Use of (enhanced) PEPPOL eDelivery network for exchange of ISO 20022-based payment messages

Difi guidance document

Version: 1.0

2017/18/01

Erlend Klakegg Bergheim and Olav Astad Kristiansen

Direktoratet for forvaltning og IKT

Table of context

List of figures

List of diagrams

List of acronyms

Acronym	Description
AS2	Applicability Statement 2 (Protocol used in OpenPEPPOL)
ASiC-e	Associated Signature Containers – extended
CEF	Connecting European Facility
MDN	Message Disposition Notification
OpenPEPPOL	Non-profit international association under Belgian law (AISBL)
PEPPOL	Pan-European Public Procurement Online
SBDH	Single Business Document Header
TLS	Transport Layer Security – used in combination with HTTP
SHA 256	Secure Hash Algorithm 2
ETSI	European Telecommunications Standards Institute

1 Background

The PEPPOL eDelivery network is widely used for exchange of invoices and credit notes in EHF and PEPPOL BIS formats in the Norwegian market. It is also used for exchange of product catalogues and orders in the same formats. A vast majority of the public sector, most banks and more than 65.000 private sector entities can receive documents through the network, and in 2016 more than 35 million invoices was exchanged. More than 50 service providers offer connectivity to the PEPPOL eDelivery network through access points, and more than 60 service providers offer solutions to issue or receive EHF or PEPPOL BIS business documents.

From 2017, an enhanced version of the PEPPOL eDelivery network will be used for submission of tenders in public procurement.

In 2015, Norwegian banks agreed to use ISO 20022-based messages for handling of payments, such as payment instructions from customers or notifications sent to customers.

Both banks and ERP vendors need one common solution for ISO 20022 compliance for public and private sector entities to handle secure files and file transport. By using a common method for handling ISO 20022-based documents, both banks and ERP vendors need only to connect to the infrastructure once in order to serve the entire market.
It will also reduce the number of payment formats and transport channels.

The Government Agency for Financial Management (DFØ) is in charge of the framework agreement with banks for central government agencies in Norway. DFØ requires a common standard for securing and transporting file-based ISO20022 payment transactions in their future agreements with banks, by using the PEPPOL eDelivery network and an ASiC archive. This solution reuses Open Source software components based on open standards, and utilises experience and knowledge from the existing use of the PEPPOL eDelivery network in Norway.

The use of an enhanced version of the PEPPOL eDelivery network for handling of ISO 20022-based payments, is expected to reduce end-user lock-in and make it less complicated for entities to change both their bank service providers and ERP vendors.

2 Objectives

The purpose of this document is to provide information on the usage of an enhanced version of the PEPPOL eDelivery network (the PEPPOL profile of CEF eDelivery DSI) for handling of secure end-to-end message exchange. The document focuses on the handling of ISO 20022-based payment messages as a use-case. However, the PEPPOL eDelivery network and the concepts described in this document are generic in their nature, and are in principle applicable for other document types/use-cases where end-to-end security and traceability is needed. The document's main target groups are business process owners and technical developers. In order to bridge the gap between the business process design and process implementation, Business Process Management and Notation (BPMN) is used.

This document is based on the current technological solution; a solution subject to dialogue with stakeholders such as banks and ERP providers. The overall picture communicated in this document is seen as agreed.

This document also cover how public authorities can cooperate with Norwegian banks in order to collect data about entities. Such cooperation will use other document types than those based on ISO 20022.

3 Solution overview

3.1 OpenPEPPOL

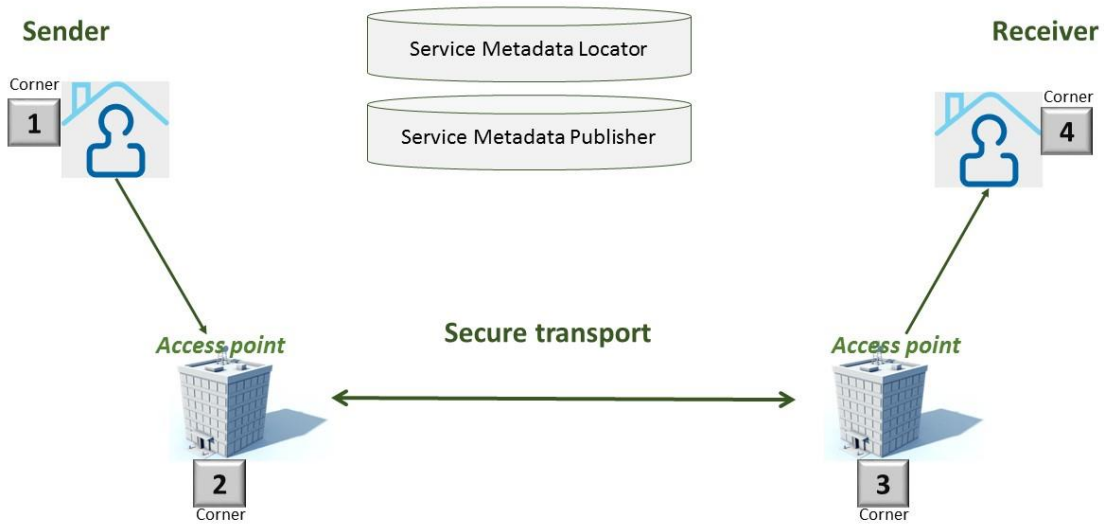
The EU co-funded Pan-European Public Procurement Online (PEPPOL) project started in 2008. In 2012 the specifications and operations of the PEPPOL eDelivery network were transferred to OpenPEPPOL AISBL, a member driven non-profit international association under Belgian law with both public and private sector members.

The purpose of OpenPEPPOL is to enable European businesses to easily deal electronically with any European public sector buyers in their end-to-end procurement processes, thereby increasing opportunities for greater competition for government contracts and providing better value for tax payers' money.

3.2 The PEPPOL eDelivery network

The PEPPOL eDelivery network is a profile of the European Commission (EC) Connecting Europe Facility (CEF) eDelivery Digital Service Infrastructure (DSI), or a PEPPOL profile of CEF eDelivery DSI for short. The network is based on a four-corner model with dynamic recipient capability look-up.

The PEPPOL eDelivery network is a combination of a message exchange model, discovery model (capability look-up), a PKI-based security model and a legal framework that enables the exchange of structured information through the internet, wrapped in a messaging envelope.



PEPPOL profile of CEF eDelivery

In the four-corner model, the back-end systems of end-users do not exchange data directly with each other, but transport data through Access Points. These Access Points are conformant to the same technical specifications and are therefore capable of communicating with each other.

As a result, end-users can easily and safely exchange data, even when their IT systems are developed independently from each other.

This is also known as the Mesh network, where all mesh nodes cooperate in the distribution of data in the network.

3.2.1 Service Metadata Locator

The role of the SML (Service Metadata Locator) is to manage the resource records of the participants and the SMPs (Service Metadata Publishers) in the DNS (Domain Name System). The SML is the only centralised component in the PEPPOL eDelivery network, and is currently operated by the EC unit DG DIGIT.

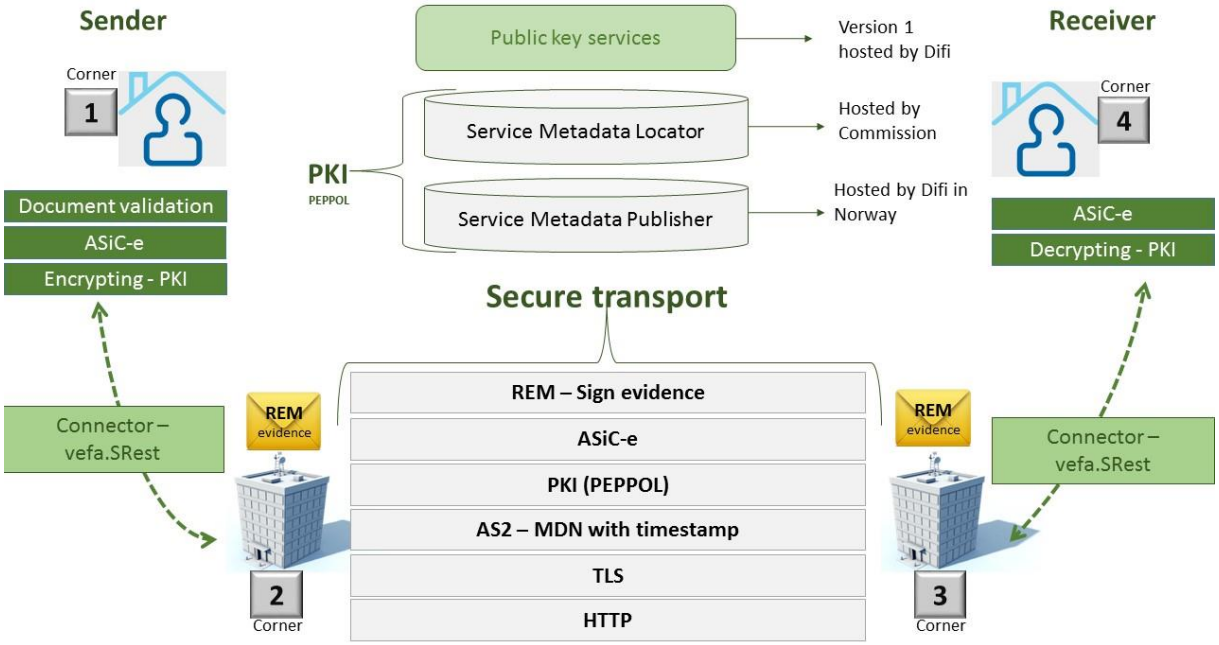
3.2.2 Service Metadata Publisher

After using the SML to discover the endpoint address of the receiver's SMP (Service Metadata Publisher), it is able to retrieve the needed information (i.e. metadata) about the receiver. The SMP is a distributed component in the PEPPOL eDelivery network.

3.3 Use of an enhanced PEPPOL eDelivery network

The current PEPPOL eDelivery network was established to ensure secure and reliable messaging between access points. In order to provide support for end-to-end security and reliable messaging as required for electronic communication by the public procurement directives, an enhanced version of the PEPPOL eDelivery network has been established. The specifications of this enhanced version are developed and tested as part of the e-SENS project as well as by Difi, and are expected to become a part of the PEPPOL eDelivery network specifications. The enhanced PEPPOL eDelivery network contains additional security components delivered as Building Blocks, which are available as open source software.

Business to business use of the PEPPOL eDelivery network and use of PEPPOL-components in other areas beyond procurement are also recognised as important and are encouraged by OpenPEPPOL. PEPPOL is currently being implemented in several European countries and interest is now increasing outside of the EU as well.



Enhanced PEPPOL profile of CEF eDelivery

The main features of the enhanced PEPPOL eDelivery network is that it supports a higher level of security (in addition to the existing security), and the ability to track and trace all messages sent throughout the transport infrastructure. The sender must encrypt the document and only the receiver can decrypt the document. A short description of the different components is given below.

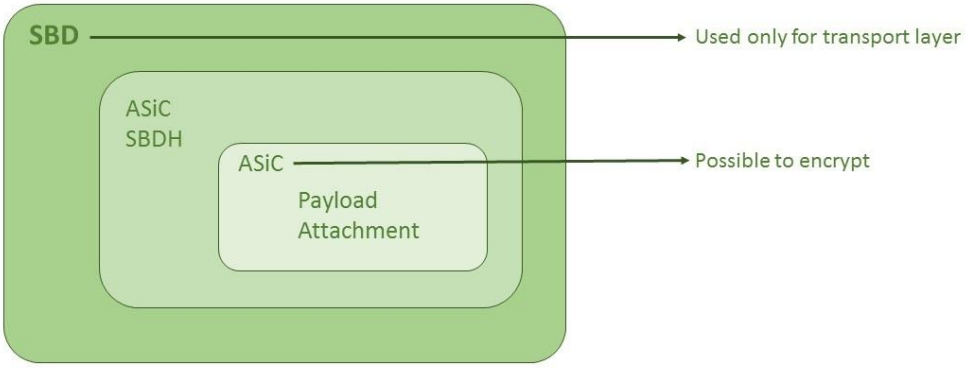
3.3.1 Public key services (Certificate server)

The role of the Public key services (Certificate server) is to store the public key of a qualified certificate for a receiver who wishes to receive encrypted documents. This makes it possible to introduce end-to-end security. The server is able to retrieve the needed public key when a valid combination of participant identifiers and business profiles are used for the lookup. Business profiles is used to separate areas like payments and invoices. The Certificate server is either a centralised or a distributed component in the enhanced PEPPOL eDelivery network.

The first version of certificate server is hosted by Difi, and receivers will have access to store their public keys within a profile. In the future, the SMP will be used to locate an organization's published certificate.

3.3.2 ASiC-E archive

The ASiC-E (Associated Signature Containers – Extended) is a file format to package data of various types standardized by ETSI. This container is capable of holding one or more signatures. Each file object can have payload, additional information or metadata associated with it that can be protected by the signature.



ASiC-E

The method for securing ISO 20022-based payments is to first make an ASiC-E archive and then encrypt this archive. The purpose is to exploit the rate of compression of the payload and attachments in an ASiC-E. Encrypting documents before compression will result in the compression rate to be much lower.

3.3.3 REM evidences

This solution uses a part of REM (Registered Electronic Mail) standardized by ETSI. To secure non-repudiation for ISO 20022-based payments is the MDN (Message Disposition Notification) put into the REM evidence. The REM evidence needs to be signed and stored for at least 1 year. This is the evidence for sending and receiving documents.

3.3.4 vefa.SRest

In order to encourage competition amongst the Access Point vendors and prevent vendor lock-in, Difi now provides a REST-based interface to the Access Point named «vefa-srest».

«vefa-srest» is an add-on product to the well-established open source PEPPOL Access Point software package named «Oxalis».

«vefa-srest» provides a secure REST-interface allowing users to upload and download PEPPOL business documents. Users may implement their own client within their business software or use the client software package to manage the electronic collaboration with their peers in the PEPPOL network.

«vefa-srest» is published as open source software and may be downloaded and used freely by the PEPPOL constituents. The REST-based protocol, named SREST, is well documented and easy to use by any party wishing to integrate with the PEPPOL eDelivery network.

Vendors may choose to use «vefa-srest» as-is or use their own implementation using the SREST-protocol. «vefa-srest» comes with built-in support for «Oxalis» and as such is a perfect fit for any access point running «Oxalis».

4 The PEPPOL legal framework

The PEPPOL eDelivery network requires that several actors work together in a trusted environment. Governance for the PEPPOL eDelivery network is secured through a legal framework known as the PEPPOL transport Infrastructure Agreement.

In this governance model OpenPEPPOL ASBL, in its role as the PEPPOL Coordinating Authority, has authority over all central components of the PEPPOL eDelivery network: the technical and service specifications, the Service Metadata Locator and the Transport Infrastructure Agreements and its annexes. Through signed agreements the PEPPOL Coordinating Authority will delegate authority over the implementation and use of the PEPPOL eDelivery network within a defined domain to a PEPPOL Authority. The PEPPOL Authority Agreement defines the general principles of cooperation between these two parties.

The PEPPOL Authority must ensure that Access Point (AP) and Service Metadata Publisher (SMP) services are provided in conformance to the technical standards and service specifications by entering into separate AP and SMP agreements with each of the respective providers within their domain.

The regime of agreements and the governance structure ensure that:

- the role and responsibilities of each actor are clearly described and openly available, making PEPPOL an open and transparent community;
- sufficient information is made available through the SML/SMP, allowing a Participant to make this its sole source of information for conducting e-procurement with its trading partners.

Through these measures, a set of minimum requirements and criteria is established and consistently applied throughout the full PEPPOL eDelivery network.

The enhanced PEPPOL eDelivery network introduces new actors (the certificate server) as well as new features that need to be reflected in an enhanced version of the PEPPOL Transport Infrastructure Agreement.

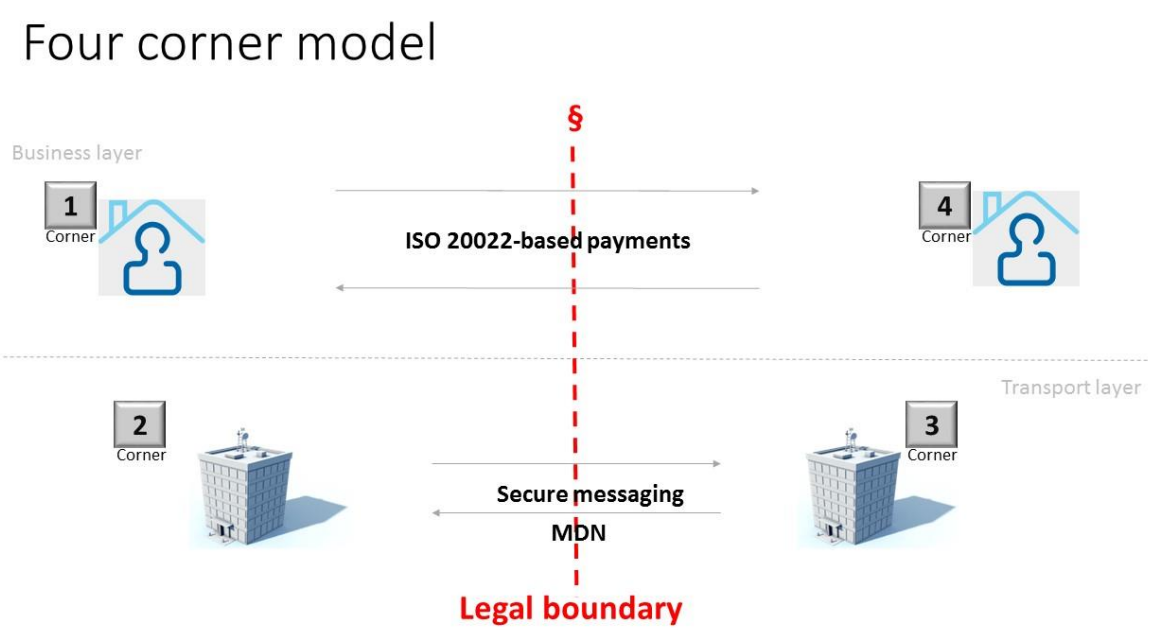
The enhanced version of the PEPPOL Transport Infrastructure Agreement need to cater for:

- the new actors/roles identified, their responsibilities and services, as well as service level requirements;
- new services and increased service levels (such as mandatory support for encryption, ASiC-E and REM evidence) required from existing actors/roles in order to support the new uses cases.

5 Handling of secure messaging

The following chapter explains how ISO 20022-based payments can be processed at the customer level and at the bank level.

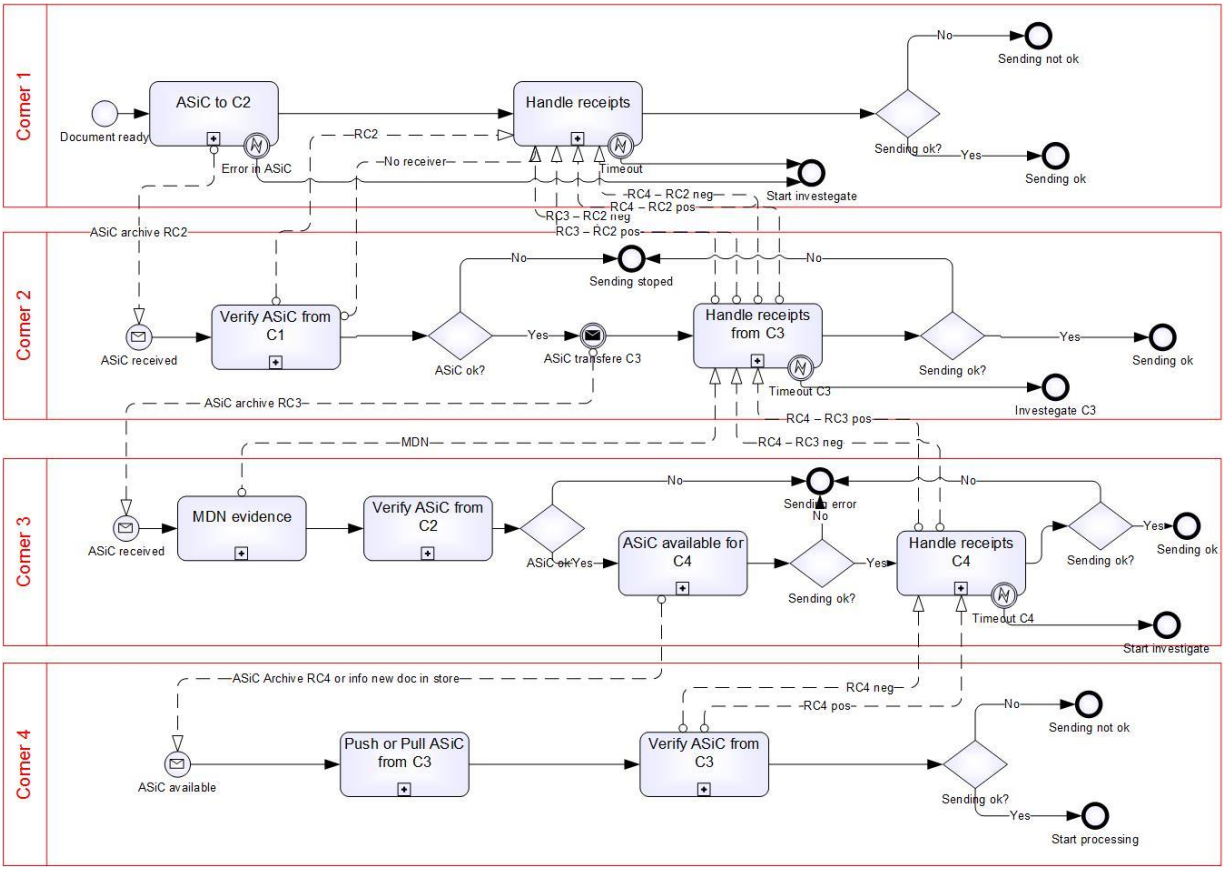
This document is a guideline for how to set up a secure solution for messaging with both business process owners and technical developers as target group.



Four-corner model

5.1 Overview of the four-corner model

In the four-corner model, all corners have their predefined processes. These processes contain tasks and here we illustrate tasks that each corner must perform as a minimum. This list is not exhaustive and every corner will have to perform additional tasks depending on the professional system used.



Collaboration diagram model 1

The tasks that appear in each corner are called sub-processes (the + inside the boxes) and they are hierarchically structured. The diamond shapes are gateways or decisions and after a test, we can see which path is possible. We will later go into more detail in the hierarchical models.

The diagram model has three final end states:

End Status	Description
Sending ok	Bank has received payment package (ASIC)
Sending not ok	Payment package has stopped on the road
Start investigate	Something in the payment package is incorrect

Sending ok - The receiver (Corner 4) has received the document and has sent a confirmation of this.
Start investigate - This error is caused by a timeout and it can occur in corner 1, 2 or 3. The document has stopped or it has taken too long to send the document to Corner 4 from

corner 2 or 3. If end state “Start investigate” occur corner 1 must contact corner 2, and corner 2 must contact corner 3, and corner 3 must contact corner 4.

Sending not ok - This is an error that can occur in all corners. There may be errors in the SBDH or ASiC-E archive. In corner 1, it may be an error in payload.

The messages sent between corners are

Message name	Description
RC2	Receipt of corner two
RC3	Receipt of corner three
RC4	Receipt of corner three
ASiC	Zip file with signature
SBDH	Standard Business Document Header

RC2 - This is a receipt from a Corner 2. The receipt must be sent with a positive or negative message. Negative receipts should be sent if there are errors in addressing, SBDH or ASIC archive. The exact manifestation of this receipt is out of scope for this document, however is it mandated to have a formalized receipt.
RC3 - This is a receipt from a Corner 3. Corner 2 must convert the MDN to an RC message. The receipt must be sent with a positive or negative message. Negative receipts should be sent if there are errors in addressing, SBDH or ASiC-E archive. REM is used as a transport neutral receipt.
RC4 - This is a receipt from a Corner 4. The receipt must be sent with a positive or negative message. Negative receipts should be sent if there are errors in addressing, SBDH or ASiC-E archive. RC4 is not a receipt for handling the payload, for receipt of payload handling, business messages are used
ASiC-E - This is a zipped file to be signed. The content is encrypted payload, manifest, public certificate and inner Standard Business Document Header (SBDH).
SBDH - Carrier of the ASiC-E archive as standardized by ETSI. Contains information regarding sender, receiver, document type identifier, process identifier, unique identifier per message and timestamp of creation. May be thought of as an envelope allowing efficient transmission without support for the individual document type in transmission layer.

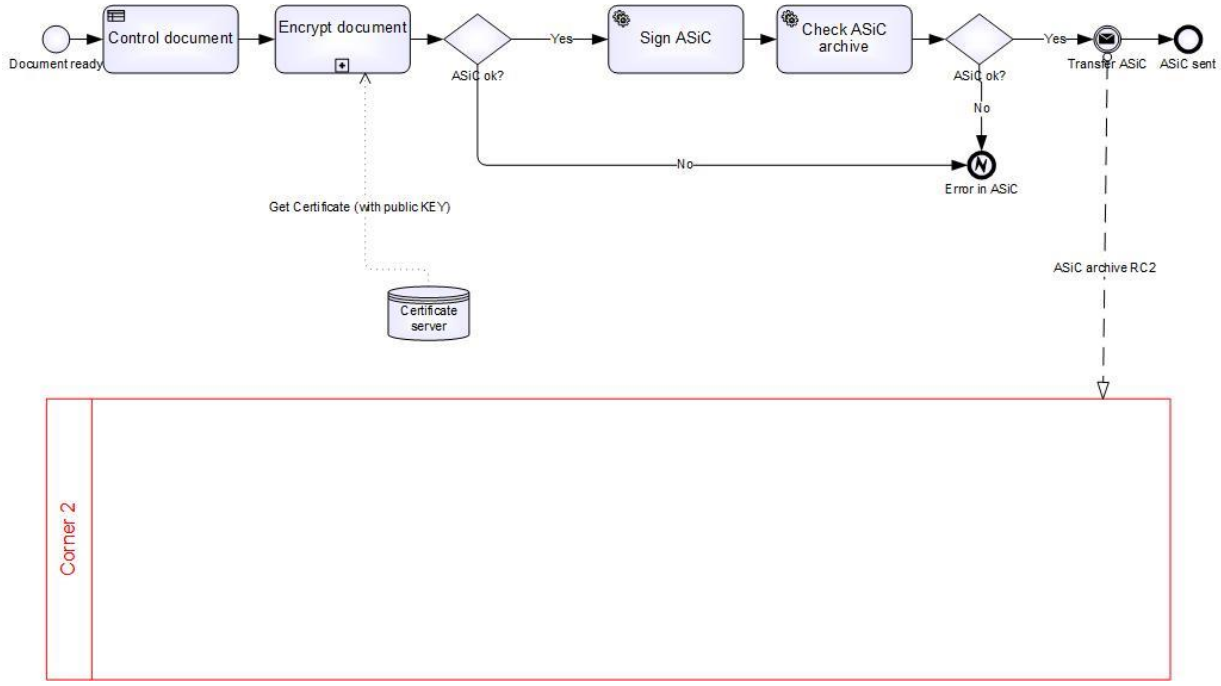
5.2 Process for corner 1

Corner 1 is always a sender of an ISO 20022-based document. It can be a payer, an ERP system or a bank. The process starts when a document is ready for sending. If this document is a payment, it will typically be a “pain.001”. If the document is a reaction to the “pain.001” it will be a “pain.002”. There are many other documents that can be distributed as well.

The most common ISO 20022-based documents today are the following:

Document type	Area
Pain001	CustomerCreditTransferInitiation
Pain002	Syntax Receipt/Content receipt
Camt029	Camt029
Camt54C	Credit advise
Camt053	Financial statement
Camt054D	Advis
Camt055	Request for cancellation

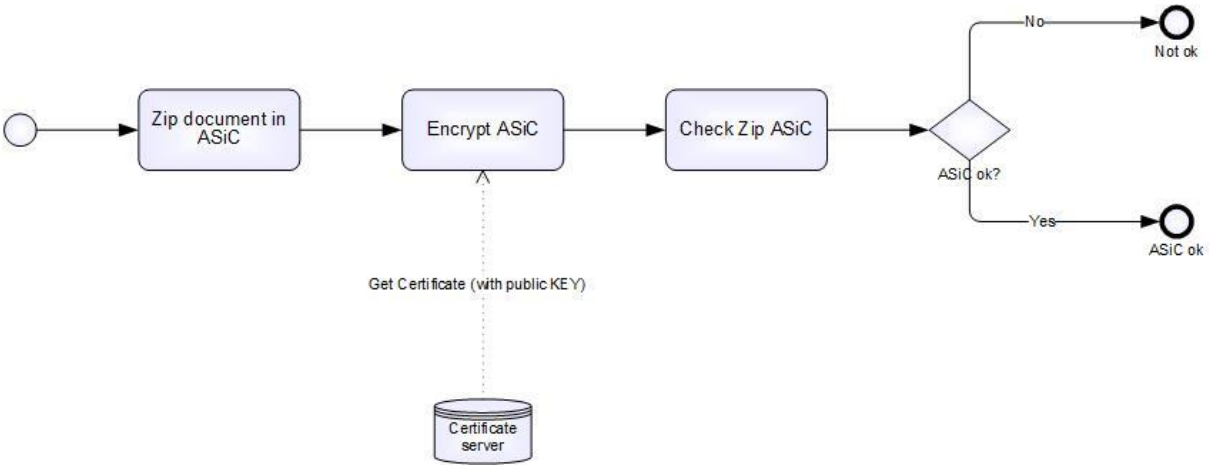
5.2.1 Diagram – Send ASiC to C2



Collaboration diagram Send ASiC to C2

This sub-process starts when a new document is ready to be sent. The first task is "**Control the document**". This control should be based on business rules for the type of document and it must be validated to be sure its valid. The next task is "**Encrypt document**" and this is a sub-process. The task uses the recipient's public key and will use a "get" command to get the key from the "**Certificate server**". If the ASiC archive is correct after the sub-process "**Encrypt document**" the next task is "**Sign ASiC**". Corner 1 will use its private key from the company certificate to sign the ASiC archive. The following task "**Check ASiC archive**" will do a control of the ASiC archive. If the result of this control yields "ok", Corner 1 will send the ASiC archive to Corner 2 and set the end state to "ASiC sent".
If the result from "**Encrypt document**" is not ok, the end state will be set to error "**Error in ASiC**" and Corner 1 will start to investigate the problem.
If the result from "**Check ASiC archive**" is not ok, the end state will be set to error "**Error in ASiC**" and Corner 1 will start to investigate the problem.

5.2.2 Diagram – Encrypt document

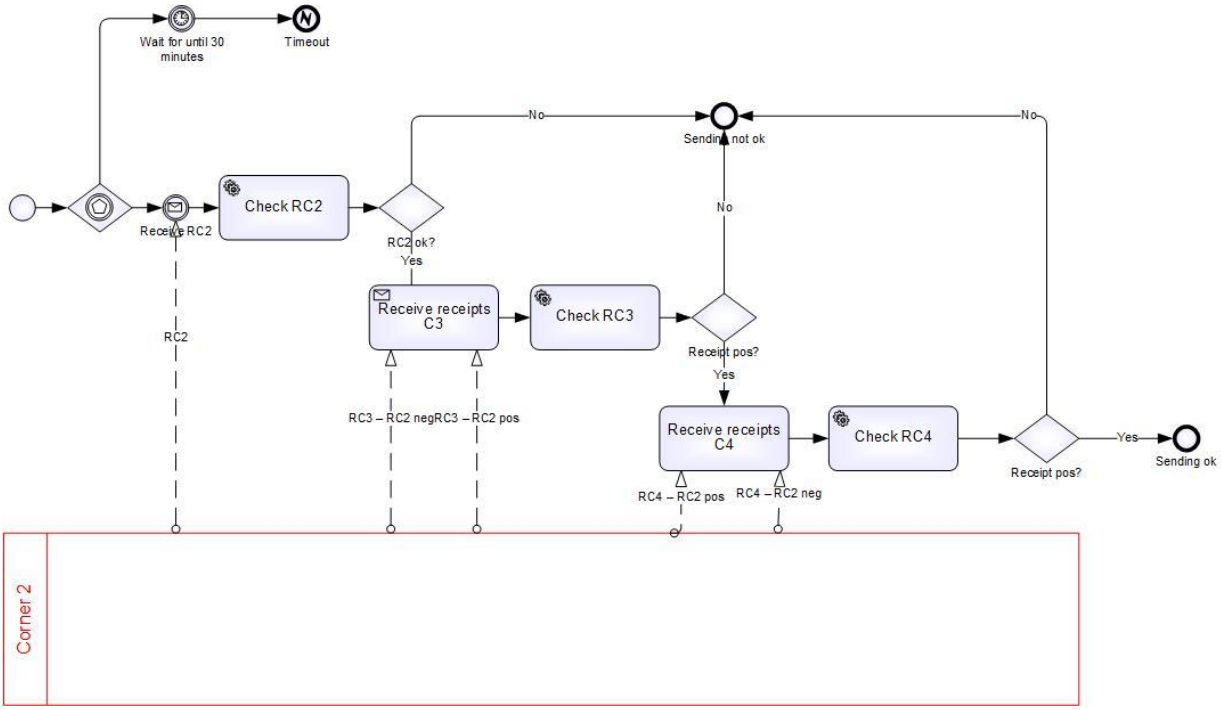


Collaboration diagram Encrypt document

Sub-Process "**Encrypt document**" starts with task "**Zip document in ASiC**". This task will compress the payload. The following task (Encrypt ASiC) will encrypt the Zip file with the receiver's certificate (public key) and then attach all documents needed and sign the ASiC with the private key from its company certificate.
When finishing the "**Encrypt ASiC**" the next task "**Check Zip ASiC**" will control if the ASiC is ok.

If the ASiC is ok, the end state is "**ASiC ok**" and the sub-process will return to "**Send ASiC to C2**". If the result is "not ok", the end state is set to "**Not ok**" and the sub-process will return to sub-process "**Send ASiC to C2**".

5.2.3 Diagram – Handle receipts



Collaboration diagram Handle receipts

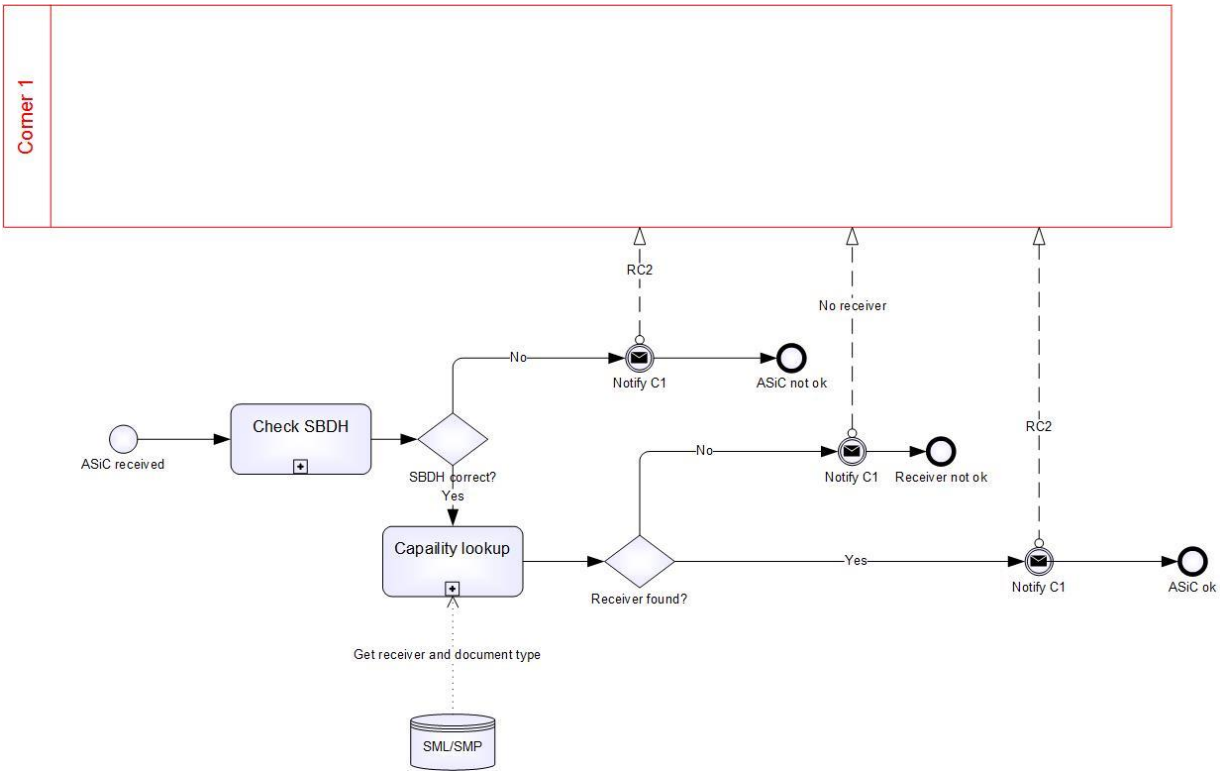
This sub-process starts when sub-process "**Send ASiC to C2**" is finished correctly.
This sub-process is a race against the clock. If all three receipts are not received within i.e. 30 minutes, the end state will be set to error "**Timeout**" and return to the main process. The main process will then start to investigate the problem.

The sub-process first task is "**Receive addressinform.**". This task will receive a positive or negative "**RC2**" message from Corner 2. If corner 2 don't find the receiver the next message will be "**No receiver**". The next task "**Check address**" will check the status of "**RC2**" and "**No receiver**". If the result is "**Address not ok**", the end state is set to "**Sending not ok**" and the sub-process will end. The main process will end as well with status "**Sending not ok**" and carry out the action needed.
If the status from "**Check address**" is ok, the next task "**Receive receipt C3**" is waiting for "**RC3**". After receiving "**RC3**" task "**Check RC3**" will start. If the status is not ok, the end state is set to "**Sending not ok**" and the sub-process will end. The main process will end as well with status "**Sending not ok**" and carry out the action needed.

If the status from "**RC3**" is "**Receipt pos**", the next task "**Receive receipt C4**" is waiting for "**RC4**". After receiving "**RC4**" task "**Check RC4**" will start. If the status is "**Receipt neg**", the end state is set to "**Sending not ok**" and the sub-process will end. The main process will end as well with a status "**Sending not ok**" and carry out the action needed.
If the status from "**RC4**" is "**Receipt pos**", the end state will be set to "**Sending ok**" and the sub-process will end. The main process will also end with a result "**Sending ok**". This result tells Corner 1 that Corner 4 has received the ASiC archive.

5.3 Process for corner 2

5.3.1 Diagram – Verify ASiC from C1



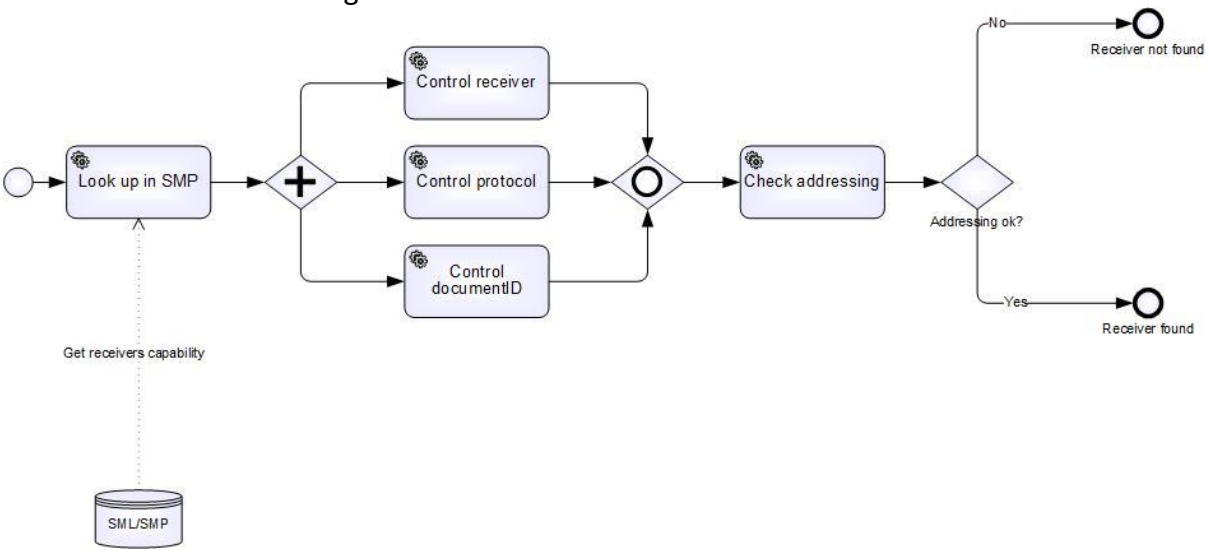
Collaboration diagram Verify ASiC from C1

The sub-process "**Verify ASiC from C1**" will first check the SBDH for errors. If there is an error in the SBDH, the task will return a negative "**RC2**" to Corner 1 and set the end state to "**ASiC not ok**".

If the SBDH is correct the sub-process "**Capability lookup**" will locate the receiver of the ASiC archive with the correct document type in the **SML/SMP** look-up.
If the receiver is not found with a correct document type, the test after this task will send a negative "**No receiver**" and set the end state to "**Receiver not ok**". If the task has found a receiver with the correct document type the test will send a positive "**RC2**" to corner 1 and set the end state to "**ASiC ok**".

5.3.2 Diagram – Capability lookup

The main reason for this sub-process is to be sure the incoming SBDH has correct information before sending it to Corner 3.



Collaboration diagram Capability lookup

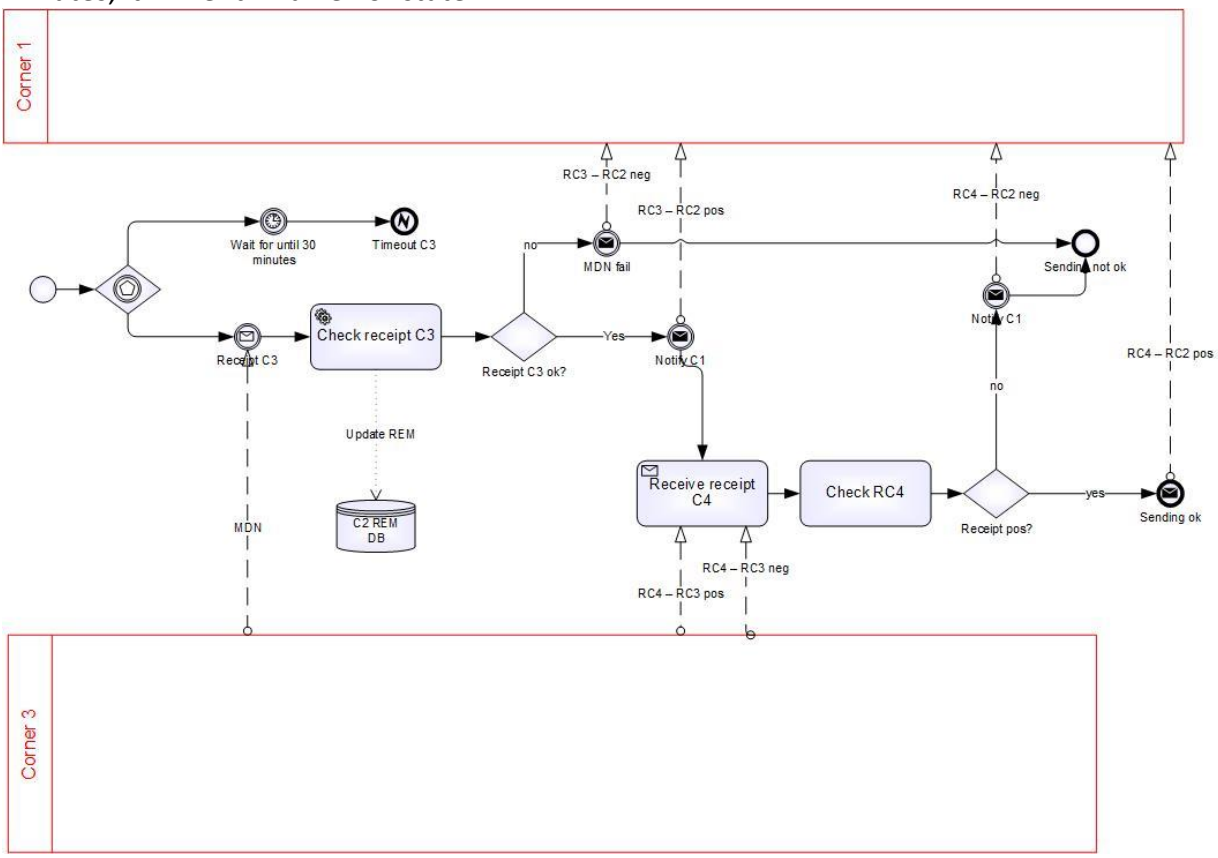
The sub-process "**Capability lookup**" first performs the task "**Look up in SMP**". This is a lookup in the SML/SMP to find the receiver, document type and protocol to exchange the payload.

The next task is divided into three different controls, running in parallel. The first is "**Control receiver**" against the **SBDH**, and the second is "**Control protocol**" that will look for which message exchange protocol to use. This is not a mandatory task today (since it is currently mandatory to use AS2). The third task is "**Control documentID**" which will check if the receiver can receive this type of document.

When all controls are done, the next task "**Check addressing**" will check if all necessary information for addressing the receiver is found. If the addressing is not complete, the end state will be "**Receiver not found**" and if the addressing is ok, the end state will be "**Receiver found**"

5.3.3 Diagram – Handle receipts from C3

After sending the ASiC to Corner 3, this sub-process will start. This sub-process is a race against the clock. If no receipts from Corner 3 and Corner 4 are received within i.e. 30 minutes, it will end in an error state.



Collaboration diagram Handle receipts from C3

This sub-process starts when the ASiC is sent to Corner 3, and the timer starts at the same time. If no message from **RC3** and **RC4** is received within 30 minutes, the end state will be set to "**Timeout C3**" with an error state. The timer started when corner 2 sent the ASiC archive to corner 3, and corner 1 should have started the inquiry for receipts. The sub-process will end, then the main process will end for Corner 2 and for Corner 1. Corner 1 needs to take action upon this message.
The first expected message to receive is (*Message Disposition Notification)* **MDN**. The task "**Check receipt C3**" will check if the **MDN** is positive or negative. The **MDN** will be included in a **REM** and signed. When the **REM** is signed, Corner 2 needs to save this **REM** evidence.

If the **MDN** is negative, the end state will be "**MDN fail**", and the sub-process will send a negative **RC3** to Corner 1 and set the end state to "**Sending not ok**". The sub-process will end, then the main process will end for Corner 2 and for Corner 1. Corner 1 needs to take action upon this message.

If the MDN is positive, the sub-process will send a positive **RC3** and wait for a receipt from Corner 4. Task "**Receive receipt C4**" receives **RC4** and checks it in task "**Check RC4**". If **RC4** is negative, the sub-process will send a negative **RC4** to Corner 1 and set the end state to "**Sending not ok**". The sub-process will end, then the main process will end for Corner 2 and for Corner 1. Corner 1 needs to take action upon this message.

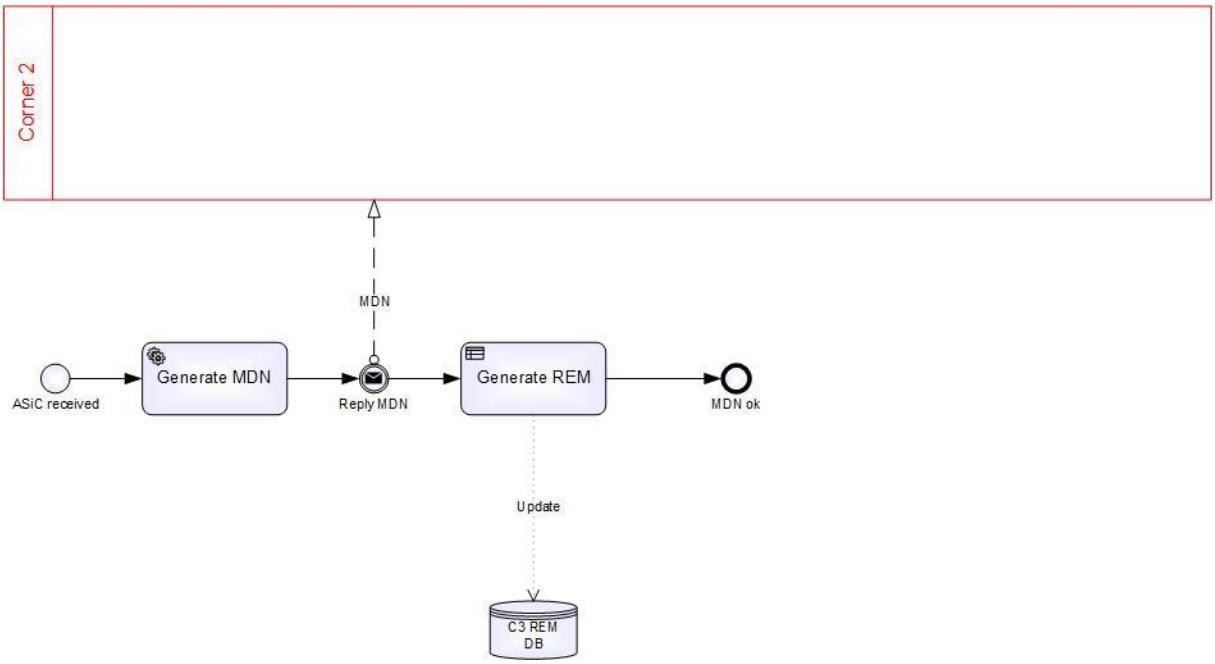
If The RC4 is positive the sub-process will send a positive **RC4** to Corner 1 and set the end state to "**Sending ok**". The sub-process will end, then the main process will end for Corner 2 and for Corner 1. Corner 1 has succeeded with sending.

5.4 Process for corner 3

The process for Corner 3 is to receive an ASiC archive from Corner 2 and verify that it is correct and make the ASiC archive available for Corner 4.

5.4.1 Diagram – MDN evidence

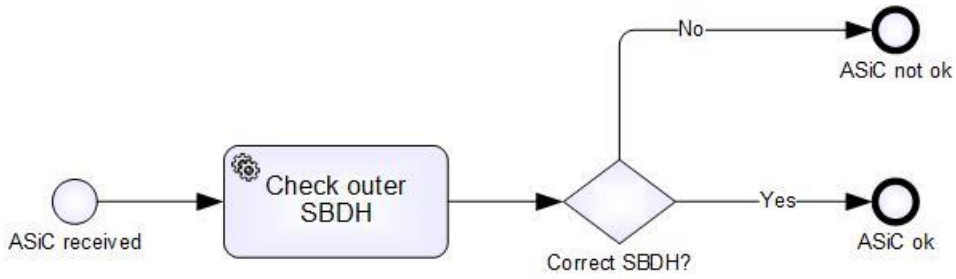
This sub-process starts with receiving of an ASiC archive from Corner 2 and to produce an MDN to Corner 2.



Collaboration diagram MDN evidence

This sub-process starts when the last byte in an ASiC archive is received. The task "**Generate MDN**" must generate a positive or negative **MDN**, dependent on whether Corner 3 successfully received the sending. After sending the MDN to Corner 2, task "Generate REM" starts to generate the **REM** based on the **MDN**. The REM must be signed and stored as an evidence, if needed later.
The end state is "**MDN ok**", meaning the MDN is sent.

5.4.2 Diagram – Verify ASiC from C2

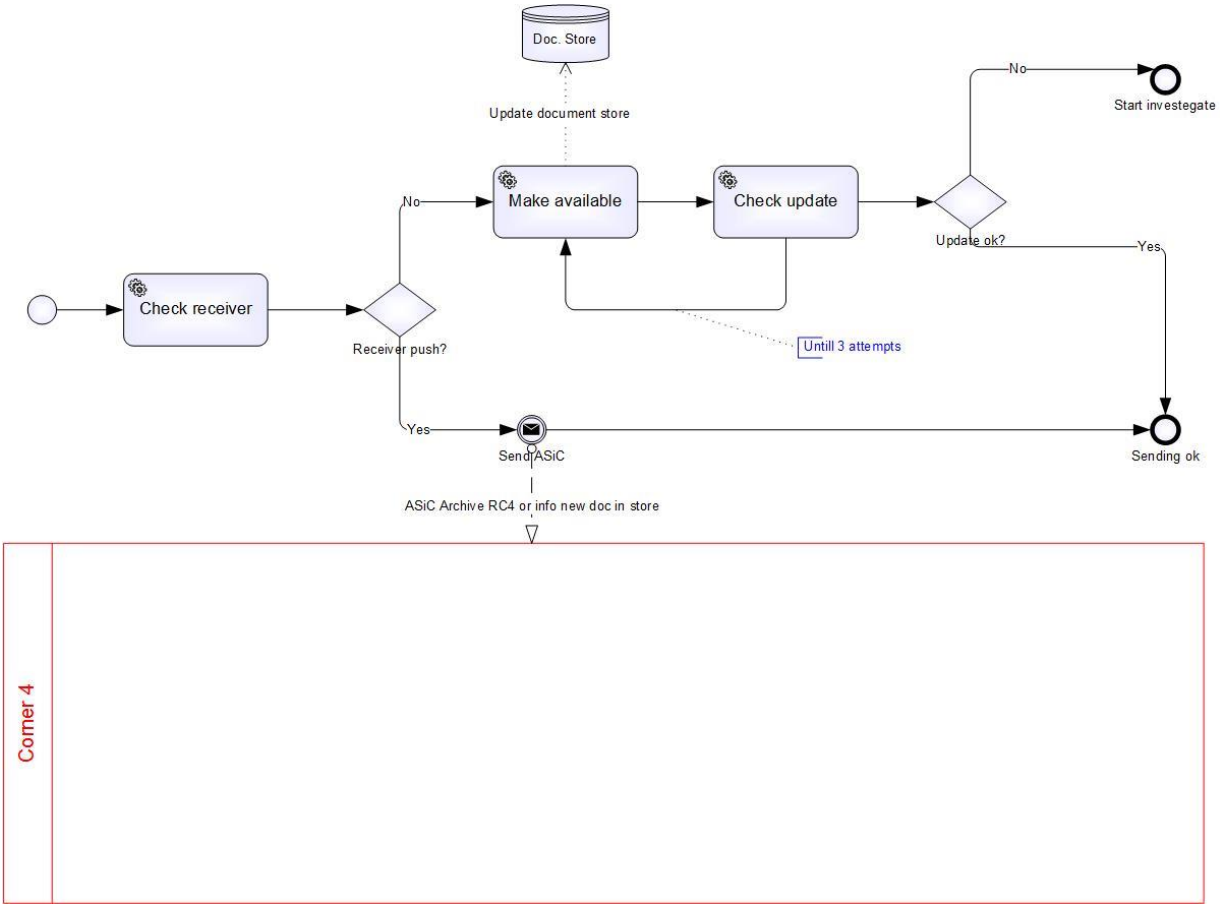


Collaboration diagram Verify ASiC from C2

The sub-process "**Verify ASiC from C2**" starts when "**MDN evidence**" is finished. The task "**Check outer SBDH**" will check if the outer SBDH is correct. If there is any error in the SBDH, the end state will be "**ASiC not ok**". The sub-process will end and return to main process, there the end state will be tested and Corner 3 will end the main process. Based on the negative MDN, both Corner 2 and Corner 1 will end the main process. Corner 1 needs to determine why the end state is "**Sending not ok**"

If the SBDH was correct the end state will be "**ASiC ok**" and the sub-process will return to main process.

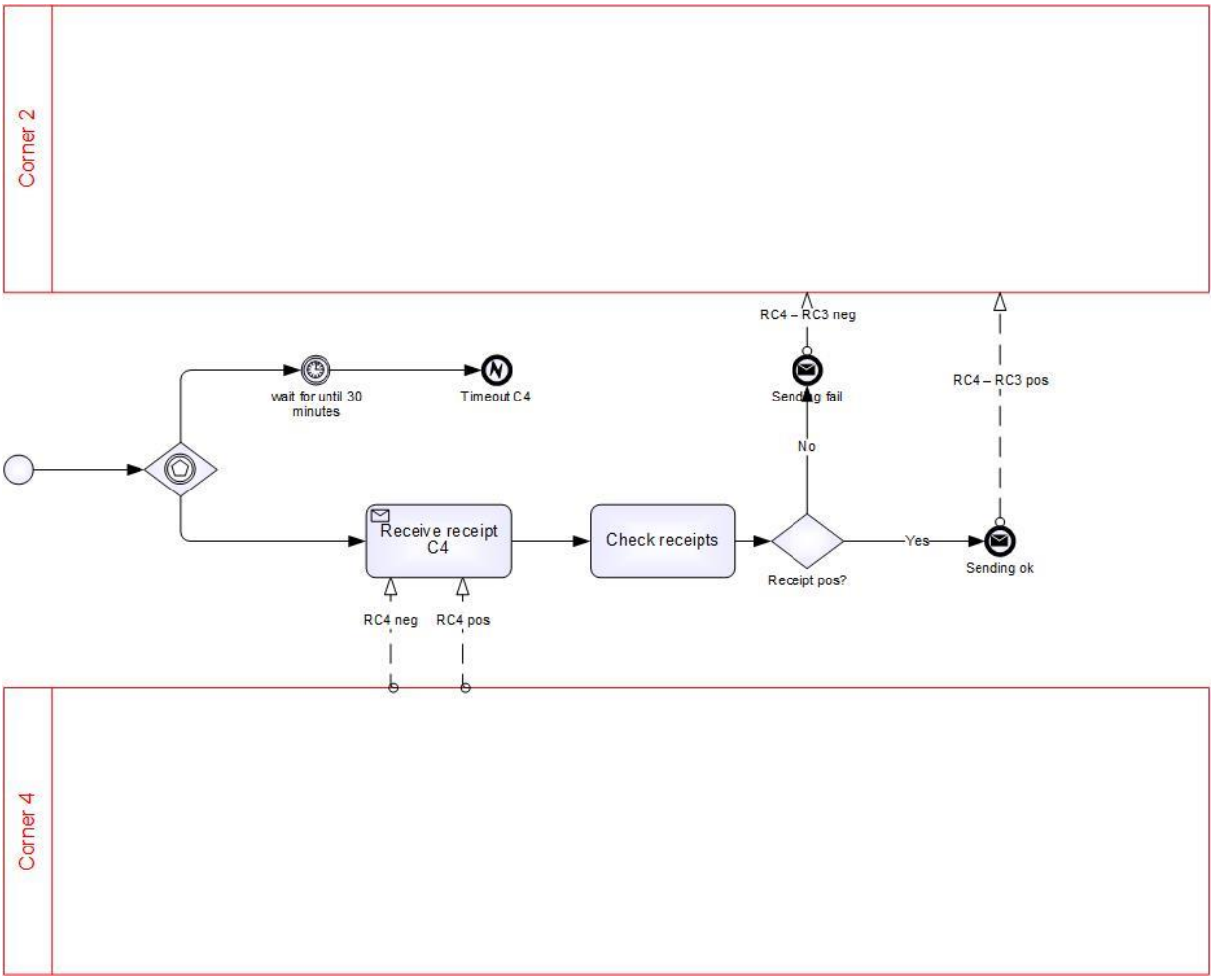
5.4.3 Diagram – ASiC available for C4



Collaboration diagram ASiC available for C4

The sub-process "**ASiC available for C4**" starts with task "**Check receiver**". The purpose of this is to determine whether the receiver is a push or pull receiver of documents.
If Corner 4 is a push receiver, Corner 3 sends the ASiC archive to Corner 4 and sets the end state to "**Sending ok**". The sub-process will end and return to main process.
If Corner 4 is a pull receiver, Corner 3 will start the task "**Make available**". This task will update the document store with the ASiC archive. It is assumed that Corner 3 has a retry function (try to upload document until 3 attempts), if not successful. The next task is "**Check update**" and this will perform an extra control to be sure that the document store is "update ok". If not updated, the end state will be "**Start investigate**". Corner 3 needs to investigate why the document store is not updated and return to main process.
If "update ok", the end state will be "**Sending ok**". The sub-process will end and return to the main process.

5.4.4 Diagram – Handle receipts C4



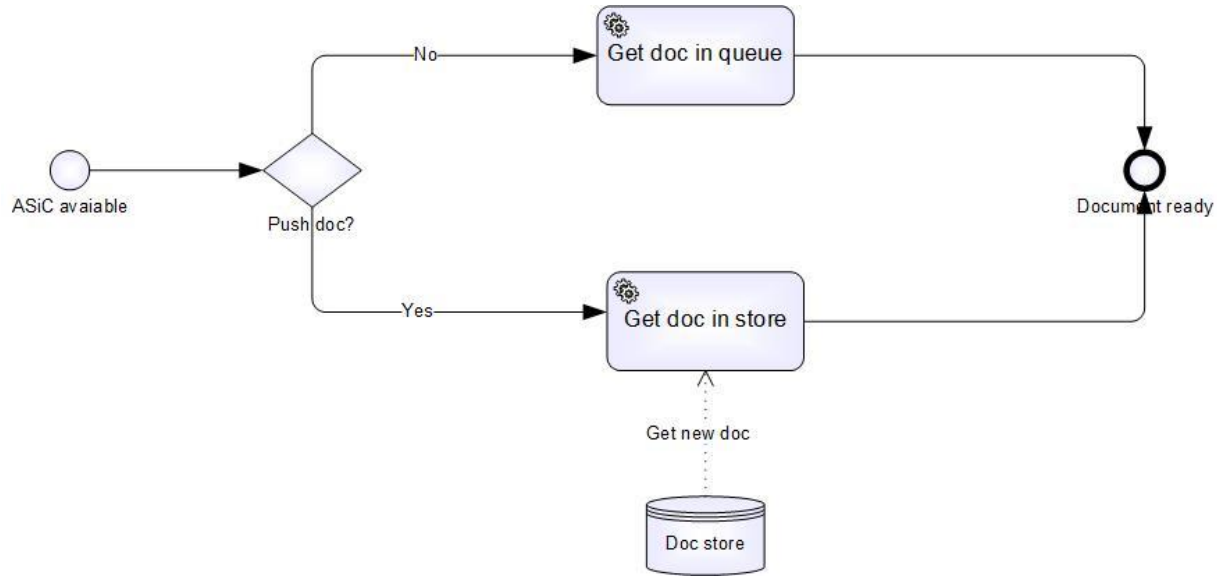
Collaboration diagram Handle receipts C4

After finishing sub-process "**ASiC available for C4**" the sub-process "**Handle receipts C4**" is a race against the clock. If Corner 4 is not sending a receipt within e.g. 30 minutes, the end state is error "**Timeout C4**". The timer started when corner 3 sent the ASiC archive to corner 4, and corner 1 and corner 2 should have started the inquiry for receipts. The sub-process will return to main process and Corner 3 needs to investigate the sending to Corner 4.
The sub-process will receive a positive or negative "**RC4**" and Corner 3 sends a negative or positive "**RC4**" to Corner 2. The sub-process will end with state "**Sending ok**" and so will the main process.

5.5 Process for corner 4

Corner 4 receives the ASiC archive either through a push or pull. It will check the ASiC-E archive and send a receipt back to Corner 3.

5.5.1 Diagram – Push or Pull ASiC from C3

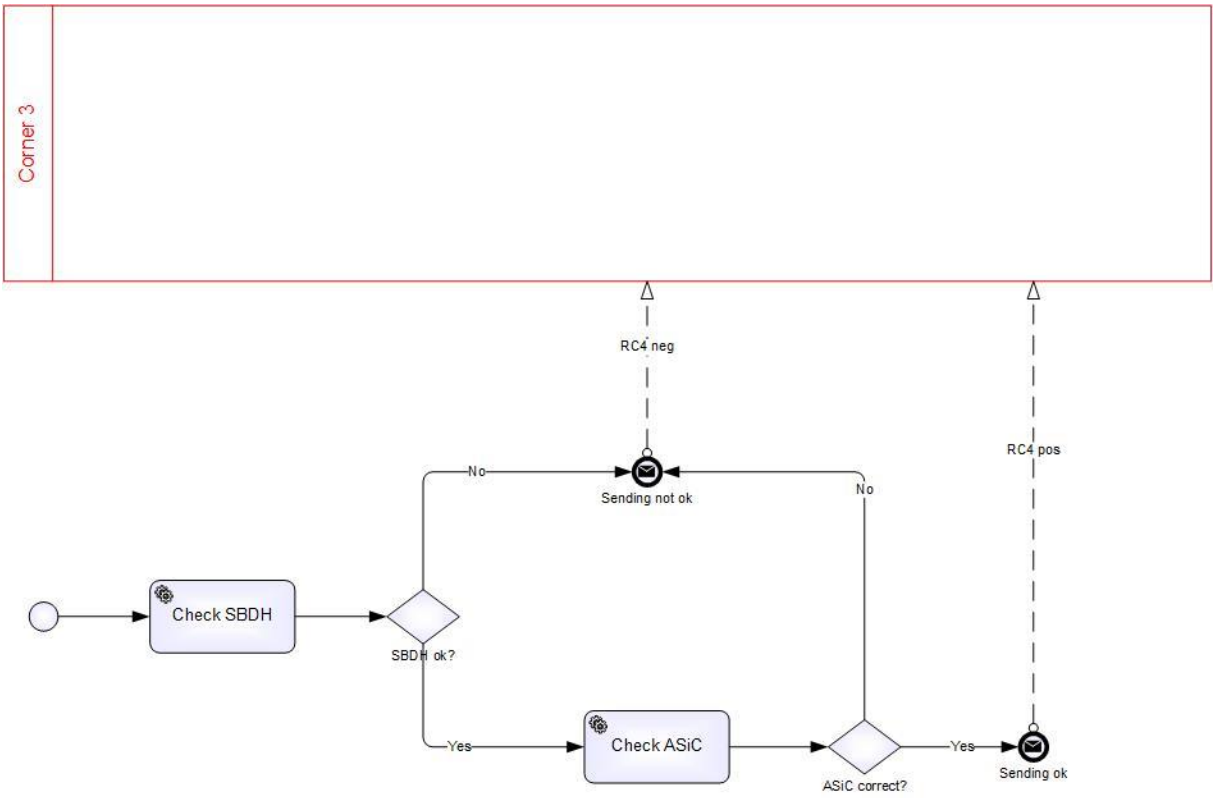


Collaboration diagram Push or Pull ASiC from C3

The sub-process "**Push or Pull ASiC from C3**" starts with a test. If the ASiC archive is sent, the task "**Get doc in queue**" will make the ASiC archive ready for verifying and end the sub-process.

If the document is available in document store, "**Get doc in store**" will pick up the ASiC archive, make it ready for verification and end the sub-process.

5.5.2 Diagram – Verify ASiC from C3



Collaboration diagram Verify ASiC from C3

Sub-process “Verify ASiC from C3” starting with task “**Check SBDH**”. If SBDH is not ok, the sub-process will send a negative “**RC4**” and set the end state to “**Sending not ok**”. This will end the sub-process and the main process.

If the SBDH is ok, the task “**Check ASiC**” will start. If the ASiC archive is not ok, the sub-process will send a negative “**RC4**” and set the end state to “**Sending not ok**”. This will end the sub-process and the main process.

If the ASiC archive is ok, the sub-process will send a positive “**RC4**” and set the end state to “**Sending ok**”. This will end the sub-process and the main process successfully.

6 Security requirements from the eIDAS regulation

REGULATION (EU) No 910/2014 on electronic identification and trust services for electronic transactions in the internal market (the "eIDAS Regulation") applies to the PEPPOL eDelivery network.
Even though the enhanced PEPPOL eDelivery network is not a qualified trust service in terms of the eIDAS Regulation, it does comply with requirements that will apply to a qualified trust service, in particular the security requirements. We have highlighted the most important requirements in the table below.

6.1 Security requirements

The eIDAS Regulation sets out security requirements for trust services and trust service providers. The enhanced PEPPOL eDelivery network is designed to be compliant with the security requirements in the regulation. The table below illustrate which part of the network solution that is relevant for complying with the respective Articles in the regulation.

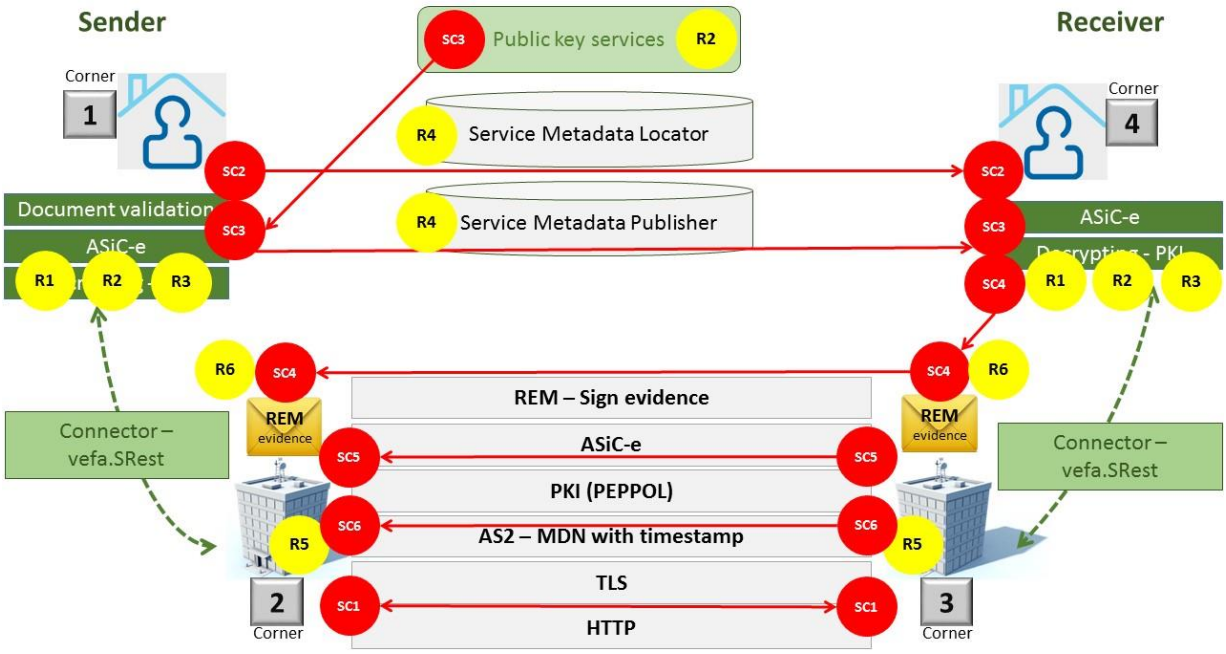
No	Requirement	Description	eIDAS reference	Solution
1	Message Integrity	Messages should be secured against any modification during transmission.	Article 3 (36) Article 19 Article 44, (d) the sending and receiving of data is secured by an advanced electronic signature or an advanced electronic seal of a qualified trust service provider in such a manner as to preclude the possibility of the data being changed undetectably.	ASiC-e archive is sign by the sender. Payload is encrypted with receivers public key from the company certificate
2	Message Confidentiality	Messages should be encrypted during transmission.	Article 5 Article 19	ASiC-e archive is sign by the sender. Payload is encrypted with receivers public key from the company certificate
3	Sender Identification	The identity of the sender should be verified.	Article 44, (b) they ensure with a high level of confidence the identification of the sender.	ASiC-e archive is sign by the sender.
4	Recipient / Addressee Identification	Recipient / address Identity should be verified before the delivery of the message.	Article 24 Article 44, (c) they ensure the identification of the addressee before the delivery of the data	Sender and receiver is validated in the SBDH Sending access point uses PEPPOL PKI and the SMP uses PEPPOL PKI
5	Time-Reference	The date and time of sending and receiving a message should be indicated via a qualified electronic timestamp.	Article 44, (f) the date and time of sending, receiving and any change of data are indicated by a qualified electronic time stamp.	Extended MDN have qualified timestamp
6	Proof of Send/Receive	Sender and receiver of the message should be provided with evidence of message sending and receiving.	Article 3 (36) "… provides evidence relating to the handling of the transmitted data, including proof of sending and receiving the data…"	REM evidence for both sending access point and Receiving access point All participants must notify sender when receiving a message

6.2 Security controls

To ensure security functions conformant with the eIDAS regulation there is established security controls in the enhanced PEPPOL eDelivery network. The security controls are presented in the table below with a description and reference to the relevant legal instruments.

No	Security control	Legal implications	Description
1	Transport Layer Security (TLS) and Authentication	European General Data Protection Regulation (GDPR) and local regulations for public sector	TLS protocols ensure confidentiality by applying host to host (Corner 2 and Corner 3) cryptographic mechanisms
2	Message Encryption end-to-end	European General Data Protection Regulation (GDPR) and local regulations for public sector	Message encryption (end-to-end) ensures confidentiality of the payload so that only the correct recipient can access it.
3	Electronic Seal of message	eIDAS regulation, Article 35: "A qualified electronic seal shall enjoy the presumption of integrity of the data and correctness of the origin of that data".	From technical perspective, quilified electronic seal ensures integrity of the message header and payload and authenticity of origin.
4	Electronic Seal of evidence	eIDAS regulation, Article 35: "A qualified electronic seal shall enjoy the presumption of integrity of the data and correctness of the origin of that data".	Provides evidence (signed MDN from Corner 3 to Corner 2) to the sender (Corner 1) that the message was sent and delivered to the final recipient (Corner 4) and authenticity of destination
5	Electronic Timestamp	eIDAS regulation, Article 41: "A qualified electronic time stamp shall enjoy the presumption of the accuracy of the date and the time it indicates and the integrity of the data to witch the date and time are bound".	Data in electronic form wich binds other data in electronic form to a particular time establishing evidence that the latter data existed that time.
6	Message Encryption	European General Data Protection Regulation (GDPR)	Message encryption ensures confidentiality of the payload between sending and receiving access point, so that only the correct recipient can access it.

6.3 Coherence between requirements and security controls and the enhanced PEPPOL eDelivery network



Requirements and security controls

The yellow circles are indicating requirements in the eIDAS Regulation.

Requirement number one, marked as R1, is about “message integrity”, message should be secured against any modification during transmission. The payload is put inside an ASiC-E archive; the ASiC-E archive is signed by the sender. If the payload is being changed the receiver will see that within the checksum. Responsibility is on both the sender and the receiver side (signing the message and validating the signature respectively).

Requirement number two, marked as R2, is about “message confidentiality”, message should be encrypted during transmission. The payload is encrypted before it is included in the ASiC-E archive, using the receiver’s public key to encrypt a one-time encryption key. The receiver’s public key is retrieved from the “Public key services”. Responsibility is placed both on sender and receiver side, the sender must use the public key of the receiver, and the receiver must upload the public key in the “Public key services” and use the private key for decryption of the one-time encryption key and subsequently the payload.

Requirement number three, marked as R3, is about “sender identification”, the identity of the sender should be verified. The sender is singing the ASiC-E archive with a private key supporte by a qualified certificate before sending. Responsible is both on sender and receiver side, the sender must use a qualified signature and the receiver must validate the signature.

Requirement number four, marked as R4, is about, “Receiver identification”, recipient address should be verified before the delivery of the message. Before sending the ASiC-E archive the sender must put in a SBD. SBD include the receivers address and the sender must retrieve the address from SML and SMP. Responsibility is on the sender side.

Requirement number five, marked as R5, is about “the date and time of sending and receiving a message should be indicated via a qualified electronic timestamp”. The enhanced PEPPOL profile of CEF eDelivery have upgraded MDN with timestamp. In the first version, the timestamp is not from a qualified source. Responsibility is placed on the receiving Access Point (Corner 3) by sending the MDN to the sending Access Point.

Requirement number six, marked as R6, is about “Proof of sending/receiving”, sender and receiver of the message should be provided with evidence of message sending and receiving. REM evidence must be generated of both the sending and receiving Access Point. The REM evidence is based on the MDN and is signed by both Access points.

The red circles are indicating security controls in the enhanced PEPPOL eDelivery network.

Security control number one, marked as SC1, is about “Transport Layer Security (TLS)”. TLS protocols ensure confidentiality, authenticity and integrity of message, by applying host to host cryptographic mechanisms. Responsibility is put on the sending and the receiving Access Points respectively.

Security control number two, marked as SC2, “Message Encryption end-to-end”. The end-to-end encryption ensures that the payload is encrypted end-to-end. Responsible is sender of the document.

Security control number three, marked as SC3, “Electronic seal of message”. From a technical perspective, an electronic seal ensures the integrity of the message header and payload and authenticity of origin. Use of qualified signatures must be used by the sender. Responsibility is placed on the sender. Corner 4 will validate the electronic seal is qualified.

Security control number four, marked as SC4, “Electronic Seal evidence”. Provides evidence to the sender that the message was sent, delivered to the final recipient and authenticity of destination. In the enhanced PEPPOL eDelivery network, corner 2, 3 and 4 must send a receipt back to the sender of the document. An electronic seal must be applied to the receipt. Responsibility is placed on the sending Access Point, receiving Access Point and the receiver.

Security control number five, marked as SC5, “Electronic timestamp”. Data in electronic form, witch binds other data in electronic form to a particular time, establishing evidence that the latter data existed at that time. The timestamp is copied in to the MDN and receiving and sending Access Point must store the REM evidence for at least one year. Responsibility is placed on sending and receiving Access Point.

Security control number six, marked as SC6, “Message encryption”. Message encryption ensures confidentiality of the payload so that only the correct receipt can access it. Responsibility is placed on the sending Access Point.

7 Security techniques applied to the ISO 20022-based payment messaging

7.1.1 Transport Layer Security (TLS)

The Transport Layer Security [9] protocol is used, following ENISA security [7] and BSI [8] guidelines. All versions starting with 1.0 and higher must be supported. When versions are deemed insecure must insecure versions be removed from use. Supporting all trusted versions allows for efficient communication also in a situation where one or more versions are deemed insecure.

7.1.2 Message Encryption

Corner 1 encrypts the payload of the message using the Corner 4 public key of a company certificate using AES-256 GCM.

7.1.3 Electronic seal of message

Corner 1 establishes an ASiC-E archive and signs the ASiC-E with the private key from the company certificate. Corner 4 uses its own private key from the company certificate which guarantees integrity protection.

7.1.4 Electronic Seal of evidence

The electronic seal is applied to the receipt. Upon receipt and verification of a message from Corner 2, Corner 3 generates an evidence receipt based on message identification information (e.g., message identifier, timestamp, and sender metadata) with a new timestamp and a reference to the received message, it then applies an electronic seal and returns the sealed evidence to Corner 2. The receipt is sent automatically to Corner 2 as a Message Disposition Notification (MDN) to the initial message.
Both Corner 2 and Corner 3 will have a REM evidence.

7.1.5 Electronic Timestamp

The timestamp is placed in the MDN, and it is electronically sealed for integrity protection. At this moment, by default, it is not qualified.

8 Migration Policy

The solution described in this document is based upon principle of loose coupling, allowing changes to the solution based on security reasons and improved solutions. A way of handling change is needed, as there will be needs to update standards used and change building blocks as the PEPPOL eDelivery network, and the world in general, evolves.

8.1 Types of change

Changes may be introduced as a result of

- security threats,
- better standards,
- depression of standards,
- new best practice,
- and new requirements.

8.2 Introducing change

Change is introduced in one of two ways:

- Direct change
- Managed change

8.2.1 Direct change

Direct change is when change is performed without further reflection in the network. This is the case especially in situations of security threats.

8.2.2 Managed change

Managed change is when change is performed by support of the SMP. This option should be the natural way of changing the network, and is done by updating the individual receiver's capabilities in the SMP.

Managed change is performed using four steps:

1. New feature is optional, old feature is mandatory.
2. Both new feature and old feature is mandatory.
3. New feature is mandatory, old feature is optional.
4. Old feature is no longer supported.

8.3 Timeline for performing change

Changes related to security (direct change) should be performed within a limited time based upon a risk assessment. Urgent security changes may require change within one hour.

Changes related to non-security features (managed change) is performed in between 6 and 24 months based upon assessment of work needed to complete migration.

9 External links

9.1 Secure building blocks as Open Source

9.1.1 Oxalis

Oxalis is an open source implementation of a PEPPOL AS2 Access Point.
https://github.com/difi/oxalis

9.1.2 ASiC-E archive

ASiC-E archive is the standard used for containerisation, as defined by ETSI.
https://github.com/difi/asic

9.1.3 SBDH

Library to support handling of SBDH.
https://github.com/difi/vefa-peppol/tree/master/peppol-sbdh

9.1.4 Vefa.SRest

Vefa.SRest is the PEPPOL connector proposed by Difi.
https://github.com/difi/vefa-srest

9.1.5 Look-up Difi Certificate Server

Certificate Server is expected to be available as Open Source later this year.

9.2 Other links

9.2.1 OpenPEPPOL

www.peppol.eu

9.2.2 Open Source library

https://github.com/difi/