Guideline

OpenPEPPOL AISBL

PEPPOL Transport Infrastructure

ICT - Models

PEPPOL Directory

Non-normative introduction for SMP providers

Version: 1.0-20161128

Status: DRAFT

Editors:

Philip Helger, BRZ
Ger Clancy, IBM

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Version	Date	Description of changes	Approved by
	2016-02-23	Initial version	PH
	2016-11-28	Updated to latest specification document	PH
	2016-12-05	Updated contributor list	PH

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Contributors

Organisations

BRZ (Bundesrechenzentrum)1, Austria, http://www.brz.gv.at/

IBM, http://www.ibm.com

ESV, The Swedish National Financial Management Authority, http://www.esv.se

Persons

Philip Helger, BRZ (editor)

Ger Clancy, IBM

Martin Forsberg, ESV

Georg Birgisson, Midran Ltd.

Referenced documents

This document references the following documents:

[PDIR] PEPPOL Directory Specification, no URL available yet

[phoss] phoss SMP Server, https://github.com/phax/peppol-smp-server

[1] English: Austrian Federal Computing Centre

1 Introduction

The goal of this document is to describe the required changes for SMP Providers to be able to publish participant information to the PEPPOL Directory. This document is a guideline and contains only recommendations but is not normative. This document requires a basic understanding of how the PEPPOL Directory works [PDIR] and focuses purely on the aspects that are relevant to SMP implementers and SMP operators.

2 Terms and definitions

2.1 Service Metadata Publisher

The Service Metadata Publisher (SMP) is a decentralized registry in the PEPPOL network that is used for dynamic capability lookup.

2.2 Service Group

A Service Group is an SMP term that is the container for all PEPPOL participant information. A Service Group relates to exactly one PEPPOL participant.

2.3 PEPPOL Directory

The PEPPOL Directory is a new service introduced to the PEPPOL network with the main goal to allow for an overview of who is registered to the network and the mapping from participant identifier to name. It consists of an Indexer and a Publisher and handles Business Card data elements.

2.4 Business Card

A Business Card is the PEPPOL Directory representation of a participant's data to be published. It is an XML based format with a custom XSD.

2.5 PD Indexer

This is short for PEPPOL Directory Indexer. It is the one half of the PEPPOL Directory Server implementation that is responsible for indexing the Business Cards provided by SMPs.

3 Management summary

The necessary steps to enable interconnectivity between an SMP Server and the PEPPOL Directory Server are:

1. Provide the possibility to store 0..1 Business Card per SMP Service Group
2. Add a new REST interface to your SMP Server so that the PEPPOL Directory Server can retrieve the Business Cards
3. Implement a callback mechanism that notifies the PEPPOL Directory every time a Business Card is created, modified or deleted in the SMP Server.

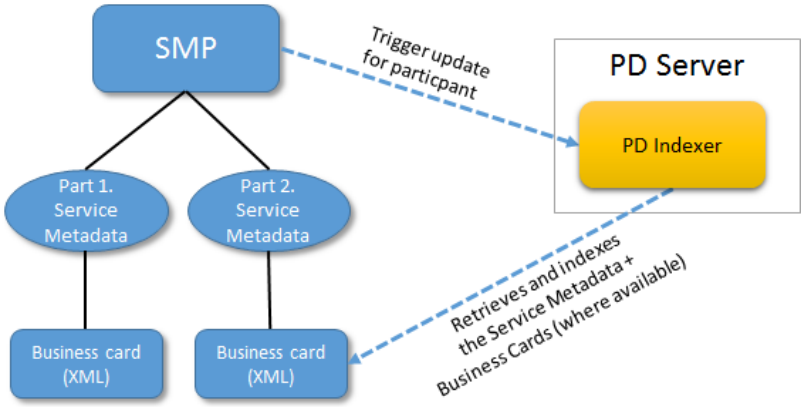

Figure 1 - Interactions between SMP and Directory Server

The above figure shows the dependencies between the different necessary steps to connect an SMP Server with the PEPPOL Directory Server.

4 Storage

This section describes how Business Cards are to be stored in an SMP. The SMP differentiates between Service Groups and Service Registrations. A Service Group is basically the PEPPOL participant identifier whereas a service registration is the combination of a participant identifier, a document type, a process identifier and an AP endpoint URL (plus some additional information).

Each Business Card must be stored in relation to a single Service Group. There are no predefined rules how this is to be achieved as the data storage mechanisms of an SMP server are quite different in practice. The only binding rules are [PDIR]:

1. An SMP MUST NOT provide Business Cards for service groups not owned by this SMP.
2. Each service group MAY have zero or one associated Business Card.
3. The link between the Service Group and the Business Card MUST be the PEPPOL participant ID.

4.1 SQL based storage

When an SMP uses an SQL-based backend system (any relational database) it is recommended to create a new table for Business Card Entities. It should contain at least the following columns:

- Service group identifier
- Entity name
- Country information
- Geographical information
- Identifiers
- Registration date

Note: document type identifiers may not be stored in this table because they are already present in another table of the SMP database.

Depending on your existing data model you may split the service group identifier into two columns (based on the CIPA data model) or you may use a single column where scheme and value are combined (e.g. using "::" – the same separator as used in URLs).

4.1.1 Example DDL

The following MySQL DDL is taken from [phoss] and shows how it can be done:

```
DROP TABLE IF EXISTS `smp_bce`;
CREATE TABLE `smp_bce` (
  `id` varchar(45) NOT NULL COMMENT 'Internal ID',
  `pid` varchar(255) NOT NULL COMMENT 'Participant/Business ID',
  `name` text NOT NULL COMMENT 'Entity name',
  `country` varchar(3) NOT NULL COMMENT 'Country code',
  `geoinfo` text COMMENT 'Geographical information',
  `identifiers` text COMMENT 'Additional identifiers',
  `websites` text COMMENT 'Website URIs',
  `contacts` text COMMENT 'Contact information',
  `addon` longtext COMMENT 'Additional information',
  `regdate` date DEFAULT NULL COMMENT 'Registration date',
  PRIMARY KEY (`id`),
  KEY `FK_pid` (`pid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='SMP Business Card Entity';
```

The table is called `smp_bce` which the abbreviation of "Business Card Entity". The scheme is based on an older version of the Business Card schema hence matching the current schema, the columns `websites`, `contacts` and `addon` may be deleted. This implementation uses JSON notation to store fields that have a multiplicity of more than 1 (`identifiers`, `websites` and `contacts`) to simplify the DB schema.

> Note: a problem with this DDL is the usage of character set "latin1" (which is basically ISO-8859-1) which limits the number of allowed characters. The usage of "utf-8" would be better, but for compatibility to the rest of the scheme it was chosen to use "latin1". The reason why "utf-8" cannot be used is, that MySQL has a limit of 767 bytes for a key. In UTF-8 a character may have up to 3 bytes, so at most 255 characters (= 765 bytes) may be used for key in MySQL.

4.2 XML based storage

When an SMP uses an XML-based backend it is recommended to create a new entity for a Business Card that links to the Participant identifier of the Service group.

4.2.1 Example XML schema

The following XML schema is also taken from [phoss]:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace=""
           elementFormDefault="unqualified"
           attributeFormDefault="unqualified"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="IdentifierType">
    <xs:attribute name="id" type="xs:string" use="required" />
    <xs:attribute name="scheme" type="xs:string" use="required" />
    <xs:attribute name="value" type="xs:string" use="required"/>
```

```
  </xs:complexType>

  <xs:complexType name="ContactType">
    <xs:attribute name="id" type="xs:string" use="required" />
    <xs:attribute name="type" type="xs:string" use="optional" />
    <xs:attribute name="name" type="xs:string" use="optional" />
    <xs:attribute name="phone" type="xs:string" use="optional" />
    <xs:attribute name="email" type="xs:string" use="optional"/>
  </xs:complexType>

  <xs:complexType name="EntityType">
    <xs:sequence>
      <xs:element name="geoinfo" type="xs:string" minOccurs="0" />
      <xs:element name="identifier" type="IdentifierType" minOccurs="0"
maxOccurs="unbounded" />
      <xs:element name="website" type="xs:string" minOccurs="0"
maxOccurs="unbounded" />
      <xs:element name="contact" type="ContactType" minOccurs="0"
maxOccurs="unbounded" />
      <xs:element name="additional" type="xs:string" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="optional" />
    <xs:attribute name="name" type="xs:string" use="required" />
    <xs:attribute name="country" type="xs:string" use="required"/>
    <xs:attribute name="regdate" type="xs:date" use="optional" />
  </xs:complexType>

  <xs:complexType name="BusinessCardType">
    <xs:sequence>
      <xs:element name="entity" type="EntityType" minOccurs="0"
maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="servicegroupid" type="xs:string" use="required" />
  </xs:complexType>
</xs:schema>
```

> Note: again `website`, `contact` and `additional` are present because the scheme is based on an old Business Card schema.

5 Business Card retrieval REST interface

To retrieve the Business Cards from an SMP server a new REST interface must be implemented in an SMP server.

REST request: `GET /businesscard/{participantID}`

> Note: {participantID} is the placeholder for the effective PEPPOL participant identifier

REST response: the XML representation of the business card preferably in UTF-8 encoding using MIME type `application/xml`.

REST response code:

- HTTP 200 (OK) – everything was ok. A response body is send back.
- HTTP 404 (Not found) – no business card was found for the provided participant ID.
- HTTP 500 (Internal server error) – something internally went wrong. Response body contains the details in plain text.

Example querying the Business Card for PEPPOL participant `0088:gln123` on the SMP server running at `http://smp.example.org`:

```
http://smp.example.org/businesscard/iso6523-actorid-upis%3A%3A0088%3Agln123
```

Note: using PEPPOL participants directly in URLs may impose problems. So please ensure that the colon character (":") is escaped as %3A in the URL.

Note: this interface must also work with the computed "B-….edelivery.tech.ec.europa.eu" URLs.

6 Notify PEPPOL Directory Server on changes

This chapter describes the technical details on how to notify the PD Indexer if the Business Card of a Service Group is added, changed or deleted. All URLs are provided server-relative in this document.

Note: the complete URLs to the PEPPOL Directory are not yet fixed and must be prepended to create working URLs.

6.1 Authentication and authorization

Note: this section is only applicable, it the *PD Indexer* runs on a server that offers secure HTTP connections (https).

For security reasons, only legitimate PEPPOL SMPs are allowed to request modifications in the *PD Indexer*. To ensure this *all* HTTP calls to the *PD Indexer* interface must provide a client X.509 certificate. This is the same technology that is already used in the SMP to SML communication and should therefore be implementable in a quick and easy way. Requests to the *PD Indexer* without a client certificate will result in an error.

The provided client certificate must be the PEPPOL SMP certificate as used for the communication with the SML.

6.2 Adding a participant

For adding a participant, only the participant identifier must be passed to the *PD Indexer*. The Business Card is read directly from the respective SMP (determined via DNS lookup) and is not passed in this call. This allows the *PD Indexer* to build a queue of items to be updated in an optimized way and also avoids overwriting data of PEPPOL participants that are owned by different SMPs.

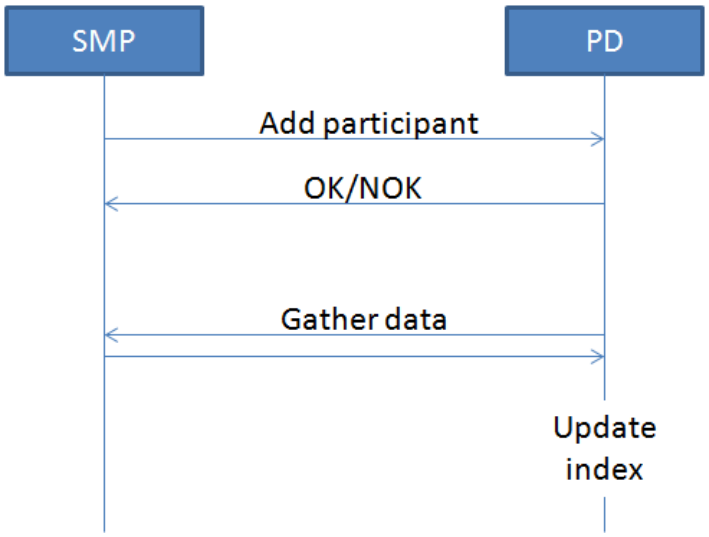

Figure 2: Add participant workflow

REST request: `PUT /indexer/1.0/`

Request body: `{participantID}`

> Note: {participantID} is the placeholder for the effective PEPPOL participant identifier in URL encoded form

Example request:

- URL: `PUT /indexer/1.0/`
- Body: `iso6523-actorid-upis%3A%3A0088%3Agln1234`

REST response code:

- HTTP 204 (OK, No content) – everything was ok. No response body is send back.
- HTTP 403 (Forbidden) – no client certificate or an invalid client certificate provided
- HTTP 500 (Internal server error) – something internally went wrong. Response body contains the details in plain text.

> Note: This requires the DNS entry of the added PEPPOL participant already being available publicly to resolve the owning SMP. Therefore an SMP MUST call the PD after the registration at the SML. The *PD Indexer* will handle added participants gracefully if the respective DNS entry is not yet present and will retry at a later point in time. If a new participant DNS entry is not present within 24 hours of the original indexing request, this particular request is discarded and therefore no indexing takes place. If previous indexed information of that participant is present (if it is an updating call) they are left unchanged.

6.3 Modifying an existing participant

If the business card of an existing participant is modified the *PD Indexer* must be informed about the change. The API and the constraints are identical to “Adding a participant” (see chapter 6.2).

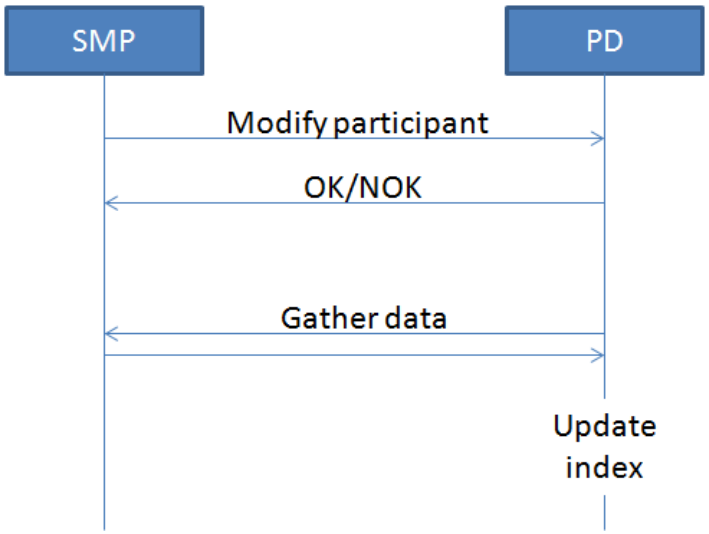

Figure 3: Modify participant workflow

> Note: there is no possibility to identify whether the participant was added or updated by the response. To check for existence, use the GET operation defined below.

6.4 Deletion of a participant

When a service group in the SMP is about to be deleted (either because the participant leaves the PEPPOL network or because an SMP migration takes place), the *PD Indexer* must be notified. To delete participant information in the *PD Indexer* it is suitable to provide only the respective PEPPOL identifier.

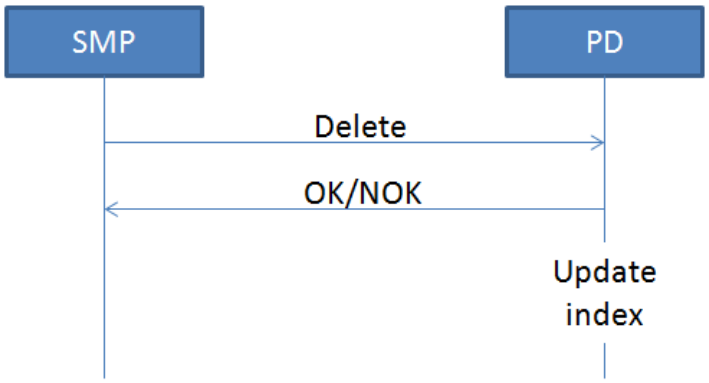

Figure 4: Delete participant workflow

REST request: `DELETE /indexer/1.0/{participantID}`

> Note: {participantID} is the placeholder for the effective PEPPOL participant identifier in URL encoded form

Example request:

- `DELETE /indexer/1.0/iso6523-actorid-upis%3A%3A0088%3Agln1234`

> Note: using PEPPOL participants directly in URLs may impose problems. So please ensure that the colon character (":") is escaped as %3A in the URL.

REST response code:

- HTTP 204 (OK, No content) – everything was ok. No response body is send back.
- HTTP 403 (Forbidden) – no client certificate or an invalid client certificate provided
- HTTP 500 (Internal server error) – something internally went wrong. Response body contains the details in plain text.

> Note: if a participant is moved from SMP to another it must first be deleted by the old SMP and then re-created by the new SMP.

> Note: the delete operation may impose a security problem because one SMP can delete the information of a participant created by a different SMP. Therefore the deletion does not directly delete the information in the index but only marks the respective records internally as "deleted" so that the data can be restored in case of a misuse.

6.5 Existence check of a participant

Checking whether a business card of a PEPPOL participant is present in the *PD Indexer* can be performed via the following interface:

REST request: `GET /indexer/1.0/{participantID}`

> Note: {participantID} is the placeholder for the effective PEPPOL participant identifier

Example request:

- `GET /indexer/1.0/iso6523-actorid-upis%3A%3A0088%3Agln1234`

> Note: using PEPPOL participants directly in URLs may impose problems. So please ensure that the colon character (":") is escaped as %3A in the URL.

REST response code:

- HTTP 204 (OK, No content) – Yes, the participant is already in the *PD Indexer*.
- HTTP 403 (Forbidden) – no client certificate or an invalid client certificate provided
- HTTP 404 (Not found) –the participant is not in the *PD Indexer*.
- HTTP 500 (Internal server error) – something internally went wrong. Response body contains the details in plain text.

> Note: because of the internal asynchronous processing, it might take some time after an index request until the participant is available in search results.

7 Annex A – Business Card XSD

The current Business Card XML Schema can be found on GitHub:

https://github.com/phax/peppol-directory/blob/master/peppol-directory-businesscard/src/main/resources/schemas/peppol-directory-business-card-20161123.xsd