Specification

OpenPEPPOL AISBL

PEPPOL Transport Infrastructure

ICT - Models

PEPPOL Directory

(formerly PEPPOL Yellow Pages)

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
	2015-08-03	Updated details on the PD-SML connection Editorial changes in 5.2.2, 6.1.1 Changed link in chapter 7	PH
	2015-09-10	Moved chapter 4.1.5 to become 4.1.1 Fixed example requests in the PD Indexer section Fixed HTTP return codes in Indexer section Added chapter 4.1.7 on internal processing	PH
	2016-01-20	Updated to PEPPOL Directory Adopted to separate SMP interface Adopted to new XSD	PH
	2016-04-22	Introduction less technical Logical separation between Publisher and Indexer less important	PH
	2016-11-28	Updated to MC decisions Stripped down Business Card data	PH
	2016-12-05	Updated contributor list	PH

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.

[1] English: Austrian Federal Computing Centre

1 Introduction

The goal of this document is to describe the architecture and interfaces of the PEPPOL Directory (PD; formerly known as PEPPOL Yellow Pages) project. The goal of the PD project is to create a publicly available, searchable list of all PEPPOL participants with their respective metadata like company name, country code, etc. (for details see chapter 4.1). The PD is not meant to replace existing PEPPOL components but to be an aggregator for data that is contained in existing PEPPOL SMPs.

An additional singleton service is added to the PEPPOL infrastructure: the so called ***PD Server***. It is filled with electronic **Business Cards** of the PEPPOL participants on a voluntary basis[2] meaning that SMP providers can (but are not forced to) publish their client's metadata in the PD. The data is stored in correlation with the SMP entry of the respective participant (aka service group). Details are described in chapter 4.2.

This document describes the architecture of the PD server, the interfaces to and from it as well as the data format for the Business Cards (see chapter 4) within the SMP. This document concludes with a technical proposal on how the PD Server could be implemented.

2 Why PEPPOL Directory?

Due to variations between countries and markets, there are no shared models on how to know the PEPPOL Participant ID (PPID) of the sender, further enforced by the lack of open national business registries. Knowing each other in domains of limited size, for example e-CODEX project in e-Justice, is easy, however in domains like PEPPOL having potentially millions of organizations it is impossible.

Trying to solve the problem of finding each other, PEPPOL Directory (PD) is introduced, a central service to query based on given metadata. Querying may be part of a manual or automated process before performing lookup in SML (Service Metadata Locator) and SMP (Service Metadata Publisher). PD contains indexed PEPPOL Directory Business Cards (BC) containing metadata related to a given PPID. The lack of a PEPPOL Directory is a constraint to wider scale adoption of PEPPOL by small and medium sized enterprises.

2.1 Use Cases

The PEPPOL Directory is intended to support business cases that are concerned with finding PEPPOL participants registered on the PEPPOL network in order to start exchanging business documents with them. Some of the possible business cases are identified below.

2.1.1 New PEPPOL BIS support - Matching

An organization that has recently become a PEPPOL participant to exchange a particular PEPPOL BIS, as a Customer or a Supplier, will want to find who of their trading partners are capable of exchanging the same BIS documents in the opposing role.

[2] National PEPPOL authorities however may force participants to show up in the PD.

As example an organization that is starting to send invoices may want to know which of their customers can receive them and an organization that is starting to receive invoices will want to know which of their suppliers can send them.

2.1.2 Monitoring new PEPPOL users - Alerting

An organization that is using PEPPOL to exchange one or more PEPPOL BIS may want to monitor when more of their trading partners become PEPPOL participants and consequently to automate their trading relations with them by using PEPPOL.

2.2 Planned key functions of PEPPOL Directory

The following key functions are planned for the PEPPOL Directory and will be implemented through different releases of the Directory. These features are intended to support the business use cases described in the previous chapter.

2.2.1 Free text search

A free text search allows the Directory user to enter a text string into an online form and get a list of result for all listings in the Directory where that string appears. As an example if the user enters the word "Acme" he will get a list of all participants who's name contains the word “Acme” as well as participants where the word “Acme” appears in other elements of the Business Card.

The user can browse the list to find the PEPPOL participant he is looking for and then click on his choice to see the full details.

2.2.2 Identifier search

The directory specification supports the use of qualified identifiers for the search. The objective is to enable single match searches where the user submits a query on whether there exists a user with a particular identifier and BIS capabilities. This enables searching by VAT, legal identifiers and others parameters that are commonly known but may differ from the PEPPOL end point identifiers. As an example a user may want to find the end point identifier for a customer who has a particular VAT identifier. By restricting the search to a particular capability he can use the query to monitor when that customer starts to support the given documents.

2.2.3 API connection

The PEPPOL Directory will also enable Directory users to let their systems connect automatically instead of manually browsing through a web interface. This supports automated searches that can be integrated into the sending process.

A drawback to be considered is that the publication of the Business Cards in the PEPPOL Directory happens on a voluntary basis.

2.3 Considerations

The following considerations that may influence the ongoing development of the PEPPOL Directory have been identified but may require additional analysis.

2.3.1 Searching for senders

The current architecture of the PEPPOL network does not require PEPPOL Participants who are only sending documents to be registered in the SMP's and consequently they are not in the SML. This limits the capability of the PEPPOL Directory to include these PEPPOL participants in search results. This relates to other issues that are currently being addressed in other PEPPOL initiatives. A potential change in the PEPPOL policy that requires registration of senders would benefit the PEPPOL Directory without requiring additional changes to the PD.

Alternatively sending only participants may be registered to an SMP with an empty service group which allows them to publish Business Cards for the PEPPOL Directory as well.

3 PD Server architecture

This section describes the overall architecture of the PD Server. It logically consists of two major parts: a *PD Indexer* which is responsible for creating, updating, deleting and indexing the Business Card data and the *PD Publisher* which is the public web frontend to the PD for both humans and machines.

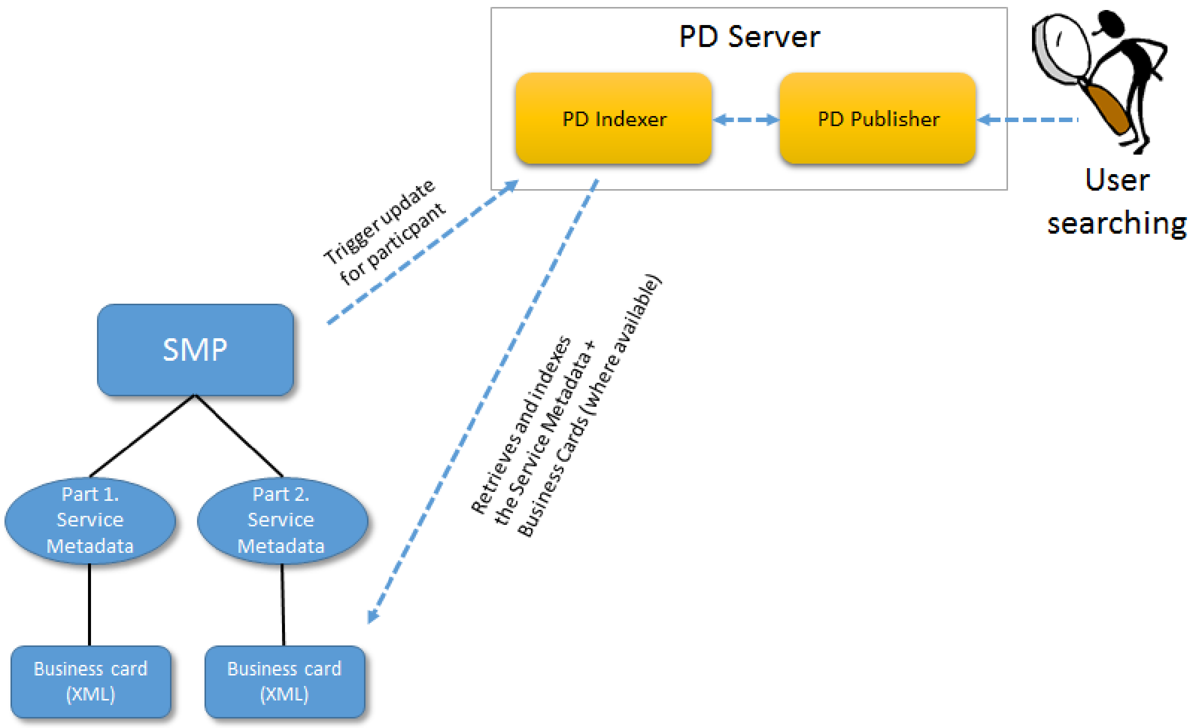

Figure 1: PD big picture without SML

The above big picture outlines the information flow. If a participant’s business card is added to, updated to or deleted from an SMP, the SMP MAY trigger an update to the *PD Indexer* (see arrow from SMP to the *PD Indexer* in the figure) even if the Business Card contained in the SMP is empty. If data is to be added or updated on the PD, the *PD Indexer* will retrieve the complete Business Card from the respective SMP and index it for searchability (see arrow from PD Indexer to *Business card* in the figure).

If a user wants to know whether a certain company is registered in the PEPPOL network he opens the web site of the *PD Publisher*, types the search term (e.g. the company name) and a list of potential hits (including the PEPPOL participant identifier and the supported PEPPOL document types) shows up. Additionally to the human interface a REST interface for automatic searching is offered. The *PD Publisher* retrieves all relevant information directly from the *PD Indexer* so that no interaction with the concerned SMPs is necessary.

An extension to the *PD Indexer* is the direct connection to the SML to retrieve a list of **all** registered PEPPOL participants. In this case the PD Indexer will query the SML regularly (e.g. once a week) for a complete participant list and queries the respective SMPs independent of the SMP provided update status.

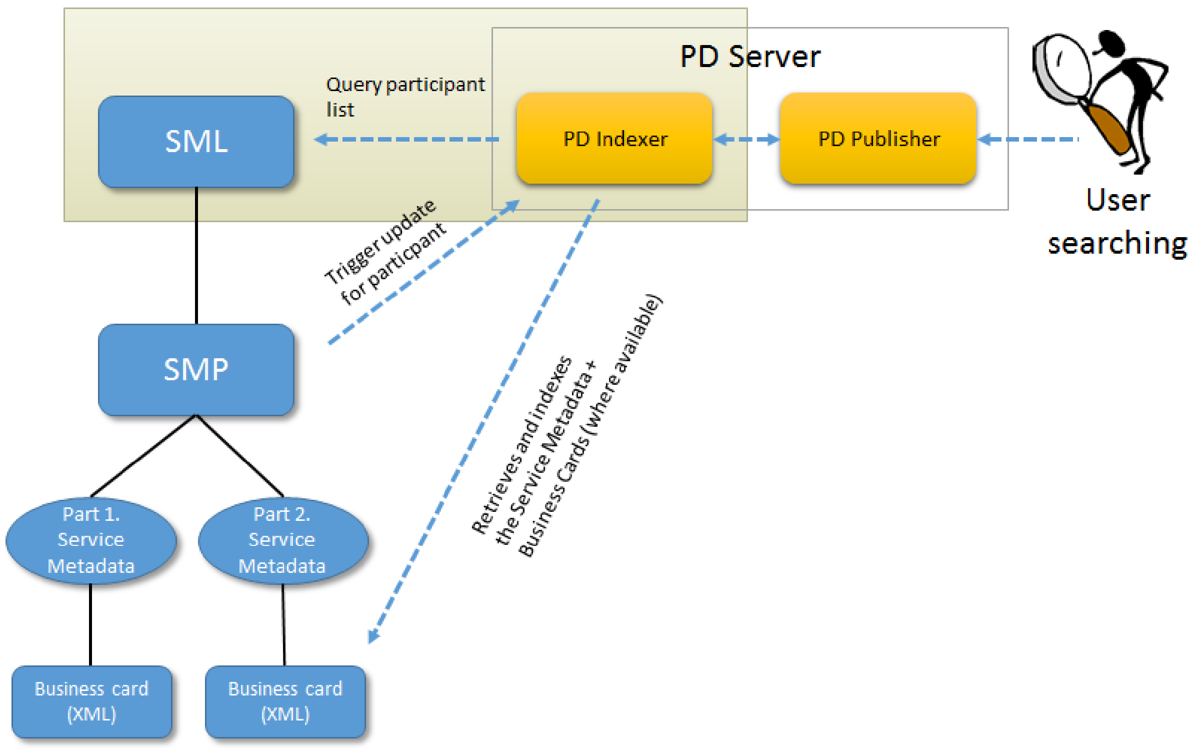

Figure 2: PD big picture with SML

As shown in the previous figure the overall architecture is only extended to interconnect with the SML and no other changes are necessary. The SML already offers an interface to retrieve a list of all registered PEPPOL participants and is therefore prepared to be interconnected with the PD.

Early benchmarks on the SML test machine (being slower than the production machine) showed that a list with 100.000 entries can be created in 16 seconds and 150.000 entries took 34 seconds. By end of 2015 approx. 40.000 entries were in the production database.

4 Business card

4.1 Data format

This section describes the layout of the business card data that is stored in an SMP. Because the scope of a single PEPPOL participant identifier within an SMP can be very broad, the data format must be capable of handling information for more than one business entity in a structured way. Sometimes a PEPPOL participant may even link to different entities in different countries.

Existing formats like vCard, xCard or the UBL 2.1 Party type were not considered because they are either not XML or too complex to interpret fully. Instead a new minimal XML-based format is created because PEPPOL participant identifiers are used very differently it was decided to use a very flexible scheme that can represent multiple business entities at once.

The format defines a single business card consisting of the following fields:

- PEPPOL participant ID
  - Description: PEPPOL participant identifier corresponding to a service group hosted on the same SMP
  - Multiplicity: 1..1 (mandatory)
- PEPPOL document type ID
  - Descriptions: all PEPPOL document type identifiers as indicated by the default SMP service group query.
  - Multiplicity: 0..n (optional but potentially many)
- Business entity
  - Description: a business entity that can be reached via the provided PEPPOL participant ID
  - Multiplicity: 0..n (optional but potentially many)

Each business entity consists of the following fields:

- Entity name
  - Description: the company name or the name of the governmental entity
  - Multiplicity: 1..1 (mandatory)
- Country code
  - Description: the country code in ISO 3166-2 format (e.g. “AT” for Austria)
  - Multiplicity: 1..1 (mandatory)
- Geographic information
  - Description: describes the location or region of the entity that is usually used to identify the entity. This may be an address, a state name etc.
  - Multiplicity: 0..1 (optional)
- Identifier
  - Description: additional (non-PEPPOL) identifiers of the entity that are not part of the PEPPOL participant identifier. It consists of a type and a value. This can e.g. be a

national VAT identification number; a national company register number etc. The following identifier types (case insensitive) must at least be supported by the Directory:
        - "vat" – VAT identification number including the national prefix
        - "orgnr" – the national organisation number
        - "gln" – Global Location Number (GLN)
        - "duns" – DUNS number
    - Multiplicity: 0..n (optional but potentially many)
- Registration date
    - Description: the date when the participant joined the PEPPOL network
    - Multiplicity: 0..1 (optional)

The XML Schema describing the layout of the Business Card can be found in chapter 7 of this document. To support future updates of this Business Card scheme the XML root element (`BusinessCard`) has an XML namespace URI that allows for easy versioning of the contained data. Version 1 of the XML schema for the business card uses the XML namespace URI `http://www.peppol.eu/schema/pd/businesscard/20161123/`.

A non-normative example Business Card with a single entity looks like this:

```
<BusinessCard
    xmlns="http://www.peppol.eu/schema/pd/businesscard/20161123/">
  <ParticipantIdentifier
      scheme="iso6523-actorid-upis">0088:example</ParticipantIdentifier>
  <BusinessEntity registrationDate="2010-07-06">
    <Name>ACME Inc.</Name>
    <CountryCode>AT</CountryCode>
    <GeographicalInformation>ACME street 123</GeographicalInformation>
    <Identifier scheme="VAT">ATU12345678</Identifier>
    <Identifier scheme="OrgNr">hjdh7as9ds</Identifier>
  </BusinessEntity>
</BusinessCard>
```

4.2 SMP impacts

This chapter describes the constraints for storing Business Cards in an SMP and how to access the Business Cards from the outside.[3]

4.2.1 Storage

This section describes how and where 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, a transport protocol and an AP endpoint URL (plus some additional information).

[3] phoss SMP and IBM SMP have already implemented support for the BusinessCard API in their solutions.

Each Business Card must be stored in relation to a single SMP 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:

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.

Originally it was considered to store the Business Card information in the `Extension` element of an SMP Service Group. The positive aspects of this solution are that the data model of existing SMPs does not need to be altered and that no new APIs for the SMP must be provided. The negative aspects of this solution are that the network traffic for non-PD queries would heavily increase and the general performance of SMPs might be downgraded and that non-relevant information would be returned in regular Service Group queries. An additional problem with this solution is that the PEPPOL SMP specification is lacking support for multiple extensions in a single service group which in turn would require an additional non-standard “extension container” to maintain extensibility. OASIS BDXR SMP CS03 adds supported for multiple extensions.

4.2.2 Public REST interface

To retrieve the Business Cards from an SMP server a new REST interface is introduced. This interface must be provided by all SMP servers that want to serve Business Card data for the PD. REST was chosen because the existing SMP interfaces are already REST based and therefore no new technology is introduced.

4.2.2.1 Retrieve Business Card interface

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

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

REST response: the XML representation of the business card (according to the XSD specified in chapter 7) 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.

Non-normative example to query the business card for PEPPOL participant `9915:test` on the SMP server running at `http://smp.example.org`:

```
http://smp.example.org/businesscard/iso6523-actorid-upis%3A%3A9915%3Atest
```

The response may look like the example provided in section 4.1.

Note: using PEPPOL participants directly in URLs may impose problems. It must be ensured 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.

Note: as a future extension, the response of the SMP may be signed with the PEPPOL SMP certificate.

5 PD Indexer

This chapter describes the technical details of the *PD Indexer*. It describes the data elements that must be passed to the *PD Indexer* so that Business Cards can be created, updated, deleted or retrieved. This is a REST interface, because the SMP server (that will trigger this interface) is also a REST server and therefore the technology is well known and supported.

All REST interface URLs contain a version number so that it will be easy to provide updated interfaces in the future without breaking the existing ones.

5.1 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.

5.2 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.

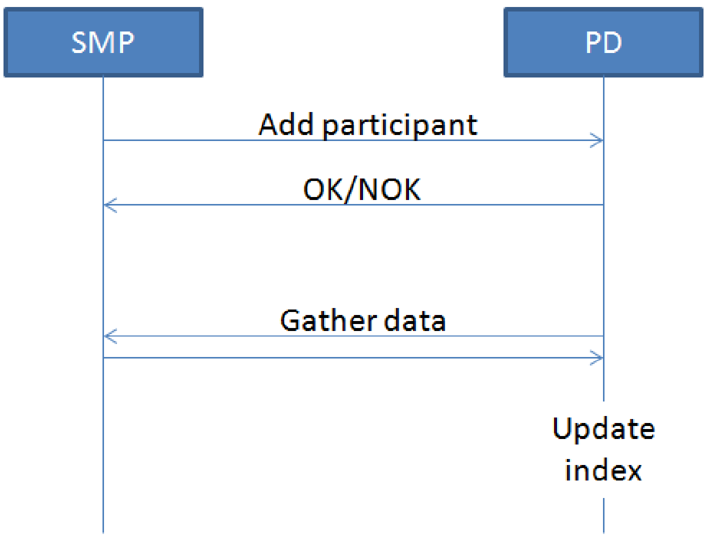

Figure 3: Add participant workflow

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.

5.3 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 5.2).

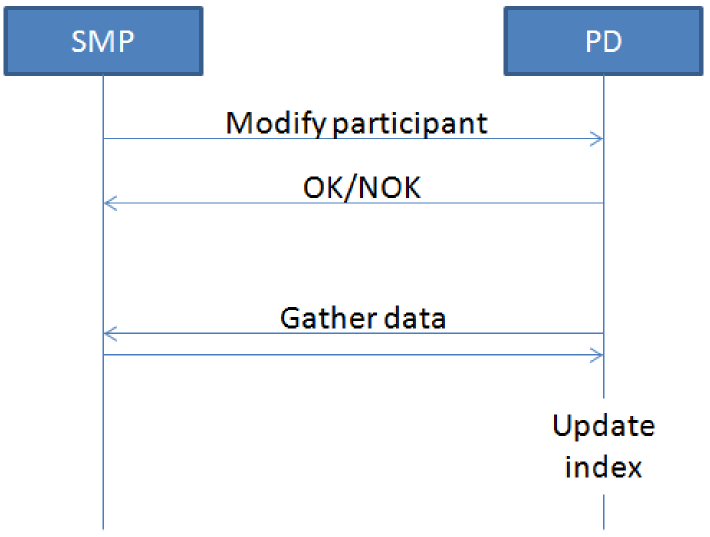

Figure 4: Modify participant workflow

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.

5.4 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.

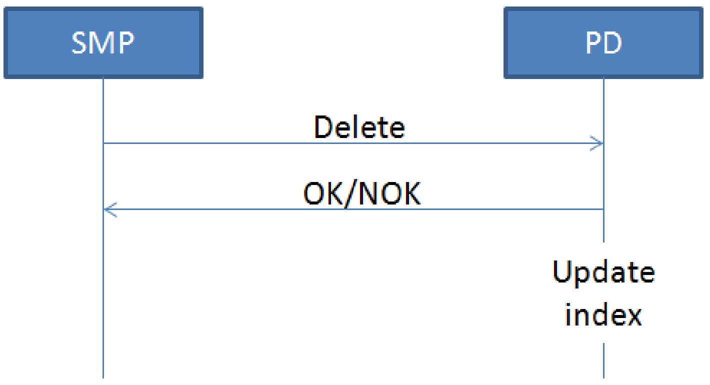

Figure 5: Delete participant workflow

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.

5.5 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. See chapter 5.7 for more details.

5.6 Auditing and Logging

All successful calls to the *PD Indexer* (create/update/delete/get) are logged together with the timestamp, the source IP address and some information from the provided certificate (country, subject name and serial number) to ensure traceability of the performed actions.

5.7 Internal processing of the data

Internally the Indexer keeps a FIFO work queue that is processed asynchronously. All new indexing requests (create/update/delete) are put into that queue and wait for their serial processing to avoid overloading a single SMP with queries. That's why deletion (see chapter 5.4) may not trigger an immediate return code like "not found" because the result is not known synchronously.

If the data retrieval from the SMP fails (for whatever reason) the work item is put into a special "retry queue" and the data retrieval is retried some time later (suggested duration until retry is 5 minutes – must be configurable). If an entry cannot be indexed after a certain period of time (suggested period is 24 hours – must also be configurable), it is moved to a "dead work item queue". In case of a permanent failure manual intervention is necessary. E.g. the PD administrator may re-trigger the work item manually or choose to drop it completely.

The asynchronous processing may impose problems when trying to check for the existence of a certain PEPPOL participant identifier in the index. This check will only return success if the item was already processed by the worker queue but not if it is still in the work queue.

5.8 Internal data structure

The internal data structure of the *PD Indexer* is slightly different from the Business Card entities defined in chapter 4.1. Besides the Business Card content the following data elements should also be stored:

- All supported PEPPOL document type identifiers as listed by the PEPPOL service group interface. Therefore a separate SMP query on the ServiceGroup must be performed and the document types must be extracted.
- The unique identifier taken from the client certificate that triggered the indexing of the document (the "requestor"). This can e.g. consist of the certificates subject name, serial number and country code.
- The date and time when the Business Card was last indexed.

6 PD Publisher

This section describes the components of the *PD Publisher*. It consists of a machine-to-machine search interface as well as a search interface for humans as well as a list of registered PEPPOL participants for download. Additional features can be integrated into the Publisher after the initial version.

6.1 Search interface

This section only describes the machine-to-machine search interface. It uses REST as the protocol and responds (at least) with XML data[4].

6.1.1 Request

The relative base URL of the REST search service is `/search/1.0/[format]` which is then followed by a list of query parameters as outlined below. The `[format]` placeholder in the request API denotes the desired response format. Initially only `xml` format (for XML output) will be used but other formats (like JSON) might be added as a future extension. All search REST requests are HTTP GET requests. Other HTTP methods like POST, PUT etc. are not supported.

The search routines should use the following text matching algorithms:

- *Exact match*: the search term and the indexed values must be completely equal, including case sensitivity.
- *Partial match*: the search term must be equal or fully contained in the indexed value in a case insensitive way (e.g. searching for “tici” or “TICI” in the indexed value “participant” will be a match)
- *Starts with match*: a special version of the partial match that requires the indexed value to begin with the search term in a case insensitive way (e.g. search for “part” or “PART” will match “participant” but “art” won’t match “participant”)

Parameter name	Explanation
q	General purpose query term. This term is searched in all fields with the matching rules of the respective fields. Multiple search terms can be provided separated by a whitespace character.
participant	Searches for *exact matches* in the *PEPPOL participant identifier* field (the identifier scheme must be part of the value)
name	Searches for *partial matches* in the *entity name* field. Multiple search terms can be provided separated by a whitespace character. Only search terms consisting of at least 3 characters are used for search.
country	Searches for *exact matches* in the *country code* field
geoinfo	Searches for *partial matches* in the *geographic information* field. Multiple search terms can be provided separated by a whitespace character. Only search terms consisting of at least 3 characters are used for search.
identifier	Searches for *exact matches* in the *identifier* fields (only the value, not the type)

[4] Upon request additional data formats like JSON can be added

regdate	Searches for matches in the *registration date* field. The value of the date to search must be provided in the format 'YYYY-MM-DD' (ISO 8601 based date format). Optionally a syntax for comparison queries (<, ≤, >, ≥ and range) may be supported. Similar solutions should be evaluated and best practices should be used.
doctype	Searches for exact matches in the *PEPPOL document type identifier* field.
resultPageIndex	The result page to be shown. If this parameter is not present the first page is returned. The result page index is 0-based meaning that the first page has index 0. The index of the first search result returned is calculated by `resultPageIndex * resultPageCount`
resultPageCount	The number of results to be returned on a single page. If this parameter is not present 20 results are returned by default.

If multiple of the query parameters are used together only the results matching ALL query terms are returned.

6.1.2 Response

If no query term parameter (see table in chapter 6.1.1) is provided the return value is HTTP 400 (Bad Request).

Note: the PD Publisher will deliver at most the top 1000 results. If the combination of resultPageIndex and resultPageCount results in too small (< 0) or too large values (> 1000) the return value is HTTP 400 (Bad Request). The index of the first search result returned is `resultPageIndex * resultPageCount`. The index of the last search result returned is `(resultPageIndex + 1) * resultPageCount - 1`.

6.2 User interface

6.2.1 Use case Search

The PD Publisher must offer a publicly available web page where the user can enter search terms to search for one or more PEPPOL participants. It should provide a simple search form where only a set of terms can be entered and the *PD Publisher* will search for the best possible matches. Additionally an extended search form with all fields (as outlined in chapter 6.1.1) should be available.

The search results will be shown on the website and will also be made available for download.

6.2.2 Use case Browse

The *PD Publisher* should offer a list of all registered business entities so that the information is browsable or even downloadable as e.g. an Excel document. This of course implies that the full data must be stored in the *PD Indexer*.

This use case has slightly lower priority for implementation than the Search use case but is definitively a valuable extension.

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

8 Annex B - Implementation proposal (non-normative)

This section roughly describes, how the *PD Server* could be implemented and how existing SMP servers could be modified to interact with the PD server.

All data described in this document must be stored and/or transmitted in UTF-8 character encoding set. Using other character encodings is prohibited.

The rest of this chapter assumes that the development will be done with Java.

8.1 PD Server

For simplicity the *PD Server* should be implemented as a regular Java web application that is runnable on a regular servlet container like Apache Tomcat or Jetty. It internally consists of two main parts: the *PD Indexer* and the *PD Publisher*. Both components have to expose a component to the outside world but need to fulfil different tasks.

8.1.1 PD Indexer

The *PD Indexer* is responsible for gathering the business cards from the different SMPs and storing it into a searchable index. It is also responsible for periodically grabbing all participants from the SML.

The basic components are:

- A “work queue” that handles the requested actions for certain participants with a certain priority handling (requests from SMPs have a higher priority than SML crawling results). The work queue must be able to filter out duplicate requests and leave only the ones with the highest priority.
- A “fetcher” that grabs action items from the work queue and queries the SMP for the corresponding data of a participant
- An “indexer” that takes the fetch results and stores them into a searchable index
- A scheduled “SML retriever” that retrieves the participant list from the SML and stores all entries for updating in the work queue.
- A REST server implementing the interfaces as defined in chapter 5 and accordingly filling the work queue. Only HTTP requests providing a valid PEPPOL SMP client certificate are accepted.
- An “auditor” that keeps track of all indexing actions together with some meta information

The *PD Indexer* should be based on Apache Lucene (https://lucene.apache.org/core/ - Apache 2 License) for the indexing. The REST interface is to be done with Jersey (https://jersey.java.net/ - CDDL

1.1 or GPL 2 with Classpath exception) like with the SMP. The scheduling functionality is provided by Quartz (http://quartz-scheduler.org/ - Apache 2 License).

8.1.2 PD Publisher

A simple *PD Publisher* can be built with the ph-oton library (https://github.com/phax/ph-oton - Apache 2 License) which offers capabilities to create state of the art (responsive, fast, nice looking) web applications quickly. For the main searching Apache Lucene will be used (must be identical to the version used for indexing).

The basic components of the *PD Publisher* are:

- A REST based search interface as described in chapter 6.1
- A public web page for the simple search
- A public web page for the extended search
- A public web page with the most recently added participants
- A secure web site to see the log and audit entries

8.2 SMP-PD interface

The PD software suite should ship with a library that can be used to trigger the indexing in the *PD Indexer*. SMP software providers can use this library to simplify the process of integrating their software with the PD as they just need to call this when relevant information changes (new participant, Business Card update, participant deletion).

It is proposed to provide a patch for the CIPA SMP server (https://joinup.ec.europa.eu/software/cipaedelivery/description).

phoss SMP (https://github.com/phax/peppol-smp-server) already supports the new Business Card API.