# How to Setup a Peppol AS4 Access Point

Greg Kolinski
Sr. Developer
Loren Data Corp.

As you are probably aware the deadline for the Peppol transition to AS4 is fast approaching. For those stilling needing to get certified or for anyone looking to create their own access point, here is a brief overview of what Loren Data Corp. went through to get AS4 certified for the Peppol communication network. It is intended to act as a rough guide or outline containing some general information about our Peppol connection process. You will have to handle the nuances and special exceptions on your own.

Start by getting registered and receiving your testing certificate, just follow the setup guide. I will be addressing the implementation and acceptance testing portion of the guide.

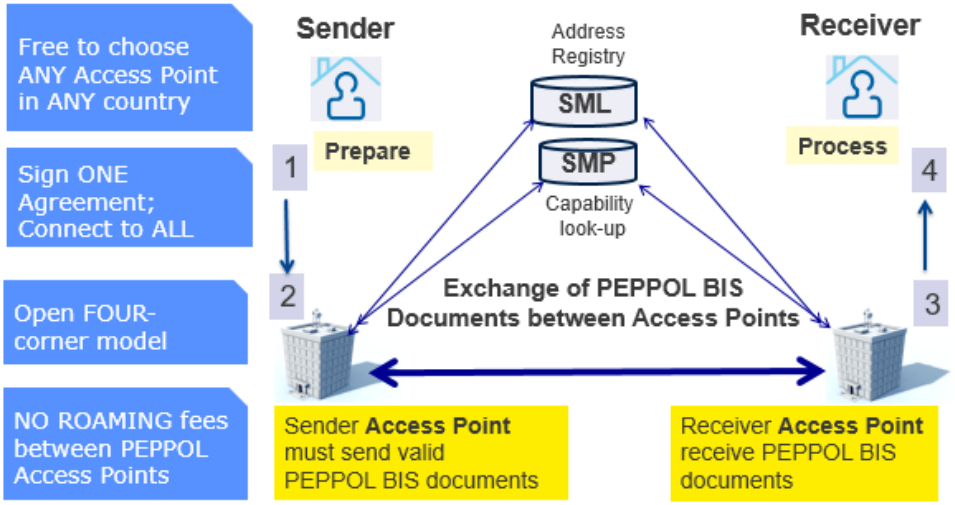
https://peppol.eu/wp-content/uploads/2019/07/How-to-set-up-a-Post-Award-PEPPOL-Access-Point_v1.4.pdf

## The 4-Corners

Peppol AS4 follows the 4-corner model and relies on the Peppol SMP for dynamic discovery of the endpoint information and certificates used in communication. This information is found in the SBDH (<StandardBusinessDocumentHeader>) of the AS4 payload when sending documents and it can be found in the XML SoapHeader/Messaging/UserMessage section when receiving an AS4 payload.

If you are unfamiliar with the 4-corner model, it would benefit you to learn, as it is referred to in the documentation often.

https://peppol.eu/what-is-peppol/peppol-transport-infrastructure/





©2020, Loren Data Corp.
POWERED BY INNOVATION

The CEF Digital website on eDelivery has all the documentation you could need on this. As you go through the AS4 documentation you will find out that this is the specification that Peppol is based on or extended from. If you are familiar with the CEF standards you are ahead of the game already.

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Message+exchange

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+AS4+-+1.14#eDeliveryAS4-1.14-FourCornerTopology

## SMP Lookup

If you haven't dealt with the SMP dynamic discover portion before, here are some basics that might help you. The goal is to dynamically discover the access point Corner1 or Corner4 is using, you need to get the AS4 ID, URL, and certificate to use in communication. Please refer to the Peppol documentation for a more in-depth explanation.

https://github.com/OpenPEPPOL/documentation/blob/master/TransportInfrastructure/ICT-Transport-SML_Service_Specification-101.pdf

https://github.com/OpenPEPPOL/documentation/blob/master/TransportInfrastructure/ICT-Transport-SMP_Service_Specification-110.pdf

As stated in the documentation the first thing to do is to establish the location of the Service Metadata. This typically is a listing for the document capabilities for the Peppol Participant's ID (PPID) and the SMP URL for the needed Metadata for each document type.

To get it, start by assembling the dynamic URL for a listing, you only need the PPID and PPID-Scheme. You can include the document type to go straight to the Service Metadata information.

**Remember to URL Encode the dynamic variables.*

Listing

"http://B-" + hexstring(md5(lowercase(PPID-VALUE))) + "." + PPID-SCHEME + "." + SML-ZONE-NAME + "/" + PPID-SCHEME + "::" + PPID-VALUE

Assuming PPID: `<Identifier Authority="iso6523-actorid-upis">9915:test</Identifier>`
Production SML Zone = edelivery.tech.ec.europa.eu
Test SML Zone = acc.edelivery.tech.ec.europa.eu

You should end up with something like:

> http://B-85008b8279e07ab0392da75fa55856a2.iso6523-actorid-upis.acc.edelivery.tech.ec.europa.eu/iso6523-actorid-upis%3A%3A9915%3Atest

As you can see this is a collection of SMP URLs for each of the documents that are serviced. Match the path with the document type you need, and you have the SMP Service Metadata final endpoint.

## Direct

Directly getting the Service Metadata information is the same as the listing with the addition of the document type appended to the end of the dynamic URL.


©2020, Loren Data Corp.

```
"http://B-" + hexstring(md5(lowercase(PPID-VALUE))) + "." + PPID-SCHEME + "." + SML-ZONE-NAME + "/" + PPID-SCHEME + "::" + PPID-VALUE + "/services/" + DOCUMENT-IDENTIFIER + "::" + DOCUMENT-IINSTANCEIDENTIFIER
```

If the SBDH does not have a Scope/Type(DOCUMENTID)/Identifier use "busdox-docid-qns"

```
<BusinessScope>
  <Scope>
    <Type>DOCUMENTID</Type>
    <InstanceIdentifier>urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.peppol.eu:bis:peppol5a:ver2.0::2.1
    </InstanceIdentifier>
```

You should end up with something like:

> http://B-85008b8279e07ab0392da75fa55856a2.iso6523-actorid-upis.acc.edelivery.tech.ec.europa.eu/iso6523-actorid-upis%3A%3A9915%3Atest/services/busdox-docid-qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice-2%3A%3AInvoice%23%23urn%3Awww.cenbii.eu%3Atransaction%3Abiitrns010%3Aver2.0%3Aextended%3Aurn%3Awww.peppol.eu%3Abis%3Apeppol5a%3Aver2.0%3A%3A2.1

The result is the SMP Service Metadata you need to make the AS4 HTTP request.

To construct the Hex string after MD5 hashing for .Net you can use the following…

```
System.Text.StringBuilder hexString = new System.Text.StringBuilder();

// Create hash object
using (System.Security.Cryptography.MD5 hasher =
System.Security.Cryptography.MD5.Create()){
	// Convert string to byte array and get hash
	byte[] bytes =
hasher.ComputeHash(System.Text.Encoding.UTF8.GetBytes(string2hash));

	// Convert hashed byte data to hex string
	for (int i = 0; i < bytes.Length-1; i++){
		hexString.Append(bytes[i].ToString("X2"));
	}
}
return hexString.ToString();
```

Check out how the DNS portion of the SML/SMP process works by swapping out the computed hash and SML Zone portion with the SMP domain from the document listing. Notice they both point to the same place.


©2020, Loren Data Corp.

With SML Domain

> http://B-85008b8279e07ab0392da75fa55856a2.iso6523-actorid-upis.acc.edelivery.tech.ec.europa.eu/iso6523-actorid-upis%3A%3A9915%3Atest/services/busdox-docid-qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice-2%3A%3AInvoice%23%23urn%3Awww.cenbii.eu%3Atransaction%3Abiitrns010%3Aver2.0%3Aextended%3Aurn%3Awww.peppol.eu%3Abis%3Apeppol5a%3Aver2.0%3A%3A2.1

With SMP Domain

> http://test-infra.peppol.at/iso6523-actorid-upis%3A%3A9915%3Atest/services/busdox-docid-qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice-2%3A%3AInvoice%23%23urn%3Awww.cenbii.eu%3Atransaction%3Abiitrns010%3Aver2.0%3Aextended%3Aurn%3Awww.peppol.eu%3Abis%3Apeppol5a%3Aver2.0%3A%3A2.1

After obtaining the Service Metadata you can parse the required information for AS4 communication by finding the matching ProcessID and Transport protocol (peppol-transport-as4-v2_0). Listed for each protocol is the EndpointReference/Address (Access Point) URL to send the AS4 Request to and the Public Certificate for signing/encryption depending on if you are sending or receiving.

There is more to the SML/SMP, with redirects, registering, updating, and switching SMPs, but this covers the basics.

## Peppol AS4 Profile

You may notice that the Peppol AS4 documentation [Peppol AS4 Profile] is limited in scope. As stated it is simply an extension of CEF eDelivery AS4 Profile v1.14 [CEFeDeliveryAS4] which in turn is a modular profile of the ebMS3 and AS4 OASIS specifications. While this documentation extension method limits what Peppol needs to maintain, I found myself constantly searching up and down the different technical documents to find exactly what I needed.

If you are already CEF AS4 compliant this process might be easier for you to implement, but for us, we were starting from starch and only had practical experience with AS2. While having the technical documentation is good, what I would have really found helpful was examples. There is a lot of reading and a lot to sort through.

The good news is that the Peppol requirement deals with one AS4 exchange pattern, the One-Way Push. This is very similar to how AS2 works with HTTP messages being sent over TLS 1.2 using IDs and certificates (In a SOAP Header instead of the Request Header) and a receipt (like an MDN) being returned.

If your AS4 library or software is able, you can set the AS4 profile to CEF eDelivery e-SENS and save some configuration work. Otherwise the basic configuration we used is listed below.

> Require Encryption: True
> Encryption Algorithm: AES128GCM
> Payload Compression Format: GZIP
> Payload: MIME Attachments (Not in the SOAP Body)
> Require Signature: True


©2020, Loren Data Corp.

Signature Algorithm: SHA256
OAEP RSA and MGF1 Hash Algorithms: SHA256
Encryption and Signing Security Token Format: Binary

## AS4 Sending Procedure

For sending documents we broke the process down into 3 basic parts, the information from the SMP, the AS4 Request, and the AS4 Receipt.

1. Just like in the Peppol AS2 process when your outside corner (C1) wants to send a document through your access point (C2) you need to gather the necessary information using the SMP. Our first step was to parse this information from the document SBDH. We get the Sender/Receiver Authority and Identifier as well as the DocumentID InstanceIdentifier and ProcessID InstanceIdentifier.

Example SBDH

```
<StandardBusinessDocumentHeader>
  <HeaderVersion>1.0</HeaderVersion>
  <Sender>
   <Identifier Authority="iso6523-actorid-upis">0088:LORENDATA</Identifier>
  </Sender>
  <Receiver>
   <Identifier Authority="iso6523-actorid-upis">0088:TBCNTRL00002</Identifier>
  </Receiver>
  <DocumentIdentification>
   <Standard>urn:oasis:names:specification:ubl:schema:xsd:Invoice-2</Standard>
   <TypeVersion>2.1</TypeVersion>
   <InstanceIdentifier>POP000276-14-20200110T173241</InstanceIdentifier>
   <Type>Invoice</Type>
   <CreationDateAndTime>2020-01-10T17:32:41.587Z</CreationDateAndTime>
  </DocumentIdentification>
  <BusinessScope>
   <Scope>
    <Type>DOCUMENTID</Type>
    <InstanceIdentifier>urn:oasis:names:specification:ubl:schema:xsd:Invoice-
2::Invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.peppol.eu:bis:peppol5a:ver2.0::2.1</InstanceIdent
ifier>
   </Scope>
   <Scope>
    <Type>PROCESSID</Type>
    <InstanceIdentifier>urn:www.cenbii.eu:profile:bii05:ver2.0</InstanceIdentifier>
   </Scope>
  </BusinessScope>
 </StandardBusinessDocumentHeader>
```

Refer to the above section for the SMP process. For Sending lookup the Receivers PPID. You will need match the Document and Process Identifier to get the EndpointReference/Address and the Certificate from the Endpoint transportProfile for peppol-transport-as4-v2_0.


©2020, Loren Data Corp.

Example SMP

```
<smp:ServiceInformation>
 <id:ParticipantIdentifier scheme="iso6523-actorid-upis">0088:tbcntrl00002</id:ParticipantIdentifier>
 <id:DocumentIdentifier scheme="busdox-docid-qns">
urn:oasis:names:specification:ubl:schema:xsd:Invoice-
2::Invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.peppol.eu:bis:peppol5a:ver2.0::2.1
 </id:DocumentIdentifier>
 <smp:ProcessList>
  <smp:Process>
   <id:ProcessIdentifier scheme="cenbii-procid-ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</id:ProcessIdentifier>
   <smp:ServiceEndpointList>
   <smp:Endpoint transportProfile="peppol-transport-as4-v2_0">
    <wsa:EndpointReference>
     <wsa:Address>https://proxy.peppol.eu/proxy/as4</wsa:Address>
    </wsa:EndpointReference>
    <smp:RequireBusinessLevelSignature>false</smp:RequireBusinessLevelSignature>
    <smp:MinimumAuthenticationLevel/>
    <smp:ServiceActivationDate>2019-07-01T00:00:00.000</smp:ServiceActivationDate>
    <smp:ServiceExpirationDate>2021-12-31T00:00:00.000</smp:ServiceExpirationDate>
    <smp:Certificate>MIIFwjC…</smp:Certificate>
    <smp:ServiceDescription>OpenPEPPOL Operations Oxalis AS4</smp:ServiceDescription>
    <smp:TechnicalContactUrl>mikael+openpeppol@aksamit.se</smp:TechnicalContactUrl>i
    <smp:TechnicalInformationUrl>https://testbed.peppol.eu</smp:TechnicalInformationUrl>
   </smp:Endpoint>
   </smp:ServiceEndpointList>
  </smp:Process>
 </smp:ProcessList>
</smp:ServiceInformation>
```

2. With all the information gathered you can construct the AS4 Request.

The Peppol AS4 documentation does a pretty good job of laying out what static values need to be filled. The AS4To and AS4From PartyId come from the Certificate Subjects (CN) use the receivers CN from their public certificate for the AS4To and your Peppol Access Point Certificate CN for the AS4From. The rest of the information is static or from the SBDH and SMP.

You need to create 2 custom message properties, the orginalSender and finalRecipient hold the Peppol Participant IDs and scheme from the SBDH. Other needed values are the ProcessIdentifier from the SMP as the Service, ProcessIdentifier scheme from the SMP as the Service Type, and the Action as a combination of the SMP DocumentIdentifier Scheme and the Document Identifier.

**Be sure the values are not URL Encoded*

This information is key for the recipient Access Point (C3), because the data/payload is encrypted they cannot use the SBDH to make the SMP lookup. Instead they will use the values set in the SOAP Header.

Filenames in the MIME Content-Disposition is not used or is defaulted to “payload.xml”

```
--NSMIMEBoundary__edb2d5af-c191-4b8d-8b21-957a1d572c5d
Content-Disposition: attachment; name="payload.xml"; filename="payload.xml"
Content-ID: <_a3d6d4a4-4a82-439e-9aae-c19cb503f090@nsoftware>
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```


©2020, Loren Data Corp.

Here is an example of the UserMessage for reference.

```
<eb3:UserMessage>
 <eb3:MessageInfo>
  <eb3:Timestamp>2020-01-10T18:54:39.551Z</eb3:Timestamp>
  <eb3:MessageId>614846880.559.0.2020011018542562@as4.peppolas4.azurewebsites.net</eb3:MessageId>
 </eb3:MessageInfo>
 <eb3:PartyInfo>
  <eb3:From>
   <eb3:PartyId type="urn:fdc:peppol.eu:2017:identifiers:ap">XXXXXXXX – CN from Cert your AP Peppol ID</eb3:PartyId>
   <eb3:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</eb3:Role>
  </eb3:From>
  <eb3:To>
   <eb3:PartyId type="urn:fdc:peppol.eu:2017:identifiers:ap">XXXXXXXX – CN from Cert found from SMP Lookup</eb3:PartyId>
   <eb3:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</eb3:Role>
  </eb3:To>
 </eb3:PartyInfo>
 <eb3:CollaborationInfo>
  <eb3:AgreementRef>urn:fdc:peppol.eu:2017:agreements:tia:ap_provider</eb3:AgreementRef>
  <eb3:Service type="cenbii-procid-ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</eb3:Service>
  <eb3:Action>busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:invoice-
2::invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.peppol.eu:bis:peppol5a:ver2.0::2.1</eb3:Action>
  <eb3:ConversationId>614846880.559.0.2020011018542562@as4.peppolas4.azurewebsites.net</eb3:ConversationId>
 </eb3:CollaborationInfo>
 <eb3:MessageProperties>
  <eb3:Property name="originalSender">iso6523-actorid-upis::0088:LORENDATA</eb3:Property>
  <eb3:Property name="finalRecipient">iso6523-actorid-upis::0088:TBCNTRL00002</eb3:Property>
 </eb3:MessageProperties>
 <eb3:PayloadInfo>
  <eb3:PartInfo href="cid:_a3d6d4a4-4a82-439e-9aae-c19cb503f090@nsoftware">
   <eb3:PartProperties>
    <eb3:Property name="CompressionType">application/gzip</eb3:Property>
    <eb3:Property name="MimeType">application/xml</eb3:Property>
    <eb3:Property name="CharacterSet">utf-8</eb3:Property>
   </eb3:PartProperties>
  </eb3:PartInfo>
 </eb3:PayloadInfo>
</eb3:UseriMessage>
```

Use your Peppol Access Point Private certificate to sign the outgoing message/file and the SMP public certificate for the receiver to encrypt it.

3. Finally, once the AS4 request is sent to the receiving access point (C3) we receive back a Receipt in the HTTP response as a Signal Message and match the RefToMessageID to the MessageInfo/MessageId from our request. Use the SMP receiver certificate to verify the signed receipt.

```
<eb:SignalMessage xmlns="" xmlns:ns3="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
xmlns:ns4="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns6="http://www.w3.org/2000/09/xmldsig#"
xmlns:ns7="http://docs.oasis-open.org/ebxml-bp/ebbp-signals-2.0">
 <eb:MessageInfo>
  <eb:Timestamp>2020-01-10T18:54:44.645Z</eb:Timestamp>
  <eb:MessageId>78297a93-c882-48e3-8208-3b6ae74d3d89@ip-10-10-208-232.eu-west-1.compute.internal</eb:MessageId>
  <eb:RefToMessageId>614846880.559.0.2020011018542562@as4.peppolas4.azurewebsites.net</eb:RefToMessageId>
 </eb:MessageInfo>
 <eb:Receipt>
  <ns7:NonRepudiationInformation>.....
```


©2020, Loren Data Corp.

## AS4 Receiving Procedure

For the receiving procedure we followed another 3-part breakdown, read the request (SoapHeader), lookup the SMP information, and send the Receipt.

1. We started by reading the AS4 HTTP request sent (from C2) to our access point (C3). Check that it is a Peppol Agreement using the AgreementRef value and then parse out all the values needed for a SMP lookup. The MPC is either not present or defaulted. Service and Service Type for Process information, Action for Document Information and the Message Properties for the PPIDs. Refer to the Peppol AS4 Profile for exact values.

Example UserMessage from Received Request

```
<ns2:UserMessage>
 <ns2:MessageInfo>
  <ns2:Timestamp>2020-01-10T17:32:41.916Z</ns2:Timestamp>
  <ns2:MessageId>807a3bba-5a6d-4bc9-980b-d4baf2afe005@ip-10-10-208-232.eu-west-1.compute.internal</ns2:MessageId>
 </ns2:MessageInfo>
 <ns2:PartyInfo>
  <ns2:From>
   <ns2:PartyId type="urn:fdc:peppol.eu:2017:identifiers:ap">XXXXXXXX</ns2:PartyId>
   <ns2:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</ns2:Role>
  </ns2:From>
  <ns2:To>
   <ns2:PartyId type="urn:fdc:peppol.eu:2017:identifiers:ap">XXXXXXXX</ns2:PartyId>
   <ns2:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</ns2:Role>
  </ns2:To>
 </ns2:PartyInfo>
 <ns2:CollaborationInfo>
  <ns2:AgreementRef>urn:fdc:peppol.eu:2017:agreements:tia:ap_provider</ns2:AgreementRef>
  <ns2:Service type="cenbii-procid-ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</ns2:Service>
  <ns2:Action>busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-
2::Invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.peppol.eu:bis:peppol5a:ver2.0::2.1</ns2:Action>
  <ns2:ConversationId>c5311808-f847-40b8-8d29-afb0d9a0988d@ip-10-10-208-232.eu-west-
1.compute.internal</ns2:ConversationId>
 </ns2:CollaborationInfo>
 <ns2:MessageProperties>
  <ns2:Property name="originalSender">iso6523-actorid-upis::0088:tbcntrl00002</ns2:Property>
  <ns2:Property name="finalRecipient">iso6523-actorid-upis::0088:lorendata</ns2:Property>
 </ns2:MessageProperties>
 <ns2:PayloadInfo>
  <ns2:PartInfo href="cid:1bb0377a-0a49-4330-b9d7-80aa2667b271@ip-10-10-208-232.eu-west-1.compute.internal">
   <ns2:PartProperties>
    <ns2:Property name="CompressionType">application/gzip</ns2:Property>
    <ns2:Property name="MimeType">application/xml</ns2:Property>
   </ns2:PartProperties>
  </ns2:PartInfo>
 </ns2:PayloadInfo>
</ns2:UserMessage>
```

2. Next, we used that information to lookup the SMP for the originalSender PPID. Once we have the public certificate from the SMP we can use it verify the message/file signature and decrypt the payload with our Peppol Access Point private certificate.

Refer to the above section for the SMP process.


©2020, Loren Data Corp.

After decryption we decompress the Gzip payload for archiving and delivery to our outside corner (C4).

```
--uuid:efcf8e3a-ab82-4232-a781-7d9451364f5b
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <1bb0377a-0a49-4330-b9d7-80aa2667b271@ip-10-10-208-232.eu-west-1.compute.internal>
CompressionType: application/gzip
MimeType: application/xml
```

3. Finally, we send the Receipt signed with our access point private certificate back in the HTTP response as a SignalMessage referencing the MessageId from the request in the RefToMessageId.

```
<ns2:SignalMessage>
 <ns2:MessageInfo>
  <ns2:Timestamp>2020-01-10T16:05:49.336Z</ns2:Timestamp>
  <ns2:MessageId>_4f3fcc7d-b84a-43a4-8cc6-96154cdc214a@nsoftware</ns2:MessageId>
  <ns2:RefToMessageId>e6181d33-c480-4e5f-96b2-f4a682965a68@ip-10-10-208-232.eu-west-
1.compute.internal</ns2:RefToMessageId>
 </ns2:MessageInfo>
 <ns2:Receipt xmlns:ns2="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/">
 <ebbp:NonRepudiationInformation xmlns:ebbp="http://docs.oasis-open.org/ebxml-bp/ebbp-signals-2.0">....
```

## Testing

Using the Peppol Test Bed went smoothly, no real surprises, and it was fast to complete once everything was in place. Make sure you check your receiving endpoint for SSL compliance using their recommended tester https://www.ssllabs.com/ssltest/ if it is anything less than an “A” rating you fail automatically.

One thing to note is when sending back the files you receive, in addition to switching the sender and receiver ID in the SBDH, be sure to also switch the values in the document sections, even though it is not stated in the instructions at the beginning of the sending tests.

```
<cac:AccountingSupplierParty>
 <cac:Party>
  <cbc:EndpointID schemeID="GLN">LORENDATA</cbc:EndpointID>
  <cac:PartyIdentification>
      <cbc:ID schemeID="GLN">LORENDATA</cbc:ID>
  </cac:PartyIdentification>

<cac:AccountingCustomerParty>
 <cac:Party>
  <cbc:EndpointID schemeID="GLN">TBCNTRL00002</cbc:EndpointID>
  <cac:PartyIdentification>
   <cbc:ID schemeID="GLN">TBCNTRL00002</cbc:ID>
  </cac:PartyIdentification>
```

I would also recommend adding some additional logging and error catching into your process as the error results from the test bed are sometime generalized and don’t fully explain what the issue was.

However, once we had our report, submission and verification of AS4 certification took less than two business days.


©2020, Loren Data Corp.

## Summary

Overall migrating to AS4 and setting up an access point for Peppol was not a major ordeal. Our biggest difficulties with the migration came from the multiple technical documents we needed to drill down into and the lack of real examples to reference. There is only so much internal testing one can do, sending and receiving within an internal system, before needing an external access point and real requests. At the time of writing this AS4 is not a highly adopted communication protocol in the U.S., so options are limited. A basic walkthrough with examples in addition to the in-depth technical documentation would have helped me focus on what we needed to do. Hopefully this article will meet that need for you.


©2020, Loren Data Corp.