|  |
| --- |
|  |



European eInvoicing Standard in Italy

|  |  |
| --- | --- |
| **Project number** | INEA/CEF/ICT/A2017/1560867 2017-IT-IA-0150 |
| **Project acronym** | EeISI |
| **Project title** | European eInvoicing Standard in Italy |
| **Starting date** | 1 May 2018 |
| **Ending date** | 31 December 2019 |
| **Programme** | Connecting Europe Facility (CEF) CEF-TC-2017-3: eInvoicing |

Design of the eDelivery architecture for NoTI-ER

Deliverable D3.9

|  |  |
| --- | --- |
| **Related WP** | WP3 – Design – Task 3.3 |
| **Deliverable number** | D3.9 |
| **Due date** | 30/11/2018 |
| **Revision date** | 30/11/2018 |
| **Actual date** | 30/11/2018 |

This Page Intentionally Left Blank

Deliverable Info

|  |  |
| --- | --- |
| **Editor (s)** | **Intercent-ER** |
| **Contributors** |  |
|  |  |
|  |  |
| **Acknowledgement** | This work was partially supported by the European Commission (EC) through the Connecting Europe Facility (CEF) programme under project EeISI.(grant agreement no. INEA/CEF/ICT/A2017/1560867 2017-IT-IA-0150) |
| **Disclaimer** | The sole responsibility of this publication lies with the author(s). The European Union is not responsible for any use that may be made of the information contained therein. |
| **Confidentiality** | The information in this document is confidential and restricted only to the members of the EeISI consortium  (including the Commission Services). |
|  |  |
| **Note** | - |
|  |  |

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of change |
| 0.4 | 30/11/2018 | Intercent-ER |  |
| 0.4.1 | 31/03/2020 | Roberto Reale | Technical review and quality assessment |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

This Page Intentionally Left Blank

Table of contents

[Deliverable Info 3](#_Toc27738294)

[Table of contents 5](#_Toc27738295)

[Executive Summary 7](#_Toc27738296)

[Glossary 7](#_Toc27738297)

[1. Introduction 9](#_Toc27738298)

[2. eDelivery infrastructure – AS2/AS4 9](#_Toc27738299)

[2.1. Software architecture 9](#_Toc27738300)

[2.1.1. Inbound 9](#_Toc27738301)

[2.1.2. Outbound 10](#_Toc27738302)

[2.1.3. Oxalis Quartz scheduling 11](#_Toc27738303)

[2.2. Integration between NoTI-ER and Oxalis 11](#_Toc27738304)

[2.2.1. NoTI-ER “UrnList” REST web service 11](#_Toc27738305)

[2.2.2. NoTI-ER “SingleDocument” REST web service 11](#_Toc27738306)

[2.2.3. NoTI-ER “SendStatus” REST web service 12](#_Toc27738307)

[2.2.4. NoTI-ER “ReceiveDocument” REST web service 12](#_Toc27738308)

[2.2.5. Oxalis “QuartzConsole” Servlet web service 12](#_Toc27738309)

[2.2.6. Oxalis “Lookup” Servlet web service 13](#_Toc27738310)

[2.2.7. Oxalis “Outbound” Servlet web service 13](#_Toc27738311)

[3. Inbound Scenario for NoTI-ER – Cross-border B2G 13](#_Toc27738312)

[3.1. Receiving from PEPPOL Network 13](#_Toc27738313)

[3.2. Receiving from Italian Exchange System 14](#_Toc27738314)

[4. Outbound scenario for NoTI-ER 14](#_Toc27738315)

[4.1. Sending to Italian Exchange System 14](#_Toc27738316)

[4.1.1. Sending UBL invoice 14](#_Toc27738317)

[4.2. Sending to PEPPOL Network 14](#_Toc27738318)

[4.2.1. Sending IMR 15](#_Toc27738319)

This Page Intentionally Left Blank

Executive Summary

This document aims to describe the design of the eDelivery architecture for NoTI-ER, in relation to the EeISI project. The analysis mainly interests:

* eDelivery infrastructure – AS2/AS4;
* eDelivery infrastructure for the new scenarios:
  + inbound cross-border B2G;
  + outbound via SATER.

Glossary

|  |  |
| --- | --- |
| **B2B** | Business to Business |
| **B2C** | Business to Consumer/Citizen |
| **B2G** | Business to Government |
| **BII** | Business Interoperability Interfaces |
| **C2G** | Citizen to Government |
| **CCTS** | Core Component Technical Specification |
| **CEF** | Connecting Europe Facility |
| **CEM** | Certified Electronic Mail – Legal Mail (PEC Posta Elettronica Certificata in Italy) |
| **CEN** | European Committee for Standardisation |
| **CII** | Cross Industry electronic Invoice |
| **CIUS** | Core Invoice Usage Specification |
| **DSI** | Digital Service Infrastructures |
| **EDIFACT** | Electronic Data Interchange For Administration, Commerce and Transport |
| **EMSFEI** | European Multi-Stakeholder Forum on eInvoicing |
| **e-SENS** | Electronic Simple European Networked Services |
| **FatturaPA** | Public administration electronic invoice framework (FatturaPubblica Amministrazione) |
| **G2B** | Government to Business |
| **G2G** | Government to Government |
| **INEA** | Innovation and Networks Executive Agency |
| **NoTI-ER** | Nodo Telematico di Interscambio Emilia-Romagna (Telematic Interchange Node) |
| **OASIS** | Organization for the Advancement of Structured Information Standards |
| **PEPPOL** | Pan-European Public Procurement Online |
| **PEPPOL-BIS** | Pan-European Public Procurement Online Business Interoperability Specifications |
| **SATER** | Sistema Acquisti Telematici Emilia-Romagna (Telematic Purchasing System) |
| **SDI** | Electronic exchange system in Italy (Sistema Di Interscambio) |
| **UBL** | Universal Business Language |
| **UN/CEFACT** | United Nations Centre for Trade Facilitation and Electronic Business |
| **UNTDID** | UN Trade Data Interchange Directory |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Location |
| **URN** | Uniform Resource Name |
| **XML** | Extensible Mark-up Language |

1. Introduction

The following chapters show the changes implemented in the NoTI-ER eDelivery infrastructure to align with EeISI’s specific requests.

1. eDelivery infrastructure – AS2/AS4

The aim of this first section is to describe the technical structure whose NoTI-ER is composed by.

* 1. Software architecture

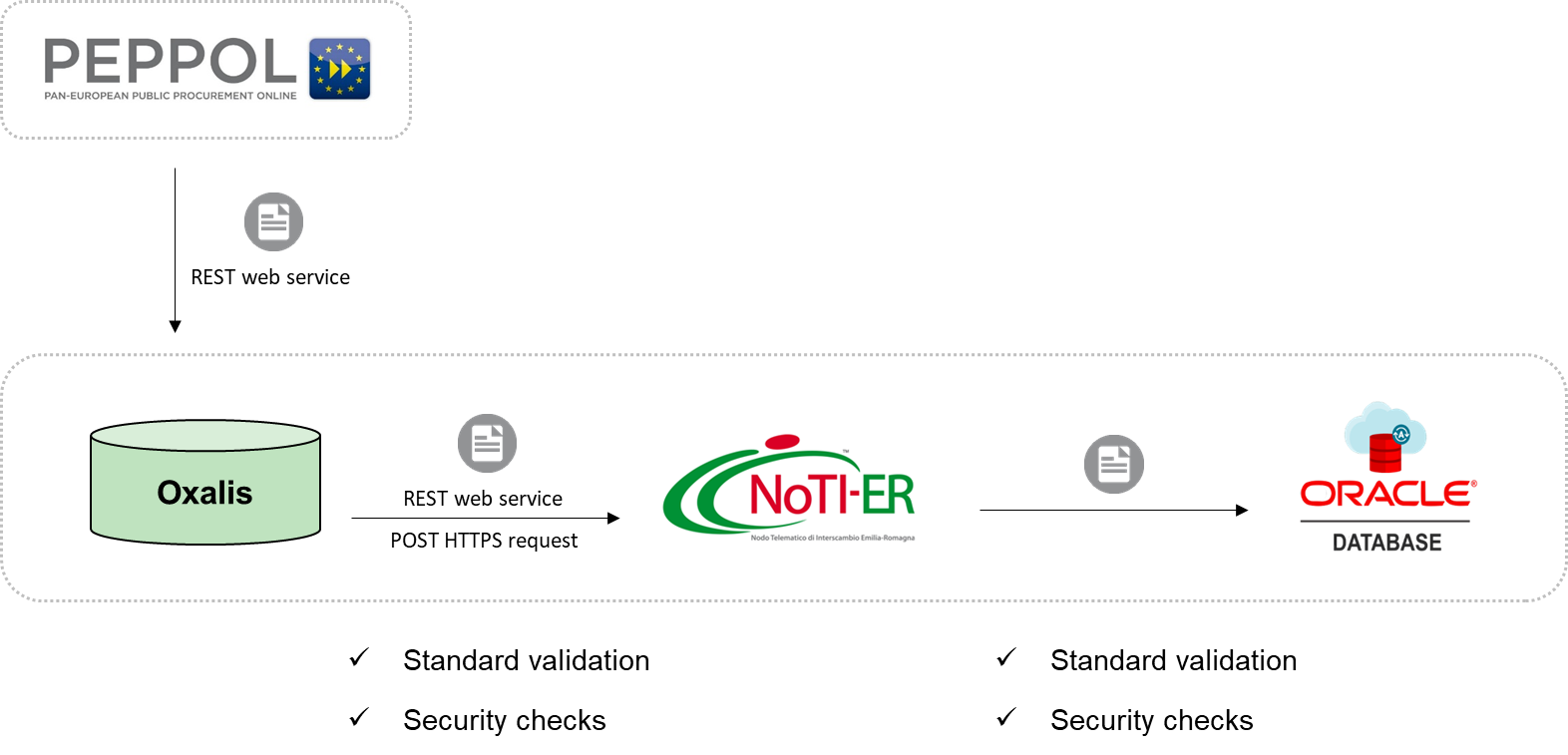
The software architecture of NoTI-ER has changed significantly. Since 2014 NoTI-ER kept Oxalis implementation integrated inside its own web application. Oxalis has now been unbundled from NoTI-ER. As a result, now we have set up two different web application.

* + 1. Inbound

The inbound workflow is handled both by Oxalis and by NoTI-ER. The goal of Oxalis is to receive document from PEPPOL Network, forwarding them to NoTI-ER, using REST web service in a synchronized way. The goal of NoTI-ER is to handle the documents received from Oxalis, persisting them into its own Oracle database.

The inbound workflow works as it follows:

* Oxalis receives a document from PEPPOL Network, performing standard validation and security checks;
* If document checks are passed, Oxalis builds a JSON String representing the received document and sends it on NoTI-ER composing a POST HTTPS request;
* NoTI-ER acquires the document and persists it on database, if XSD and SCHEMATRON validations ends up successfully;
* If NoTI-ER persists to fail due to some SQL errors or NoTI-ER unreachability, Oxalis provides a file system persist, sending an e-mail notification to NoTI-ER technical support reporting the issue.

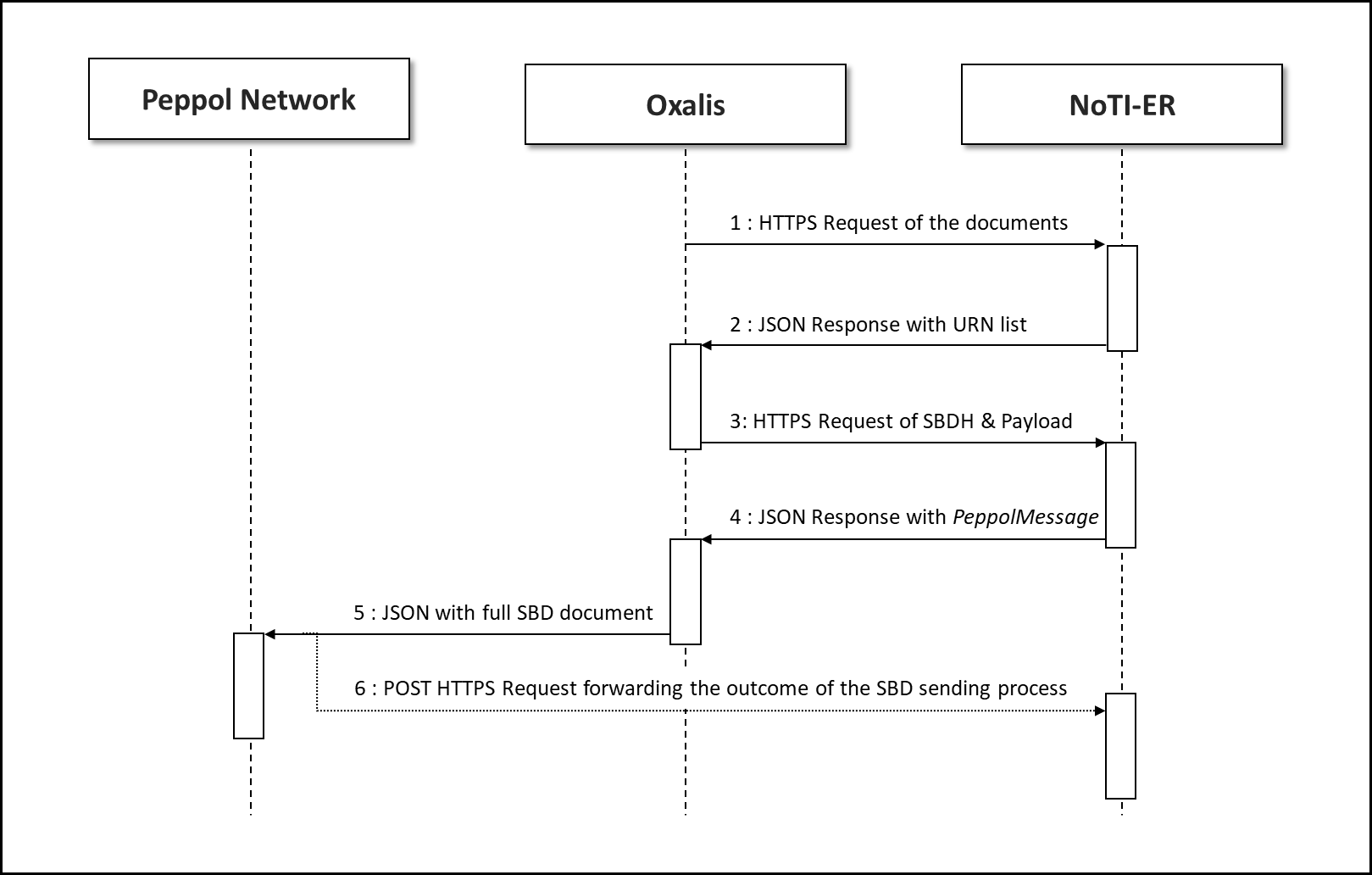


* + 1. Outbound

The outbound workflow is entirely handled by Oxalis. Oxalis uses Quartz (v. 2.2.1) in order to schedule the execution of the outbound workflow.

Oxalis outbound job works as it follows:

* Oxalis sends an HTTPS request to NoTI-ER, querying for documents to send on PEPPOL Network;
* NoTI-ER sends a JSON response to Oxalis, defining an URN list representing an index that lists documents that needs to be sent on PEPPOL by Oxalis itself;
* Using each URN of the retrieved list, Oxalis sends an HTTPS request to NoTI-ER querying for both the SBDH and payload, related to the document represented by the given URN;
* NoTI-ER sends a JSON response to Oxalis defining a *PeppolMessage* composed by three objects:
  + *PeppolDetails header* object defines SBDH (header information used by Oxalis in order to create the whole SBD to send on PEPPOL);
  + *Boolean performLookup* object (used by Oxalis in order to perform lookup against the SML or not) set to “true” as default value;
  + *ByteArrayInputStream payload* object that contains the whole byte sequence representing the document that needs to be sent (the document can be an invoice either IMR);
* Oxalis builds the full SBD document using the JSON previously received from NoTI-ER and sends it on PEPPOL Network;
* Oxalis catches the outcome of the SBD sending process and parses it in JSON String;
* Oxalis sends a POST HTTPS request to NoTI-ER, forwarding the outcome previously catched;
* Once NoTI-ER receives the outcome, it updates the document status on NoTI-ER Oracle database.



* + 1. Oxalis Quartz scheduling

Oxalis workflow is scheduled by Quartz (v. 2.2.1). The timing is set up to one minute. Every 60 seconds Oxalis starts the outbound workflow process described above in *2.1.2. Outbound* paragraph.

* 1. Integration between NoTI-ER and Oxalis

Integration between NoTI-ER and Oxalis is set up using REST web service technology. HTTP communications are secured under HTTPS protocol.

NoTI-ER REST web services are handled using Spring Web (version 3.2.7-RELEASE) Framework technology.

In the whole document workflow NoTI-ER acts as a server and Oxalis acts as a client, anyway Oxalis exposes three different web services using servlet technology. Oxalis servlets are intended to expose the functionalities of Dynamic Discovery Lookup, Static Discovery document outbound and Oxalis Quartz scheduler management (NoTI-ER can send commands to Oxalis Quartz in order to stop and resume the Outbound workflow scheduling).

* + 1. NoTI-ER “UrnList” REST web service

URL endpoint: [https://NoTI-ER-regione.emilia-romagna.it:8443/NoTI-ER/rest/v1.0/oxalis/urnList](https://notier-regione.emilia-romagna.it:8443/notier/rest/v1.0/oxalis/urnList)

Path parameters ({name} – {Java type} – {Description} – {Required}): none

Request headers ({name} – {Java type} – {Description} – {Required}):

* {X-FwdCertSubject\_0} – {java.lang.String} – {It represents the distinguish name related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true};
* {X-FwdCertSerialNumber\_0} – {java.lang.String} – {It represents the serial number related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true}.

*Notes: both the headers are automatically deducted from the PKCS12 client certificate used by Oxalis.*

Request method: GET

Produces: application/JSON

* + 1. NoTI-ER “SingleDocument” REST web service

URL endpoint: [https://NoTI-ER.regione.emilia-romagna.it:8443/NoTI-ER/v1.0/oxalis/singleDocument/{documentUrn}](https://notier.regione.emilia-romagna.it:8443/notier/v1.0/oxalis/singleDocument/%7bdocumentUrn%7d)

Path parameters ({name} – {Java type} – {Description} – {Required}):

* {documentUrn} – {java.lang.String} – {It represents the URN related to the document whose SBD JSON representation Oxalis is asking for} – {true}.

Request headers ({name} – {Java type} – {Description} – {Required}):

* {X-FwdCertSubject\_0} – {java.lang.String} – {It represents the distinguish name related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true};
* {X-FwdCertSerialNumber\_0} – {java.lang.String} – {It represents the serial number related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true}.

*Notes: both the headers are automatically deducted from the PKCS12 client certificate used by Oxalis.*

Request method: GET

Produces: application/JSON

* + 1. NoTI-ER “SendStatus” REST web service

URL endpoint: [https://NoTI-ER.regione.emilia-romagna.it:8443/NoTI-ER/v1.0/oxalis/sendStatus](https://notier.regione.emilia-romagna.it:8443/notier/v1.0/oxalis/sendStatus)

Path parameters ({name} – {Java type} – {Description} – {Required}): none

Request headers ({name} – {Java type} – {Description} – {Required}):

* {X-FwdCertSubject\_0} – {java.lang.String} – {It represents the distinguish name related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true};
* {X-FwdCertSerialNumber\_0} – {java.lang.String} – {It represents the serial number related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true}.

*Notes: both the headers are automatically deducted from the PKCS12 client certificate used by Oxalis.*

Request parameters ({name} – {Java type} – {Description} – {Required}):

* {oxalisContent} – {java.lang.String} – {It represents the JSON String containg the outcome related to the outbound process of the given document} – {true}.

Request method: POST

Produces: application/JSON

* + 1. NoTI-ER “ReceiveDocument” REST web service

URL endpoint: [https://NoTI-ER.regione.emilia-romagna.it:8443/NoTI-ER/v1.0/oxalis/receivingDocument](https://notier.regione.emilia-romagna.it:8443/notier/v1.0/oxalis/receivingDocument)

Path parameters ({name} – {Java type} – {Description} – {Required}): none

Request headers ({name} – {Java type} – {Description} – {Required}):

* {X-FwdCertSubject\_0} – {java.lang.String} – {It represents the distinguish name related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true};
* {X-FwdCertSerialNumber\_0} – {java.lang.String} – {It represents the serial number related to the PKCS12 client certificate used by Oxalis in order to communicate with NoTI-ER} – {true}.

*Notes: both the headers are automatically deducted from the PKCS12 client certificate used by Oxalis.*

Request parameters ({name} – {Java type} – {Description} – {Required}):

* {document} – {java.lang.String} – {It represents the JSON String containing the SBDH} – {true};
* {peppolPayload} – {java.lang.String} – {It represents the JSON String containing the payload of the document} – {true};
* {isInternal} – {java.lang.String} – {It represents the String version of the Boolean value (true/false), defines when the document is related to internal flow} – {true}.

Request method: POST

Produces: application/JSON

* + 1. Oxalis “QuartzConsole” Servlet web service

URL endpoint: [https://NoTI-ER.regione.emilia-romagna.it/oxalis/quartzConsole](https://notier.regione.emilia-romagna.it/oxalis/quartzConsole)

Path parameters ({name} – {Java type} – {Description} – {Required}): none

Content-Type: application/JSON

Entity: is the JSON String representation that defines the Oxalis Quartz command that needs to be executed on Oxalis side.

Request method: POST

Produces: application/JSON

* + 1. Oxalis “Lookup” Servlet web service

URL endpoint: [https://NoTI-ER.regione.emilia-romagna.it/oxalis/lookup](https://notier.regione.emilia-romagna.it/oxalis/lookup)

Path parameters ({name} – {Java type} – {Description} – {Required}): none

Content-Type: application/JSON

Entity: is the JSON String representation that defines PEPPOL details that needs to be queried on SML by Oxalis.

Request method: POST

Produces: application/JSON

* + 1. Oxalis “Outbound” Servlet web service

URL endpoint: [https://NoTI-ER.regione.emilia-romagna.it/oxalis/sendOutnbound](https://notier.regione.emilia-romagna.it/oxalis/sendOutnbound)

Path parameters ({name} – {Java type} – {Description} – {Required}): none

Content-Type: application/JSON

Entity: is the JSON String representation that defines a full SBD that needs to be sent on PEPPOL by Oxalis.

Request method: POST

Produces: application/JSON

1. Inbound Scenario for NoTI-ER – Cross-border B2G

This section examines the inbound process, which includes receiving BIS 3.0 invoices from both PEPPOL Network and Italian Exchange System.

* 1. Receiving from PEPPOL Network

NoTI-ER can now receive BIS 3.0 invoices from PEPPOL Network. Once the invoice is received, NoTI-ER sends it to the Italian Exchange System.

The whole workflow is set up as follows:

* Oxalis receives an invoice from PEPPOL Network and forwards it to NoTI-ER, using REST web services described in paragraph 2;
* NoTI-ER receives the same invoice and identifies it as an XML UBL 2.1 BIS 3.0 invoice, using XPath parsing technologies;
* NoTI-ER persists the XML UBL 2.1 BIS 3.0 invoice inside its own Oracle database;
* NoTI-ER sends the XML UBL 2.1 BIS 3.0 invoice to Italian Exchange System.

Italian Exchange System will keep sending outcome notifications related to the invoice. NoTI-ER will keep receiving these notifications using standard SOAP web services communication.

* 1. Receiving from Italian Exchange System

NoTI-ER can now identify BIS 3.0 invoices received from Italian Exchange System. The documents are sent from the Italian Exchange System to NoTI-ER in *FatturaPA Italian e-invoicing* format.

The received invoice will also contain, as an attachment, the original XML UBL 2.1 BIS 3.0 invoice (coded in base64).

The whole workflow is set up as follows:

* Italian Exchange System sends an XML FatturaPA 1.2 format invoice, containing the original XML UBL 2.1 BIS 3.0 invoice format, as an attachment to NoTI-ER;
* NoTI-ER searches for attachments named “*fattura\_originale*” inside the received invoice and, if some occurrences are found, decodes the base64 String format, obtaining the original XML UBL 2.1 BIS 3.0 invoice;
* Once the XML UBL 2.1 BIS 3.0 invoice is obtained, NoTI-ER persists it inside its own Oracle database;
* NoTI-ER persists XML FatturaPA 1.2 invoice format inside its own Oracle database;
* NoTI-ER converts XML Fattura PA 1.2 into XML UBL 2.1 BIS 2.1 invoice format and persists the conversion result inside its own Oracle database.

Finally, NoTI-ER will expose to the associated Public Administrations the same invoice in three different formats:

* XML UBL 2.1 BIS 3.0;
* XML UBL 2.1 BIS 2.1;
* XML FatturaPA 1.2.

1. Outbound scenario for NoTI-ER

In this section the Outbound process is going to be examined, paying attention to the phase in which documents reaches Italian Exchange System or PEPPOL Network from NoTI-ER.

* 1. Sending to Italian Exchange System

NoTI-ER is now able to send XML UBL 2.1 BIS 3.0 invoices to Italian Exchange System.

* + 1. Sending UBL invoice

NoTI-ER sends XML UBL 2.1 BIS 3.0 invoices, received from PEPPOL Network, to Italian Exchange System using the standard web service communication channel.

It is important to assert that no conversion process will be executed during this workflow. NoTI-ER will send to Italian Exchange System the original UBL invoice, without append any signature on the invoice.

* 1. Sending to PEPPOL Network

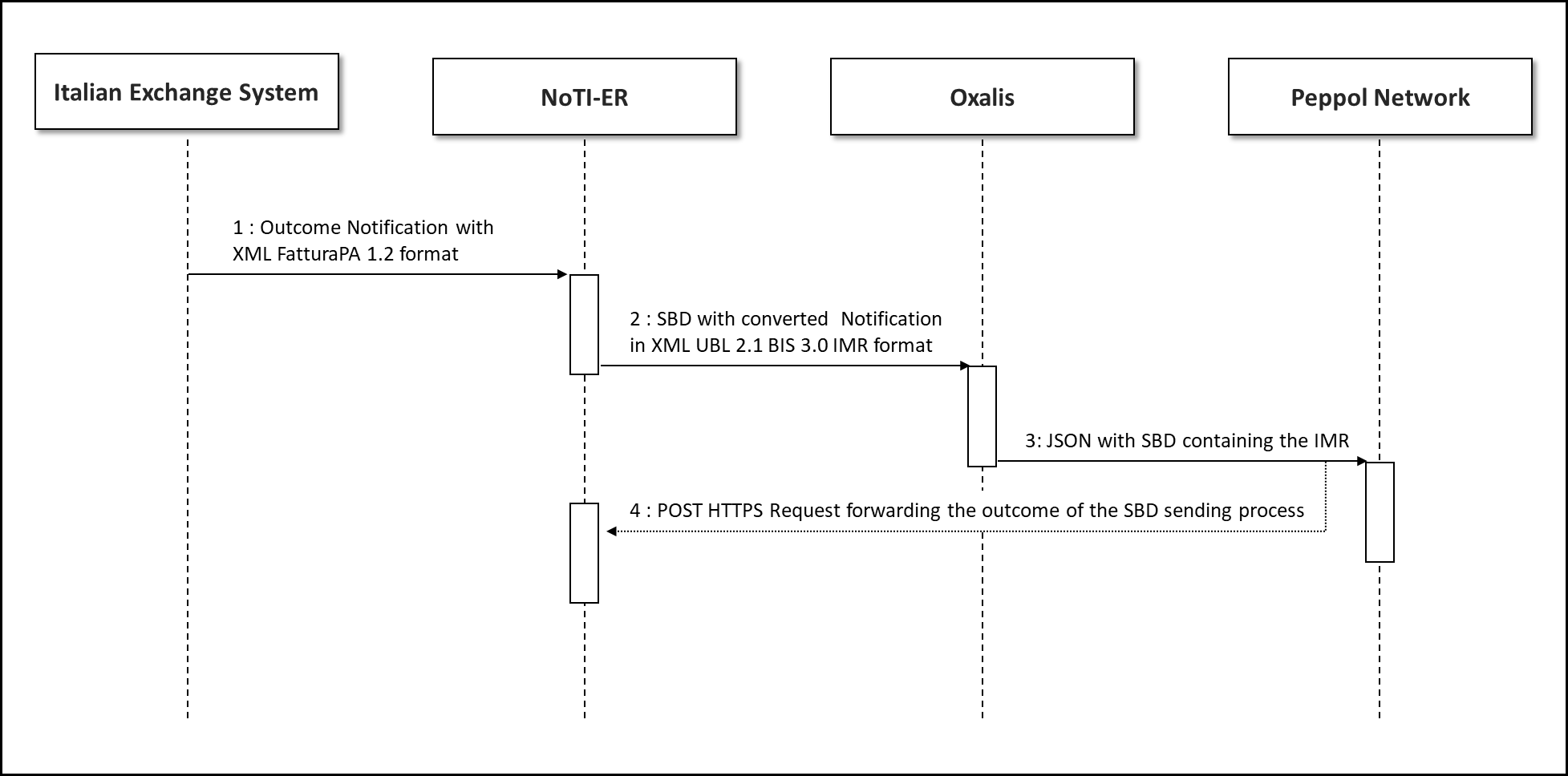
NoTI-ER is now able to produce XML UBL 2.1 BIS 3.0 IMR notifications, sending them to PEPPOL Network.

* + 1. Sending IMR

The whole workflow is set up as follows:

* Italian Exchange System sends an outcome notification to NoTI-ER using the XML FatturaPA 1.2 format;
* NoTI-ER identifies that the received XML FatturaPA 1.2 notification is linked to a BIS 3.0 invoice originally received from PEPPOL;
* NoTI-ER converts the XML FatturaPA 1.2 notification into XML UBL 2.1 BIS 3.0 IMR format;
* NoTI-ER processes XSD and SCHEMATRON validation against the conversion result;
* NoTI-ER sends back the IMR to the original sender of the linked BIS 3.0 invoice using the PEPPOL Network;
  + In order to accomplish this, NoTI-ER builds the whole SBD containing IMR;
  + NoTI-ER sends the SBD to Oxalis;
  + Oxalis forwards the SBD on PEPPOL Network;
  + Oxalis catches the sending outcome and forwards it to NoTI-ER.

*Note: the whole process described here refers to the paragraph 2.*

**