Collected information on e-Codex messages within the e-Codex environment

Table of contents

[Table of contents 2](#_Toc529872813)

[1. Introduction 3](#_Toc529872814)

[1.1. Scope and Objective of this document 3](#_Toc529872815)

[1.2. Preconditions 3](#_Toc529872816)

[1.3. Structure of the document 3](#_Toc529872817)

[2. In the beginning… 4](#_Toc529872818)

[2.1. E-Codex 4](#_Toc529872819)

[2.1.1. The DOMIBUS gateway 4](#_Toc529872820)

[2.1.2. The domibusConnector 5](#_Toc529872821)

[2.1.3. The national backend side 5](#_Toc529872822)

[2.1.4. Business use cases 5](#_Toc529872823)

[3. Message parts provided by the national backend 6](#_Toc529872824)

[3.1. Business document as a PDF 6](#_Toc529872825)

[3.2. Business document as structured data (XML) 7](#_Toc529872826)

[3.3. Business attachments 7](#_Toc529872827)

[3.4. Additional metadata 7](#_Toc529872828)

[4. The domibusConnectorAPI message structure 8](#_Toc529872829)

# Introduction

## Scope and Objective of this document

When participating and working with e-Codex several questions will rise on

* What are the messages processed within e-Codex decoupled from the business use case?
* What structures do those messages have?
* Where, how and why are there transformation processes on messages?
* What kind of security on what layer of a message is given/provided?
* What is happening with a message while being processed by e-Codex components?
* When, where and how are ETSI-REM evidences generated and treated?

This document should answer those questions and give an overview on the processing.

## Preconditions

Else than other documents that focus on certain components within the e-Codex environment, this document focuses on messages over the borders of components used. Therefore it is recommended to have know-how of the following e-Codex components first:

* The DOMIBUS gateway.
* The domibusConnector.
* The national backend side (whether it is a domibusConnectorClient or a self-developed application).

As this document is rather technically focused, it is meant to be missing link between technically knowing the e-Codex components and getting to know the big picture a little better.

## Structure of the document

At least in its first versions this document may seem quite unstructured. This is due to the fact that it starts as some sort of information collection.

With time the writers of the document will try to bring more structure and “red band” into the document. Anyway the information this document contains is very useful and helps to understand some other documentation better.

# In the beginning…

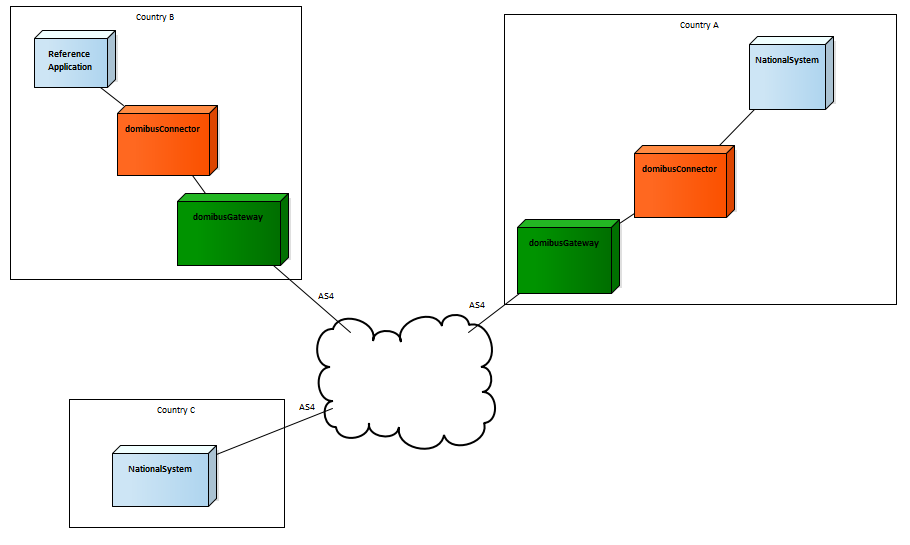
.. it all starts with a practitioner or citizen that has the need to start a cross-border procedure in the context of e-Justice.

## E-Codex

As there is a lot of documentation on what e-Codex stands for and what it is, this document sees the idea and contents of e-Codex as given.

E-Codex started as a long-running project in 2010 funded by the European Union to enable electronic transmission of procedures over country borders in the justice domain.

Together with the invention of such procedures an own kind of “infrastructure” was implemented that ensures the secure and standardized transmission of such procedures. To bring it all together the e-Codex architecture was invented:



Countries (or participants) are connected in B2B style with each other following a standard. As a reference two major components were developed during e-Codex to support the standards and give a link between common understandings and already used national applications to support such procedures nationally.

### The DOMIBUS gateway

Though not required mandatorily the DOMIBUS gateway implements all defined standards to connect participants to be able to transmit cross-border procedures.

The gateway component supports the transport mechanisms only not being aware of the contents of messages.

This means that a message that contains a procedure has to have a certain structure to be able to be transmitted. There also needs to be a definition of the cross-border procedures supported. But what is inside this message is not of interest for the gateway. As long as the structure of the message follows the given parameters it will be transmitted, no matter what the particular contents contain.

### The domibusConnector

Other than the gateway, the domibusConnector supports the processing of the message between the national environments and the gateway. Therefore the domibusConnector also slightly touches the contents of the message and builds the structure of the message that can be transmitted by the gateway. For this purpose the domibusConnector provides interfaces towards the national side that allow in an understandable structure to start the processing of messages. The domibusConnector is also aware of the national backend applications connected and which procedure is supported by which backend application.

Additionally the domibusConnector provides tracking of the message by using ETSI-REM evidences. Those evidences give information for the original sender of such a message on where the message currently is and if the delivery of a message to the final recipient of the message was successful. As e-Codex is placed in the justice domain, such evidences can have legal impact on the procedures.

### The national backend side

Though the national side could have several faces and also not only one application but also chains of services, the e-Codex procedure sees the national backend side as a black box.

Anyways, the national backend side triggers messages with documents that already need to be provided by the national backend. Also, received messages, are transmitted to the national backend. The way of how such procedures and documents are triggered nationally or how the message received reaches its final recipient are completely transparent for e-Codex components.

There are, depending on the procedure, supporting tools to implement national applications, but in the end it has to be done by every participant itself.

### Business use cases

What this document calls “procedure” up to this chapter is commonly known as business use cases.

Such a business use case defines the procedure, the domain, the content and also the connectivity in some way.

Examples for business use cases are the European Payment Order (EPO) or Small Claims (SC) in civil domain or the European Investigation Order (EIO) in criminal domain.

# Message parts provided by the national backend

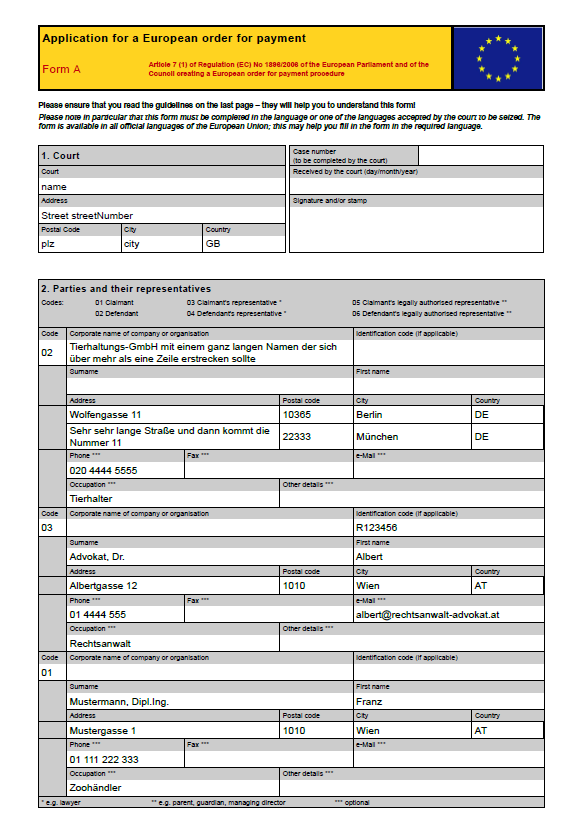
To even start the processing of a message containing a business use case, the main parts of the message have to be provided by the national backend.

## Business document as a PDF

The business use case usually is defined by having a form that must be filled. This can be a file that a practitioner or participant claims to a cross-border court or a request for investigation from one practitioner to another for example.

This form has to have a defined form and must contain defined information.

Here is an example of a European Payment Order form that initiates the file at the receiving court:



This form is defined usually by EU regulations and look the same in every country (besides that the texts are in the language of the court’s country).

This business document is the legally binding part of the message. Therefore this PDF must not be changed during the processing of the message from one end (the original sender) to the other (the final recipient).

To ensure the durability and validity of the document e-Codex defines two mechanisms:

* The document must be signed by official electronic certificate. This electronic signature can be embedded inside this document, but it can also be provided together with the document as a detached signature.
* The document was created inside a trusted network. In that case the sender of the message guarantees that the document origins from an advanced electronic system where no access by non-authenticated and non-authorized people is given at any time.

## Business document as structured data (XML)

A message that is electronically created, transmitted and received usually is treated by applications. For those applications, to support the practitioners, data is provided electronically.

The business document as XML represents the contents of the business document as PDF.

The structure of the XML document is defined by the business use case and strongly relies on the structure of the PDF forms that are defined relying on EU regulations.

Usually the schemas defining the structure of the XML business documents are provided by the maintainer of the business use case.

## Business attachments

Some claims, applications or files can be supported by attachments.

In the example of having an Application for a European Payment Order (the form shown in the screenshot above), the practitioner maybe wants to provide also a scan of an invoice, or some other evidence.

Those business attachments are not limited by e-Codex. It can be any format that can be electronically transmitted.

As such attachments usually are linked inside the business document some metadata for the attachments have to be provided as well (and must match the data published inside the business document).

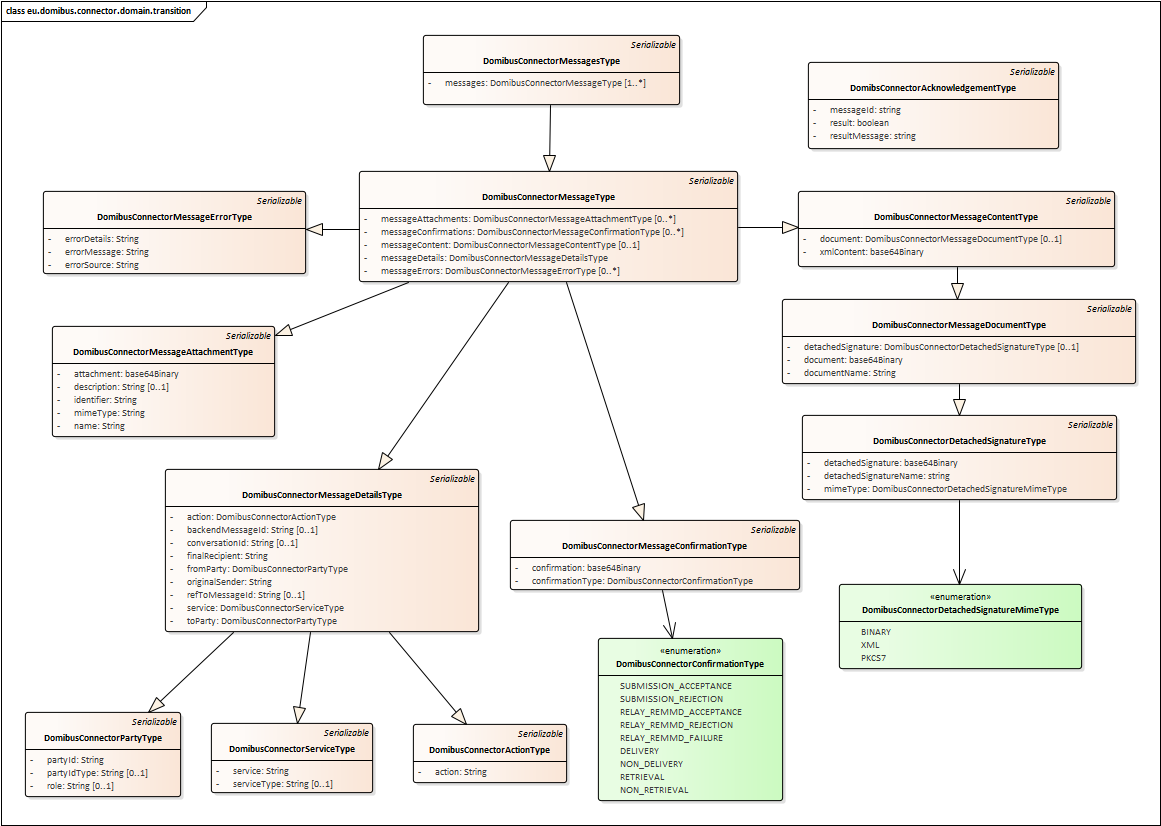
## Additional metadata

To be able to process and transmit the message to a final recipient some metadata on message level needs to be provided. This metadata is defined by the public interfaces of the domibusConnector and described in the following chapter.

# The domibusConnectorAPI message structure

To give the domibusConnector a new message for processing, the national backend application must use the domibusConnectorAPI. This API not only defines the methods that the domibusConnector offers, but also the structure of the message the way that the connector can process it.

## The domibusConnectorMessageType



The most important object types to be filled within the DomibusConnectorMessageType by the national backend application are:

* DomibusConnectorMessageDetailsType: to fill the routing information. Essential are
  + Service -> the business use case
  + Action -> the sub defined action within the business use case
  + fromParty -> the sender’s gateway data
  + toParty -> the receiver’s gateway data
  + originalSender -> the initial sender of the message, the practitioner, the court
  + finalRecipient -> the final receiver of the message, the court, the department
* DomibusConnectorMessageContentType: to place the business document. Essential are
  + document
  + xmlContent
* DomibusConnectorMessageAttachmentType: to add the Business attachments of the message.