readers are familiar with the EUDAMED MDR general purpose and CEF eDelivery building block (EUSEND) of the Connecting Europe Facility (CEF).[1]

2.2 Modes of data input

A user can register data in EUDAMED by using:

- The EUDAMED user interface. For any further assistance visit the EUDAMED Information Centre
- The XML bulk upload/download. For more information visit the Bulk uploading-downloading page of the EUDAMED Information Centre
- The M2M data exchange.

NOTE

Not sure which way to use for registering data? Here is a tip:

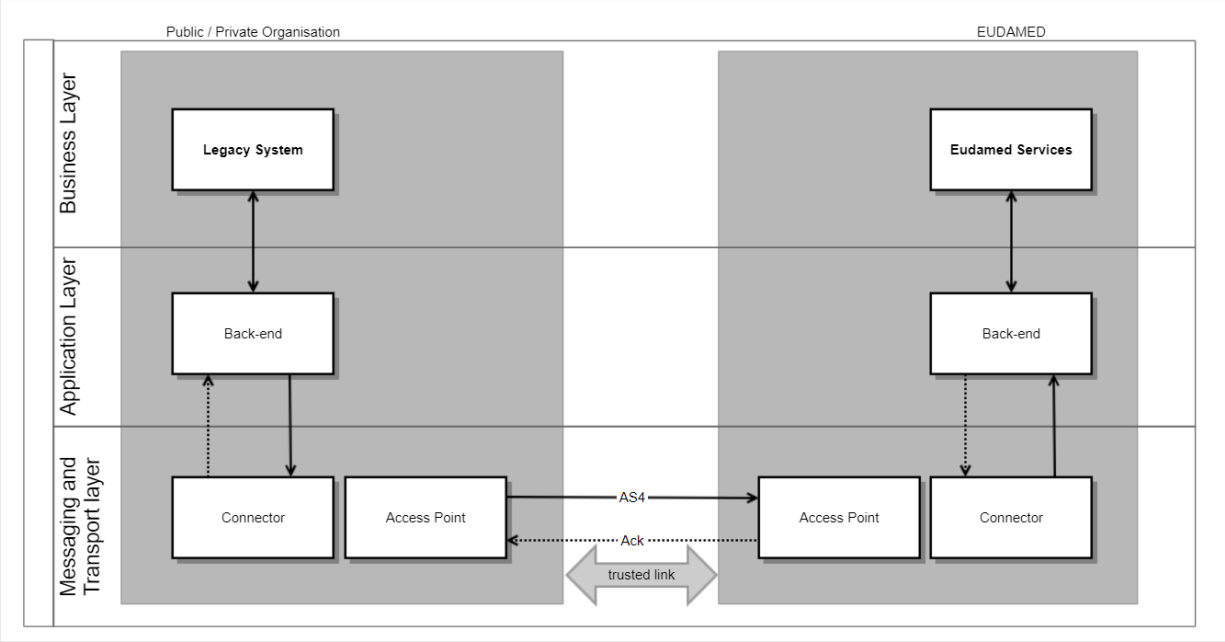
- If you must register up to 100 devices, then use the EUDAMED user interface
- If you must register 100-1000 devices, then use the bulk upload/download
- If you must register more than 1000 devices, then use the M2M data exchange.

[1]For a wider understanding on how to use the platform, including FAQs and process infographics, visit the EUDAMED Information Centre.

3 M2M data exchange architecture

The M2M data exchange allows the automatic data exchange between a public/private organisation (external system) and EUDAMED. To achieve this, the external system must convert its data into the XML format that EUDAMED supports. Moreover, the external system must install and configure a dedicated CEF eDelivery Access Point and a secure AS4-compliant communication protocol on its premises in order to establish a secure link with EUDAMED.

The EUDAMED M2M data exchange defines a set of building blocks that should be specified and implemented to enable the information exchange between public/private organisations and EUDAMED.



Organisation Backend: Represents a dedicated information exchange gateway. It implements a specific communication protocol, service and entity data exchanged between the organisations and EUDAMED, compliant with the EUDAMED service and data model.

Access Point: The organisation's AP will deliver the messages to the EUDAMED AP. The EUDAMED AP will acknowledge or respond to the organisation's requests. Both must be AS4-compliant. Domibus is used on the EUDAMED side.

EUDAMED Backend: Is responsible for the data exchange message requests, including the security, access control and reliability aspects, and for constructing the messageresponses.

4 Prerequisites

Before performing M2M data exchange, the following conditions must be met:

1. The public/private organisation must be registered as an Actor in EUDAMED (see EUDAMED Information Centre) and have an active user with Local Actor Administrator (LAA) profile.

2. A user with LAA profile acting on behalf of the Actor must know or obtain the Party ID that identifies the Access Point that will be used for data exchange with EUDAMED.

3. A user with LAA profile acting on behalf of the Actor must obtain a security key (token) for every module that the user intends to perform M2M data exchange (see Section Generate your security key [31] of the current guide).

4. A user with LAA profile acting on behalf of the Actor must have successfully completed the onboarding procedure in the Playground environment, because before having an AP in the Production environment all participants must apply for an AP in Playground in order to test the service (see Section M2M Onboarding of the Playground user guide).

5 M2M Onboarding

5.1 How to configure your M2M settings in EUDAMED

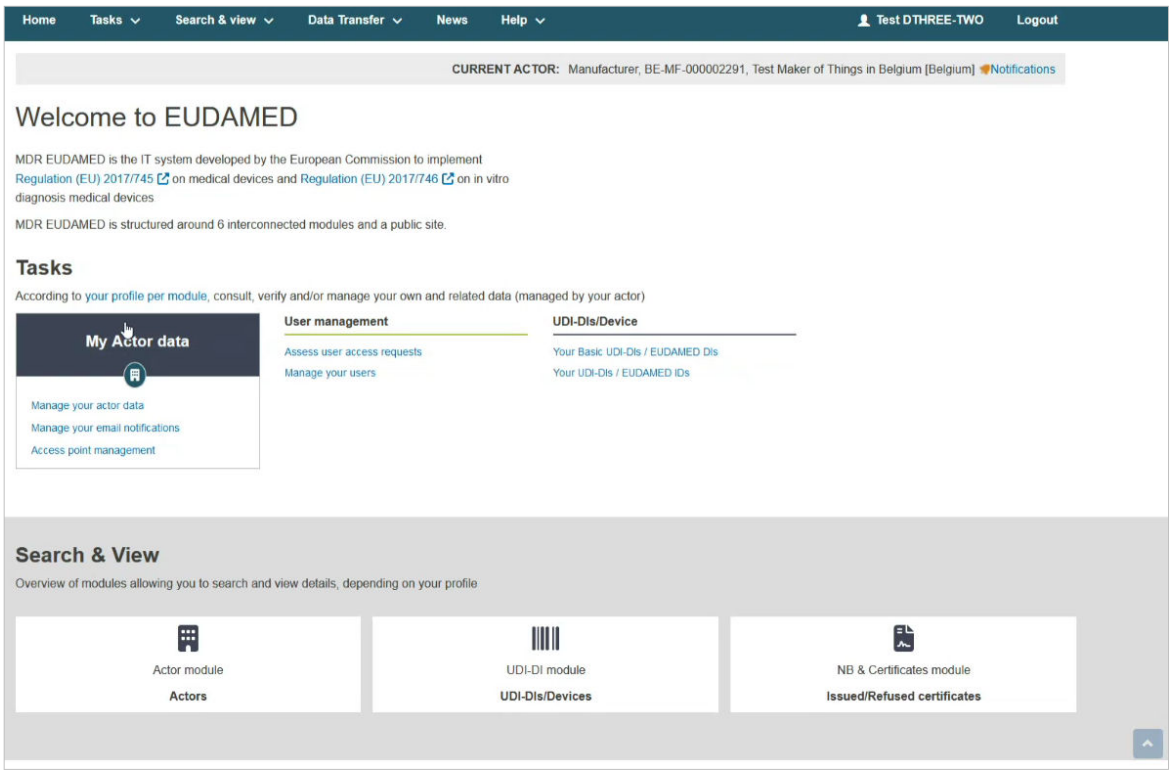
NOTE
If you wish to activate your M2M services in EUDAMED Production environment you must first apply for an Access Point (AP) in EUDAMED Playground environment, complete the onboarding and test the service.

5.1.1 Request the use of a new AP

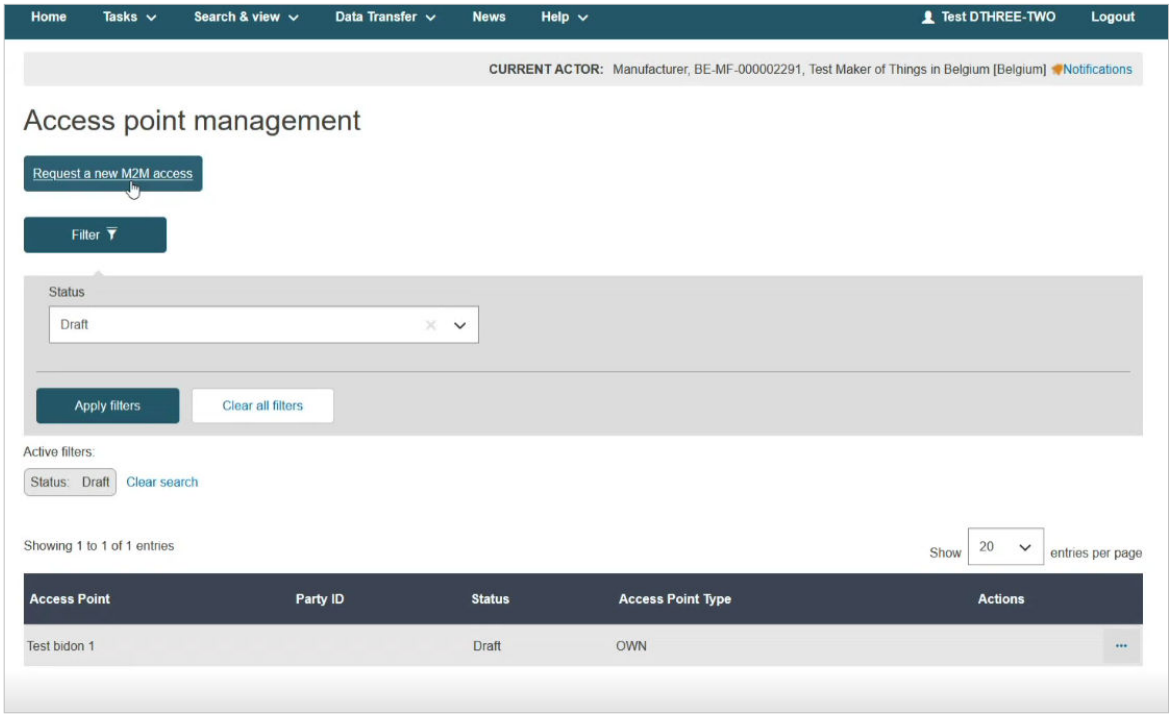
VIDEO: eDelivery access points

NOTE
Economic Operators can have **up to two** APs in *active/submitted/connectivity under validation* statuses while Competent Authorities, Designating Authorities and Notified Bodies can have **only one** AP in *active/submitted/connectivity under validation* statuses. See Annex 3 (AP statuses) [60] of the current guide for further information about the AP statuses.

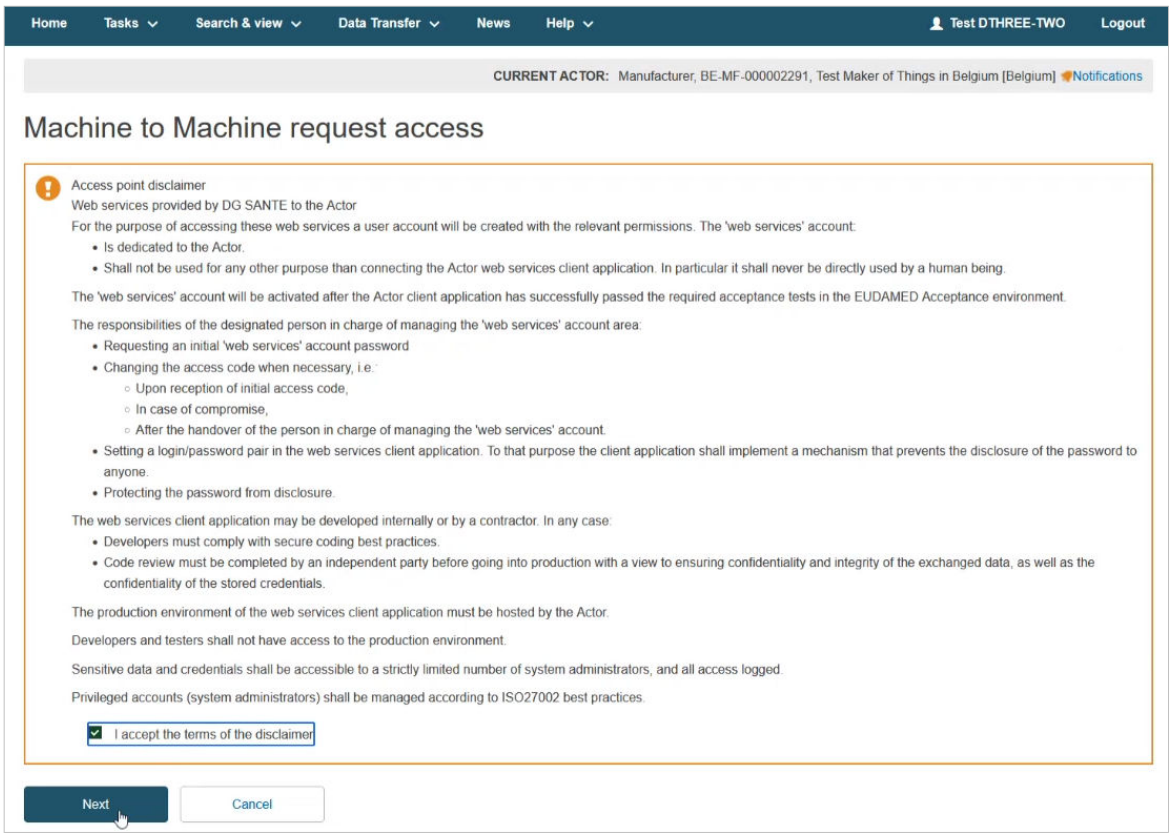
1. Log into EUDAMED as a Local Actor Administrator (LAA). Under the *My Actor data* section click on the *Access point management* link:



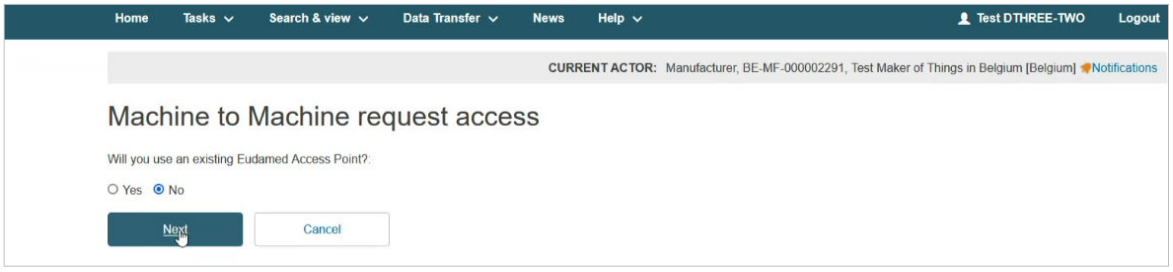
2. Click on the **Request a new M2M access** button:



3. Agree to the disclaimer and click on the **Next** button:

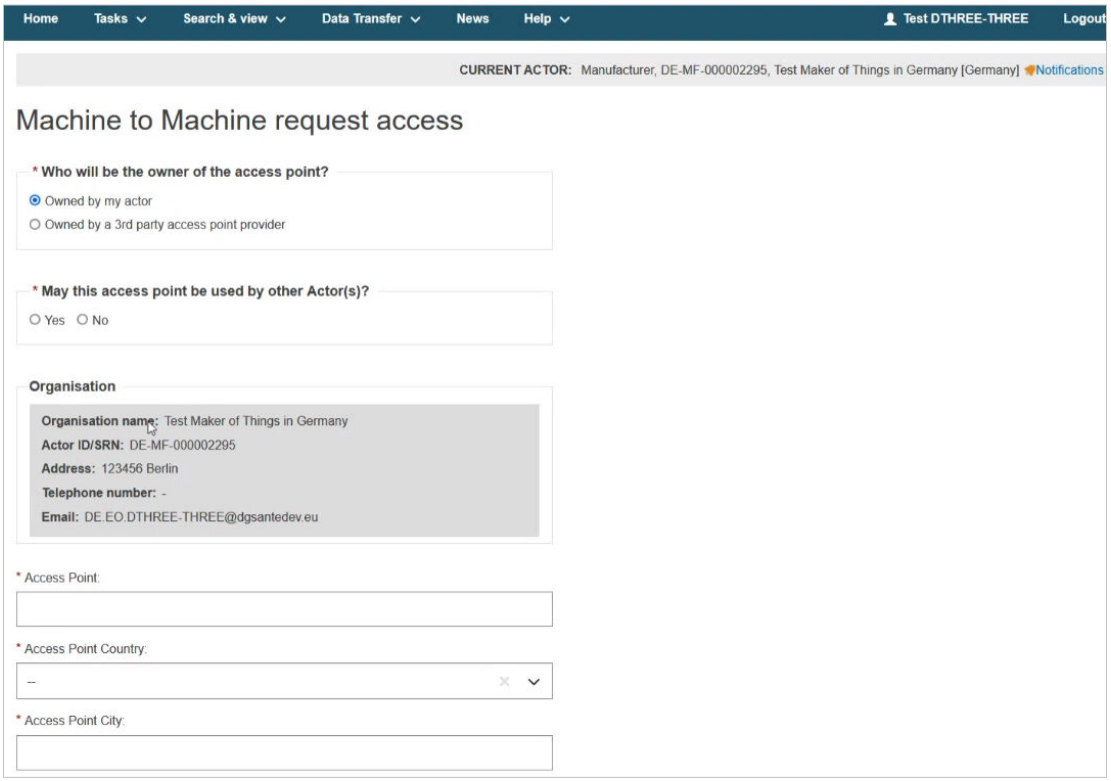


4. In the next screen select **No** to create a new AP and click on the **Next** button:

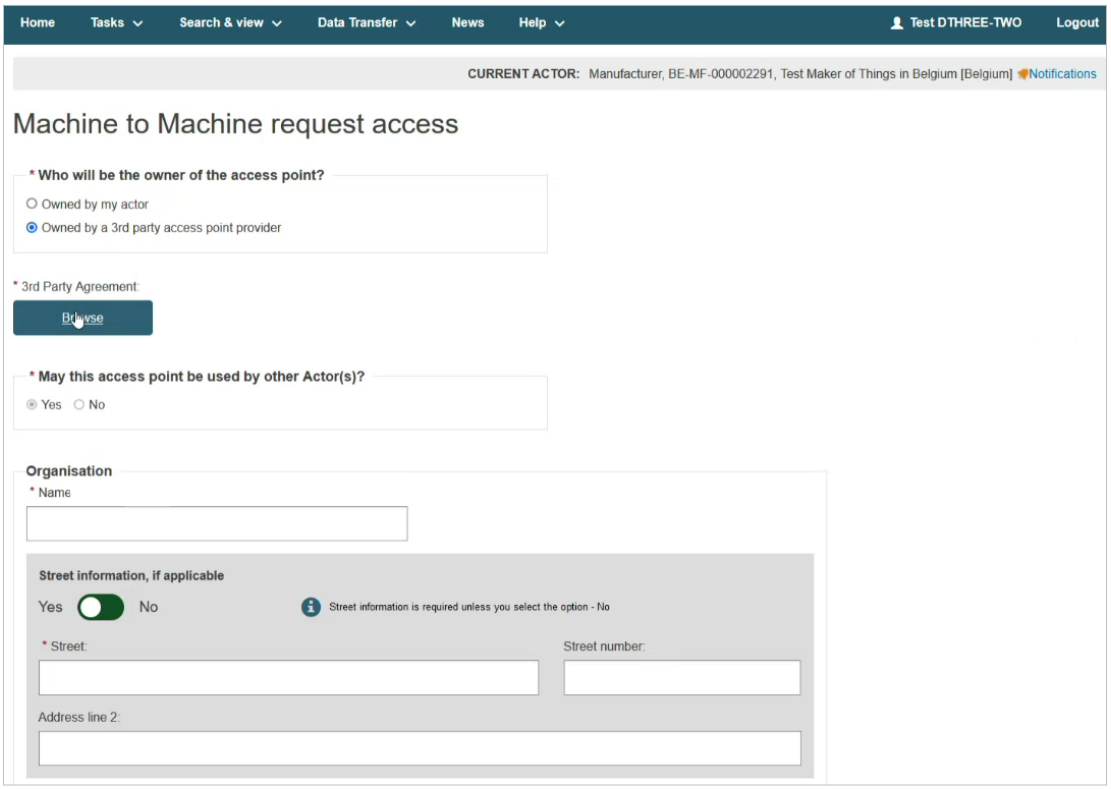


5. Select the owner of the AP:

a. If you own the AP, you must select if you will share this AP or not. The Organisation details will be prefilled:



b. If the AP is owned by a 3rd party company, the *May this access point be used by other Actor(s)?* field will be greyed out and you will be asked to upload the 3rd Party Agreement:



NOTE
3rd Party Agreement

The 3rd Party Agreement document is to confirm that a company (either a 3rd party provider or another Actor registered in EUDAMED having already an AP) agrees to act as 3rd party service provider for the Actor organisation that submits the request for the purposes of M2M DTX with EUDAMED.

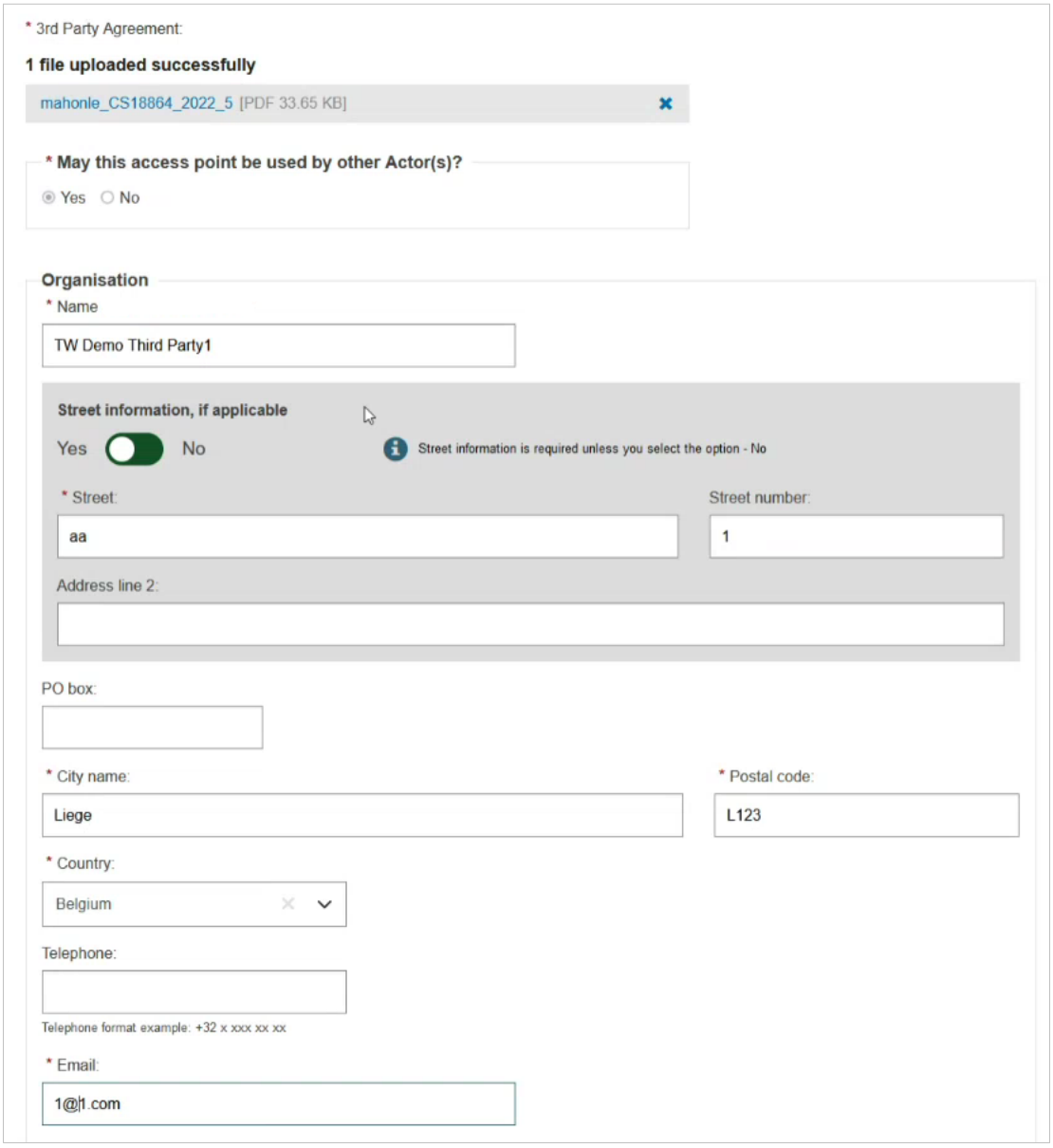
This document will be used to assess the validation of the AP access/link request and must be provided for each environment where an AP access/link is requested (Playground and Production).

This document must identify clearly the two parties (Actor and 3rd party service provider) and must be signed by both parties for confirming the agreement from the 3rd party service provider toward the Actor.

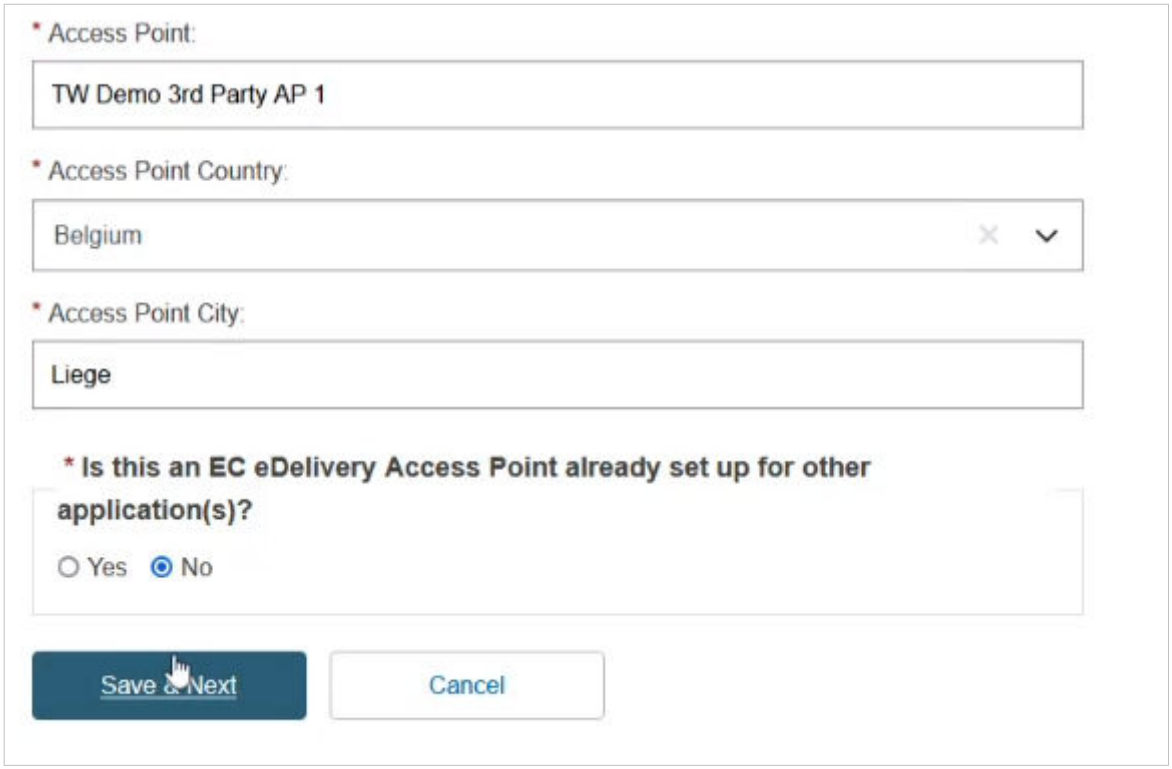
You can download the 3rd Party Agreement template here.

You must upload a PDF file in the *3rd Party Agreement* field.

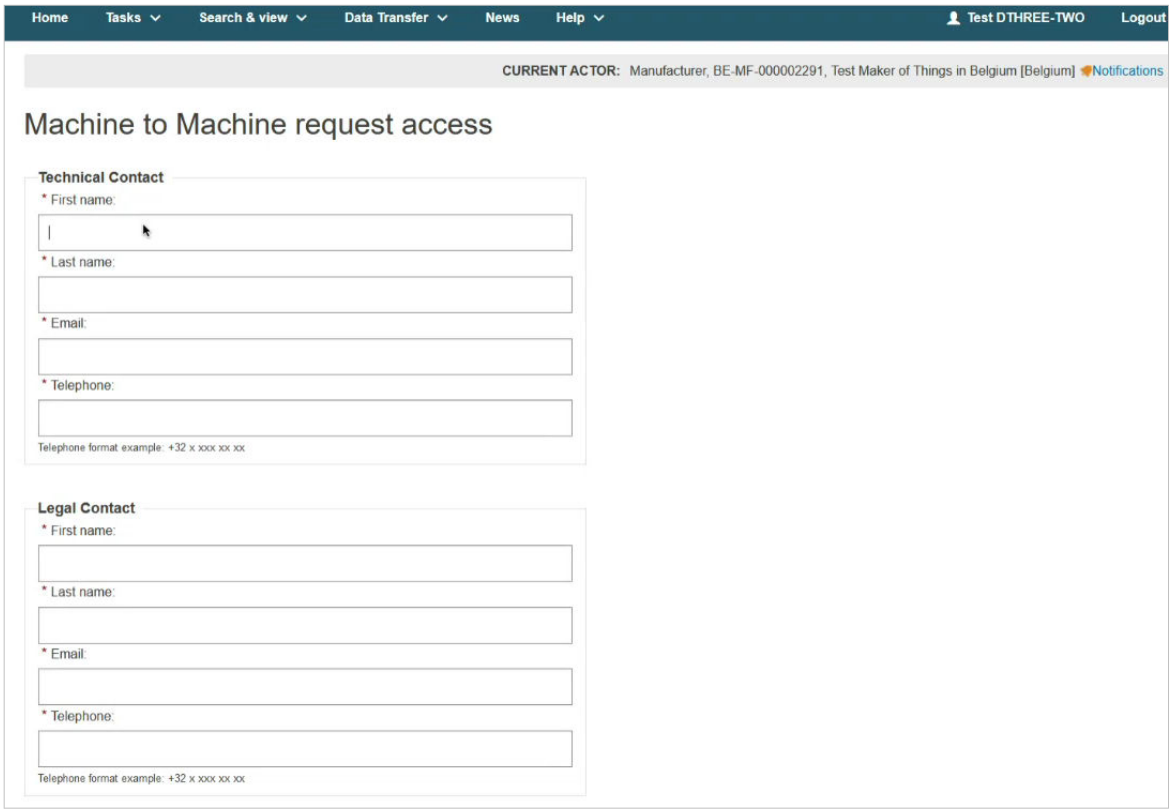
Enter information related to the *Organisation*:



6. Enter the name of the AP in the *Access Point* field, the country and the city of the AP, then select **Yes** if you use this AP in other eDelivery application or **No** otherwise and click on the **Save & Next** button:



7. In the next screen fill in your technical and legal contact details:



NOTE

Technical contact: The technical person to contact for setting up the AP and for any related technical issues.

This person must belong to the organisation that owns the AP (if the owner is a 3rd party provider, the technical contact should belong to the 3rd party provider; if the owner is an Economic Operator, the technical contact should belong to the Economic Operator).

Legal Contact: The person to contact in case there is a non-technical issue with the way the AP is used (e.g., misuse) or for notifications in case of maintenance/bug etc. that impacts the use of the AP.

This person must belong to the Actor's organisation submitting the request.

8. Upload your Proof of testing:

NOTE

In the *Business justification / Proof of testing* field use the Business justification document if you are a Playground user and the Proof of testing document if you are a user of the Production environment. See the EUDAMED environments page for further information about the existing environments accessible to the users.

IMPORTANT

Proof of testing

The Proof of testing is required to set-up an AP in Production. Its purpose is to confirm that the prerequisite of a successful AP access/link in Playground is fulfilled by getting concrete information on the successful data exchange in Playground in order to assess whether the Actor can have an active AP in Production.

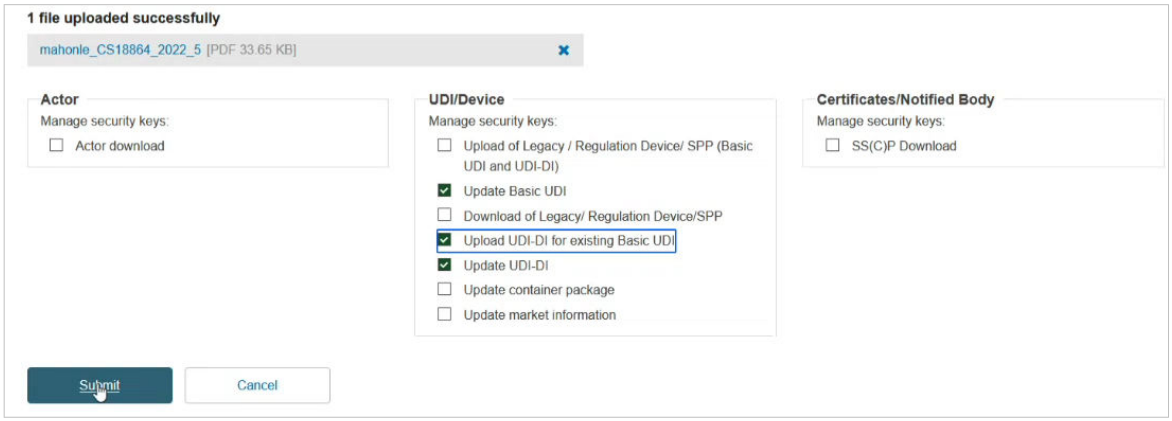
There is no template. However, it should contain the information listed below in a single PDF document:

1. Your Actor's SRN/ID and Party ID in Playground.
2. The XML request and response files that prove successful upload and/or download via M2M in Playground environment.

In case you request the upload of Legacy/Regulation Device/SPP (Basic UDI and UDI-DI) service, please also provide:

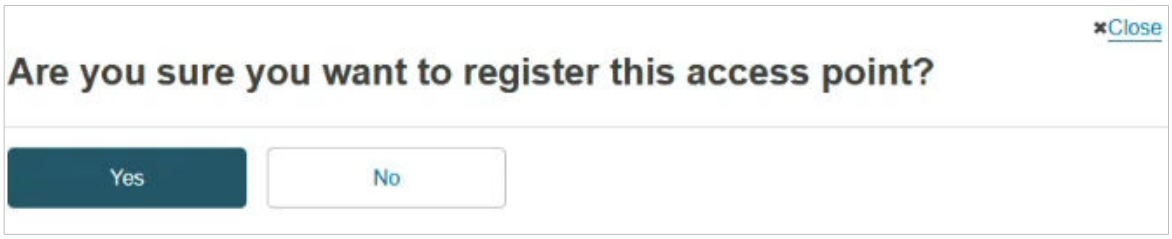
1. The Basic UDI-DI and UDI-DI of a successfully registered device in Playground.
2. The XML request and response files that prove that you have successfully registered this specific device (see point 1 above) via M2M in Playground.

9. Select the services that you want to use this AP for and click on the **Submit** button:



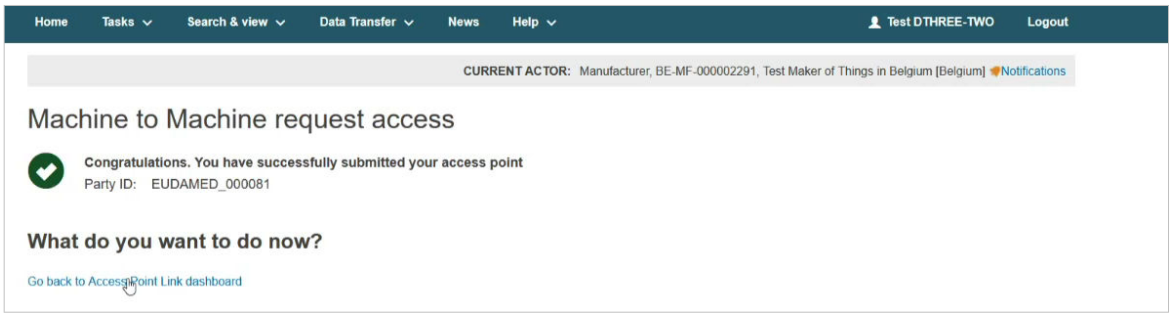
NOTE
You will be able to edit the selected services after the activation of your AP. Each service can be linked to **only** one AP. See section Generate your security key [31] of the current guide for more information.

10. Select **Yes** in the pop-up window to complete the AP registration:



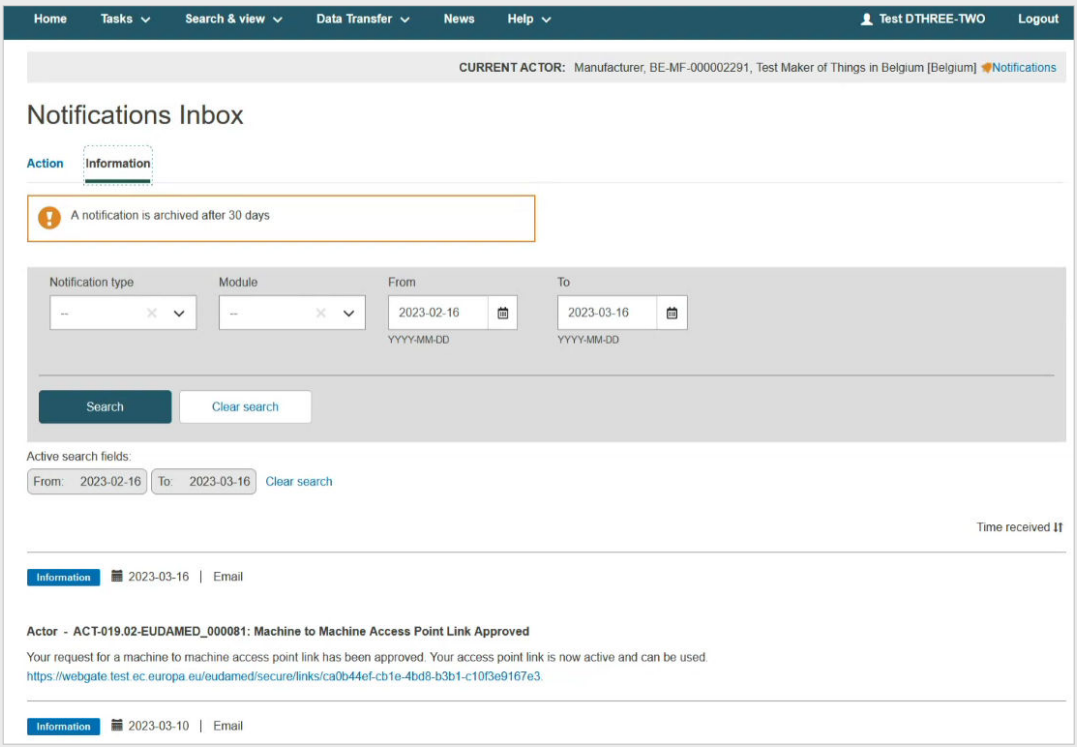
NOTE
If you select **Yes** your request will be stored as *submitted*. Otherwise, your request will be stored as *draft* (see Annex 3 (AP statuses) [60] of the current guide).

11. In the next screen you can see a confirmation message and the AP's Party ID. Click on the *Go back to Access Point Link dashboard* link to manage your AP:



NOTE

EUDAMED team at the Commission will either accept or reject your request. If your request is rejected then the status of your request will be *rejected* and you will be able to request again. Otherwise, the status of your request will be *connectivity under validation* (see Annex 3 (AP statuses) [60] of the current guide). Upon EUDAMED team's decision (request accepted/rejected), you will get a notification in your *Notifications Inbox*:



IMPORTANT

After your request is approved and the status of your request is *connectivity under validation* you must configure your AP to proceed with the onboarding. See How to configure your AP [36] for further information.

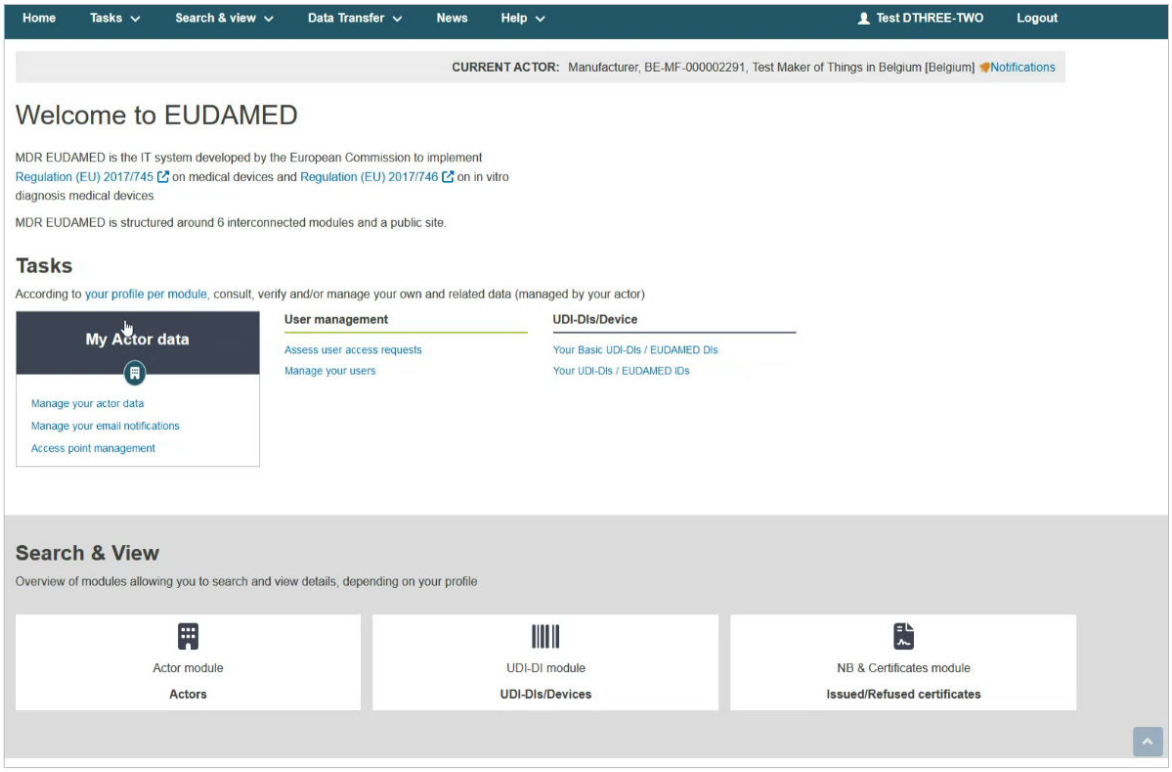
5.1.2 Request the use of an existing AP

VIDEO: Link to an existing AP

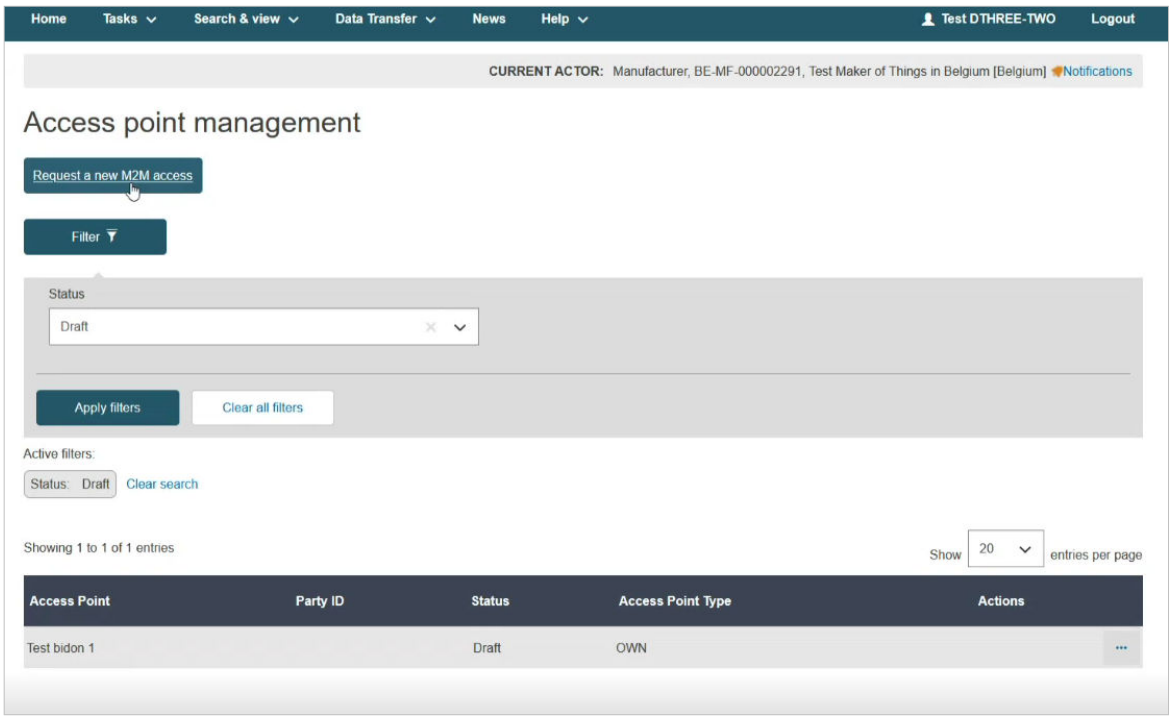
NOTE

Economic Operators can have **up to two** APs in *active/submitted/connectivity under validation* statuses while Competent Authorities, Designating Authorities and Notified Bodies can have **only one** AP in active/submitted/connectivity under validation statuses. See Annex 3 (AP statuses) [60] of the current guide for further information about the AP statuses.

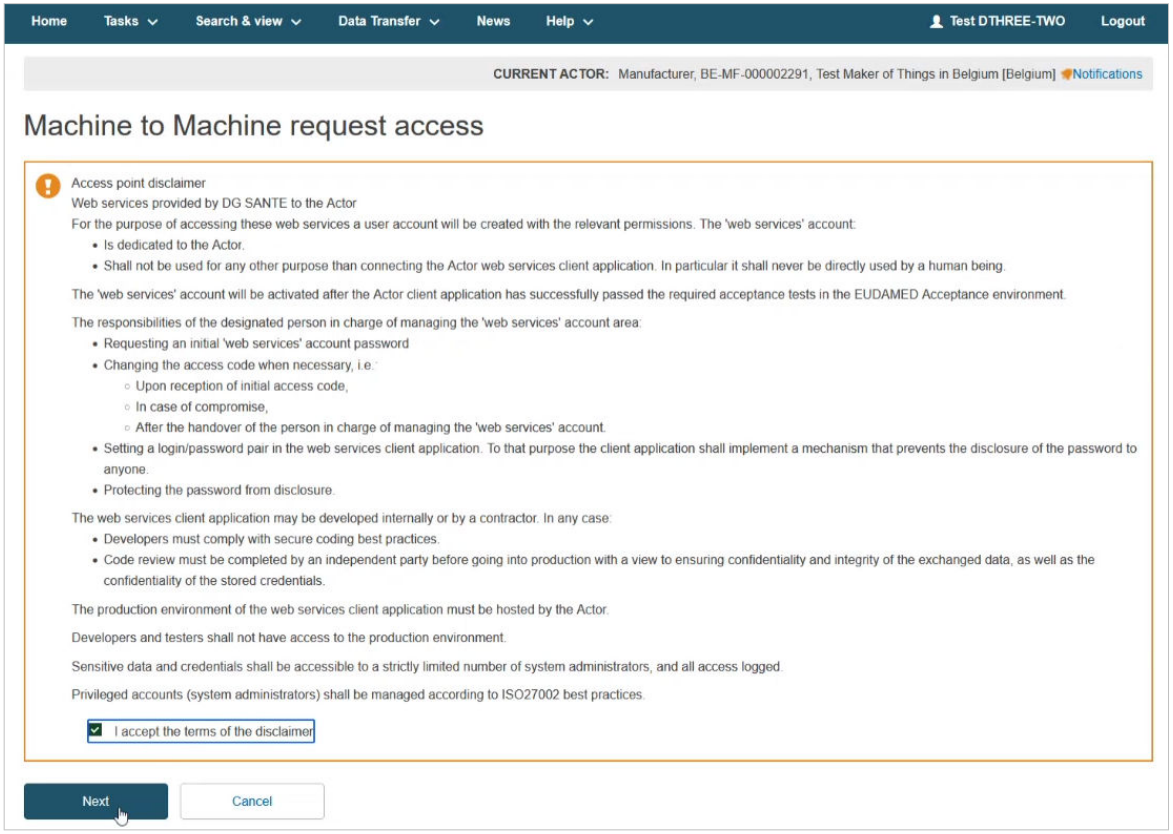
1. Log into EUDAMED as a Local Actor Administrator (LAA). Under the *My Actor data* section click on the *Access point management* link:



2. Click on the **Request a new M2M access** button:

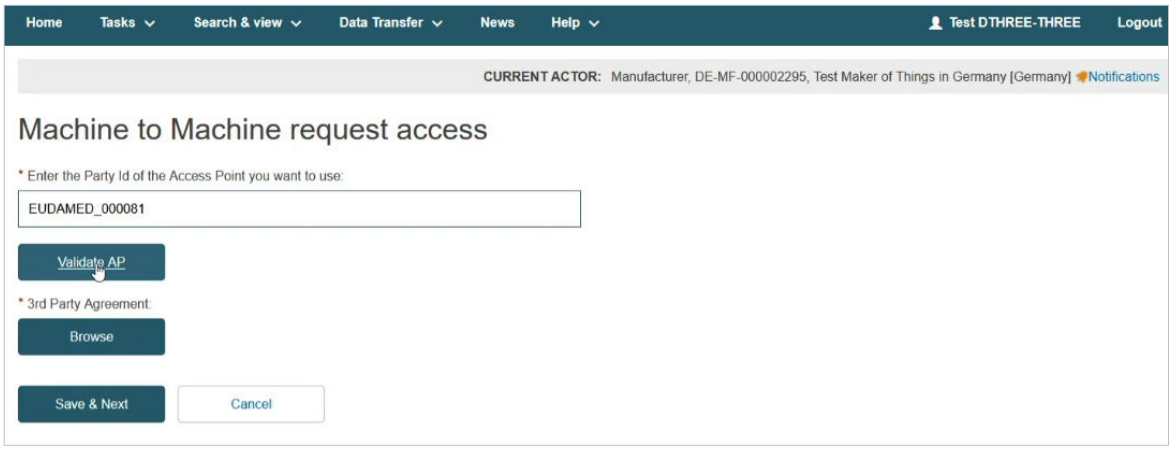


3. Agree to the disclaimer and click on the **Next** button:

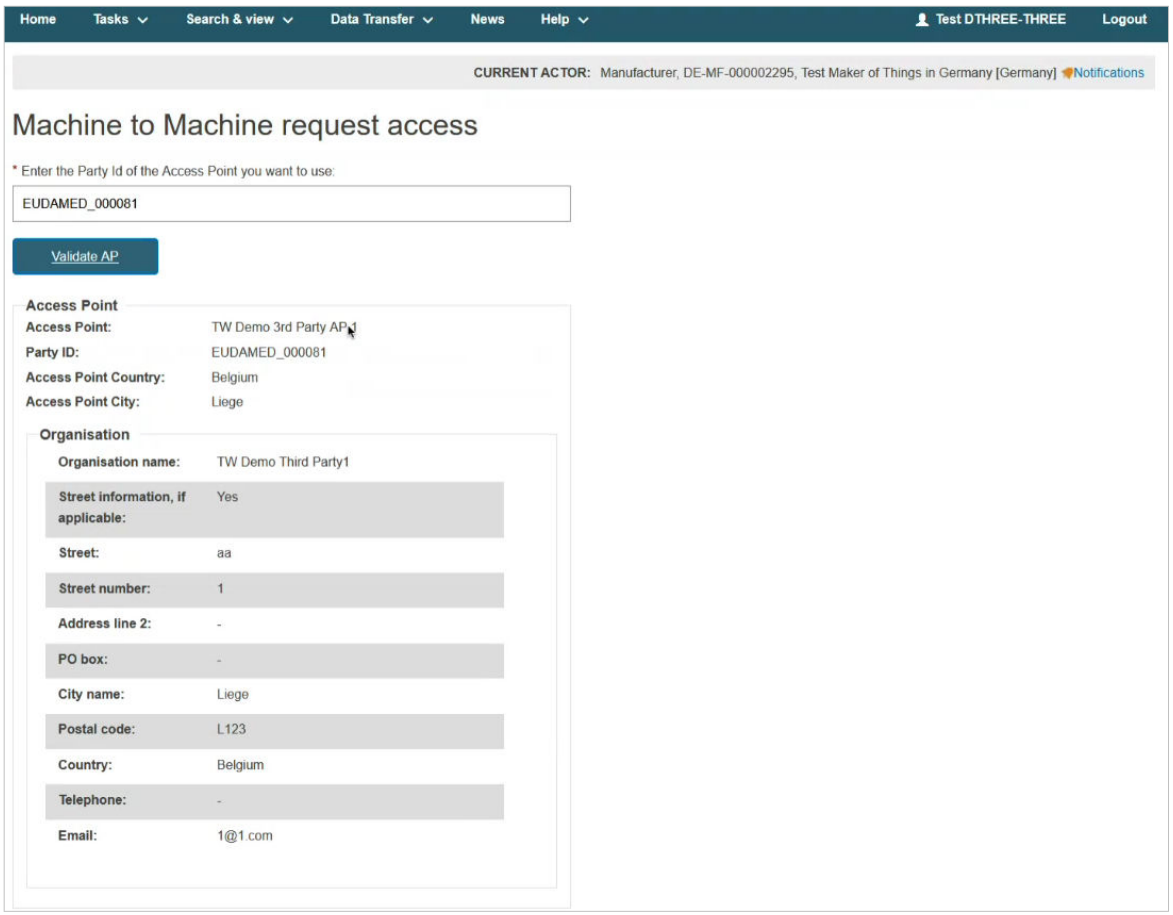


4. In the next screen select **Yes** to use an existing AP and click on the **Next** button:

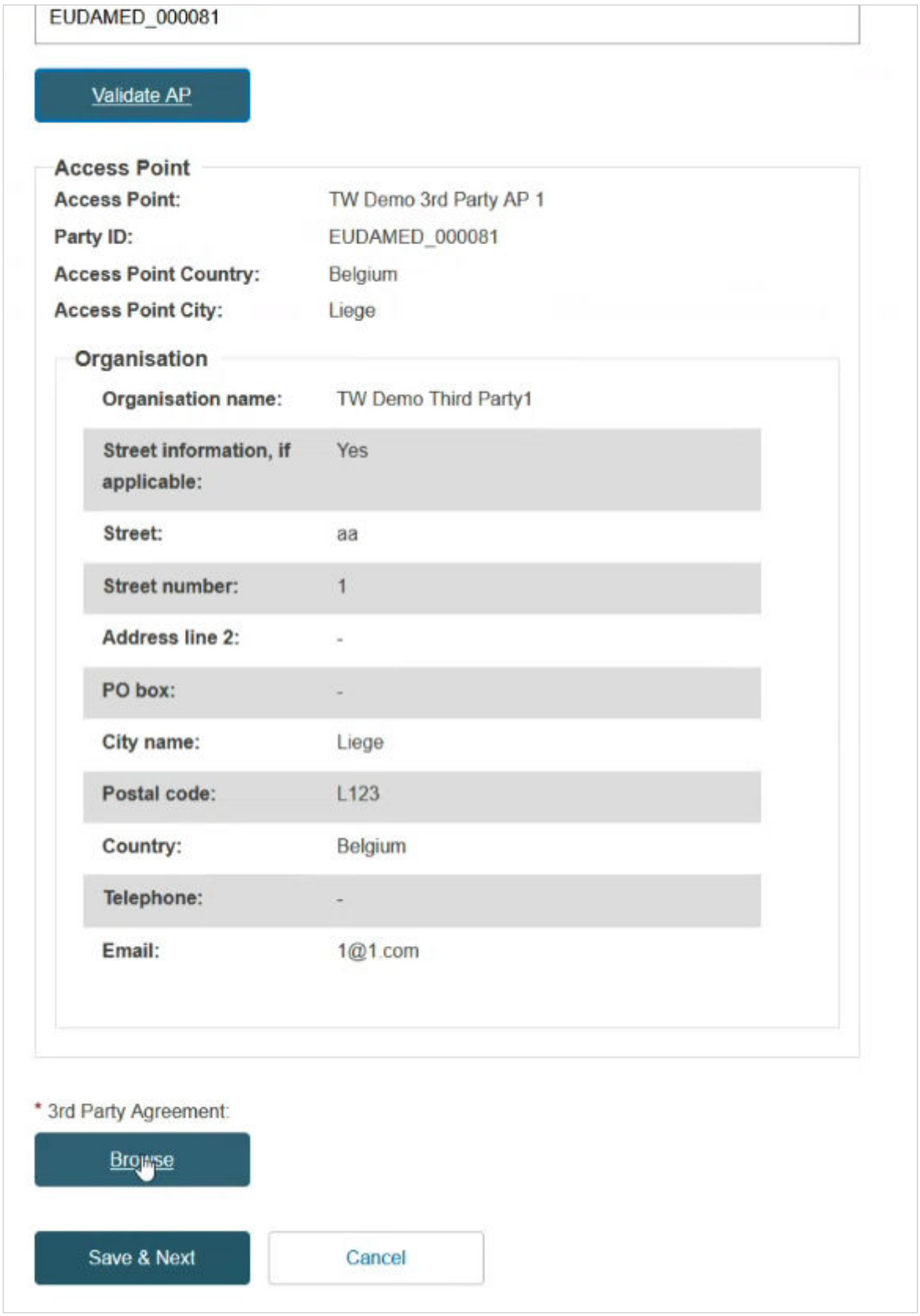
5. Enter the Party ID of the Access Point that you want to use and click on the **Validate AP** button:



6. You will be able to view information regarding the AP with the specified Party ID. You will not be able to edit that information:



7. Upload the 3rd Party Agreement and click on the **Save & Next** button:


EUDAMED_000081
Validate AP
Access Point
Access Point: TW Demo 3rd Party AP 1
Party ID: EUDAMED_000081
Access Point Country: Belgium
Access Point City: Liege
Organisation
Organisation name: TW Demo Third Party1
Street information, if applicable: Yes
Street: aa
Street number: 1
Address line 2: -
PO box: -
City name: Liege
Postal code: L123
Country: Belgium
Telephone: -
Email: 1@1.com
* 3rd Party Agreement:
Browse
Save & Next
Cancel

NOTE
3rd Party Agreement

The 3rd Party Agreement document is to confirm that a company (either a 3rd party provider or another Actor registered in EUDAMED having already an AP) agrees to act as 3rd party service provider for the Actor organisation that submits the request for the purposes of M2M DTX with EUDAMED.

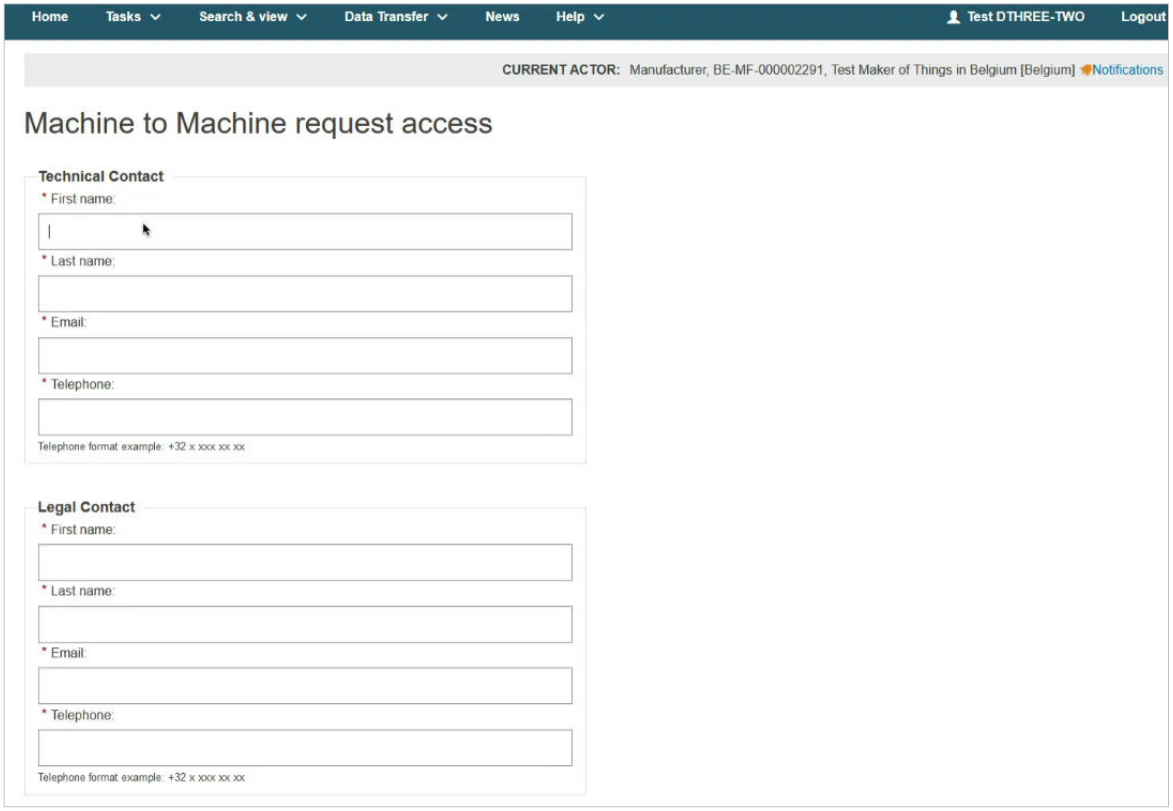
This document will be used to assess the validation of the AP access/link request and must be provided for each environment where an AP access/link is requested (Playground and Production).

This document must identify clearly the two parties (Actor and 3rd party service provider) and must be signed by both parties for confirming the agreement from the 3rd party service provider toward the Actor.

You can download the 3rd Party Agreement template here.

You must upload a PDF file in the *3rd Party Agreement* field.

8. In the next screen fill in your technical and legal contact details:



NOTE

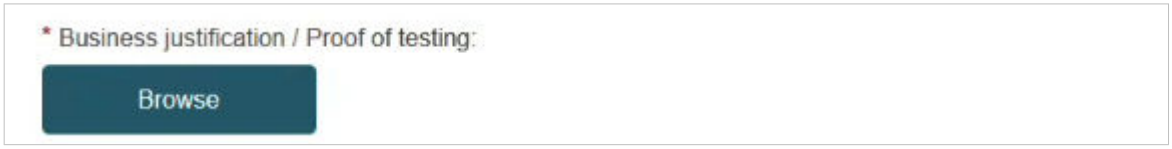
Technical contact: The technical person to contact for setting up the AP and for any related technical issues.

This person must belong to the organisation that owns the AP (if the owner is a 3rd party provider, the technical contact should belong to the 3rd party provider; if the owner is an Economic Operator, the technical contact should belong to the Economic Operator).

Legal Contact: The person to contact in case there is a non-technical issue with the way the AP is used (e.g., misuse) or for notifications in case of maintenance/bug etc. that impacts the use of the AP.

This person must belong to the Actor's organisation submitting the request.

9. Upload your Proof of testing:



NOTE

In the *Business justification / Proof of testing* field use the Business justification document if you are a Playground user and the Proof of testing document if you are a user of the Production environment. See the EUDAMED environments page for further information about the existing environments accessible to the users.

IMPORTANT

Proof of testing

The Proof of testing is required to set-up an AP in Production. Its purpose is to confirm that the prerequisite of a successful AP access/link in Playground is fulfilled by getting concrete information on the successful data exchange in Playground in order to assess whether the Actor can have an active AP in Production.

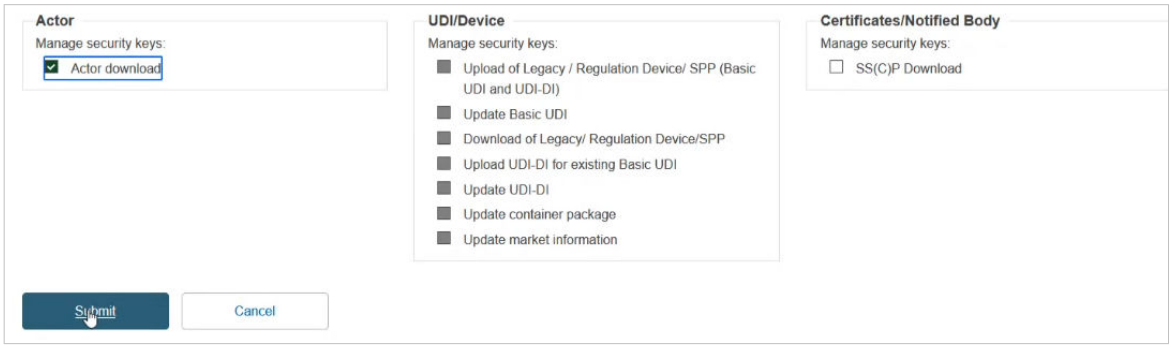
There is no template. However, it should contain the information listed below in a single PDF document:

1. Your Actor's SRN/ID and Party ID in Playground.
2. The XML request and response files that prove successful upload and/or download via M2M in Playground environment.

In case you request the upload of Legacy/Regulation Device/SPP (Basic UDI and UDI-DI) service, please also provide:

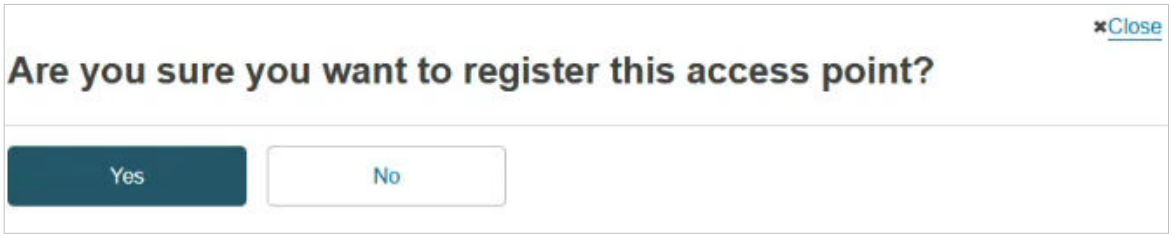
1. The Basic UDI-DI and UDI-DI of a successfully registered device in Playground.
2. The XML request and response files that prove that you have successfully registered this specific device (see point 1 above) via M2M in Playground.

10. Select the services that you want to use this AP for and click on the **Submit** button:



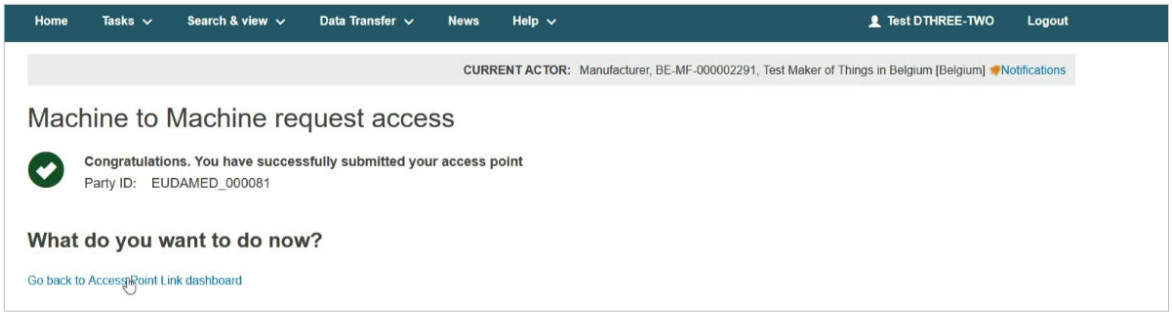
NOTE
You can only select services that are not assigned to another AP. Services assigned to another AP are greyed out. You will be able to edit the selected services after the activation of your AP. See section Generate your security key [31] of the current guide for more information.

11. Select **Yes** in the pop-up window to complete the AP registration process:



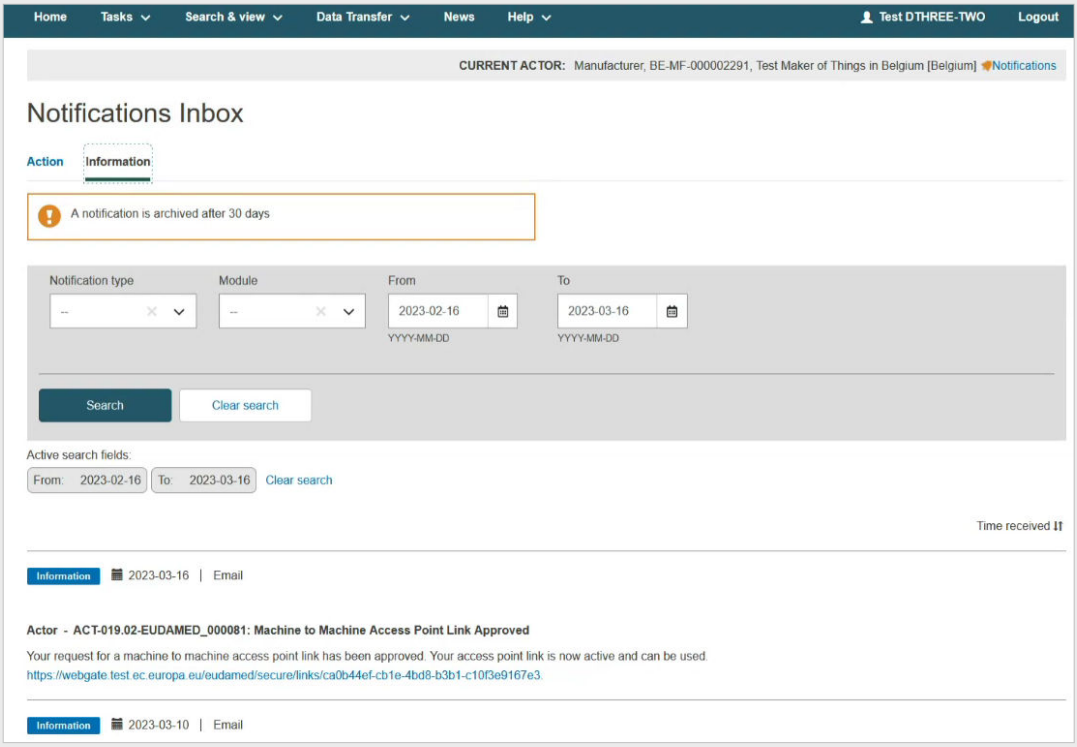
NOTE
If you select **Yes** your request will be stored as *submitted*. Otherwise, your request will be stored as *draft* (see Annex 3 (AP statuses) [60] of the current guide).

12. In the next screen you can see a confirmation message and the AP's Party ID. Click on the *Go back to Access Point Link dashboard link* to manage your AP:



NOTE

EUDAMED team at the Commission will either accept or reject your request. If your request is rejected then the status of your request will be *rejected* and you will be able to request again. Otherwise, the status of your request will be *active* (see Annex 3 (AP statuses) [60] of the current guide). Upon EUDAMED team's decision (request accepted/rejected), you will get a notification in your *Notifications Inbox*:

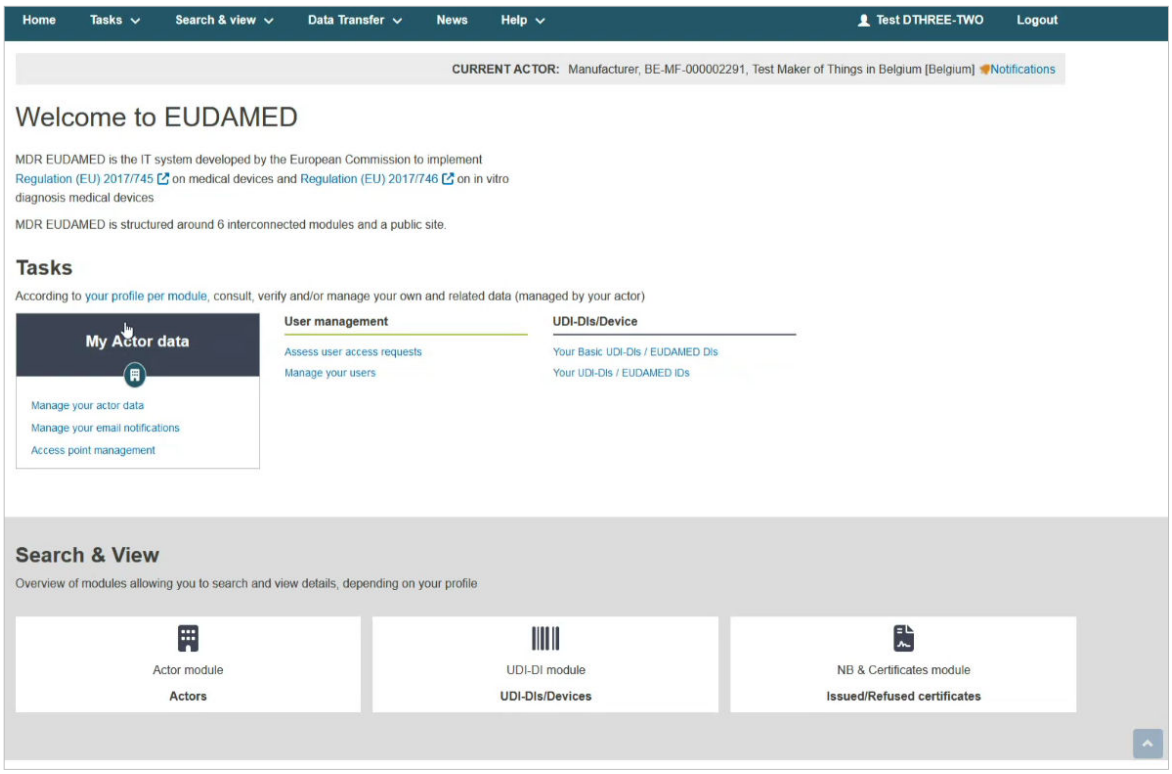


IMPORTANT

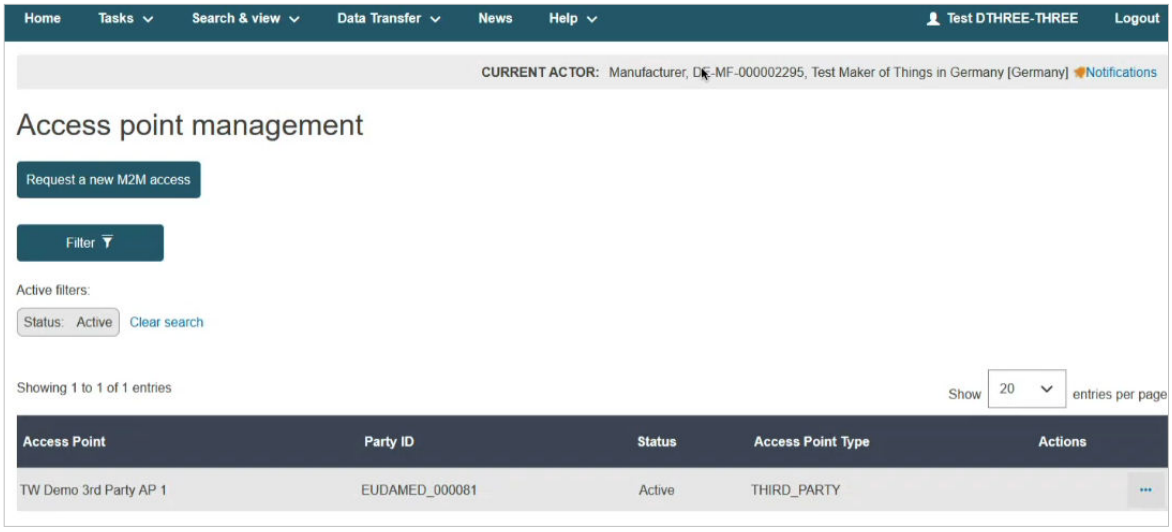
After your request is approved, you must generate your security key before using the M2M services. See Generate your security key [31] for further information.

5.1.3 View your APs

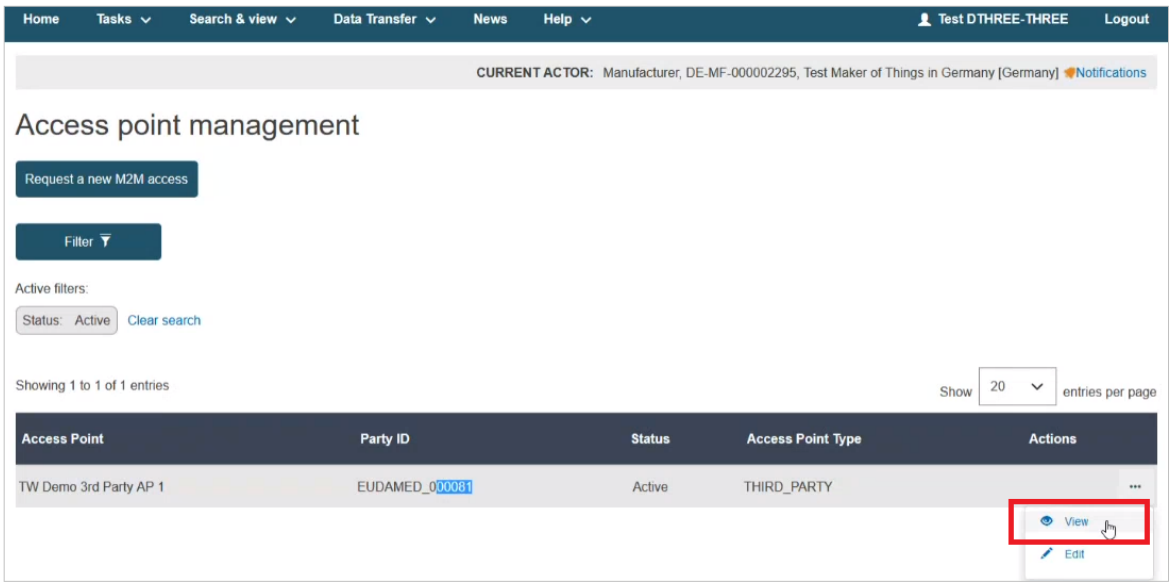
1. Log into EUDAMED as a Local Actor Administrator (LAA). Click on the *Access point management* link under the *My Actor data* section to view your APs:



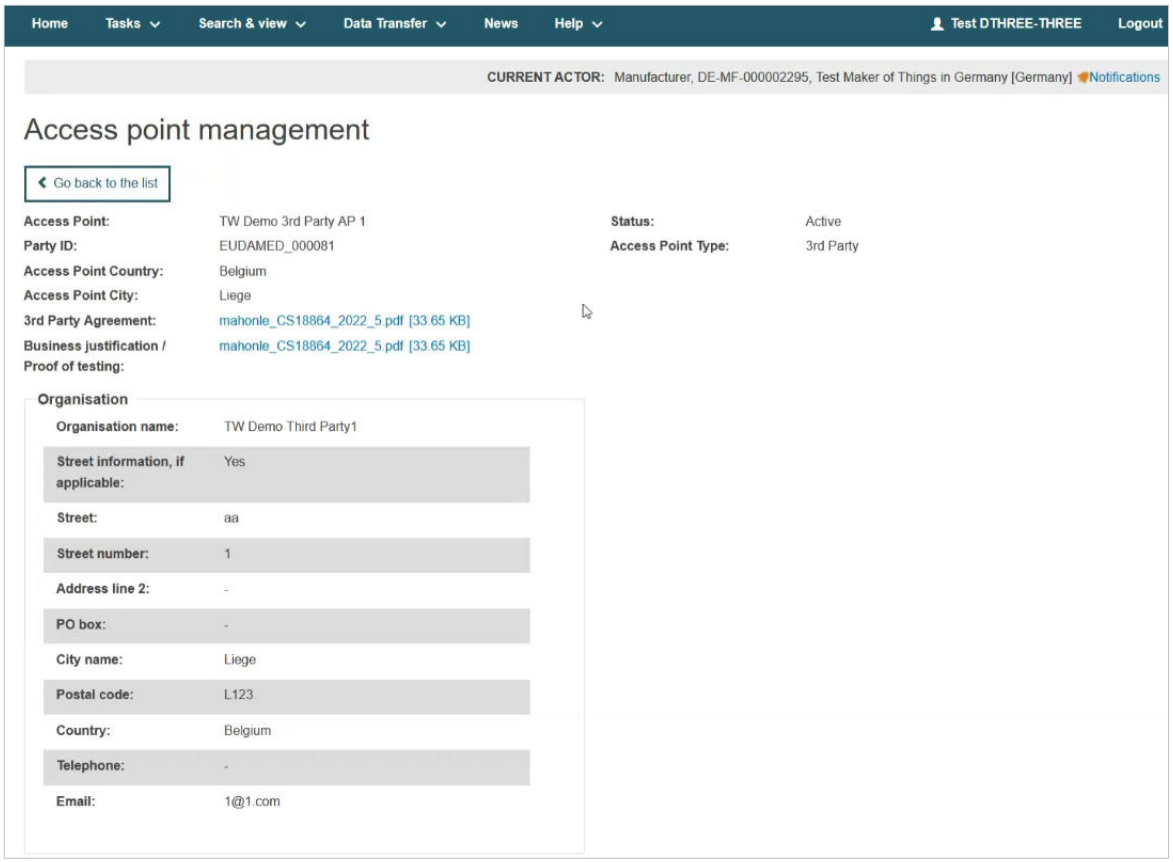
2. In the *Access Point management* page, you can view all your APs listed in the table:

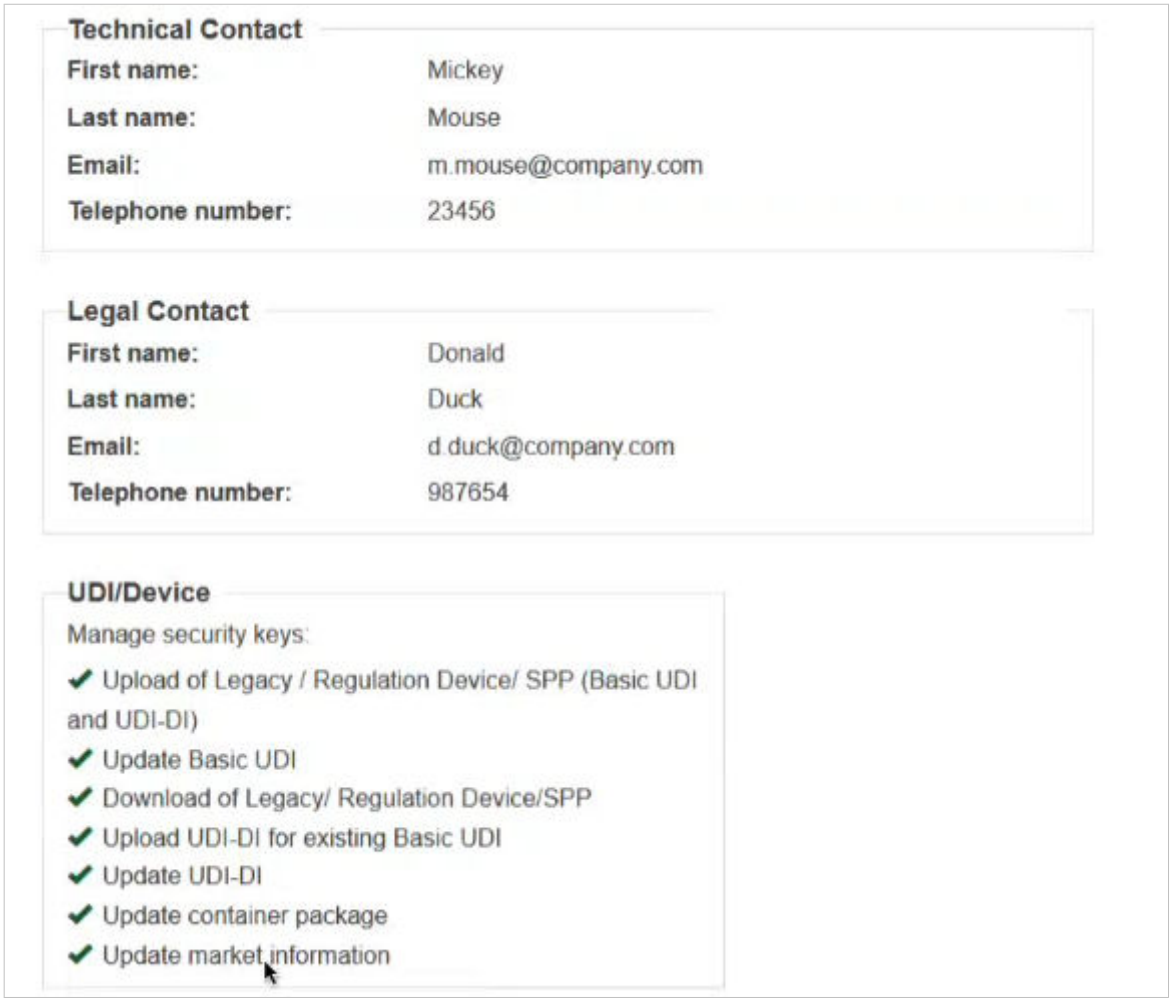


3. To view further details about an AP, click on the *View* link under the three dots:



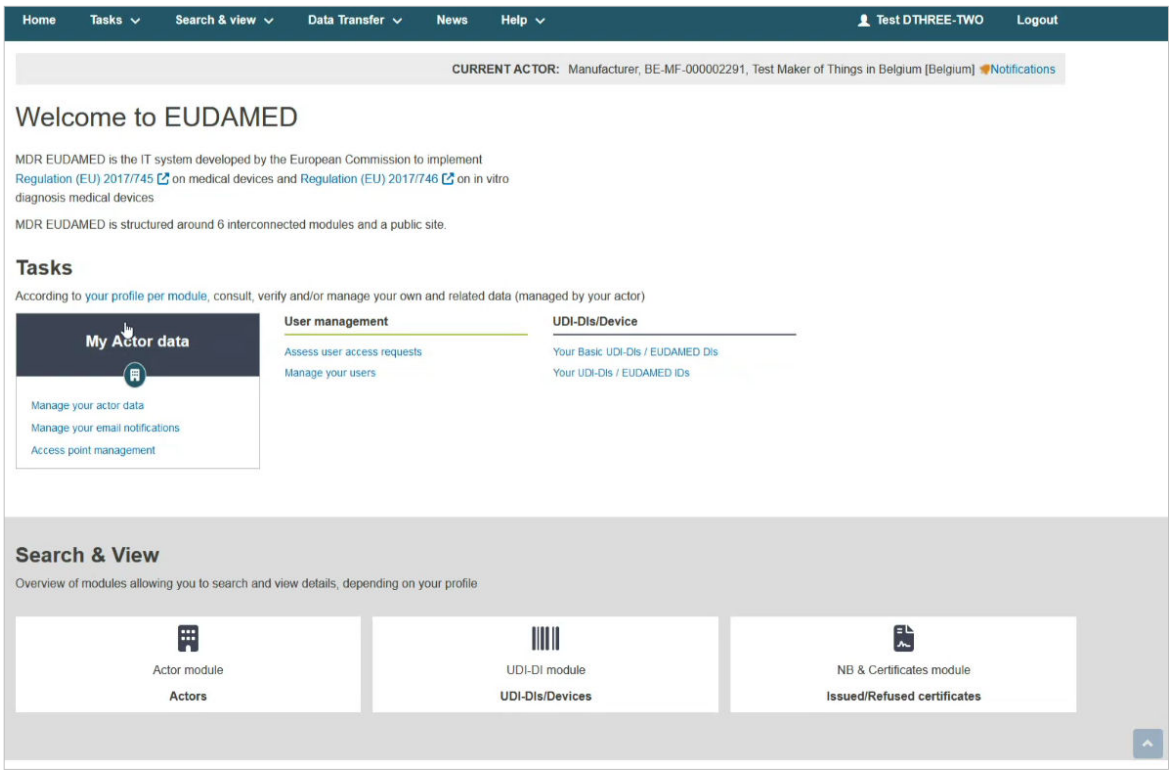
4. In the next screen you can only view details about the selected AP:



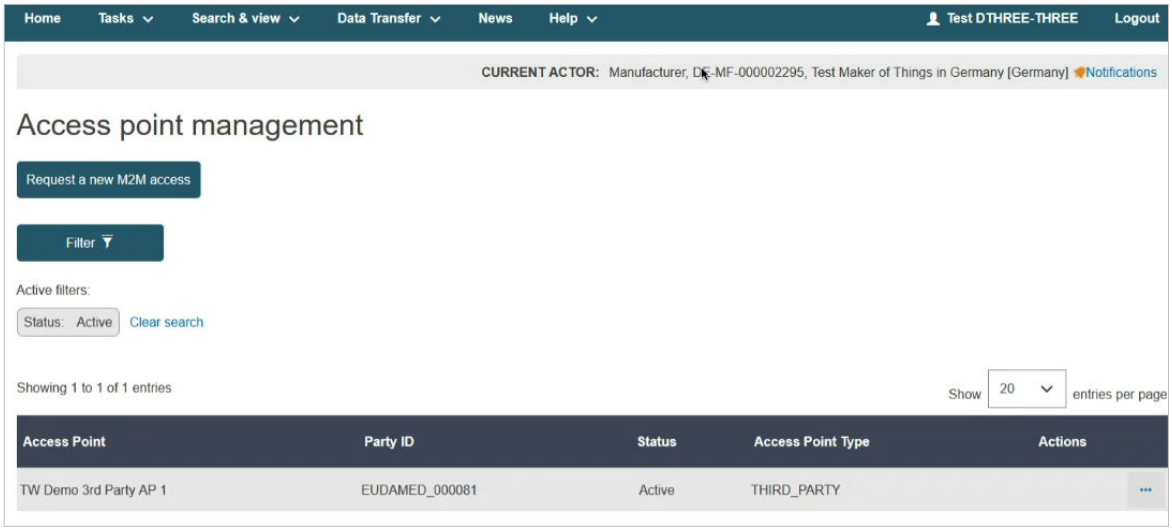


5.1.4 Edit your APs

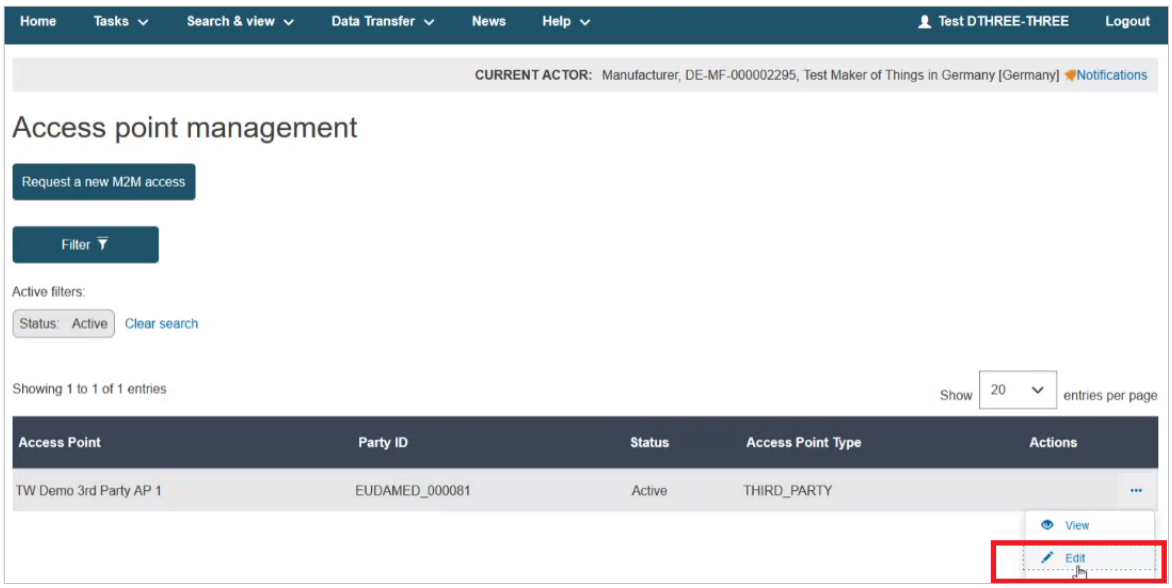
1. Log into EUDAMED as a Local Actor Administrator (LAA). Click on the *Access point management* link under the *My Actor data* section to view your APs:



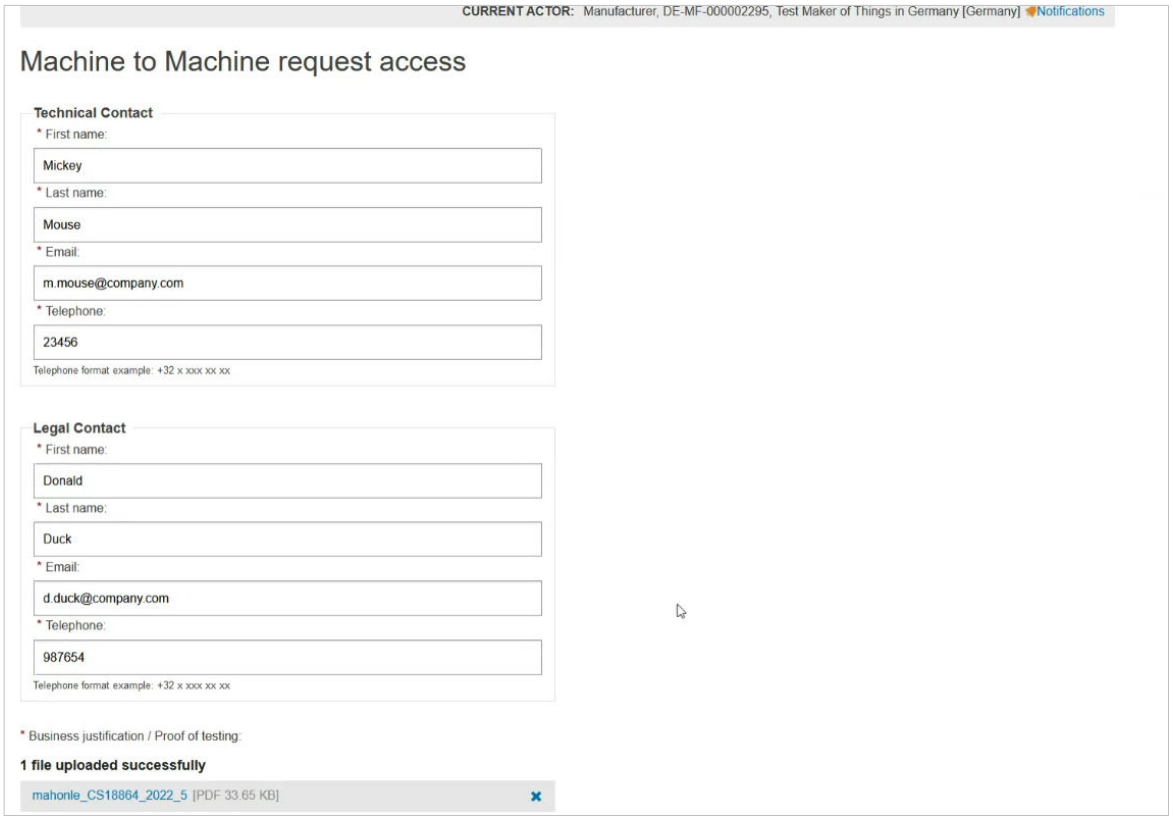
2. In the *Access Point management* page, you can view all your APs listed in the table:



3. Click on the *Edit* link under the three dots to edit your AP's settings:

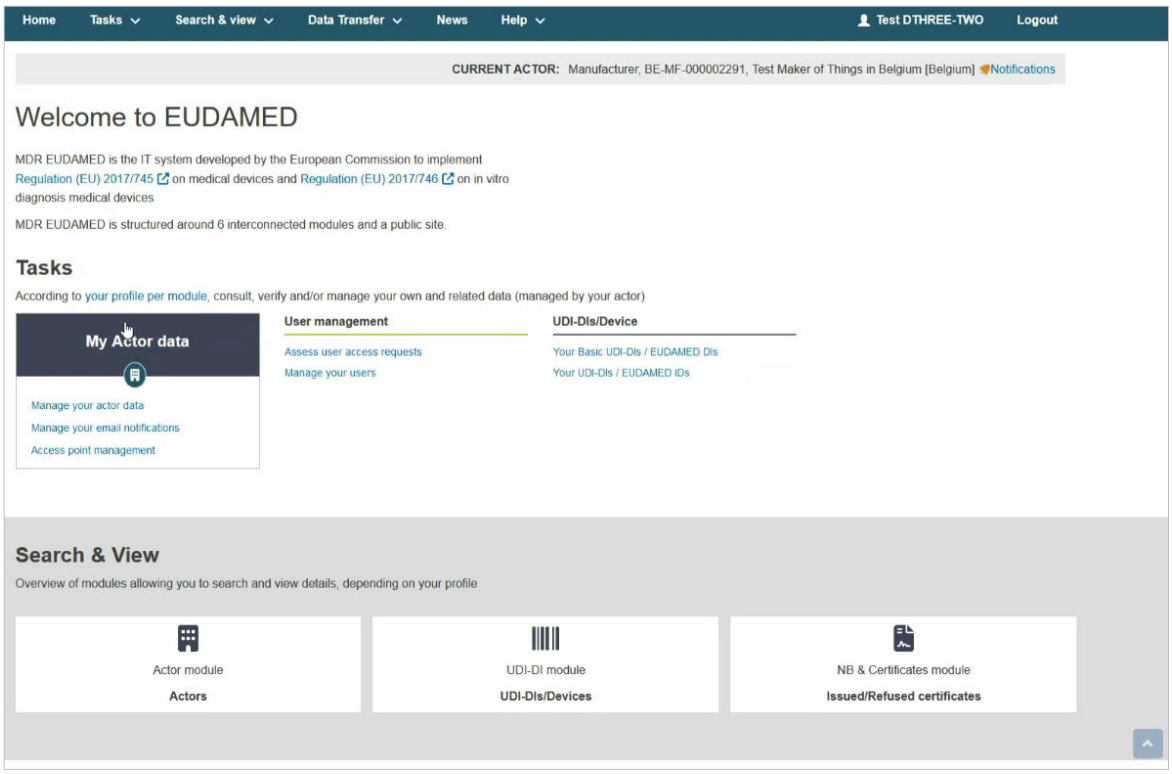


4. In the next screen you can edit your AP details:

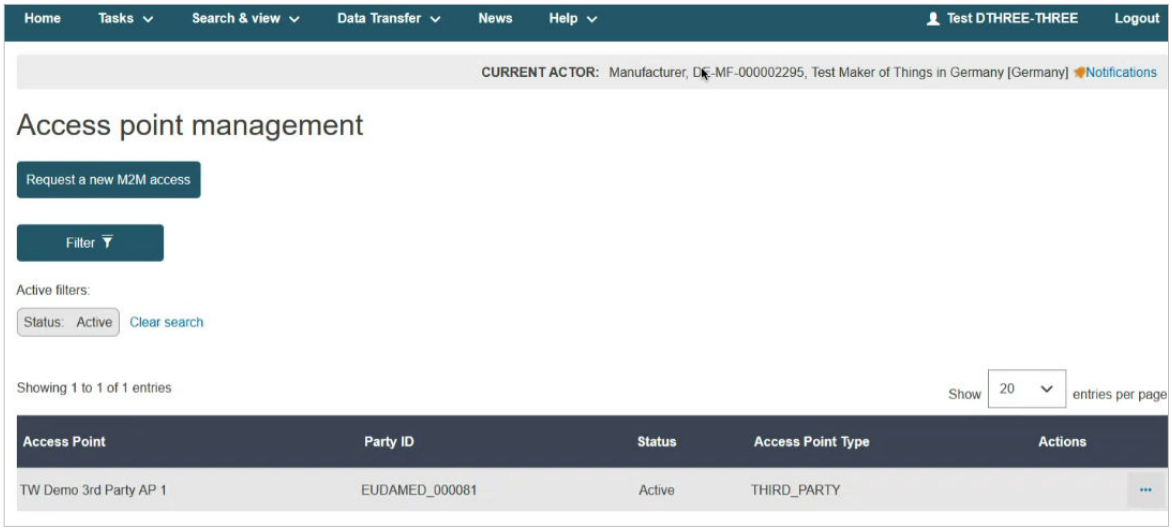


5.1.5 Terminate your AP

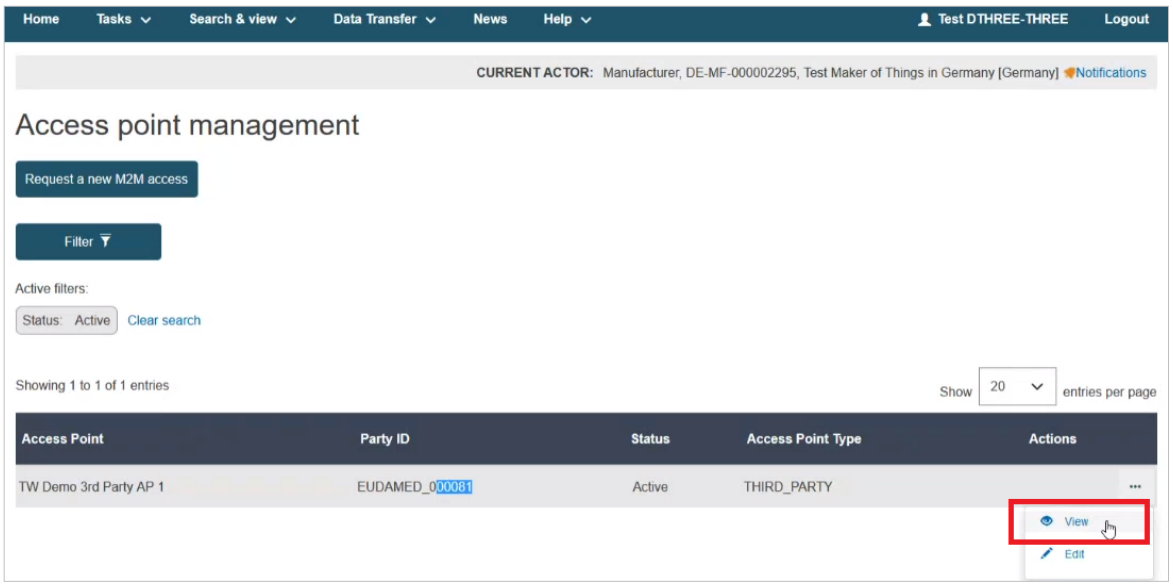
1. Log into EUDAMED as a Local Actor Administrator (LAA). Click on the *Access point management* link under the *My Actor data* section to view your APs:



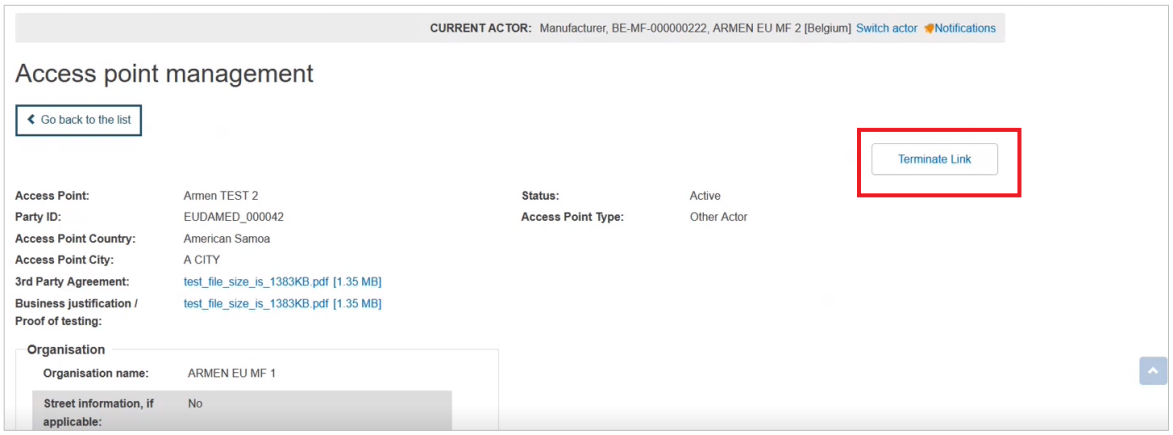
2. In the *Access Point management* page, you can view all your APs listed in the table:



3. Click on the *View* link under the three dots to view further details about your AP:



4. In the next screen click on the **Terminate link** button to terminate your AP:



NOTE

The status of your AP is now *terminated* (see Annex 3 (AP statuses) [60] of the current guide). You will not be able to use this AP anymore nor revert to the previous state.

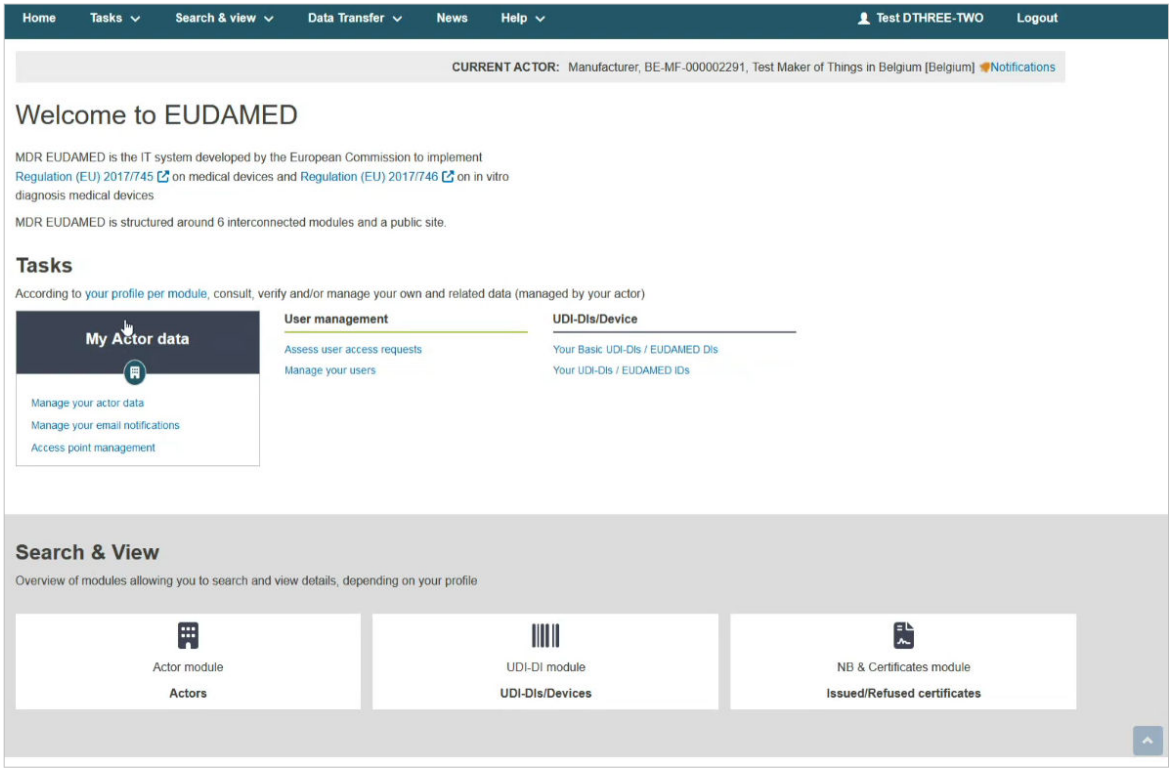
5.1.6 Generate your security key

IMPORTANT

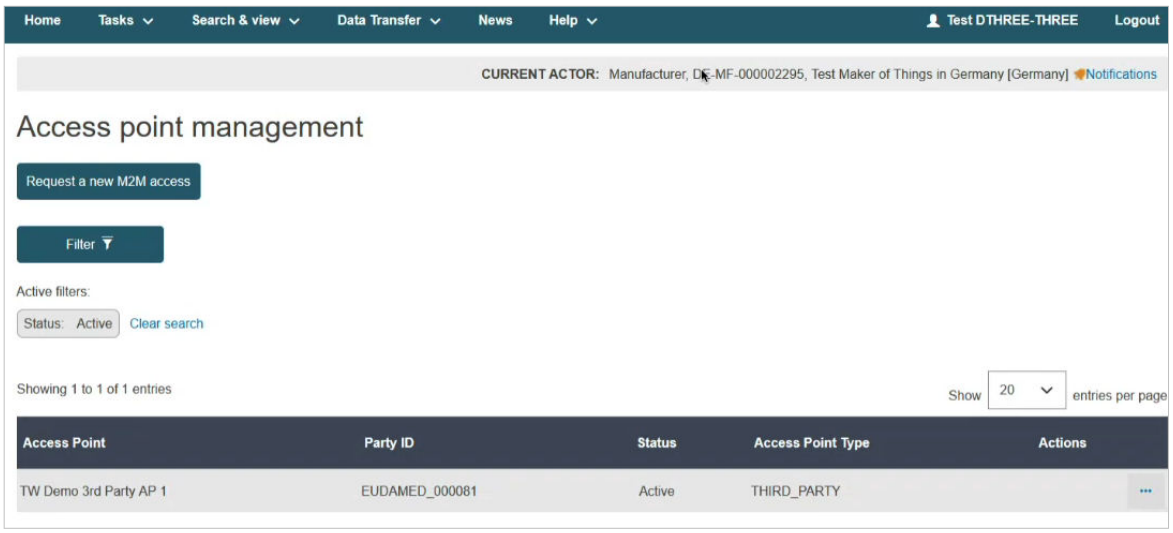
The security key is specific for each EUDAMED module, and the combination of this key and the SRN of the sender is used as security to ensure that the message is received from the correct actor. Without this key, your message will not pass our security checks. An incorrect key will give the following error response:

```
    <message:elementReport>
      <message:operationErrorCode>M-50000</message:operationErrorCode>
      <message:operationErrorDetail>The access point code is not valid: please contact the support</message:operationErrorDetail>
    </message:elementReport>
  </message:report>
  <message:responseCode>SERVER_ERROR</message:responseCode>
</message:PullResponse>
```

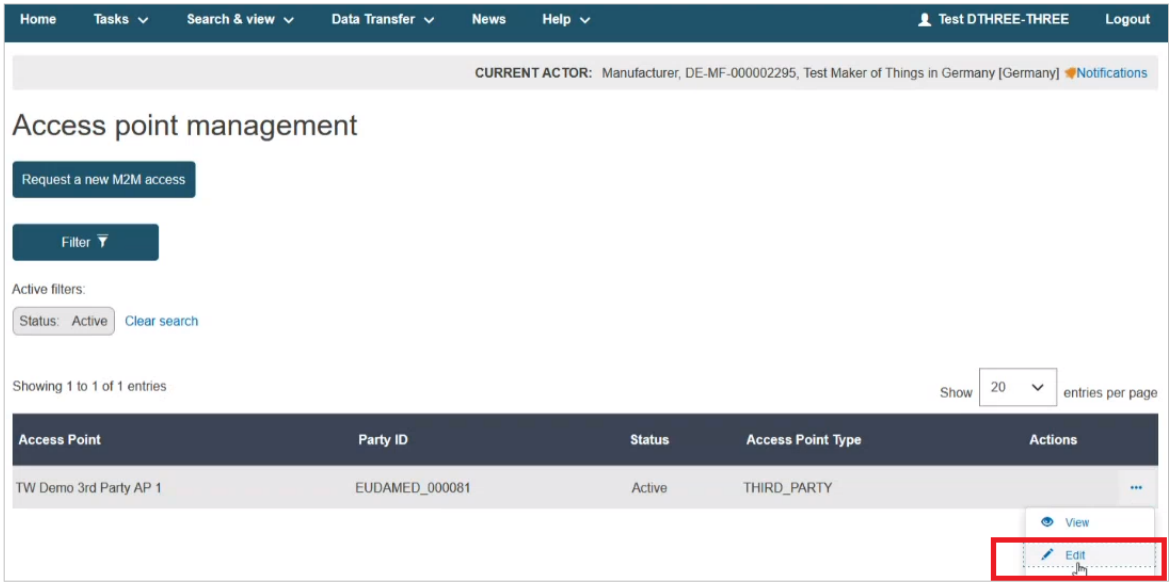
1. Log into EUDAMED as a Local Actor Administrator (LAA). Click on the *Access point management* link under the *My Actor data* section to view your APs:



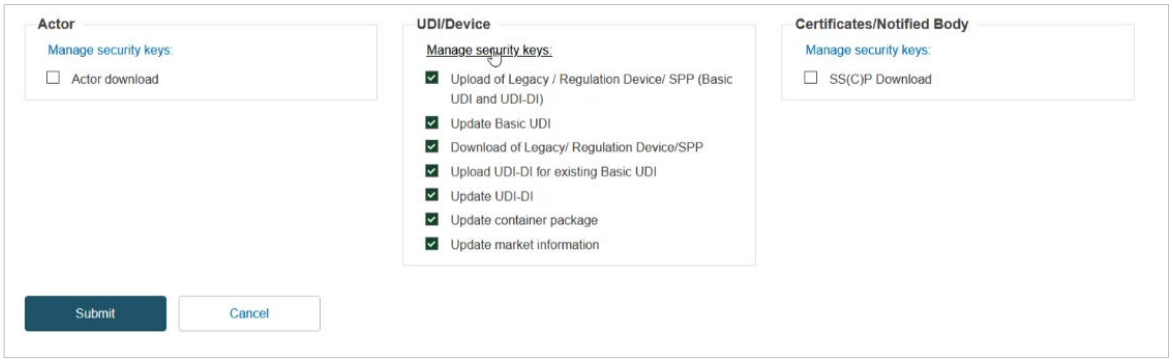
2. In the *Access Point management* page, you can view all your APs listed in the table:



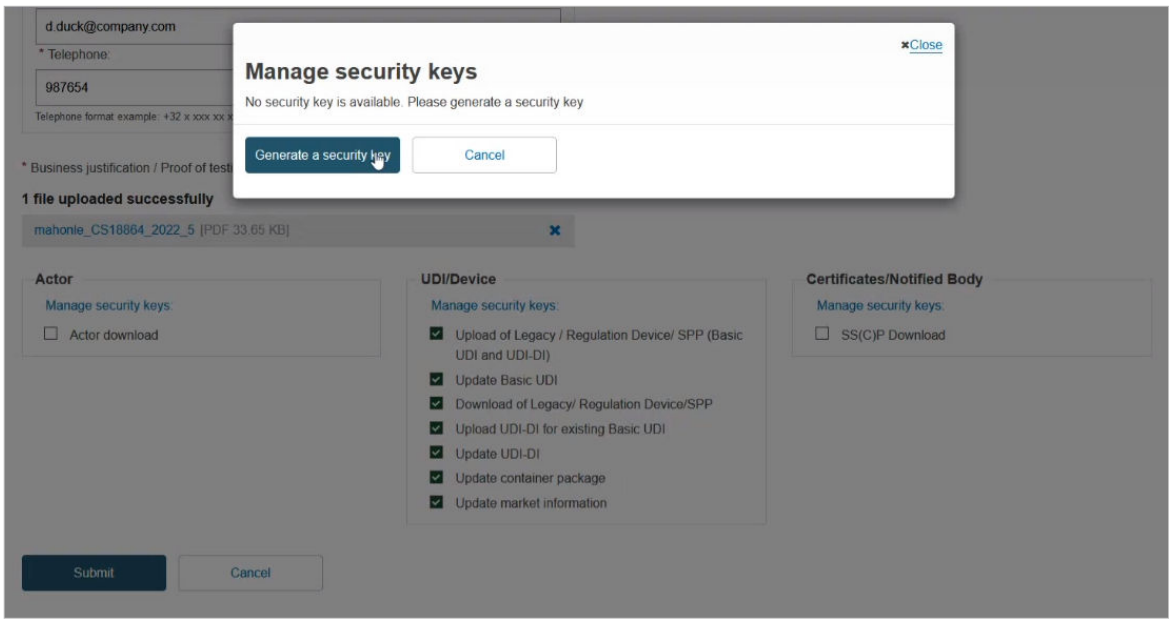
3. Click on the *Edit* link under the three dots to view your AP's settings:



4. In the displayed screen you can view and edit the details you entered regarding the selected AP. Moreover, you can now change the services attached to your AP and manage your security keys by clicking on the *Manage security keys* link:

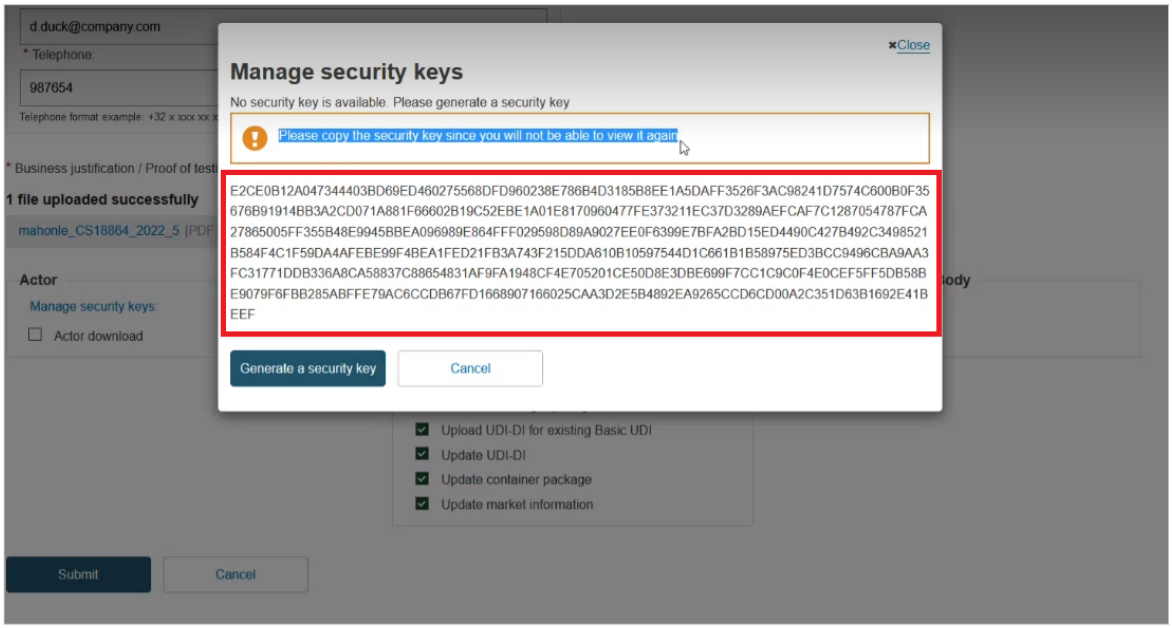


5. In the pop-up window click on the **Generate a security key** button to generate the security key for the selected module:



NOTE
The security keys are generated per module.

6. In the pop-up window you will be able to view the generated security key:

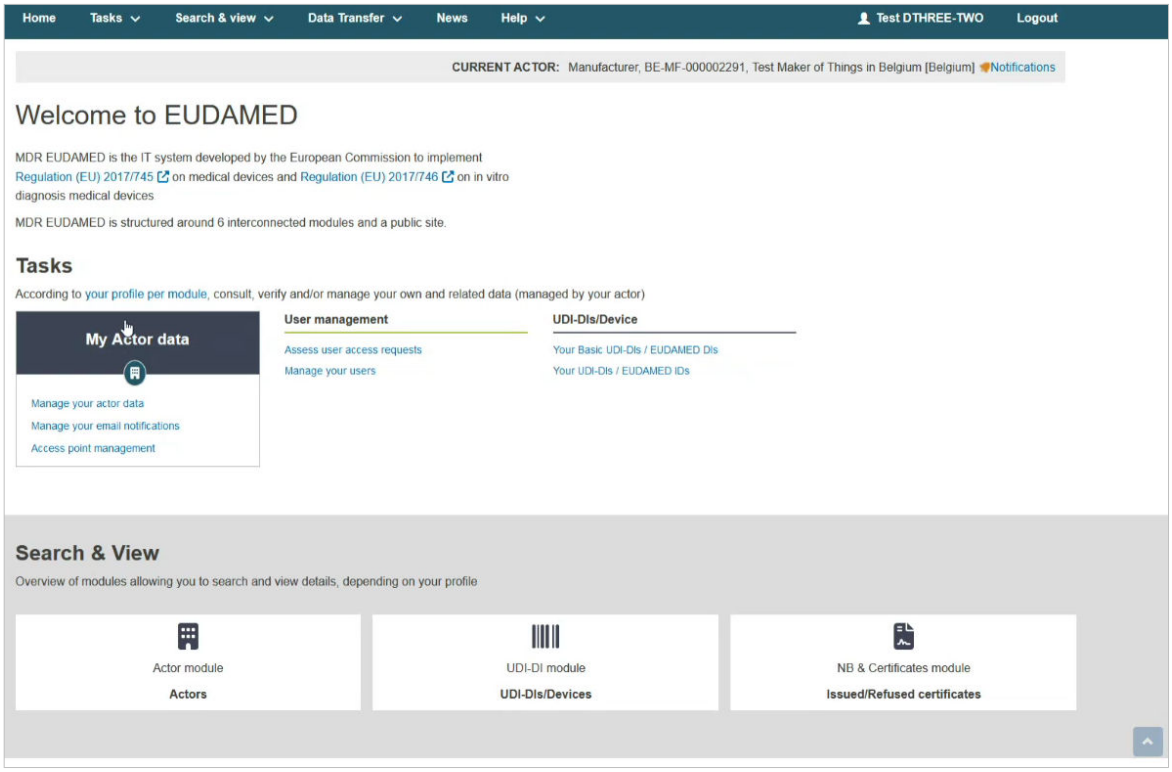


IMPORTANT
Please copy the security key. You will not be able to view your security key after closing the pop-up window. If you lose your security key, you must regenerate it (see Section Regenerate your security key [34] of the current guide for further details).

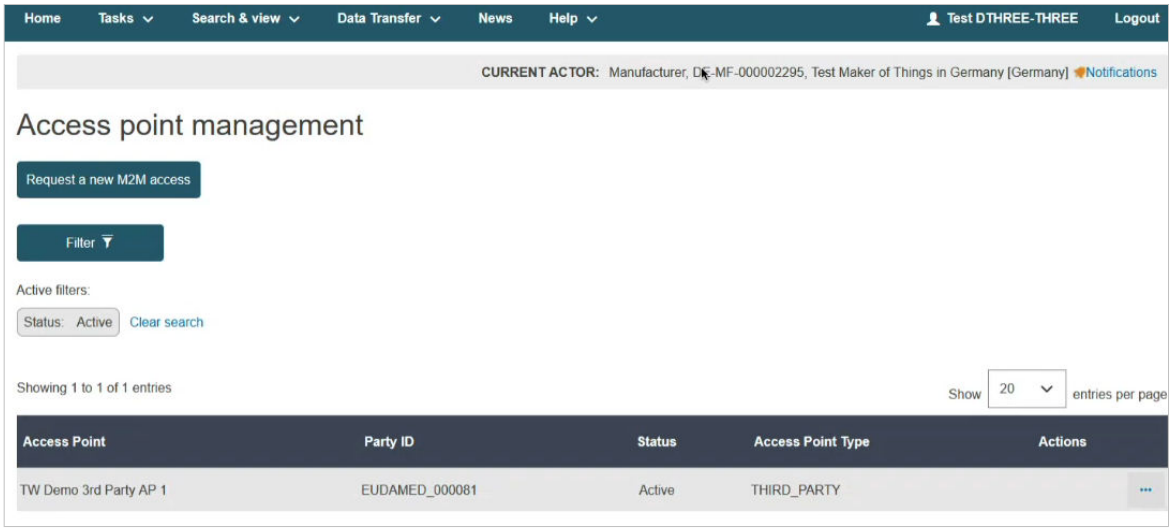
IMPORTANT
Go to How to configure your AP [36] to proceed with the onboarding.

5.1.7 Regenerate your security key

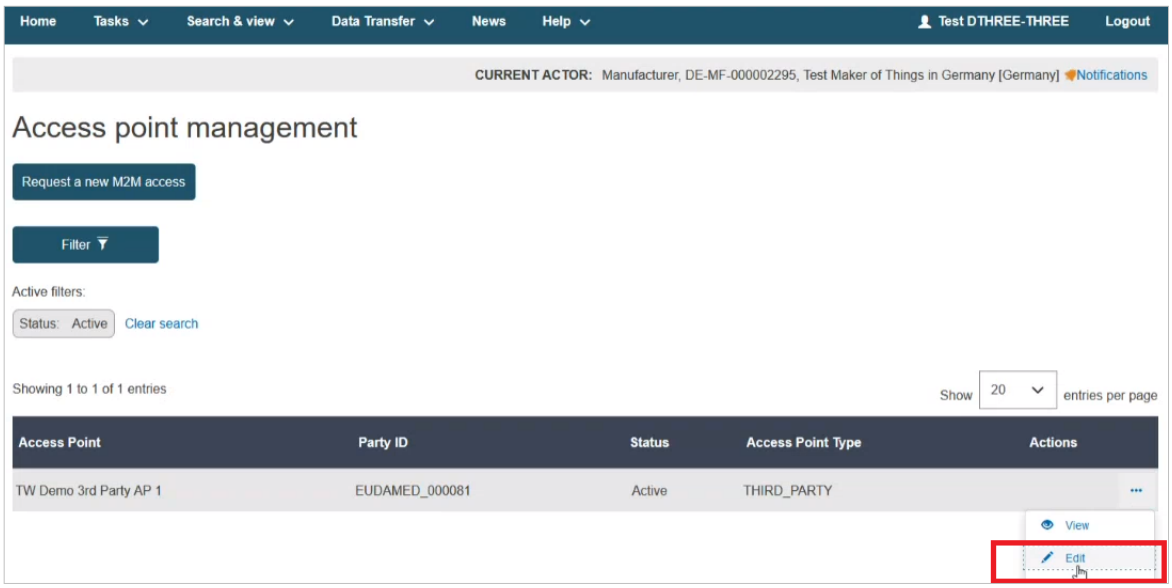
1. Log into EUDAMED as a Local Actor Administrator (LAA). Click on the *Access point management* link under the *My Actor data* section to view your APs:



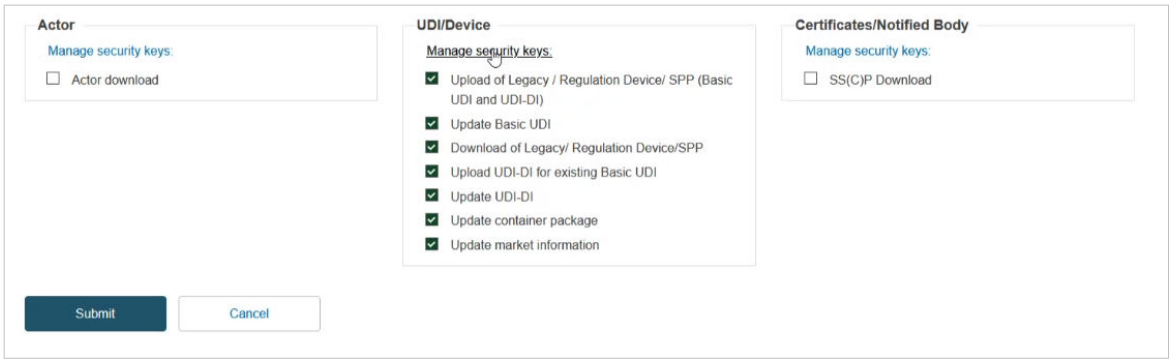
2. In the *Access Point management* page, you can view all your APs listed in the table:



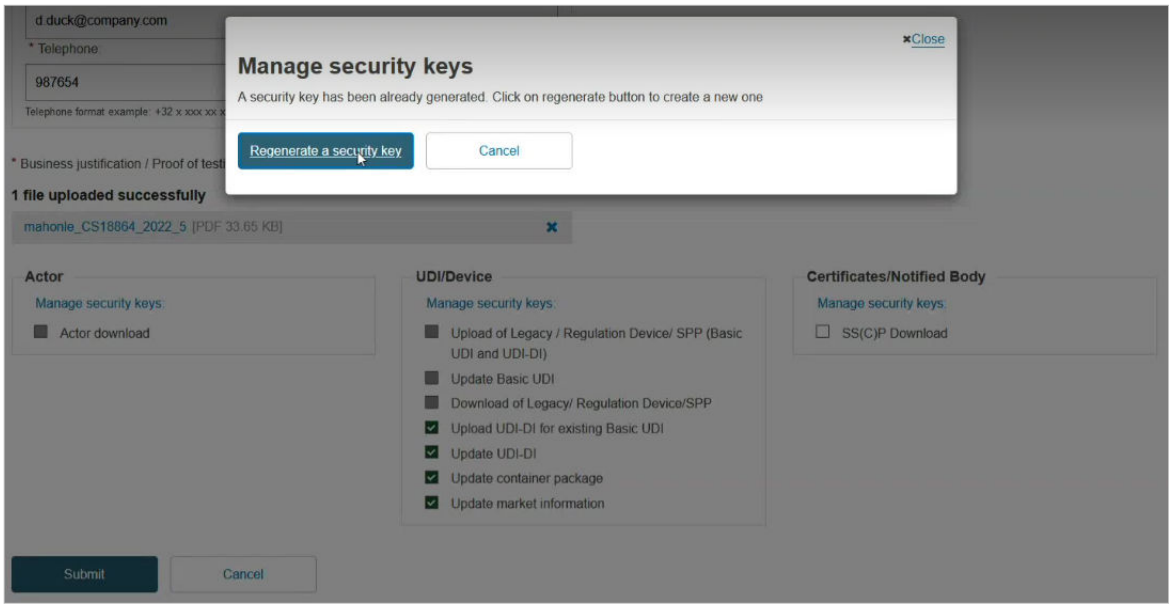
3. Click on the *Edit* link under the three dots to view your AP's settings:



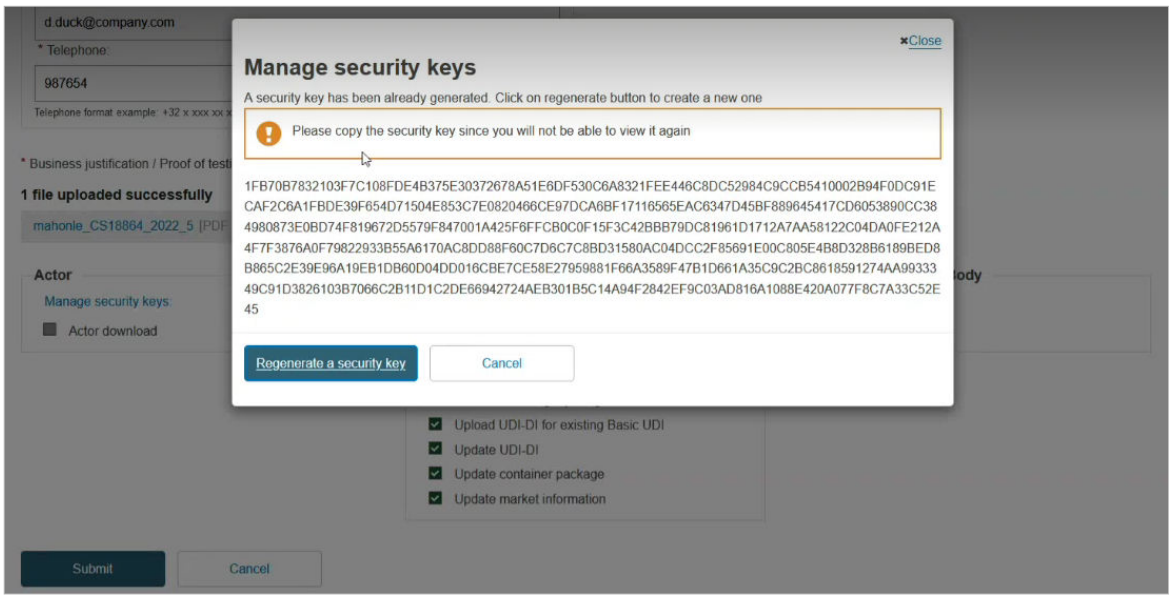
4. In the displayed screen click on the *Manage security keys* link:



5. In the pop-up window click on the **Regenerate a security key** button to regenerate the security key for the selected module:



6. In the pop-up window you will be able to view the regenerated security key:



IMPORTANT
Please copy the regenerated security key. You will not be able to view your security key after closing the pop-up window. If you lose your security key, you must regenerate it.

NOTE
When a security key is regenerated, all new XML files must include the new security key for that module, otherwise they will be rejected.

5.2 How to configure your AP

After submitting your request (the status of the request should be *connectivity under validation*) for M2M services, you must follow the steps listed below in order to complete the onboarding:

1. Download the Technical Onboarding template here.
2. Fill in the Technical Onboarding template given the instructions described in the document.
3. Send the filled in document via email to the EUDAMED support team.

EUDAMED support team will process your request and will send you an email message consisting of further useful information that will help you configure your AP and complete the onboarding.

NOTE
Only after completing the steps described in the email message that you will receive by EUDAMED support team your AP will be active.

IMPORTANT
Only after your AP is active you will be able to generate your security keys. See Generate your security key [31] for further information.

6 M2M data exchange process

6.1 General concepts and terminology

It is worth mentioning that the concepts listed below will assist you in constructing the XML files that will be used for the M2M data exchange (see Section Technical documentation [56] of the current guide for further details):

- **Entity model**: Represents the canonical model of business entities (e.g. actor, actor registration, device, certificate, etc.)
- **Service model**: Models how data is shaped in order to be transported, encapsulated and delivered to the right service
- **Data exchange communication patterns**: Defines how the data is exchanged between participants
- **Service definition**: Describes how to represent and access the business capability.

Entity model

Most main business entities contain the abstract superclass Entity to express the versioning capability.

Generally, most of the entities exchanged (through the exposed services) are interrelated (have dependencies between them). To model this, a concept of link has been introduced. In this way, it is possible to isolate the information related to a specific sub-domain and only reference to the other related entities.

Service model

The service model describes the EUDAMED data structures used to perform information exchange and consists of the following elements:

- Information about the type of message (from the data exchange messaging patterns pull/push)
- Service destination information (each operation will represent a specific functionality)
- Means to query / filter (criteria) the request
- Security rules that may apply depending on the actor that performs it
- Information available as response, number of provided entities, pagination and versioning capabilities.

Supported services operation

The following services operation are supported by EUDAMED:

1. **GET**: The GET method requests transfer of a current selected representation for the target resource. GET is the primary mechanism of information retrieval and the focus of almost all performance optimisations. Hence, when people speak of retrieving some identifiable information via HTTP, they are generally referring to making a GET request (PULL message type).

2. **POST**: The POST method requests that the target resource process the representation enclosed in the request according to the resource's own specific semantics (PUSH message type).

3. **PUT**: The PUT method requests that the state of the target resource be created or replaced with the state defined by the representation enclosed in the request message payload. A successful PUT of a given representation would suggest that a subsequent GET on that same target resource will result in an equivalent representation being sent in a 200 (OK) response. However, there is no guarantee that such a state change will be observable, since the target resource might be acted upon by other user agents in parallel, or might be subject to dynamic processing by the origin server, before any subsequent GET is received. Thus, a PUT request always contains a full resource. This is necessary because, a necessary quality of PUT requests is idempotence – the quality of producing the same result even if the same request is made multiple times (PUSH message type).

4. **PATCH**: The PATCH method requests that a set of changes described in the request entity be applied to the resource identified by the Request. With PATCH, however, the enclosed entity contains a set of instructions describing how a resource currently residing on the origin server should be modified to produce a new version. The PATCH method affects the resource identified by the Request-URI, and it also may have side effects on other resources; i.e., new resources may be created, or existing ones modified, by the application of a PATCH. To ensure an idempotent behaviour, clients using this kind of patch application should use a conditional request such that the request will fail if the resource has been updated since the client last accessed the resource (PUSH message type).

More information for each operation is presented in the table below.

Operation	Description	Supported communication patterns	Payload	Return value
Download (GET)	This operation is invoked to request information using a criteria mechanism.	Pull	Search criteria	A PullResponse type message containing a list of the entity(ies) and all related fields based on the criteria specified in the request message
Upload (POST)	This operation is invoked to upload (create) information to EUDAMED.	Push	For a specific entity fields to be filled to create the entity	Acknowledgement
Update (PUT/ PATCH)	This operation is invoked to update information to EUDAMED.	Push	For a specific entity fields to be updated	Acknowledgement

6.2 Message exchange

SOAP messages will be used for data exchange. An example of a SOAP message is presented in the following image:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:ns="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
               xmlns:_1="http://org.ecodex.backend/1_1/">
    <soap:Header>
        <ns:Messaging>
            <ns:UserMessage>
                <ns:PartyInfo>
                    <ns:From>
                        <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered:EUDAMED">domibus-blue</ns:PartyId>
                        <ns:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</ns:Role>
                    </ns:From>
                    <ns:To>
                        <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered:EUDAMED">domibus-blue</ns:PartyId>
                        <ns:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</ns:Role>
                    </ns:To>
                </ns:PartyInfo>
                <ns:CollaborationInfo>
                    <ns:Service type="e-delivery">http://ec.europa.eu/e-delivery/services/eudamed</ns:Service>
                    <ns:Action>submitMessage</ns:Action>
                </ns:CollaborationInfo>
                <ns:MessageProperties>
                    <ns:Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C1</ns:Property>
                    <ns:Property name="finalRecipient">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C4</ns:Property>
                </ns:MessageProperties>
                <ns:PayloadInfo>
                    <ns:PartInfo href="cid:eudamedxmlAttachment">
                        <ns:PartProperties>
                            <ns:Property name="MimeType">text/xml</ns:Property>
                        </ns:PartProperties>
                    </ns:PartInfo>
                </ns:PayloadInfo>
            </ns:UserMessage>
        </ns:Messaging>
    </soap:Header>
    <soap:Body>
        <_1:submitRequest>
            <payload payloadId="cid:eudamedxmlAttachment" contentType="text/xml">
                <value>
                    PG1lc3NhZ2U6UHVsbFJlcXVlc3QgdmVyc2lvbj0iMi4wIgoJeG1sbnM6bWVzc2FnZT0iaHR0cHM6Ly9lYy5ldXJvcGEuZXUvdG9vbHMvZXVkYW1lZC9kdHgvc2Vydml
                </value>
            </payload>
        </_1:submitRequest>
    </soap:Body>
```

IMPORTANT
The entire XML message must be base64 encoded. The payload element contains the encoded message. The highlighted line shows the encoded message.

A basic message structure would consist of the following main elements:

```
<PullRequest/PullResponse/Acknowlegment>
    .. ..
  <messageID>
    .. ..
  <sender>
    <node>
    <service>
  <recipient>
    <node>
    <service>
  <payload>
    <Entities xsi:type="MFActorType"</Entities>
    <Entities xsi:type="MFActorType"</Entities>
</PullRequest/PullResponse/Acknowlegment>
```

Messaging (MessageType)

Message types and related properties, such as:

1. **Message type (Push, PullRequest, PullResponse, Acknowledge)**: Identification, description of the root of the messages

2. **Payload (Criteria Payload)**: Canonical entities data models and related attached resources (e.g. attachments, etc.), or querying criteria

3. **Capabilities requested/fulfilled metadata**: How the information should be provided to the requester (e.g., max. delay for the request, query type, max number of entities to be provided, pagination, etc.)

4. **Reports (ElementReportsType)**: How the reports should be submitted with the processing / request status in case of an initial Push or PullRequest messages.

Service definition (ServiceType)

Description of the services provided by EUDAMED:

1. **serviceID**: A unique identifier of a service in EUDAMED
2. **serviceOperation**: Supported by the service (e.g. download, upload, update, etc.)
3. **serviceAccessToken**: Bearer security authentication token (security key) to gain access permission to the requested data by the requester as EUDAMED actor or third party (acting on behalf of the actor mentioned in nodeCode)
4. **serviceVersion**: In case of multi-versioned compatible service, it allows to specify what version of the current service is invoked.

Addressing (NodeType)

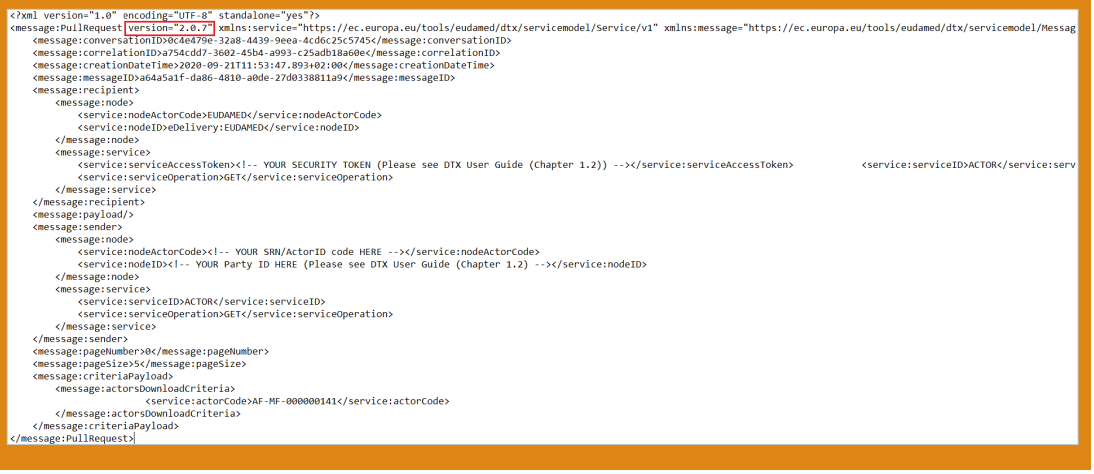
Logical networking information related to the actors and AP used in DTX process:

1. **nodeCode**: Contains the EUDAMED code (e.g. SRN, CA identifier, NB code etc.) of the party that performs the call of service. In case of multi-profile endpoint (e.g. third-party companies), it contains the actor code on behalf of the request that is performed.
2. **nodeID**: Identify the requester (actor/third-party integrators) of the message eDelivery endpoint, it contains the partyID of the requester.

IMPORTANT

In all your XML files you must include the version of the XSD schema that was used. You can visit EUDAMED Information Centre to check the latest version of the XSD schema in use. The version number included in your XML files must match the latest version number of the XSD schema in use, otherwise your XML files will be rejected.

The following figure displays the place that the version number must be included in the XML file.



6.3 Service definition

The service definition contains information about types of messages (from the data exchange messaging patterns) compatible, available operations at service level (each

operation will represent a specific functionality), means to query / filter (criteria) the request, security rules that may apply depending on the actor that perform it and finally the information available as response, eventually number of provided entities, pagination and versioning capabilities. For further details see the Service definitions documents in the Technical Documentation page of the Information Centre.

7 Example of the process

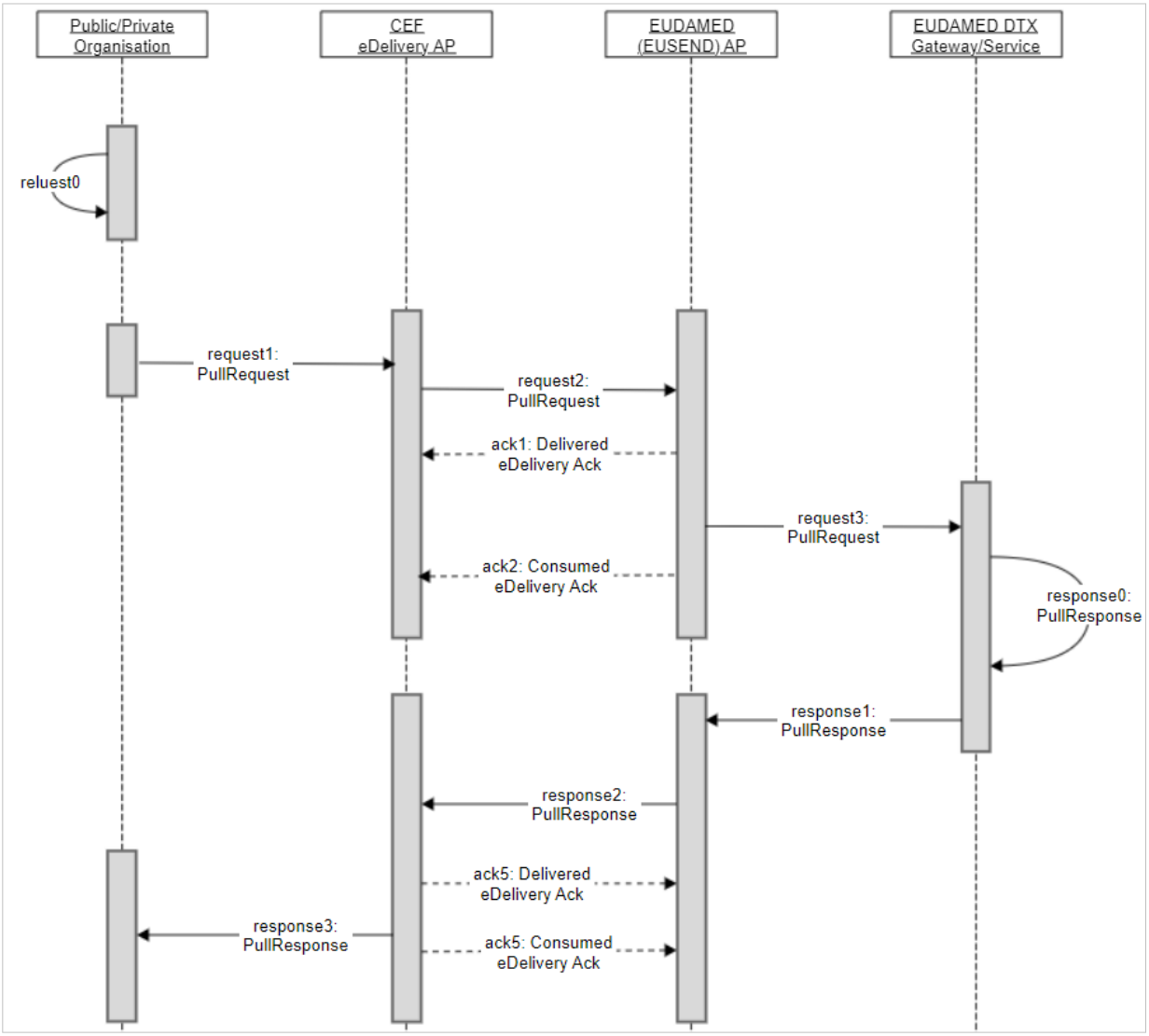
7.1 PULL

Scenario

In this example, a user of a public/private organisation with a CA profile requests from EUDAMED a list with all registered devices for the Manufacture with SRN BE-MF-000001201.

Process

The following image illustrates the message exchange between EUDAMED and the public/private organisation which performs the request.



More specifically, the following steps take place in the above-mentioned scenario:

1. A *PullRequest* message (*request0*) is created by the organisation containing the attributes listed below:
   a. messageID: A unique identifier, issued by the requester

   b. correlationID: An identifier that will correlate the request to the response or to the acknowledgements, issued by the requester

   c. sender/service/ServiceID: An identifier of the callback service (for responses and acknowledgements)

   d. sender/node/nodeCode: An identifier of the EUDAMED unique number of the requester (e.g. SRN, CA number, etc.)

   e. sender/node/nodeID: An identifier of the eDelivery partyID

   f. recipient/service/serviceID: An identifier of the EUDAMED service

   g. recipient/service/serviceOperation: An identifier of the EUDAMED service operation to uniquely define the service scope

   h. recipient/service/serviceAccessToken: The EUDAMED bearer security token attached to the requester service

   i. pageNumber: In case of paginated response, the requester can orchestrate multiple page response and request for a specific page to be provided in the response

      NOTE
      The first page is page number ‘0’.

   j. pageSize: The required maximum number of entities on a specific response page

   k. payload/Entities: It may contain the main service entity in case of a query by example criteria

2. The *PullRequest* message (*request1*), is sent by the organisation’s backend to the organisation’s eDelivery AP. As mentioned in Message exchange [40] Section of the current guide, the *request0* is encoded and embedded into the payload of a SOAP envelope. The following attributes are included in the message:
   a. envelope/header/messaging/usermessage/partyinfo/from/partyID: A unique identifier of the requester’s eDelivery AP

   b. envelope/header/messaging/usermessage/partyinfo/to/partyID: A unique identifier of the EUDAMED eDelivery AP

   c. envelope/body/submitRequest/payload: It holds the base64 format of the *request0* message

3. The *PullRequest* message (*request2*), is sent from the organisation's eDelivery AP to EUDAMED's eDelivery AP. The status of the message is now SENDING.

4. The EUDAMED eDelivery AP sends an *Acknowledge* message (*ack1*) to the organisation's eDelivery AP which implies that *request2* was successfully received by the EUDAMED eDelivery AP. The status of the message is now DELIVERED.

5. The *PullRequest* message (*request3*), is sent from EUDAMED'S eDelivery AP to EUDAMED's backend. The status of the message is now SENDING.

6. The EUDAMED eDelivery AP sends again an *Acknowledge* message (*ack2*) to the organisation's eDelivery AP which implies that *request3* was successfully received by the EUDAMED eDelivery AP. The status of the message is now DELETED.

7. A *PullResponse* message (*response0*) is created by EUDAMED containing the following attributes:
   a. messageID: A unique identifier, issued by submitter

   b. correlationID: The same as correlationID from the request message

   c. responseCode: Status code of the service call (success or matching error code)

   d. sender/service/ServiceID: An identifier of the EUDMAED initiator service

   e. sender/node/nodeCode: An identifier of the EUDAMED eDelivery party identifier

   f. sender/node/nodeID: An identifier of the eDelivery partyID

   g. recipient/service/serviceID: The requester's generic callback service

   h. recipient/service/serviceOperation: The requester's callback service operation (as specified in the request message)

   i. payload/entities: It contains the list of entities of the same type that were collected through the service invocation

8. The *PullResponse* message (*response1*), is sent by EUDAMED's backend to EUDAMED's eDelivery AP. As mentioned in Message exchange [40] Section of the current guide, the *response0* is encoded and embedded into the payload of a SOAP envelope. The following attributes are included in the message:
   a. envelope/header/messaging/usermessage/partyinfo/from/partyID: A unique identifier of the requester's eDelivery AP

   b. envelope/header/messaging/usermessage/partyinfo/to/partyID: A unique identifier of the EUDAMED eDelivery AP

   c. envelope/body/submitRequest/payload: It holds the base64 format of the *response0* message

9. The *PullResponse* message (*response2*), is sent by EUDAMED's eDelivery AP to the organisation's eDelivery AP.

10. The organisation's eDelivery AP sends an *Acknowledge* message (*ack3*) to the EUDAMED eDelivery AP which implies that *response2* was successfully received by the organisation's eDelivery AP. The status of the message is now DELIVERED.

11. The *PullResponse* message (*response3*), is sent from the organisation's eDelivery AP to organisation's backend. The status of the message is now SENDING.

12. The organisation's eDelivery AP sends again an *Acknowledge* message (*ack3*) to the EUDAMED eDelivery AP which implies that *response3* was successfully received by the organisation's eDelivery AP. The status of the message is now DELETED.

Messages

Given the above-mentioned scenario, the following messages will be created:

Request0

```
<message:PullRequest
xmlns:device="https://ec.europa.eu/tools/eudamed/dtx
    /datamodel/Entity/Device/v1"
xmlns:party="https://ec.europa.eu/tools/eudamed/dtx
    /datamodel/Party/v1">
    <message:correlationID>a754cdd7-3602-45b4-a993-c25adb18a60e
    </message:correlationID>
    <message:creationDateTime>2019-05-22T06:58:45.223+02:00
    </message:creationDateTime>
    <message:messageID>a64a5a1f-da86-4810-a0de-27d0338811a9
    </message:messageID>
    <message:recipient>
        <message:node>
            <service:nodeActorCode>EUDAMED
            </service:nodeActorCode>
            <service:nodeID>eDelivery:EUDAMED
            </service:nodeID>
        </message:node>
        <message:service>
          <service:serviceAccessToken>343452423422523423423 4
          </service:serviceAccessToken>
          <service:serviceID>DEVICE</service:serviceID>
          <service:serviceOperation>GET
          </service:serviceOperation>
        </message:service>
    </message:recipient>
    <message:payload/>
    <message:sender>
        <message:node>
            <service:nodeActorCode>CA-BE-000000555
            </service:nodeActorCode>
            <service:nodeID>eDelivery:CA-BE-000000555
            </service:nodeID>
```

```
        </message:node>
        <message:service>
            <service:serviceID>REPLY_SERVICE</service:serviceID>
            <service:serviceOperation>GET</service:serviceOperation>
        </message:service>
    </message:sender>
    <message:pageNumber>1</message:pageNumber>
    <message:pageSize>10</message:pageSize>
    <message:criteriaPayload>
       <message:diDownloadCriteria>
            <service:MFActorCode>BE-MF-000001201</service:MFActorCode>
       </message:diDownloadCriteria>
    </message:criteriaPayload>
</message:PullRequest>
```

Response0

```
<message:PullResponse
xmlns:device="https://ec.europa.eu/tools/eudamed/dtx
    /datamodel/Entity/Device/v1">
    <message:correlationID>a754cdd7-3602-45b4-a993-c25adb18a60e
    </message:correlationID>
    <message:creationDateTime>2019-05-22T06:43:12.459+02:00
    </message:creationDateTime>
    <message:messageID>a64a5a1f-da86-4810-a0de-27d0338811a9
    </message:messageID>
    <message:recipient>
        <message:node>
            <service:nodeActorCode>AC-BE-000000555
            </service:nodeActorCode>
            <service:nodeID>eDelivery:AC-BE-000000555
            </service:nodeID>
        </message:node>
        <message:service>
            <service:serviceID>REPLY_SERVICE</service:serviceID>
            <service:serviceOperation>GET</service:serviceOperation>
        </message:service>
    </message:recipient>
    <message:payload>
        <device:Device xsi:type="device:IVDRDeviceType"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <device:IVDRBasicUDI>
                .......
                <budi:modelName>
                    <budi:model>1001/202</budi:model>
                    <budi:name>Clear-View Sub-Q</budi:name>
                </budi:modelName>
                <budi:MFActorCode>BE-MF-000001201</budi:MFActorCode>
            </device:IVDRBasicUDI>
```

```
            <device:IVDRUDIDIData>
                <udidi:identifier>
                    <udidi:DICode>UDICODE1</udidi:DICode>
                    <udidi:issuingEntityCode>GS1
                    </udidi:issuingEntityCode>
                </udidi:identifier>
                .........
            </device:IVDRUDIDIData>
        </device:Device>
        <device:Device xsi:type="device:MDRDeviceType"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <device:MDRBasicUDI>
                ................
                <budi:riskClass>IIb</budi:riskClass>
                <budi:model>SCE8-03-05</budi:model>
                <budi:identifier>
                    <udidi:DICode>M991CVS1277777</udidi:DICode>
                    <udidi:issuingEntityCode>GS1
                    </udidi:issuingEntityCode>
                </budi:identifier>
                <budi:MFActorCode>BE-MF-000001201
                </budi:MFActorCode>
                <budi:applicableLegislation>MDR
                </budi:applicableLegislation>
                ..............
            </device:MDRBasicUDI>
            <device:MDRUDIDIData>
                ..............
            </device:MDRUDIDIData>
        </device:Device>
    </message:payload>
    ......
    <message:maxPageNumber>1</message:maxPageNumber>
    <message:pageNumber>1</message:pageNumber>
    <message:pageSize>10</message:pageSize>
    <message:responseCode>SUCCESS</message:responseCode>
</message:PullResponse>
```

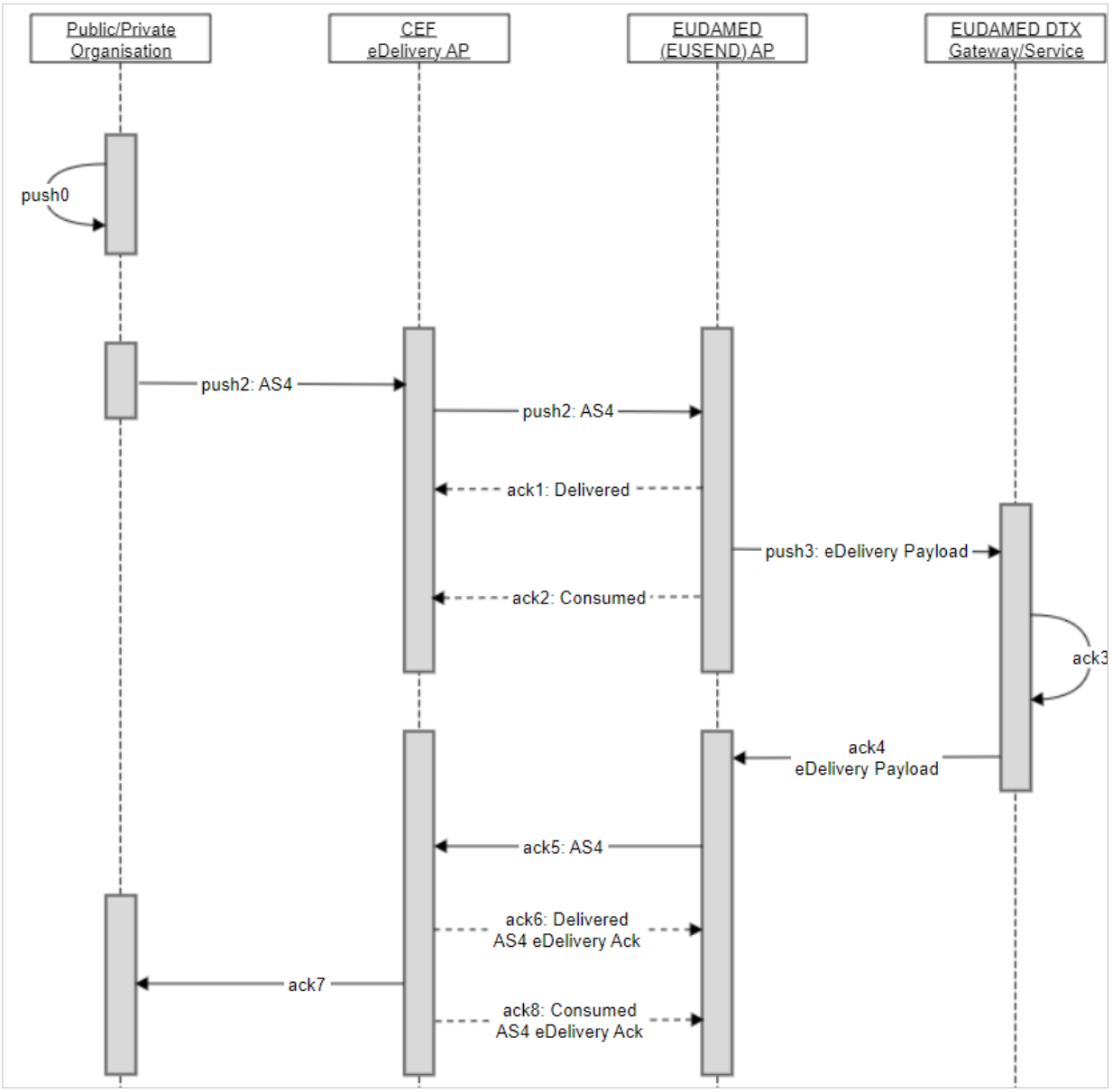
7.2 PUSH

Scenario

In this scenario a user of a public/private organisation with a non-EU Manufacturer profile uploads the organisation’s devices in EUDAMED.

Process

The following image displays the message exchange between EUDAMED and the public/private organisation which performs the upload.



More specifically, the following steps take place in the above-mentioned scenario:

1. A *Push* message (*push0*) is created by the organisation containing the following attributes:
   a. messageID: A unique identifier, issued by the requester

   b. correlationID: An identifier that will correlate the request to the response or to the acknowledgements, issued by the requester

   c. sender/service/ServiceID: An identifier of the callback service (for responses and acknowledgements)

   d. sender/node/nodeCode: An identifier of the EUDAMED unique number of the requester (e.g.SRN, CA number, etc.)

   e. sender/node/nodeID: An identifier of the eDelivery partyID

f. sender/node/nodeProfileToken: The EUDAMED bearer security token attached to the requester

g. recipient/service/serviceID: An identifier of the EUDAMED service

h. recipient/service/serviceOperation: An identifier of the EUDAMED service operation to uniquely define the service scope

i. payload/Entities: It contains the main service accepted entity

2. The *Push* message (*push1*) is sent from the organisation's backend to the organisation's eDelivery AP. As mentioned in Message exchange [40] Section of the current guide, the *push0* message is encoded and embedded into the payload of a SOAP envelope. The following attributes are included in the message:
   a. envelope/header/messaging/usermessage/partyinfo/from/partyID: A unique identifier of the organisation's eDelivery AP

   b. envelope/header/messaging/usermessage/partyinfo/to/partyID: A unique identifier of the EUDAMED eDelivery AP

   c. envelope/body/submitRequest/payload: It holds the base64 format of the *push0* message

3. The *Push* message (*push2*) is sent from the organisation's eDelivery AP to the EUDAMED eDelivery AP. The status of the message is now SENDING.

4. An *Acknowledge* message (*ack1*) is sent from the EUDAMED eDelivery AP to the organisation's eDelivery AP which implies that *push2* was successfully received by the EUDAMED eDelivery AP. The status of the message is now DELIVERED.

5. The *Push* message (*push3*) is sent from the EUDAMED eDelivery AP to EUDAMED's backend.

6. An *Acknowledge* message (*ack2*) is sent from the EUDAMED eDelivery AP to the organisation's eDelivery AP which implies that *push2* was successfully received by the EUDAMED eDelivery AP. The status of the message is now DELETED.

7. An *Acknowledge* message (*ack3*) is created by EUDAMED containing the following attributes:
   a. messageID: A unique identifier, issued by the requester

   b. correlationID: The same as correlationID from the request message

   c. sender/service/ServiceID: An identifier of the EUDMAED initiator service

   d. sender/node/nodeCode: An identifier of the EUDAMED eDelivery party identifier

   e. sender/node/nodeID: An identifier of the eDelivery partyID

   f. recipient/service/serviceID: The requester's generic callback service

g. recipient/service/serviceOperation: The requester's callback service operation (as specified in the request message)

h. ackCode: The status code of the service call (success or matching error code)

i. payload/report: It contains a report of processing statuses and details for each entity that have been sent through the Push message.

8. An *Acknowledge* message (*ack4*) is sent from the EUDAMED backend to the EUDAMED eDelivery AP with the attributes listed below:
   a. envelope/header/messaging/usermessage/partyinfo/to/partyID: A unique identifier of the requester's eDelivery AP

   b. envelope/header/messaging/usermessage/partyinfo/from/partyID: A unique identifier of the EUDAMED eDelivery AP

   c. envelope/body/submitRequest/payload: It holds the base64 format of the *ack0* message.

9. An *Acknowledge* message (*ack5*) is sent from the EUDAMED eDelivery AP to the organisation's eDelivery.

10. An *Acknowledge* message (*ack6*) is sent from the organisation's eDelivery AP to the EUDAMED eDelivery AP. The status of the message is now DELIVERED.

11. An *Acknowledge* message (*ack7*) is sent from the organisation's eDelivery AP to the organisation's backend. As mentioned in Message exchange [40] Section of the current guide, the *ack7* message is encoded and embedded into the payload of a SOAP envelope. The following attributes are included in the message:
    a. envelope/header/messaging/usermessage/partyinfo/from/partyID: A unique identifier of the organisation's eDelivery AP

    b. envelope/header/messaging/usermessage/partyinfo/to/partyID: A unique identifier of the EUDAMED eDelivery AP

    c. envelope/body/submitRequest/payload:It holds the base64 format of the *push0* message

12. An *Acknowledgment* message (*ack8*) is sent from the organisation's eDelivery AP to the organisation's backend. The status of the message is now DELETED.

Messages

Given the above-mentioned scenario, the following messages will be created:

Push0

```
<message:Push>
    <message:correlationID>a754cdd7-3602-45b4-a993-c25adb18a60e
    </message:correlationID>
    <message:creationDateTime>2019-05-22T07:00:28.066+02:00
```

```
</message:creationDateTime>
<message:messageID>a64a5a1f-da86-4810-a0de-27d0338811a9
</message:messageID>
<message:recipient>
    <message:node>
        <service:nodeActorCode>EUDAMED</service:nodeActorCode>
        <service:nodeID>eDelivery:EUDAMED</service:nodeID>
    </message:node>
    <message:service>
        <service:serviceAccessToken>943452423422523423239
        </service:serviceAccessToken>
        <service:serviceID>DEVICE</service:serviceID>
        <service:serviceOperation>POST</service:serviceOperation>
    </message:service>
</message:recipient>
<message:payload>
    <device:Device xsi:type="device:IVDRDeviceType"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <device:IVDRBasicUDI>
            ..............................
            <budi:ARActorCode>BE-AR-000033010
            </budi:ARActorCode>
            <budi:MFActorCode>JP-MF-000033020
            </budi:MFActorCode>
            <budi:specialDevice>ORTHOPAEDIC</budi:specialDevice>
            <budi:applicableLegislation>IVDD
            </budi:applicableLegislation>
            ..................................
        </device:IVDRBasicUDI>
        <device:IVDRUDIDIData>
            <udidi:identifier>
                <udidi:DICode>UDICODE1</udidi:DICode>
                <udidi:issuingEntityCode>GS1
                </udidi:issuingEntityCode>
            </udidi:identifier>
            ........................................
        </device:IVDRUDIDIData>
    </device:Device>
    <device:Device xsi:type="device:MDRDeviceType"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <device:MDRBasicUDI>
            <budi:riskClass>IIb</budi:riskClass>
            <budi:model>SCE8-03-05</budi:model>
            <budi:identifier>
                <udidi:DICode>M991CVS1277777</udidi:DICode>
                <udidi:issuingEntityCode>GS1
                </udidi:issuingEntityCode>
            </budi:identifier>
```

```
                <budi:ARActorCode>BE-AR-000000077</budi:ARActorCode>
                <budi:MFActorCode>JP-MF-000033020</budi:MFActorCode>
                <budi:applicableLegislation>MDR
                </budi:applicableLegislation>
                ........................................
            </device:MDRBasicUDI>
            <device:MDRUDIDIData>
                ..............................
            </device:MDRUDIDIData>
        </device:Device>
    </message:payload>
    <message:sender>
        <message:node>
            <service:nodeActorCode>JP-MF-000033020
            </service:nodeActorCode>
            <service:nodeID>eDelivery:JP-MF-000033020
            </service:nodeID>
        </message:node>
        <message:service>
            <service:serviceID>REPLY_SERVICE</service:serviceID>
            <service:serviceOperation>GET</service:serviceOperation>
        </message:service>
    </message:sender>
</message:Push>
```

Ack3

```
<message:Acknowledgement>
    <message:conversationID>0c4e479e-32a8-4439-9eea-4cd6c25c5745
    </message:conversationID>
    <message:correlationID>a754cdd7-3602-45b4-a993-c25adb18a60e
    </message:correlationID>
    <message:creationDateTime>2019-05-29T00:53:47.165+02:00
    </message:creationDateTime>
    <message:messageID>a64a5a1f-da86-4810-a0de-27d0338811a9
    </message:messageID>
    <message:recipient>
        <message:node>
            <service:nodeActorCode>JP-MF-000033020
            </service:nodeActorCode>
            <service:nodeID>eDelivery:JP-MF-000033020
            </service:nodeID>
        </message:node>
        <message:service>
            <service:serviceID>REPLY_SERVICE</service:serviceID>
            <service:serviceOperation>GET</service:serviceOperation>
        </message:service>
    </message:recipient>
    <message:payload/>
```

```
    <message:sender>
        <message:node>
            <service:nodeActorCode>EUDAMED</service:nodeActorCode>
            <service:nodeID>eDelivery:EUDAMED</service:nodeID>
        </message:node>
        <message:service>
            <service:serviceID>DEVICE</service:serviceID>
            <service:serviceOperation>GET</service:serviceOperation>
        </message:service>
    </message:sender>
    <message:responseCode>SUCCESS</message:responseCode>
</message:Acknowledgement>
```

8 Support

8.1 EUDAMED team

You can contact the EUDAMED support team by sending an email message to SANTE-EUDAMED-SUPPORT@ec.europa.eu for the services listed below:

1. M2M onboarding
2. Generating/regenerating the security key
3. XSD/XML validation and error responses
4. Data dictionary queries
5. Bug reporting and feature suggestions.

8.2 DIGIT AP team

You can contact the DIGIT AP support team by sending an email message to EC-EDELIVERY-SUPPORT@ec.europa.eu for the services listed below:

1. Access point connection (once approved by EUDAMED)
2. Public Key Infrastructure (PKI) queries
3. Pmode configuration
4. Advice on AS4 conformant solutions
5. Configurations for getting the message to EUDAMED.

8.3 Technical documentation

The following table summarises the content of the supporting documents that can be found in the EUDAMED Information Centre:

Document	Content
DTX XSD	The XML Schema Definition for the available services and entities
DTX XML samples	Bulk upload/download samples covering the most used scenarios per service. These are general samples, not containing personalised data
DTX Service Definition	A business level view of the available service
DTX business rules	All constraints related to the services
Enumerations	Value lists for multiple choice fields
Data dictionaries	Explains the Entity structure and meta data about their attributes
Third-party agreement	Confirms the agreement between an Actor requesting an AP and a 3rd party service provider
Technical onboarding	Contains data that need to be filled in by the Actor requesting an AP for the onboarding procedure

8.4 Useful links

1. Domibus
2. Connecting Europe Facility (CEF) Digital eDelivery access point documentation
3. EUDAMED Information Centre
4. EUDAMED – European Medical Database on Medical Devices
5. DIGIT Access Point Support

9 Annexes

9.1 Annex 1 (Error codes)

GENERIC_INTERNAL_50000("50000"),
UNSUPPORTED_MESSAGE_TYPE_50001("50001"),
EDELIVERY_BRIDGE_READ_NULL_ID_50001("E-B-50001"),
INBOUND_VALIDATION_GENERIC_NULL_OBJECT_I_50001("I-50001"),
INBOUND_VALIDATION_DOWNLOAD_ACTOR_NULL_IMESSAGE_I_50002("I-50002"),
INBOUND_VALIDATION_ACTOR_DOWNLOAD_UNSUPPORTED_SRN_I_50003("I-50003"),
INBOUND_VALIDATION_DOWNLOAD_ACTOR_NULL_IMESSAGE_I_50004("I-50004"),
INBOUND_VALIDATION_CREATE_DEVICE_NULL_IMESSAGE_PAYLOAD_I_50005("I-50005"),
INBOUND_VALIDATION_WRONG_DOWNLOAD_ACTOR_PAYLOAD_TYPE_I_50006("I-50006"),
INBOUND_TRANSFORM_PULL_REQUEST_NULL_MESSAGE_I_50007("I-50007"),
INBOUND_TRANSFORM_PULL_REQUEST_ACTOR_DOWNLOAD_CRITERIA_WRONG_TYPE_MESSAGE_I_50008("I-50008"),
INBOUND_VALIDATION_DOWNLOAD_DEVICE_NULL_IMESSAGE_I_50009("I-50009"),
INBOUND_VALIDATION_DOWNLOAD_DEVICE_NULL_IMESSAGE_PAYLOAD_I_50010("I-50010"),
INBOUND_TRANSFORM_PULL_REQUEST_DEVICE_DOWNLOAD_CRITERIA_NULL_MESSAGE_I_50011("I-50011"),
INBOUND_TRANSFORM_PULL_REQUEST_ACTOR_DOWNLOAD_CRITERIA_WRONG_TYPE_MESSAGE_I_50012("I-50012"),
INBOUND_TRANSFORM_PULL_REQUEST_ACTOR_REGISTRATION__NULL_MESSAGE("I-50013"),
INBOUND_VALIDATION_INVALID_XML_PAYLOAD_I_40000("I-40000"),
INBOUND_VALIDATION_NULL_XML_PAYLOAD_I_40001("I-40001"),
INBOUND_VALIDATION_NULL_OBJECT_PAYLOAD_I_40002("I-40002"),
INBOUND_VALIDATION_SERVICE_ID_I_40003("I-40003"),
INBOUND_VALIDATION_NULL_ACTOR_DOWNLOAD_CRITERIA_PAYLOAD_I_40004("I-40004"),
INBOUND_VALIDATION_NULL_CRITERIA_PAYLOAD_I_40005("I-40005"),
INBOUND_VALIDATION_NULL_DEVICE_DOWNLOAD_CRITERIA_PAYLOAD_I_40006("I-40006"),
INBOUND_VALIDATION_NULL_INBOUND_PAYLOAD_I_40007("I-40007"),
OUTBOUND_TRANSFORM_OBJ_TO_XML_NULL_MESSAGE_O_50001("O-50001"),
OUTBOUND_TRANSFORM_OBJ_TO_XML_NULL_RESPONSE_MESSAGE_O_50002("O-50002"),
OUTBOUND_TRANSFORM_PULL_RESPONSE_ACTOR_DOWNLOAD_NULL_MESSAGE_O_50003("O-50003"),
OUTBOUND_TRANSFORM_PULL_RESPONSE_ACTOR_DOWNLOAD_WRONG_PAYLOAD_TYPE_O_50005("O-50005"),
OUTBOUND_TRANSFORM_PULL_RESPONSE_DEVICE_DOWNLOAD_NULL_MESSAGE_O_50006("O-50006"),
OUTBOUND_TRANSFORM_PULL_RESPONSE_DEVICE_DOWNLOAD_WRONG_PAYLOAD_TYPE_O_50007("O-50007"),
OUTBOUND_TRANSFORM_DOWNLOAD_DEVICE_IDENTIFIER_O_50002("E-O-50002"),
OUTBOUND_TRANSFORM_DOWNLOAD_ACTOR_ACTOR_TYPE_O_50010("O-50010"),
OUTBOUND_TRANSFORM_DOWNLOAD_ACTOR_ACTOR_TYPE_O_50011("O-50011"),
OUTBOUND_TRANSFORM_DOWNLOAD_ACTOR_ACTOR_TYPE_O_50012("O-50012"),
OUTBOUND_VALIDATION_NULL_MESSAGE_O_40000("O-40000"),

9.2 Annex 2 (XML files index)

Use case	Actors	XML file	Description	Result
Download registered actors	CA,	SAMPLE_DTX_ACT_001.01.xml	Search for an individual manufacturer SRN (EU and non- EU)	Positive

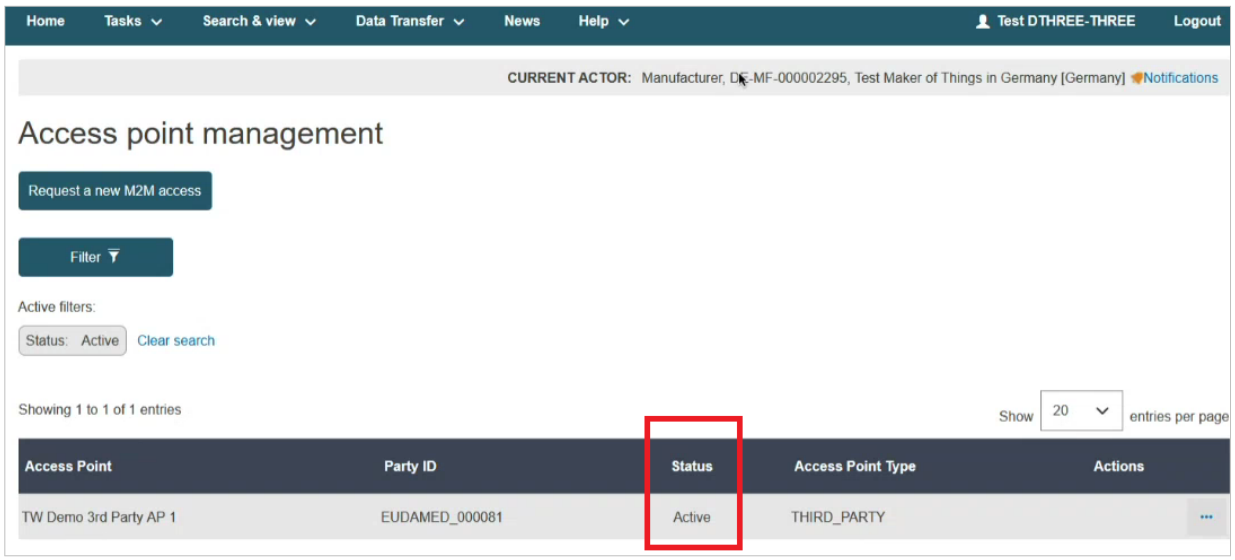
Use case	Actors	XML file	Description	Result
	EO, NB	SAMPLE_DTX_ACT_001.03.xml	Search for an individual importer	Positive
		SAMPLE_DTX_ACT_001.04.xml	Search for an individual authorised representative	Positive
		SAMPLE_DTX_ACT_001.05.xml	Search for an individual producer	Positive
		SAMPLE_DTX_ACT_001.07.xml	Request to download a registered economic operator using the wrong attribute type in the pull request	Negative
Download actor registration requests	CA	SAMPLE_DTX_ACT_002.01.xml	Download a specific application ID (includes Zip attachment)	Positive
		SAMPLE_DTX_ACT_002.02.xml	Request containing incorrect end tag – XSD invalid	Negative
		SAMPLE_DTX_ACT_002.03.xml	Request a specific initial submission date with a mix of EU and non-EU Producers (includes Zip attachments)	Positive
Upload actor registration request assessments	CA	SAMPLE_DTX_ACT_003.01.xml	Approve EU manufacturer	Positive
		SAMPLE_DTX_ACT_003.02.xml	Unknown APPLICATION-ID Provided	Negative
		SAMPLE_DTX_ACT_003.03.xml	One upload containing an approve and a correct assessment	Positive
		SAMPLE_DTX_ACT_003.04.xml	Negative assessment with reason: Duplicate	Negative
Download of Basic UDI-DI, UDI-DI and device data	CA, EO, NB	SAMPLE_DTX_UDI_004.01.xml	Request Download Device (BUDI and UDI-DI) information using a combination of the following criteria: Basic UDIDI Code, UDIDI code, MF actor code, AR actor code	Positive
		SAMPLE_DTX_UDI_004.06.xml	Request Download Device (BUDI and UDI-DI) information using a combination of the following criteria: Basic UDIDI Code, UDIDI code, MF actor code, AR actor code Criteria: Manufacturer (State is automatic Registered), Country	Positive
Upload Basic UDI-DI and UDI-DI information	EO	SAMPLE_DTX_UDI_001.01.xml	Register New MDR Device BUDI and Device (UDI-DI) information Class I Device (Special Device Type (Orthopedic), Substances, Direct Marking DI, Storage and Handling Conditions, Critical Warnings, Market Information, Clinical Sizes, Product Designer SRN)	Positive
		SAMPLE_DTX_UDI_001.02.xml	Register New MDR Device BUDI and 2 or more UDI-DI. (Special Device Type (Software), Clinical Investigations- EU and NonEU, Substances, Direct Marking DI, Storage and Handling Conditions, Critical Warnings, Market Information, Clinical Sizes, Product Designer SRN)	Positive
		SAMPLE_DTX_UDI_001.03.xml	Register New MDR Device BUDI and UDI-DI for a Risk Class III device to validate BUDI and UDI-DI will be stored in EUDAMED in “Submitted” status Class III (DeviceCertificateLinks, UnitofUseDI, Substances, Direct marking DI, Storage and Handling Conditions, Critical Warnings, Market Information, Clinical Sizes, Product Designer SRN)	Positive
		SAMPLE_DTX_UDI_001.04.xml	Register New MDR Device BUDI and UDI-DI for a Risk Class II A containing false data (e.g. containing IIb implantable exceptions – sutures) (IIb Implantable exception, Clinical Investigations- EU and NonEU, DeviceCertificateLinks, Substances, Storage and Handling Conditions, Critical Warnings, Market Information, Clinical Sizes, Product Designer SRN)	Negative
		SAMPLE_DTX_UDI_001.05.xml	Register New MDR Device BUDI and UDI-DI for a Risk Class II B containing certificate data and implantable and having IIb implantable exceptions – false Class IIB (IIB Implantable exceptions- false, DeviceCertificateLinks, Direct marking DI, Substances, Direct marking DI, Storage and Handling Conditions, Critical Warnings, Market Information, Clinical Sizes, Product	Positive
		SAMPLE_DTX_UDI_001.12.xml	Register New MDR Device BUDI and UDI-DI with multiple brand/trade names; ensure the “primary” attribute	Positive
		SAMPLE_DTX_UDI_001.13.xml	Register New MDR Device BUDI and UDI-DI with multiple packaging hierarchies (1st, 2nd, 3rd) and different packaging items on each level	Positive
		SAMPLE_DTX_UDI_002.01.xml	Register New IVDR Device BUDI and UDI-DI information Risk Class A (Storage and Handling Conditions, Critical Warnings, Market Information, Product Designer Organisation)	Positive
		SAMPLE_DTX_UDI_002.03.xml	Register New IVDR Device BUDI and UDI-DI for a Risk Class C device to see how the BUDI and UDI-DI will start in “Submitted” status and then transition to “Registered” status upon NB linking Risk Class C (DeviceCertificateLinks, Storage and Handling Conditions, Critical Warnings, Market Information, Product Designer Organisation)	Positive
		SAMPLE_DTX_UDI_003.01.xml	Register New System or Procedure Pack with only one Basic UDI	Positive
		SAMPLE_DTX_UDI_005.01.xml	Upload Legacy Device – Class I MDD Legacy Device	Positive
		SAMPLE_DTX_UDI_005.02.xml	Upload Legacy Device – AIMDD Legacy Device	Positive
		SAMPLE_DTX_UDI_005.03.xml	Upload Legacy Device – IVD_ANNEX_II_LIST_A Risk Class	Positive

Use case	Actors	XML file	Description	Result
Upload a new UDI under an existing Basic UDI-DI	EO	SAMPLE_DTX_UDI_008.02.xml	Register multiple New Device UDI-DI (MDR/IVDR) information related to an already existing (MDR/IVDR) Device B-UDI	Positive
Update Basic UDI-DI	EO	SAMPLE_DTX_UDI_009.01.xml	Update an existing MDR/IVDR Device BUDI	Positive
		SAMPLE_DTX_UDI_009.01.xml	Update an existing MDD Device BUDI	Positive
Update UDI-DI	EO	SAMPLE_DTX_UDI_010.01.xml	Update an existing MDR/IVDR Device UDI-DI	Positive
		SAMPLE_DTX_UDI_010.02.xml	Update an existing MDD Device UDI-DI	Positive
		SAMPLE_DTX_UDI_010.03.xml	Update the first EU country where the device has been put on the market	Positive
		SAMPLE_DTX_UDI_010.04.xml	Set the first EU country where the device has been placed on the market for a device that was not on the EU market at the time of its registration in EUDAMED	Positive
Update product original manufacturer	EO	SAMPLE_DTX_UDI_015.01.xml	Update information about the product original manufacturer registered as an organisation during the initial device registration	Positive
		SAMPLE_DTX_UDI_015.02.xml	Update information about the product original manufacturer with an Actor ID/SRN of a manufacturer registered in EUDAMED	Positive
Update of Market Information	EO	SAMPLE_DTX_UDI_007.01.xml	Submit a list of market info for a specific UDI-DI	Positive
		SAMPLE_DTX_UDI_007.02.xml	Submit a list of market info for a specific list of UDI-Dis	Positive
		SAMPLE_DTX_UDI_007.03.xml	Submit a list of market info for a non-existing UDI-DI	Negative
Update of Container Package Information	EO	SAMPLE_DTX_UDI_012.01.xml	Submit a list of container packages to a registered UDI	Positive
		SAMPLE_DTX_UDI_012.02.xml	Update the status of a container package	Positive
Download issued certificate (last current version)	CA, NB	SAMPLE_DTX_CRF_02.01.xml	Downloading explicitly a known certificate by its certificate number	Positive
		SAMPLE_DTX_CRF_02.02.xml	Downloading certificates that reference a specific manufacturer by specifying its SRN	Positive
		SAMPLE_DTX_CRF_02.03.xml	Downloading certificates issued by other Notified Body than the Notified Body requestor	Negative
Download refused certificate	CA, NB	SAMPLE_DTX_CRF_03.01.xml	Download refused certificates by its number	Positive
		SAMPLE_DTX_CRF_03.02.xml	Download refused certificates by type	Positive
Download SSCP	NB	SAMPLE_DTX_UDI_009.01.xml	Download SS(C)P by Basic UDI-DI	Positive
		SAMPLE_DTX_UDI_009.02.xml	Download SS(C)P by certificate ID	Positive
Download SSCP	CA, EO	SAMPLE_DTX_UDI_011.01.xml	Download SS(C)P by Basic UDI-D	Positive
		SAMPLE_DTX_UDI_011.02.xml	Download SS(C)P by certificate number / revision number	Positive

9.3 Annex 3 (AP statuses)

Status	Description
Draft	The request has not yet been submitted.
Submitted	The request is submitted. The AP is not yet active and approval is needed.
Connectivity under validation	The request has been approved but the AP is not yet configured and active.
Rejected	The request is rejected.
Active	The AP is active and ready for M2M data exchange.
Maintenance	This status is only used for migration of APs from one format to another. Contact EUDAMED support [56] if the AP is under maintenance status.
Terminated	The AP is deactivated and cannot be used for M2M data exchange.

You can see the status of your AP in the *Access point management* page:


Home
Tasks
Search & view
Data Transfer
News
Help
Test DTHREE-THREE
Logout
CURRENT ACTOR: Manufacturer, DE-MF-000002295, Test Maker of Things in Germany [Germany]
Notifications
Access point management
Request a new M2M access
Filter
Active filters:
Status: Active
Clear search
Showing 1 to 1 of 1 entries
Show 20 entries per page
Access Point
Party ID
Status
Access Point Type
Actions
TW Demo 3rd Party AP 1
EUDAMED_000081
Active
THIRD_PARTY


European
Commission