Open Patent Services v.2
Web Service Definition and Description

Version 1.0.5

Revision History

Date	Version	Description	Authors
11/05/2009	1.0.0-draft	Initial release of a draft document	Adam Grodowski (agrodowski@epo.org)
29/05/2009	1.0.0	The first release of a full document	Adam Grodowski (agrodowski@epo.org), Davide Lingua (dlingua@epo.org) Michel Plungjan (mplungjan@epo.org)
09/06/2009	1.0.1	The update of schema links within References section due to the changes in ops.epo.org website performed 07/06/2009	Adam Grodowski (agrodowski@epo.org),
04/08/2009	1.0.2	The update of FAQ to address a change in Document Retrieval service	Adam Grodowski (agrodowski@epo.org),
17/08/2009	1.0.3	Number service description was added.	Alexander Khval (akhval@epo.org)
22/09/2009	1.0.4	The documentation update for the 2.6.1 release	Adam Grodowski (agrodowski@epo.org),
11/12/2009	1.0.5	REST interface update	Alexander Khval (akhval@epo.org)

Table of Contents

Web Service Definition and Description

1. Introduction

1.1 Purpose

This document provides a description of all web services provided by Open Patent Services system (hence referred as OPS, OPS v.2, the system, or the project). It is intended to capture and convey the details of possible interactions. It has been addressed to all current and future users of OPS v.2 which would like to find out how to utilize published web services to obtain patent publication data.

1.2 Scope

This document is focused on the definition and usage of OPS v.2 web services only. It is aimed to provide all relevant information which may enhance the user experience with the system, especially simplify interactions between user and OPS v.2 system

1.3 Overview

The current chapter provides the general information about the content of this document and introduces the list of definitions, acronyms and abbreviations being used inside. The list of related resources is provided which is advised to learn.

The second chapter summarizes the architecture of OPS and how it is related to Web Service Architecture and Service Oriented Architecture together with some implementation details. This section is addressed for those who would like to receive some background information about web services and service oriented solutions in general. It is not required to read it for learning how to use the web services, but it helps to understanding what they are.

The third chapter presents the shared aspects of all services, such as: the data structure of requests/responses, the kind of request/responses used by web services, how the data is transported and encapsulated, and some rules for handling errors.

The fourth chapter covers all available web services providing detailed information how to call each service together with some input/output call examples. In case when there are several ways to utilize the service all possible call scenarios are described.

The fifth chapter presents the Frequently Asked Questions with answers.

1.4 Acronyms, Abbreviations and Concepts

The following abbreviations are frequently used inside this document (sorted alphabetically):

Abbreviation	Description
BNF	Backus–Naur Form
CQL	Contextual Query Language or Common Query Language
DTD	Document Type Definition
EBNF	Extended Backus–Naur Form
ECLA	European Classification system
EPO	European Patent Office
HTTP	Hypertext Transfer Protocol
IPC*digit*	International Patent Classification, *digit*-th release (if skipped, 8th is assumed)

MIME	Multipurpose Internet Mail Extensions
OPS	Open Patent Services
OPS v.1	Open Patent Services Version 1
OPS v.2	Open Patent Services Version 2
PDF	Portable Document Format
PNG	Portable Network Graphics
SOA	Service Oriented Architecture, or Service Orientation (Approach)
SOAP	Simple Object Access Protocol
ST.36	WIPO Standard 36: Processing of patent information using XML (Nov, 2007)
TIFF	Tagged Image File Format
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
WS-*term*	*a term* in context of the Web Services (Architecture)
XML	Extensible Markup Language
XSD	XML Schema Definition Language

The following concepts are crucial to fully understand the nature of the project. There are described for the sole purpose of information only. Since they are not formally conceptualized, in no way they might be used as a legal definition.

Concept	Description
Patent application	The formal paperwork filed by an inventor (or by a patent attorney or patent agent on the inventor's behalf) seeking a patent for a specific invention. There are several parts to a patent application, the most important of which are the sections containing the claims and the specification.
Patent publication	A patent application published generally 18 months after a priority date.
Patent priority	The Paris Convention for the Protection of Industrial Property was established in 1883. It established the system of priority rights, under which applicants have up to 12 months from first filing their patent application (usually in their own country) in which to make further subsequent applications in each signatory country and claim the original priority date. The first filing data (country, number and date) is thus known as priority data for that invention. *For further details, especially for the details of priority number formats, please refer to the Priority Number help topic in esp@cenet portal.*
Patent publication kind code	A code which includes a letter and in many cases a number, used to distinguish the kind of patent document (example, publication of an application for a patent with/without search document) and the level of publication (example first publication, second

	publication or corrected publication). *For further details, especially for the details of kind codes formats depend on countries, please refer to the Kind Codes help topic in esp@cenet portal.*
Patent publication date	A date of a publication, meaning, a date when a described invention becomes publicly available.
Patent application claims	The part of the patent which define the scope of the legal protection sought for the invention. The description and drawings are used to interpret the claims. *For further details, please refer to the Claims help topic in esp@cenet portal.*
The (simple) Patent family	All documents sharing exactly the same set of priorities.
INPADOC Patent family	All documents sharing directly or indirectly at least one priority.

1.5 References

It is not mandatory, however strongly recommended, to learn from those resources to fully understand the background and environment of Open Patent Services system.

[CQL-INT] **A Gentle Introduction to CQL**
Location: http://zing.z3950.org/cql/intro.html
Description: Provides the general information about CQL, together with some query examples.

[CQL-SPC] **CQL: Common Query Language, Version 1.1**
Location: http://www.loc.gov/standards/sru/sru1-1archive/cql.html
Description: Defines CQL, a formal language for representing queries to information retrieval systems such as web indexes, bibliographic catalogs and museum collection information. The design objective is that queries be human readable and writable, and that the language be intuitive while maintaining the expressiveness of more complex languages.

[DTD] **Extensible Markup Language (XML) 1.0 (5th Edit.): Document Type Definition**
Location: http://www.w3.org/TR/REC-xml/#dt-doctype
Description: Describes how to contain or point to markup declarations that provide a grammar for a class of documents.

[ECLA] **ECLA search system**
Location: http://v3.espacenet.com/eclasrch
Description: The European Classification system (ECLA) is used by the EPO for carrying out patent application searches. It is built on top of the International Patent Classification system (IPC), and is constantly being revised and updated. The current 8th edition of the IPC covers 70 000 groups, while ECLA covers 134 000 groups.

[EPO-EXCH] **EPO Exchange Document 2.1.4**
Location: http://documents.epo.org/projects/babylon/eponet.nsf/0/2D51D85DBF1AE4CBC125 757400290738/$File/T09_01_ST36_User_Documentation_vs_2_1_4.pdf
Description: Defines structures for exchange of patent documents used in Patent Information Services, together with an abstract.

[EPO-FAIR] **Fair use charter for esp@cenet and OPS**

Location: http://www.epo.org/patents/patent-information/free/fair-use.html
Description: A set of rules for using web services by clients of Open Patent Services.

[EPO-FTXT] **EPO Full Text Schema 1.0**
Location: http://ops.epo.org/schema/fulltext-documents.xsd
Description: Describes the structures used for exchange of a patent descriptions, claims and drawings, compliant with ST.36

[ESP-HELP] **esp@cenet Help index**
Location: http://ep.espacenet.com/help?topic=index&locale=en_ep&method=handleHelpTopic
Description: Presents the topics of esp@cenet help system, especially glossary of words and terms related to patent domain.

[IPC8] **IPC Version 8 [2006.01]**
Location: http://www.wipo.int/classifications/ipc/ipc8/?lang=en
Description: A database which contains all IPC classes (8th release)

[IPC-ST8] **Standard ST.8 - Standard recording of International Patent Classification (IPC) symbols on machine-readable records**
Location: http://www.wipo.int/standards/en/pdf/03-08-01.pdf
Description: Provides that symbols of the International Patent Classification (IPC) should be presented on machine-readable records for the exchange of information in machine-readable form in a fixed-length field in 50 positions, each part of the International Classification symbol being recorded in specific positions and in the manner prescribed

[OPS-IOS] **Open Patent Services Input and Output Schema**
Location: http://ops.epo.org/schema/ops.xsd
Description: Defines structures of input and output documents for OPS v.2

[OPS-WSDL] **Open Patent Services WSDL**
Location: http://ops.epo.org/wsdl/ops.wsdl
Description: Provides the definitions for all web services available in OPS architecture in WSDL format.

[OPS-WWW] **Open Patent Services**
Location: http://ops.epo.org
Description: OPS system delivers production stable and high quality raw patent data via Web Services.

[SOA-RA] **Reference Architecture for Service Oriented Architecture, 1.0**
Location: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.html
Description: Specifies the OASIS Reference Architecture for SOA. While it remains abstract in nature, it describes one possible template upon which a SOA concrete architecture can be built.

[SOA-RM] **Reference Model for Service Oriented Architecture, 1.0**
Location: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html
Description: An abstract framework for understanding significant entities and relationships between them within a service-oriented environment, and for the development of consistent standards or specifications supporting that environment.

[SOAP] **SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)**

Location: http://www.w3.org/TR/soap12-part1/

Description: Defines, using XML technologies, an extensible messaging framework containing a message construct that can be exchanged over a variety of underlying protocols.

[SOAP-UI] **soapUI**

Location: http://www.soapui.org

Description: An application broadly used for testing the web services. Automatically discovers all available web services from the corresponding WSDL file.

[ST36] **WIPO ST.36 Document, Ver. 1.2, November 2007**

Location: http://www.wipo.int/export/sites/www/standards/en/pdf/03-36-01.pdf

Description: The purpose of the Standard is to provide logical, system-independent structures for patent document processing, whether for text or image data.

[WS-A] **Web Services Architecture**

Location: http://www.w3.org/TR/ws-arch/

Description: Introduces the concepts of the architecture of web services.

[WS-G] **Web Services Glossary**

Location: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

Description: Provides a glossary for web service domain.

[WSDL] **Web Services Description Language (WSDL), Version 1.1**

Location: http://www.w3.org/TR/wsdl

Description: WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME

[XSD-P1] **W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures**

Location: http://www.w3.org/TR/xmlschema11-1/

Description: Specifies the XML Schema Definition Language, which offers facilities for describing the structure and constraining the contents of XML documents, including those which exploit the XML Namespace facility. Substantially reconstructs and considerably extends the capabilities found in XML document type definitions (DTDs).

[XSD-P2] **XML Schema Part 2: Datatypes Second Edition**

Location: http://www.w3.org/TR/xmlschema-2/

Description: Defines facilities for defining datatypes to be used in XML Schemas as well as other XML specifications. The datatype language, which is itself represented in XML 1.0, provides a superset of the capabilities found in XML 1.0 document type definitions (DTDs) for specifying datatypes on elements and attributes.

2. Architecture of Open Patent Services
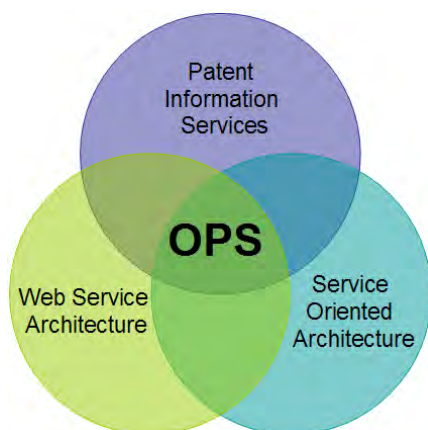
Figure 1 - Open Patent Services Architecture placement

The architecture of Open Patent Services gains from both Web Services and Service Oriented architectures. To fully understand its nature those two shall be briefly described in this chapter.

The greatest benefit from WS-Architecture is a very recognizable communication model based on standardized protocols for data encapsulation and transfer implemented by well-known technologies.

The main benefits from being focused on services rather than messages lay in reusability and interoperability, which promotes collaboration between many teams while decreasing the cost of further extensions of the system.

As it will be shown, Open Patent Services successfully implements those two approaches, promoting the easy access to the technical details of published inventions.

2.1 Web Services Architecture

The best way to describe the architecture of web services is to cite the underlying definition of a web service from W3C:
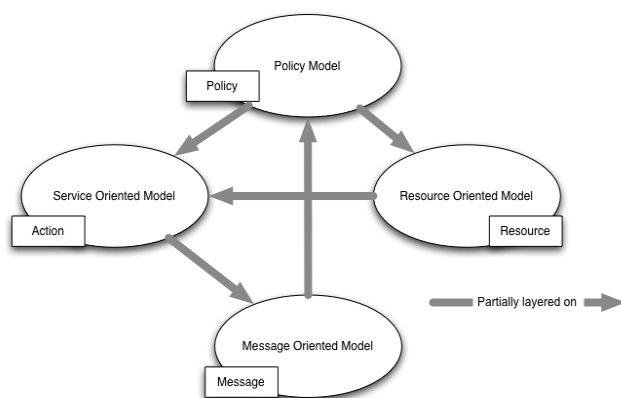
Figure 2 - Four models of Web Service Architecture
Copyright © 2004 W3C® (MIT, ERCIM, Keio), All Rights Reserved.

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Therefore, the Web Services Architecture can be projected as a structure of a system for supporting web services which comprise the system components, the way of accessing those externally and any relationships between them.

The architecture has four models presented on Figure 2, each one focusing on a different concept of the web service architecture. The underlying concepts: Action, Policy, Resource and Message are simple enough not to define them formally; they are assumed to be commonly understood.

The details of models intra-layering is out of scope of this brief description.

The four models of WS-Architecture shall now be briefly presented

1. The Message Oriented Model, which focuses on the message structure and transportation without a particular reference to the reason for messaging nor its significance.

2. The Service Oriented Model, which focuses on the service revealing its dynamic aspect (action) rather than a static content (message body) with an emphasis on how the service relates to the real world.

3. The Resource Oriented Model, which focuses on resources location, external representation and ownership rather than its content and dynamism.

4. The Policy Mode, which shows the constraints between services/resources and agents who create/process them.
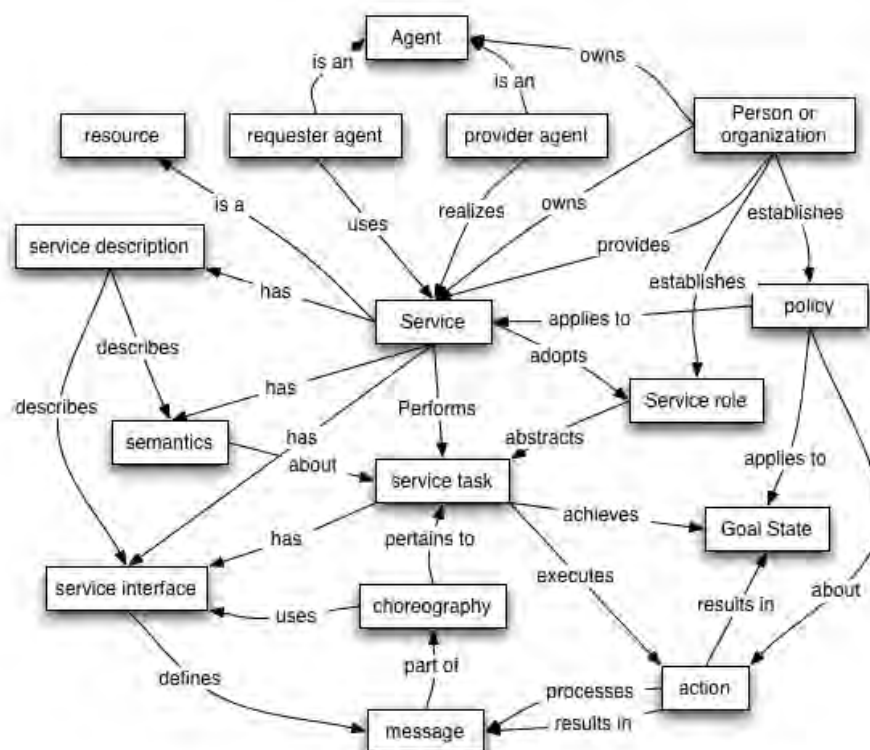
Figure 3 - Service Oriented Model
Copyright © 2004 W3C® (MIT, ERCIM, Keio), All Rights Reserved.

The most complex model is the Service Oriented Model, since each service, conceptually, is an owned resource which creates some messages in scope of a defined access policy.

How concepts of Service, Resource, Message, Policy and Action rely on each other is shown on Figure 3.

2.2 Service Oriented Architecture

SOA introduces the concept of a service as the most basic building block of the underlying system. This approach is more related to the design and system abstraction rather than to a communication model, which is a difference to WS-Architecture. Therefore, those two architectural concepts should be perceived as independent from each other , although, they quite often interact with each other and gain from this collaboration in modern systems.

To fully understand the architecture which is service oriented, its underlying concept must be introduced first and the paradigm of the system as a (virtual) machine must be abandoned for a moment.

Among many other sources, a service is well defined in [SOA-RM], which is cited below:

A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. A service is provided by an entity – the service provider – for use by others, but the eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider.

The same document conceptualize Service Oriented Architecture through the Ecosystem perspective, which is a unique definition, independent from a technical field:

Instead of visualizing a SOA as a single complex machine, it is perhaps more productive to think of it as an ecosystem: a space where people, machines and services inhabit in order to further both their own objectives and the objectives of the larger community.

[...] Taking an ecosystems perspective often means taking a step back: for example, instead of specifying an application hierarchy, we model the system as a network of peer-like entities; instead of specifying a hierarchy of control, we specify rules for the interactions between participants.

*[...] a SOA is a **medium** for **exchange of value** between independently acting **participants**;*

From IT perspective, it has been commonly accepted that to change any Enterprise environment into SOA Ecosystem means at least to adopt SOA in Software Development Process by following some design patterns and principles:

1. Standardized Service Contract
   "Service has description, semantics and interface", see on Figure 3.
   The key principle. A service needs to be formally described in a corresponding WSDL file: which data types and data models are used, and what is the network binding, meaning, the location of a service as a resource in web space.

2. Service Loose Coupling
   Following this principle means to design a service with an emphasis on reducing (loosening) dependencies between service contract, realization and clients.

3. Service Abstraction
   "Service performs a task, adopted from (business) role ", see on Figure 3.
   A service should hide as much of the underlying details as possible. By providing formal service description in WSDL the dynamic aspect of a service (action) is hidden and reduced into static request/response data structures (service task), well defined in XSD.

4. Service Reusability
   A service should be placed as an enterprise resource with clear rules of access so everyone could use it, especially another web services. Chaining web services promote business oriented modeling, where a flow of web services should correspond to the flow of business process. A chain of web services calling each other is also recognizable as "service configuration".

5. Service Autonomy
   For services to carry out their capabilities consistently and reliably, their underlying solution logic needs to have a significant degree of control over its environment and resources.

6. Service Statelessness
   A service should be carefully designed to remain "stateful" (to manage the state of carried information) only when required, to enhance its availability and scalability potential.

7. Service Discoverability
   "Service is a resource, owned by person or organization", see on Figure 3.
   To discover the service means to know how to communicate with it, and what to expect from it. By making WSDL file publicly available and promote the knowledge of processing, it is ensured that all described services are easily accessible.

8. Service Composability
   As the business described by the business process changes, new services are being designed to provide an access to new capabilities, services already present are becoming more complex, more frequently used, and service configurations turn into a more sophisticated chains. It is crucial to carefully design new services and to update already existing services to avoid massive retrofitting effort, and each service is expected to be capable of participating as a configuration member effectively, regardless of whether its need to be immediately chained.

9. Service Interoperability
   It is fundamental to every one of the principles already described, and each of the eight principles supports or contributes to it in some manner.

2.3 Summary

As a result of convergence between WS-Architecture and SOA, with respect to the responsibilities of Patent Information Systems, the Open Patent Services system has been created and advertised.

By learning from provided WSDL file, anyone with basic knowledge of SOAP and web systems can benefit by consuming provided Patent related information and chaining available services into its own automated business processes.

As a matter of fact, till the end of May 2009, the OPS v.2 is being used intensively with a transferred data amount between 30-50 GB per week.

OPS v.2: Average data transferred per day in a given week (GB)
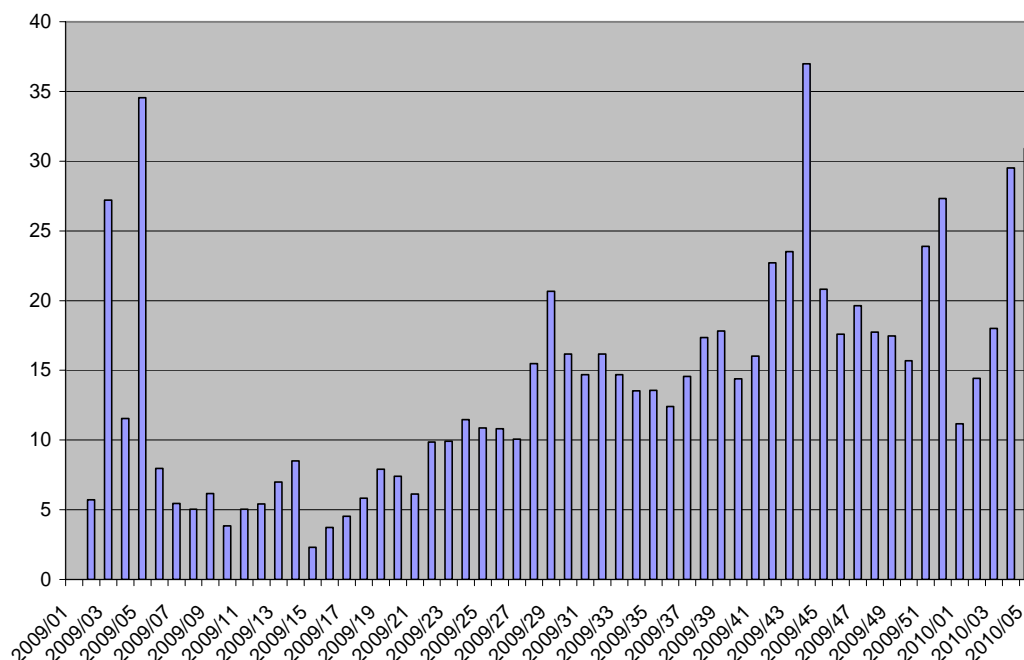
3. Open Patent Services changes

3.1 Changes between 2.6.0 and 2.6.1

3.1.1 *Revised services requests and responses of currently existing functionalities*

INPADOC services are not accepting application nor priority references in EPODOC format anymore.

Bibliography retrieval service rejects references having non alphanumeric characters except for wildcard characters.

Document Inquiry service provides a correct DOCDB publication number.

It is possible to use kind code other than %% in application/priority DOCDB references.

3.1.2 *Revised exception messages*

There are some new more detailed error messages provided instead of general error message, especially related to the CQL parsing problems.

3.1.3 *New citation types and citation category revealed in bibliography response*

In the newest release Open Patent Services supports citations during search, opposition, provided by applicant and others. A `reference-cited` element within `exchange-document` structure has been changed to provided multiple types of citations distinguished by `srep-phase` attribute of `citation` element. Citations are grouped by `srep-phase` value and numbered separately within group (in `sequence` attribute of `citation` element) and patent/non-patent citation type (in `num` attribute of either `patcit` or `nplcit` element).

The numbering of citations is not backward compatible which can be observed below in an example of `reference-cited` content between releases.

The citation category illustrates the citation importance and is provided within `category` element as a string combined from the following values:

Category letter	Category description
X	particularly relevant if taken alone
Y	particularly relevant if combined with another document of the same category
A	a technological background
O	a non-written disclosure
P	an intermediate document
T	a theory or principle underlying the invention
E	an earlier patent document, but published on, or after the filing date
D	a document cited in the application
L	a document cited for other reasons

The placement of non-patent (`nplcit`) and patent (`patcit`) citations group was exchanged, in the newest release patent citations are provided before non-patent citations.

The text of non-patent citations is stripped of multiple whitespaces and category indicator if present.

```
<references-cited>
  <citation sequence="1">
    <nplcit>
      <text>[X]     PATENT ABSTRACTS OF JAPAN vol. 2002, no. 04, 4 August
2002      (2002-08-04) -&amp; JP 2001 352271 A (HITACHI METALS LTD), 21
December 2001 (2001-12-21)</text>
    </nplcit>
  </citation>
  <citation sequence="2">
    <nplcit>
      <text>[A]     PATENT ABSTRACTS OF JAPAN vol. 1995, no. 11, 26
December     1995 (1995-12-26) -&amp; JP 07 202504 A (MURATA MFG CO
LTD), 4        August 1995 (1995-08-04)</text>
    </nplcit>
  </citation>
  <citation sequence="3">
    <nplcit>
      <text>[A]     PATENT ABSTRACTS OF JAPAN vol. 1999, no. 01, 29
January      1999 (1999-01-29) -&amp; JP 10 276003 A (HITACHI METALS
LTD),       13 October 1998 (1998-10-13)</text>
    </nplcit>
  </citation>
  <citation sequence="4">
    <nplcit>
      <text>[A]     PATENT ABSTRACTS OF JAPAN vol. 1995, no. 11, 26
December      1995 (1995-12-26) -&amp; JP 07 202502 A (MURATA MFG CO
LTD), 4       August 1995 (1995-08-04)</text>
    </nplcit>
  </citation>
  <citation sequence="1">
    <patcit>
      <text>EP1223634 A2</text>
    </patcit>
  </citation>
  <citation sequence="2">
    <patcit>
      <text>JP2001352271 A</text>
    </patcit>
  </citation>
  <citation sequence="3">
    <patcit>
      <text>JP7202504 A</text>
    </patcit>
  </citation>
  <citation sequence="4">
    <patcit>
      <text>JP10276003 A</text>
    </patcit>
  </citation>
  <citation sequence="5">
    <patcit>
      <text>JP7202502 A</text>
    </patcit>
  </citation>
</references-cited>
```

```
<references-cited>
  <citation sequence="1" srep-phase="SEA">
    <patcit num="1">
      <text>EP1223634 A2</text>
    </patcit>
    <category>A</category>
  </citation>
  <citation sequence="2" srep-phase="SEA">
    <patcit num="2">
      <text>JP2001352271 A</text>
    </patcit>
    <category>X</category>
  </citation>
  <citation sequence="3" srep-phase="SEA">
    <patcit num="3">
      <text>JP7202504 A</text>
    </patcit>
    <category>A</category>
  </citation>
  <citation sequence="4" srep-phase="SEA">
    <patcit num="4">
      <text>JP10276003 A</text>
    </patcit>
    <category>A</category>
  </citation>
  <citation sequence="5" srep-phase="SEA">
    <patcit num="5">
      <text>JP7202502 A</text>
    </patcit>
    <category>A</category>
  </citation>
  <citation sequence="6" srep-phase="SEA">
    <nplcit num="1">
      <text>PATENT ABSTRACTS OF JAPAN vol. 2002, no. 04, 4 August 2002
(2002-08-04) -&amp; JP 2001 352271 A (HITACHI METALS LTD), 21 December
2001 (2001-12-21)</text>
    </nplcit>
    <category>X</category>
  </citation>
  <citation sequence="7" srep-phase="SEA">
    <nplcit num="2">
      <text>PATENT ABSTRACTS OF JAPAN vol. 1995, no. 11, 26 December 1995
(1995-12-26) -&amp; JP 07 202504 A (MURATA MFG CO LTD), 4 August 1995
(1995-08-04)</text>
    </nplcit>
    <category>A</category>
  </citation>
  <citation sequence="8" srep-phase="SEA">
    <nplcit num="3">
      <text>PATENT ABSTRACTS OF JAPAN vol. 1999, no. 01, 29 January 1999
(1999-01-29) -&amp; JP 10 276003 A (HITACHI METALS LTD), 13 October 1998
(1998-10-13)</text>
    </nplcit>
    <category>A</category>
  </citation>
  <citation sequence="9" srep-phase="SEA">
    <nplcit num="4">
      <text>PATENT ABSTRACTS OF JAPAN vol. 1995, no. 11, 26 December 1995
(1995-12-26) -&amp; JP 07 202502 A (MURATA MFG CO LTD), 4 August 1995
(1995-08-04)</text>
    </nplcit>
    <category>A</category>
  </citation>
```

```
            <citation sequence="1" srep-phase="EXA">
              <patcit num="1">
                <text>EP0621653 A2</text>
              </patcit>
            </citation>
            <citation sequence="2" srep-phase="EXA">
              <patcit num="2">
                <text>EP0837516 A2</text>
              </patcit>
            </citation>
            <citation sequence="3" srep-phase="EXA">
              <patcit num="3">
                <text>US5767721 A</text>
              </patcit>
            </citation>
            <citation sequence="4" srep-phase="EXA">
              <patcit num="4">
                <text>EP0700169 A2</text>
              </patcit>
            </citation>
            <citation sequence="5" srep-phase="EXA">
              <nplcit num="1">
                <text>- EISENBERG J.A.; CHAMBERLAIN T.B.; SLOAN L.R.: 'High
isolation 1-20 GHz MMIC switches with on-chip drivers' IEEE 1989
MICROWAVE AND MILLIMETER-WAVE MONOLITHIC CIRCUITS SYMPOSIUM 15 June 1989,
NEW YORK USA, pages 41 - 45</text>
              </nplcit>
            </citation>
          </references-cited>
```

3.1.4   Bibliography at once

There is a new functionality provided within Bibliography Search, Equivalents Inquiry and some INPADOC services which allows to request for bibliographic data of a corresponding publication result simultaneously.

Request element of those services has been extended with an additional attribute `with-biblio`.

BI

3.1.5   Extended bibliography search and refined CQL implementation

New CQL indexes have been created for different citation types and for a simple family identifier.

3.1.6   Revised Bulk mode

Document Inquiry accepts multiple requests, up to 30.

Bibliography Request accepts more multiple requests, up to 100.

3.1.7   RESTful Bibliography Request

Bibliography Request service, additionally to SOAP, has got a REST interface.

3.1.8   New service for number translations

Number Service is designed to provide a numbers standardization facilities. It converts number in one of original/docdb/epodoc format to another format of the same list. Number Service is not delivered within a EPTOS box.

4.   General information about OPS web services

4.1   Commonly used data structures

All information retrieved from OPS v.2 are embedded within an XML document with a well-defined structure which contains several namespaces broadly used in other EPO systems. Those structures are summarized in the table below:

Data Structure document	Namespace	Description
Open Patent Services Input and Output Schema	ops	Defines the structure of the input and the output documents in OPS v.2. Additionally, defines a structure used for exchange a legal information about the patent publication.
EPO Exchange Document 2.0	exch	Defines structures for exchange of patent documents, including abstracts, in compliance with WIPO ST.36 Standard. Broadly used in almost every web service of OPS v.2, both for input and output.
EPO Full Text Schema 1.0	ftxt	Describes structures for providing description, claims and drawings from a patent publication.

Depend on the chosen web service, the response might contain elements from several namespaces.

4.2   Data transportation and encapsulation

All OPS v.2 web services requests and responses are sent via SOAP over HTTP to ensure not only standard way of data packaging but also a standard way of data transportation. Every SOAP message is a well-formed XML document having `Envelope` element which contains `Body` with the request message. Any SOAP request message might be seen as below, where `${request}` substitutes for one of a request structures described in 4.4:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ops="http://ops.epo.org" xmlns:exc="http://www.epo.org/exchange">
   <soapenv:Body>
     ${request}
   </soapenv:Body>
</soapenv:Envelope>
```

Next, the SOAP request message must be bound to a web service, using the web service address and web service name obtained from a corresponding WSDL file and send with HTTP using either GET or POST request on the web service address. The service name binding is done by adding HTTP header `SOAPAction`, as it can be observed in the example below, where the bibliography of a publication with number `EP1000000` is requested, using Bibliography Request service, described in 5.5.2. If HTTP POST request is used then `Content-Type` and `Content-Length` headers must be also properly set.

```
POST http://ops.epo.org/soap-services/biblio-retrieval HTTP/1.0
Content-Type: text/xml;charset=UTF-8
SOAPAction: "biblio-retrieval"
Host: ops.epo.org
Content-Length: 538

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ops="http://ops.epo.org" xmlns:exc="http://www.epo.org/exchange">
```

```
    <soapenv:Header/>
    <soapenv:Body>
        <ops:biblio-retrieval xmlns=http://www.epo.org/exchange
xmlns:ops="http://ops.epo.org" >
            <publication-reference data-format="epodoc">
                <document-id>
 <country></country>
 <doc-number>EP1000001</doc-number>
 <kind></kind>
                </document-id>
            </publication-reference>
        </ops:biblio-retrieval>
    </soapenv:Body>
</soapenv:Envelope>
```

The definition file for OPS v.2 (`ops.wsdl`) is published on OPS website.

Provided that the web service address is accessible and OPS system is working, `200 OK` response should be obtained with a SOAP response message having one of structures described in 4.5

```
HTTP/1.1 200 OK
Content-Type: text/xml;charset=utf-8
Content-Length: 5920
Connection: close

<SOAP-ENV:Envelope xmlns:SOAP
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
      ${response}
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In case when web service address is incorrect the system shall respond with 404 (not found) code, instructing for rechecking WSDL file:

```
HTTP/1.1 404 /soap-servicesXXX/biblio-retrieval
Content-Type: text/html
Content-Length: 434
Connection: close

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Page not found</title>
</head>
<body>
This URL is not found on the Open Patent Services home page.
<p/>
Please consult the <a href="wsdl/ops.wsdl">WSDL</a> for the correct URL
to call
Open Patent Services.

</body>
</html>
```

Additionally, system is responding with `500 Internal Server Error`, providing a `Fault` element with description of the problem in some situations. An example below simply demonstrates the system response to the incorrect document reference request, all problematic situations have been described in 4.6:

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/xml;charset=utf-8
Content-Length: 317
Connection: close

<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <SOAP-ENV:Fault>
            <faultcode>SOAP-ENV:Client</faultcode>
            <faultstring>At least one reference in the request has a wrong
value for data-format.</faultstring>
        </SOAP-ENV:Fault>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.3   SOAP clients

The general information needed to perform a call were provided in 4.2, however, instead of writing a new SOAP client, an existing one could be reused. Technically, there are many SOAP clients available on the market, among which the following are well recognizable:

- soapUI (www.soapui.org) - an application with GUI that is broadly used for web service testing

- Spring Web Services (http://static.springsource.org/spring-ws/sites/1.5/) - a Java component, one of Spring modules,

- Apache Axis 2 (http://ws.apache.org/axis2/) - a Java library,

- SOAPpy - a Python module,

4.4   Structures used for requests

In general, different groups of services (enumerated in the section 0) use different request formats, which shall be now presented. In all examples SOAP `Envelope` element is removed for clarity, as the transportation and encapsulation aspects have been already described in 4.2.

4.4.1   Request with a publication, an application or a priority reference

In almost all services (except for Bibliography Search service described in 5.5.1 and Document Retrieval service presented in 5.4.1.1) either a publication, an application or a priority reference is mandatory to perform a call. Each type of a reference requires a single document identifier element which structure is presented on Figure 4.

References elements are presented on Figure 5. Technically, all references are similar to each other, but `priority-claim` has some additional elements, and `application-reference` defines an additional attribute (`is-representative`).

Since this structure is shared between requests and responses some elements and attributes are optional and not used for requesting, such as: `name` and `date` elements inside `document-id`, all elements from `priority-claim` except of `document-id`, and all attributes except of `data-format`.
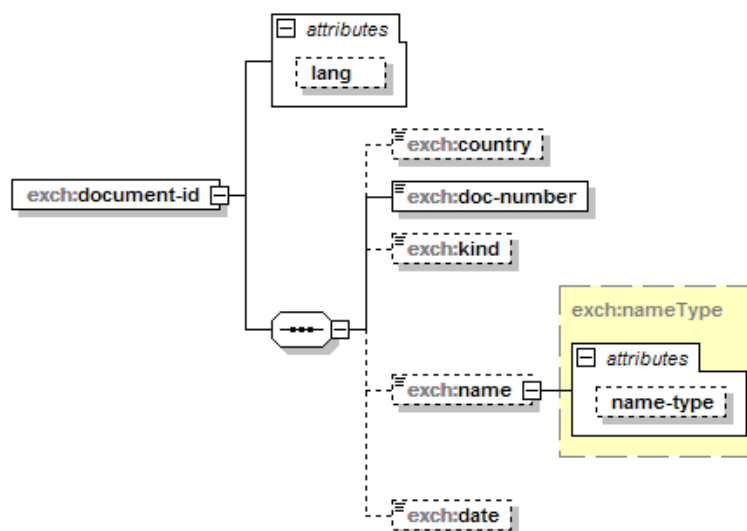
Figure 4 - The document identifier structure

The presence of other elements is determined by a `data-format` attribute and reference type.
The following rules apply:

Data Format	Reference type	Rule description
`epodoc`	any (publication, application, priority)	The `doc-number` element is mandatory. A kind code is optional and might contain up to 2 alphanumerical or pattern chars (%) which shall be resolved as 'any' character.
`docdb`	any (publication, application, priority)	All elements are mandatory. A kind code must contain up to 2 alphanumerical or pattern chars (%) which shall be resolved as 'any' character.

Since the references share the same underlying structure there are basically 2 call templates: one for a reference in `docdb` format, the latter for `epodoc` format, both presented below. All template variables written within curved brackets must be replaced with a real names corresponding to the service, as described in the table below, all variables written in square brackets must be replaced with a real value

```
<ops:${service-name} ${service-parameters}
xmlns="http://www.epo.org/exchange" xmlns:ops="http://ops.epo.org">
  <${reference-type} data-format="docdb">
    <document-id>
      <country>[country code]</country>
      <doc-number>[document number in DOCDB format]</doc-number>
      <kind>[document kind code]</kind>
    </document-id>
  </${reference-type}>
</${service-name}>
```

```
<ops:${service-name} ${service-parameters}
xmlns="http://www.epo.org/exchange" xmlns:ops="http://ops.epo.org">
  <${reference-type} data-format="epodoc">
    <document-id>
      <doc-number>[document number in EPODOC format]</doc-number>
    </document-id>
  </${reference-type}>
</${service-name}>
```

Template variable name	Possible values	Comments
${service-name}	ops:family-retrieval	Described in 5.1.1
	ops:family-and-legal-retrieval	Described in 5.1.2
	ops:family-and-biblio-retrieval	Described in 5.1.3
	ops:family-and-legal-and-biblio-retrieval	Described in 5.1.4
	ops:fulltext-inquiry	Described in 5.3.1
	ops:claims-retrieval	Described in 5.3.2
	ops:description-retrieval	Described in 5.3.3
	ops:legal-retrieval	Described in 5.2.1
	ops:legal-and-biblio-retrieval	Described in 5.2.2
	ops:document-inquiry	Described in 05.4.1
	ops:biblio-retrieval	Described in 5.5.2
${service-parameters}	depend on a chosen service	Please refer to corresponding section of a selected service
${reference-type}	exch:publication-reference	A reference to the publication document with a requested number
	exch:application-reference	A reference to the application document with a requested number
	exch:priority-claim	A reference to the priority claim document with a requested number

Service parameters correspond to the XML attributes, which are explained within the service description further below.

An example of a publication reference for the Bibliography Request service (see 5.5.2) is presented below. The docdb format has been used for a EP1000000 publication number with A1 kind code, and a proper service parameter (full-publication-cycle) is set ensuring that all kinds of the publication document are retrieved at once.

```
<ops:biblio-retrieval full-publication-cycle="true"
xmlns="http://www.epo.org/exchange" xmlns:ops="http://ops.epo.org" >
  <publication-reference data-format="docdb">
    <document-id>
      <country>EP</country>
      <doc-number>1000000</doc-number>
      <kind>A1</kind>
    </document-id>
  </publication-reference>
</ops:biblio-retrieval>
```
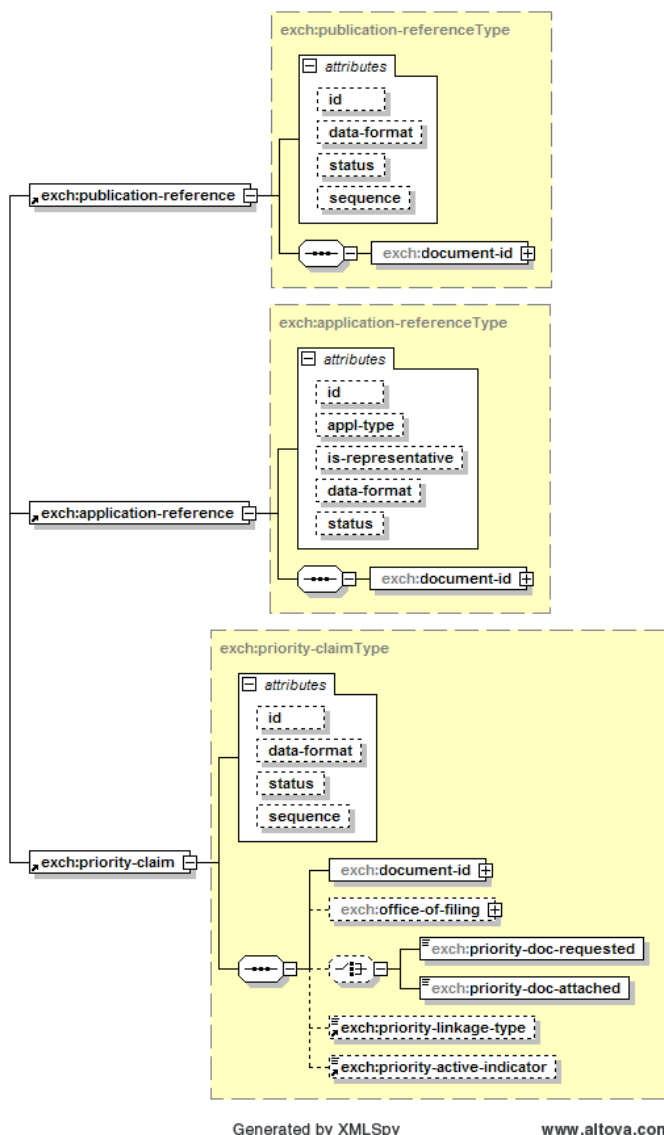
Figure 5 -Three possible kinds of a document reference

4.4.2   Standardization request

Number Service can be requested 2 different ways : using SOAP and REST approach. SOAP standardization request is nearly the same as a generic request for application, publication or priority reference described in a chapter 4.4.1. REST request is very intuitive and inline with standard way of accessing RESTful resources (for more details on

RESful standardization request interface see 5.5.4.1).
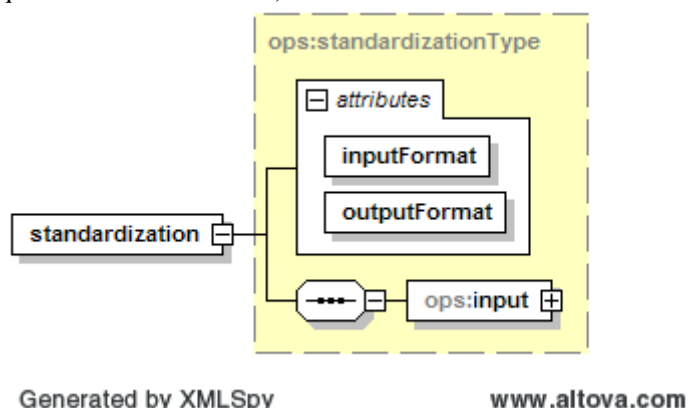
Figure 6 - SOAP standardization request structure

Standardization request duplicate a reference request described in a chapter 4.4.1 with one extension - the request reference element is wrapped into a `input` element. Additionally, input and output conversion format must be provided as a value of corresponding attributes `inputFormat` and `outputFormat`. Please make sure that `data-format` attribute of a reference element value is the same as a value of the `inputFormat` attribute. If the values are different then `inputFormat` is taking into account and `data-format` will be ignored.

4.4.2.1 Standardization request limitations

Current implementation of the Number Service has a list of limitations which will be omitted in later versions.

- List of possible values for input/output formats : original, docdb or epodoc.

Input format	Output formats
original	docdb, epodoc
docdb	epodoc
epodoc	

Figure 7 - List of conversion directions supported by Number Service

- Kind code and date elements are optional for both SOAP and REST requests and can be omitted (in case of REST interface replaced by '*' wildcard) together or each taken separately. However, the usage of these 2 additional elements increase accuracy of conversion extremely.

- Data limitations

Direction	Limitations
original to docdb	Country codes EP, FR, ES are fully supported. All other country codes expect date specified to have precise transformation.
original to epodoc	

4.4.3 Request for a document page
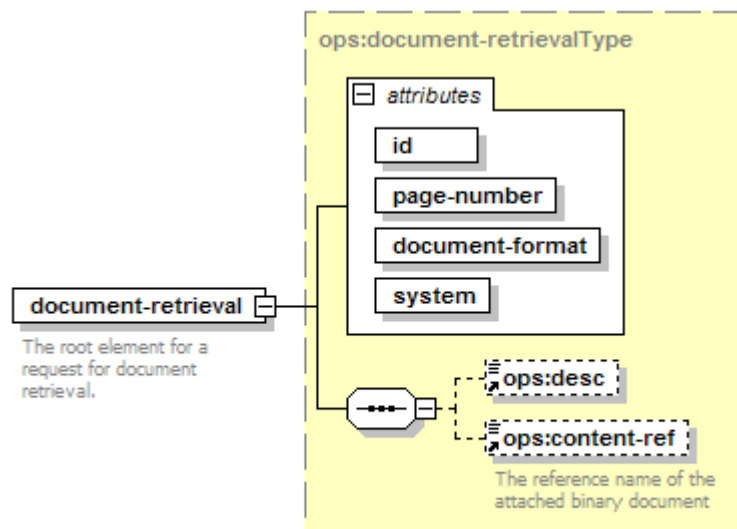
Generated by XMLSpy   www.altova.com

Figure 8 - The document retrieval structure

This kind of a request is uniquely used only by Document Retrieval Request, described in 5.4.1.1, not only for a request but also for a response. It requires a special document identifier, document format and page number to perform a call.

The example presents a request for a first page picture (often related to an abstract) of a publication EP1000000.

```
<ops:document-retrieval xmlns:ops="http://ops.epo.org"
  id="EP   1000000PAFP "
  page-number="1"
  document-format="SINGLE_PAGE_PDF"
  system="ops.epo.org"/>
```

Allowed values for each attribute are presented in a table below:

Attribute name	Possible values	Comments
id	depend on selected publication	To know the exact structure of underlying publication the publication metadata must be obtained first by calling Document Inquiry Request service, described in 5.4.1. The service returns the list of document instances, each one with a unique id, list of allowed format for this instance and number of pages.
page-number		
document-format		
system	ops.epo.org	Currently only one system value is allowed which identifies the OPS server.

When used as a response it provides additionally a reference to a single document page, without providing a page content itself. The details of the information provided within it are described below (in the order of occurrence).

Attribute name	Description
document-format	Identifies a requested page format. In OPS v.2, the following values are supported: SINGLE_PAGE_PDF, SINGLE_PAGE_TIFF and SINGLE_PAGE_PNG, corresponding respectfully to PDF, TIFF and PNG formats.
id	a unique identifier of a retrieved document instance
page-number	a number of a requested page
system	the OPS server identifier, OPS v.2 is identified by ops.epo.org

Element name	Description
desc	a short description of a retrieved document instance
content-ref	An artificial name composed of identifier and format abbreviation.

An example of a document-retrieval element referring to the third page in PDF format of a document instance with the identifier "EP   1000000A1 I " is shown below.

```
          <ops:document-retrieval document-format="SINGLE_PAGE_PDF"
     id="EP        1000000A1 I " page-number="3" system="ops.epo.org">
           <ops:desc>FullDocument</ops:desc>
            <ops:content-ref>EP        1000000A1 I .pdf</ops:content-ref>
          </ops:document-retrieval>
```

4.4.4   Request with CQL query

The call for Bibliography Search requires a query string in a CQL format (explained in 4.7). The example is presented below which queries for first 3 results of any publication having words: green and plant in the title.

```
          <ops:biblio-search xmlns:ops="http://ops.epo.org">
            <ops:query>ti=green and ti=plant</ops:query>
            <ops:range begin="1" end="3"/>
          </ops:biblio-search>
```

4.5   Structures used in responses

All responses are encapsulated within ops:world-patent-data element, but their structure highly depends on the web service type. The following section contains elements which are important parts of a service response.

4.5.1   Exchange document element

This elements holds all information related to the bibliography and an abstract of a publication. The details of the information provided are described below (in the order of occurrence). For a detailed structure information please refer to EPO Exchange Document.

Attribute name	Description
country	a country code of the underlying document,

`doc-number`	a document number (with leading zeros, if any),
`family-id`	a unique identifier of a simple family this document belongs to,
`kind`	a document kind code,
`system`	the OPS server identifier, OPS v.2 is identified by `ops.epo.org`
`status`	An optional attribute, present only when document could not be obtained.

Element	Description
`publication-reference` (in `docdb` format)	Refers to the publication in `docdb` format, with all fields properly resolved. If a pattern char (%) has been used, shall be now properly determined revealing the full kind code. Additionally, the publication date is provided in `date` element in the format `yyyyMMdd`, where `yyyy` refers to the publication year, `MM` to the publication month number (with the leading zero) and `dd` to the publication day number (with the leading zero)
`publication-reference` (in `epodoc` format)	Refers to the publication in `epodoc` format with a number field filled. Mind, that kind code attachment rules apply, therefore, a full kind code must be obtained from either `docdb` reference or from the `kind` attribute from a parent (exchange-document) element. Additionally, the publication date is provided in `date` element in the format `yyyyMMdd`, where `yyyy` refers to the publication year, `MM` to the publication month number (with the leading zero) and `dd` to the publication day number (with the leading zero)
`classification-ipc`	Provides the list of IPC1-7 classes the underlying publication has been classified into. *The content of this element is no longer supported for publication published after January 2006, please refer to `classification-ipcr` element instead which contains classes from IPC8*
`classification-ipcr`	Provides the list of IPC8 classes the underlying publication has been classified into, according to the ST.6 specification.
`classification-ecla`	Provides the list of ECLA classes the underlying publication has been classified into. *It is advised to use ECLA classification for it is more detailed in comparison to IPC8.*
`application-reference` (in `epodoc` format)	Refers to the application document in `epodoc` format with the application date.
`application-reference` (in `original` format)	An optional element which refers to the application document with a number as it was written on the original paper document, with the application date. Provided if available.

`priority-claim` (in `epodoc` format)	Contains a priority claim document reference in `epodoc` format with the priority date. The `sequence` attribute refers to the order in which priorities are printed on the published document.
`priority-claim` (in `original` format)	Contains a priority claim document reference with a number as it was written on the original paper document, with the priority date. The `sequence` attribute refers to the order in which priorities are printed on the published document. Provided if available.
`applicant` (in `epodoc` format)	Contains the applicant name(s) written in `epodoc` format.
`applicant` (in `original` format)	Contains the applicant name(s) as it is written on the original paper document, if available.
`inventor` (in `epodoc` format)	Contains the inventor name(s) written in `epodoc` format.
`inventor` (in `original` format)	Contains the inventor name(s) as it is written on the original paper document, if available.
`invention-title` (in French, German, English)	Contains the invention title in a French, German or English, if available. The language code is provided within `lang` attribute. There might be many `invention-title` elements which reflect the title structure, as printed on the publication document.
`citation`	Provides the publication references (in `epodoc` format) to the documents considered to be relevant during a single phase of a search/grant procedure. The kind code of a cited document is separated by a whitespace from the number. The `sequence` attribute refers to the order in which documents are printed on the search report document. The citation category illustrates the citation importance and is provided within `category` element as a string combined from the following values: X - particularly relevant if taken alone Y - particularly relevant if combined with another document of the same category A - technological background O - non-written disclosure P - intermediate document T - theory or principle underlying the invention E - earlier patent document, but published on, or after the filing date D - document cited in the application L - document cited for other reasons
`abstract` (in French, German, English)	Contains the abstract paragraph in a French, German or English, if available. The language code is provided within `lang` attribute of the `abstract` element. There might be many `abstract` elements which reflect the abstract structure, as printed on the publication document.

As an example, the exchange document element of a publication EP1000000 A1 is attached below. All applicant and inventor names are removed from this example.

```
<exchange-document country="EP" doc-number="1000000" family-id="19768124"
kind="A1" system="ops.epo.org">
  <bibliographic-data>
    <publication-reference data-format="docdb">
      <document-id>
        <country>EP</country>
        <doc-number>1000000</doc-number>
        <kind>A1</kind>
        <date>20000517</date>
      </document-id>
    </publication-reference>
    <publication-reference data-format="epodoc">
      <document-id>
        <doc-number>EP1000000</doc-number>
        <date>20000517</date>
      </document-id>
    </publication-reference>
    <classification-ipc>
      <text>B28B5/02</text>
      <text>B28B1/29</text>
      <text>B28B7/00</text>
    </classification-ipc>
    <classifications-ipcr>
      <classification-ipcr sequence="1">
        <text>B28B   1/   00          C I      </text>
      </classification-ipcr>
      <classification-ipcr sequence="2">
        <text>B28B   5/   00          C I      </text>
      </classification-ipcr>
      <classification-ipcr sequence="3">
        <text>B28B   7/   00          C I      </text>
      </classification-ipcr>
      <classification-ipcr sequence="4">
        <text>H02P   6/   08          C I      </text>
      </classification-ipcr>
      <classification-ipcr sequence="5">
        <text>B28B   1/   29          A I      </text>
      </classification-ipcr>
      <classification-ipcr sequence="6">
        <text>B28B   5/   02          A I      </text>
      </classification-ipcr>
      <classification-ipcr sequence="7">
        <text>B28B   7/   00          A I      </text>
      </classification-ipcr>
      <classification-ipcr sequence="8">
        <text>H02P   6/   08          A I      </text>
      </classification-ipcr>
    </classifications-ipcr>
    <classification-ecla>
      <classification-symbol>H02P6/08</classification-symbol>
    </classification-ecla>
    <classification-ecla>
      <classification-symbol>B28B1/29</classification-symbol>
    </classification-ecla>
    <classification-ecla>
      <classification-symbol>B28B5/02B2</classification-symbol>
    </classification-ecla>
    <classification-ecla>
      <classification-symbol>B28B7/00F</classification-symbol>
    </classification-ecla>
```

```
<application-reference data-format="epodoc">
  <document-id>
    <doc-number>EP19990203729</doc-number>
    <date>19991108</date>
  </document-id>
</application-reference>
<application-reference data-format="original">
  <document-id>
    <doc-number>99203729</doc-number>
  </document-id>
</application-reference>
<priority-claims>
  <priority-claim data-format="epodoc" sequence="1">
    <document-id>
      <doc-number>NL19981010536</doc-number>
      <date>19981112</date>
    </document-id>
  </priority-claim>
  <priority-claim data-format="original" sequence="1">
    <document-id>
      <doc-number>1010536</doc-number>
    </document-id>
  </priority-claim>
</priority-claims>
<parties>
  <applicants>
    <applicant data-format="epodoc" sequence="1">
      <applicant-name>
        <name><!-- 1 applicant name in epodoc format --></name>
      </applicant-name>
    </applicant>
    <applicant data-format="original" sequence="1">
      <applicant-name>
        <name><!-- 1 applicant name in a format as given --></name>
      </applicant-name>
    </applicant>
  </applicants>
  <inventors>
    <inventor data-format="epodoc" sequence="1">
      <inventor-name>
        <name><!-- 1 inventor name in epodoc format --></name>
      </inventor-name>
    </inventor>
    <inventor data-format="original" sequence="1">
      <inventor-name>
        <name><!-- 1 inventor name in a format as given --></name>
      </inventor-name>
    </inventor>
  </inventors>
</parties>
<invention-title lang="fr">Dispositif pour la fabrication de briques
crues utilisées dans l'industrie manufacturière des briques</invention-
title>
<invention-title lang="de">Vorrichtung zur Herstellung von
Steinformlingen für die Ziegelindustrie</invention-title>
<invention-title lang="en">Apparatus for manufacturing green bricks
for the brick manufacturing industry</invention-title>
<references-cited>
  <citation sequence="1">
    <patcit>
      <text>EP0680812 A1</text>
    </patcit>
  </citation>
```

```
            <citation sequence="2">
              <patcit>
                <text>NL9400663 A</text>
              </patcit>
            </citation>
            <citation sequence="3">
              <patcit>
                <text>DE3546191 A1</text>
              </patcit>
            </citation>
          </references-cited>
        </bibliographic-data>
        <abstract lang="en">
          <p>
            The invention relates to an apparatus (1) for manufacturing green
      bricks from clay for the brick manufacturing industry, comprising a
      circulating conveyor (3) carrying mould containers combined to mould
      container parts (4), a reservoir (5) for clay arranged above the mould
      containers, means for carrying clay out of the reservoir (5) into the
      mould containers, means (9) for pressing and trimming clay in the mould
      containers, means (11) for supplying and placing take-off plates for the
      green bricks (13) and means for discharging green bricks released from
      the mould containers, characterized in that the apparatus further
      comprises means (22) for moving the mould container parts (4) filled with
      green bricks such that a protruding edge is formed on at least one side
      of the green bricks. &lt;IMAGE&gt;
          </p>
        </abstract>
      </exchange-document>
```

4.5.2 Any reference.

There are 3 reference elements within `exch` namespace exist: `publication-reference`, `application-reference` and `priority-claim`. The structure of this elements is very similar and contains information about major parts of the document reference : country code, document number, kind code and a date together with reference format set as attribute of root element. Please find below examples of various types of references:

`publication-reference` :

```
        <publication-reference data-format="docdb">
          <document-id>
            <country>KR</country>
            <doc-number>20040083174</doc-number>
            <kind>A</kind>
            <date>20041001</date>
          </document-id>
        </publication-reference>
```

`application-reference`:

```
        <application-reference data-format="docdb">
          <document-id>
            <country>KR</country>
            <doc-number>20030017739</doc-number>
            <kind>A</kind>
          </document-id>
        </application-reference>
```

`priority-claim`:

```
        <priority-claim data-format="docdb">
          <document-id>
            <country>KR</country>
```

```
            <doc-number>20030017739</doc-number>
            <kind>A</kind>
            <date>20030321</date>
          </document-id>
       </priority-claim>
```

4.5.3   Publication reference element within OPS namespace

This element is used by Bibliography Search service only (see 5.5.1).

The structure of this element is almost exactly as the `publication-reference` element from `exch` namespace, mentioned in 4.5.1, but one additional attribute is added (`family-id`) which refers to the simple family identifier this publication belongs to. Exactly the same family identifier can be found in the same attribute of `exchange-document` element, described in 4.5.1.

To complete the picture, a response to a query for first 3 results of any publication having words: green and plant in the title (mentioned in 4.4.4) is presented below.

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="565"/>
  <ops:biblio-search total-result-count="320">
    <ops:query syntax="CQL">ti=green and ti=plant</ops:query>
    <ops:range begin="1" end="3"/>
    <ops:search-result>
      <ops:publication-reference data-format="docdb" family-id="40384451"
system="ops.epo.org">
        <document-id>
          <country>DE</country>
          <doc-number>102007048013</doc-number>
          <kind>A1</kind>
        </document-id>
      </ops:publication-reference>
      <ops:publication-reference data-format="docdb" family-id="37766681"
system="ops.epo.org">
        <document-id>
          <country>US</country>
          <doc-number>2009072624</doc-number>
          <kind>A1</kind>
        </document-id>
      </ops:publication-reference>
      <ops:publication-reference data-format="docdb" family-id="40340243"
system="ops.epo.org">
        <document-id>
          <country>DE</country>
          <doc-number>102008037185</doc-number>
          <kind>A1</kind>
        </document-id>
      </ops:publication-reference>
    </ops:search-result>
  </ops:biblio-search>
</ops:world-patent-data>
```

4.5.4   Legal element

The legal information is provided within `legal` element and the details of the information provided within are out of scope of this document.
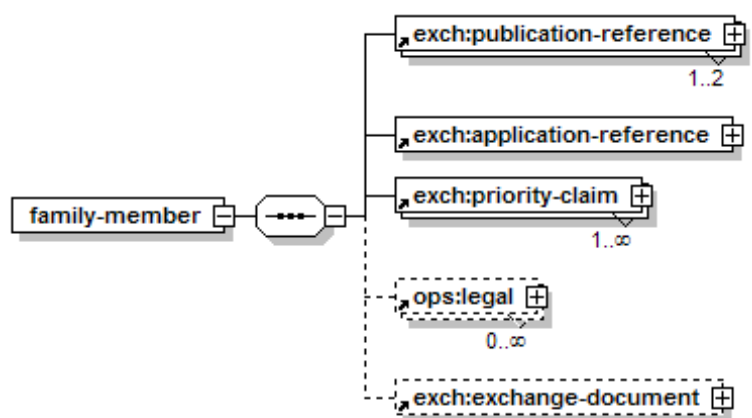
```
<ops:legal attributes>
  legal information
</ops:legal>
```

4.5.5   INPADOC Family member

Generated by XMLSpy   www.altova.com

Figure 9 - The INPADOC family member structure

This structure is broadly used in INPADOC related services (described in 5.1) and Legal status services (5.2). All elements within are already described (see 4.5.1 and 4.5.4)

The following example shows the response of a INPADOC Family Request service (described in 5.1.1) for a publication reference KR20040083174. As it is shown, the family is composed of this publication only, therefore, the response contains a single family-member element within the patent-family element.

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="230"/>
  <ops:patent-family legal="false">
    <publication-reference data-format="epodoc">
      <document-id>
        <country/>
        <doc-number>KR20040083174</doc-number>
        <kind/>
      </document-id>
    </publication-reference>
    <ops:family-member>
      <publication-reference data-format="epodoc">
        <document-id>
          <doc-number>KR20040083174</doc-number>
          <date>20041001</date>
        </document-id>
      </publication-reference>
      <publication-reference data-format="docdb">
        <document-id>
          <country>KR</country>
          <doc-number>20040083174</doc-number>
          <kind>A</kind>
          <date>20041001</date>
        </document-id>
      </publication-reference>
      <application-reference data-format="docdb">
        <document-id>
          <country>KR</country>
```

```
                    <doc-number>20030017739</doc-number>
                    <kind>A</kind>
                  </document-id>
              </application-reference>
              <priority-claim data-format="docdb">
                  <document-id>
                    <country>KR</country>
                    <doc-number>20030017739</doc-number>
                    <kind>A</kind>
                    <date>20030321</date>
                  </document-id>
              </priority-claim>
            </ops:family-member>
          </ops:patent-family>
        </ops:world-patent-data>
```

4.5.6 Inquiry result element
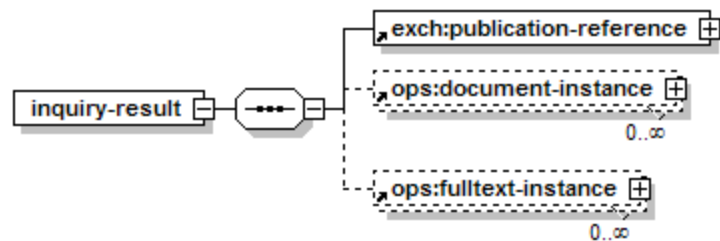
Generated by XMLSpy   www.altova.com

Figure 10 - The inquiry result element

In OPS v.2 this structure is used in 3 services: Full Text Inquiry Request service, described in 5.3.1, Document Inquiry Request service described in 05.4.1 and Equivalents Inquiry Request service described in 5.5.3.

The last one is using `fulltext-instance` elements explained in 4.5.10, the rest is using `document-instance` elements described in 4.5.7.

4.5.7   Document instance element

Figure 11 - The document instance structure

The document instance is a named part of the publication document. Every publication document must have at least one instance which corresponds to the full document structure as it has been published. Additionally, the publication document might have another instances, like: an instance corresponding to the drawings, or the instance corresponding to the magnified drawing disclosed on the first page related to the abstract.
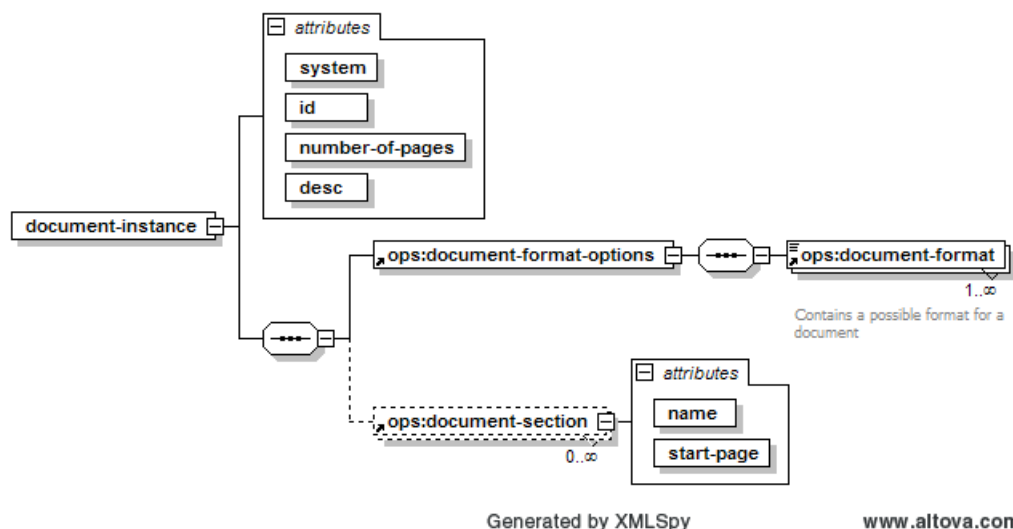
The `document-instance` element describes the structure of the publication document instance, especially its unique identifier, how many pages this instance has, what kind of page formats are available, how many sections this instance is composed of. The details of the information provided within it are described below (in the order of occurrence).

Attribute name	Description
`desc`	a short description of a document instance
`id`	a unique identifier of a document instance
`number-of-pages`	the number of pages an instance has
`system`	the OPS server identifier, OPS v.2 is identified by `ops.epo.org`

Element name	Description
`document-format`	Identifies a page format. In OPS v.2, the following values are supported: `SINGLE_PAGE_PDF`, `SINGLE_PAGE_TIFF` and `SINGLE_PAGE_PNG`, corresponding respectfully to PDF, TIFF and PNG formats.
`document-section`	Describes the section of this document instance by providing a name (in `name` attribute) and the number of a first section page (in `start-page` attribute).

An example of a document-instance element describing the full document structure of the publication `EP1000000 A1` is provided below.

```
<ops:document-instance desc="FullDocument"
id="EP       1000000A1 I " number-of-pages="12" system="ops.epo.org">
    <ops:document-format-options>
      <ops:document-format>SINGLE_PAGE_PDF</ops:document-format>
      <ops:document-format>SINGLE_PAGE_TIFF</ops:document-format>
    </ops:document-format-options>
    <ops:document-section name="ABSTRACT" start-page="1"/>
    <ops:document-section name="BIBLIOGRAPHY" start-page="1"/>
    <ops:document-section name="CLAIMS" start-page="3"/>
    <ops:document-section name="DESCRIPTION" start-page="2"/>
    <ops:document-section name="DRAWINGS" start-page="5"/>
    <ops:document-section name="SEARCH_REPORT" start-page="11"/>
</ops:document-instance>
```
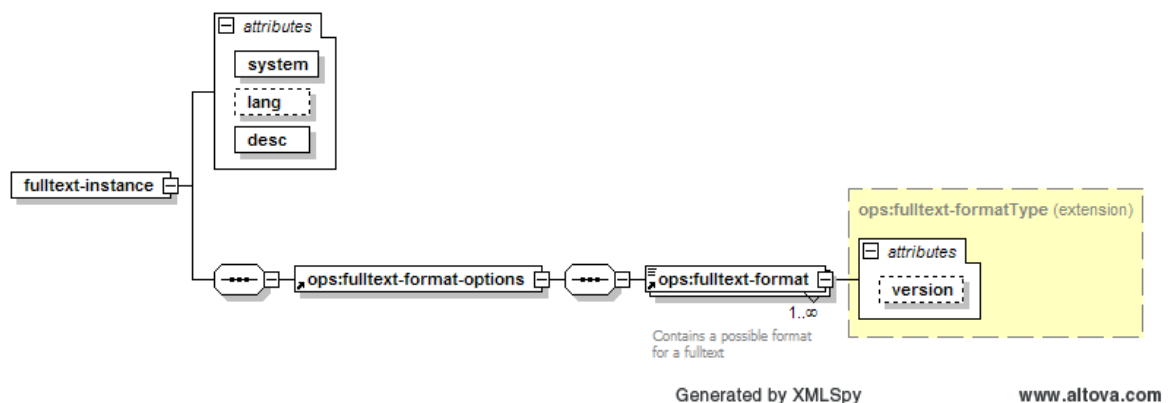
4.5.8 Document retrieval element

The `document-retrieval` element has already been described in 4.4.2.

4.5.9 Full text instance element

Figure 12 - The full text instance structure

This element is being used only by Full text Inquiry Request service, described in 5.3.1.

Comparably to the document instance, a full text instance is a part of a textual part of a publication document. This elements provides the information how this textual part is structured and what formats are available. Unlikely to the document instance, a full text instance is anonymous and cannot be further referred to.

The following examples illustrates the structure of a textual part of a publication `EP1000000 A1` as received from a Full text Inquiry Request service:

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="57"/>
  <ops:fulltext-inquiry>
    <publication-reference data-format="epodoc">
      <document-id>
        <country/>
        <doc-number>EP1000000</doc-number>
        <kind/>
      </document-id>
    </publication-reference>
```

```
<ops:inquiry-result>
  <publication-reference data-format="docdb">
    <document-id>
      <country>EP</country>
      <doc-number>1000000</doc-number>
      <kind>A1</kind>
    </document-id>
  </publication-reference>
  <ops:fulltext-instance desc="description" lang="EN"
system="ops.epo.org">
    <ops:fulltext-format-options>
      <ops:fulltext-format>text-only</ops:fulltext-format>
    </ops:fulltext-format-options>
  </ops:fulltext-instance>
  <ops:fulltext-instance desc="claims" lang="EN"
system="ops.epo.org">
    <ops:fulltext-format-options>
      <ops:fulltext-format>text-only</ops:fulltext-format>
    </ops:fulltext-format-options>
  </ops:fulltext-instance>
</ops:inquiry-result>
  </ops:fulltext-inquiry>
</ops:world-patent-data>
```

4.5.10 Full text document element
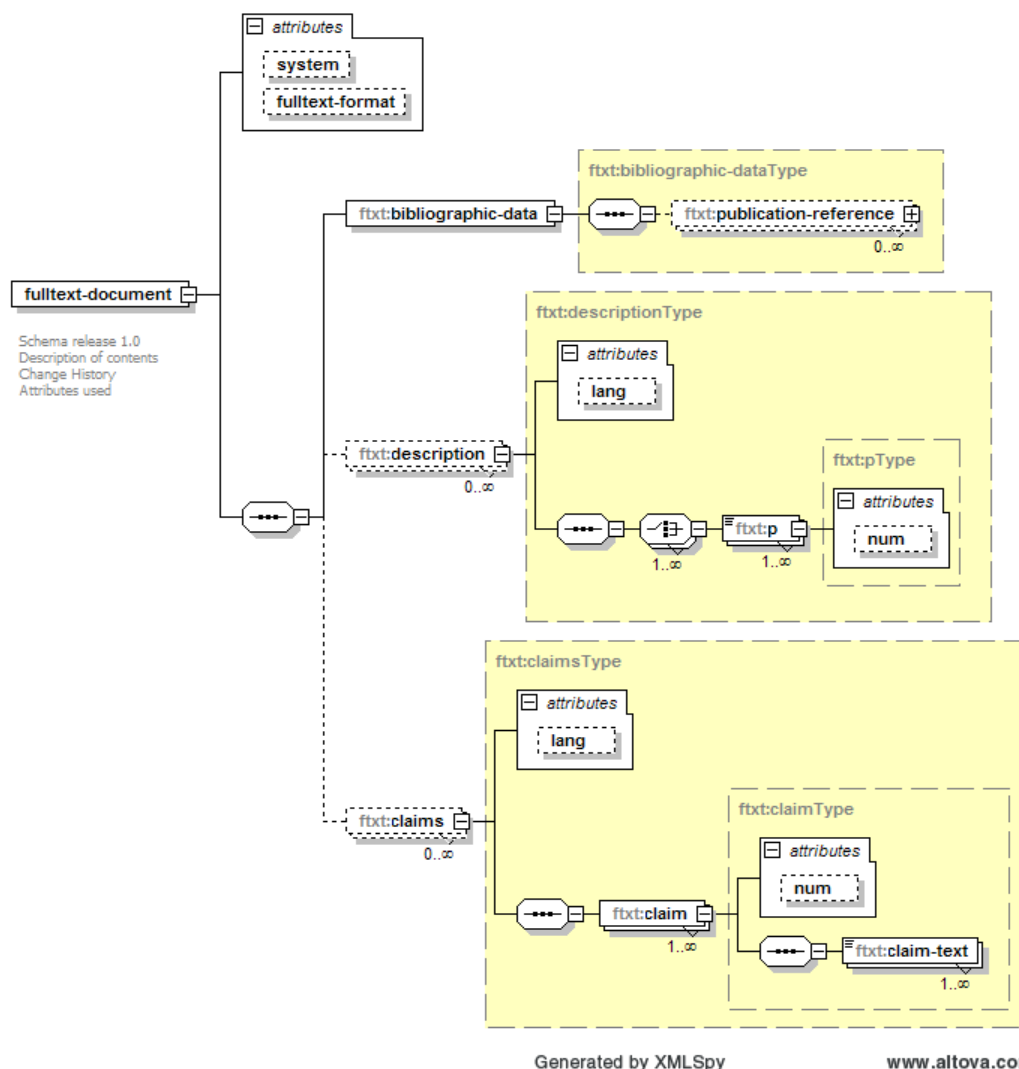
Figure 13 - The full text document structure

In OPS v.2 two services use this element shown on Figure 13 as a part of the response: Claims Request service and Description Request service described in 5.3.2 and 5.3.3.

This element contains a full textual part of a publication. The details of the information provided within it are described below (in the order of occurrence).

Attribute name	Description
fulltext-format	Identifies a text format. There is only one value supported in OPS v.2: TEXT_ONLY, but in general all available formats are provided by Full text Inquiry Request service, described in 5.3.1.
system	the OPS server identifier, OPS v.2 is identified by ops.epo.org

Element name	Description

publication-reference (in docdb format)	as described in 4.5.1, but with a different namespace: ftxt
claims	Provides the claims enclosed within the publication. The language code is provided in lang attribute.
description	An artificial name composed of identifier and format abbreviation. The language code is provided in lang attribute.

The structure of the fulltext-document element differs depend on the service: Claims Request service provides claims element, while Description Request service provides description element.

The following examples shows the structure of a claims of publication EP1000000 A1 as received from Claims Request service. Only the first claim is provided and the rest is removed from the example:

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="28"/>
  <ftxt:fulltext-documents xmlns:ftxt="http://www.epo.org/fulltext">
    <ftxt:fulltext-document fulltext-format="text-only"
system="ops.epo.org" xmlns="http://www.epo.org/fulltext">
      <bibliographic-data>
        <publication-reference data-format="docdb">
          <document-id>
            <country>EP</country>
            <doc-number>1000000</doc-number>
            <kind>A1</kind>
          </document-id>
        </publication-reference>
      </bibliographic-data>
      <claims lang="EN">
        <claim>
          <claim-text>1. Apparatus for manufacturing green bricks from
clay for the brick manufacturing industry, comprising a circulating
conveyor carrying mould containers combined to mould container parts, a
reservoir for clay arranged above the mould containers, means for
carrying clay out of the reservoir into the mould containers, means for
pressing and trimming clay in the mould containers, means for supplying
and placing take-off plates for the green bricks and means for
discharging green bricks released from the mould containers,
characterized in that the apparatus further comprises means for moving
the mould container parts filled with green bricks such that a protruding
edge is formed on at least one side of the green bricks. </claim-text>
          <!-- other claims are removed from the example -->
        </claim>
      </claims>
    </ftxt:fulltext-document>
  </ftxt:fulltext-documents>
</ops:world-patent-data>
```
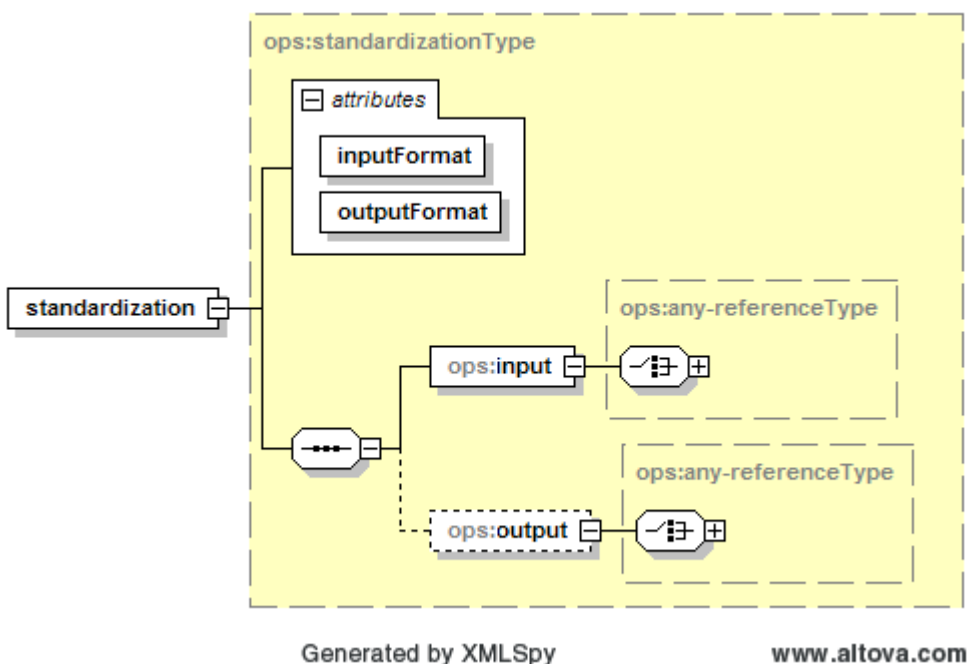
4.5.11 Standardization element

Generated by XMLSpy   www.altova.com

Figure 14 - The standardization element structure

Standardization element is produced by the Number Service as a major part of the response on a Standardization request described in chapter 4.4.2.

The element contains a reference converted into output format together with a copy of a requested reference both wrapped into corresponding `output` and `input` elements (see details below).

Attribute name	Description
inputFormat	Standardization input reference format. It can be one of `original`, `epodoc` or `docdb`.
outputFormat	Standardization output reference format. It can be one of `original`, `epodoc` or `docdb`.

For a full list of currently supported formats please refer to Figure 7 - List of conversion directions supported by Number Service.

Element name	Description
input	can be any reference in any supported format as described in a chapter 4.5.2.
output	contains a result of standardization reference in output format.

4.6 Error handling rules and exceptional non-error responses.

There are several situations when OPS system is responding with an error message embedded within SOAP `Fault` element with the description of the underlying problem. Those situations often refer to the web service unavailability, lack of server resources to process the request, wrong request format, or ambiguity.

4.6.1 Error situations catalogue

The list of the most frequent general error messages reported by OPS v.2 is presented below:

Affected services	Error message	Possible occurrence situation
all	The request could not be processed. Please try again later.	A generic message which occurs in all exceptional situations not covered by other messages.
all	The requested reference was ambiguous.	Occurs when many documents are existing for a given document reference. The `detail` element contains all found references. (see 4.6.3)
all	An invalid reference. Please check if the reference is correct for this service.	Occurs when a service is called with an invalid reference. In majority of cases caused by invalid character in one of document-id fields.
all	At least one reference in the request has a wrong value for data-format.	Occurs when some elements are missing in a document reference given in `docdb` format, or there is country and/or kind code element provided or document number element is not provided in `epodoc` format.
all	The server is too busy at the moment. Please try again later.	Occurs when the server lacks of some resources needed for a service usage.
all	There was a problem while accessing the data source. Please try again later.	Occurs when the data source is not available (for example during the data loading time).
all	Past behaviour implies you are a robot. The server is at the moment too busy to serve robots. Please try again later.	Occurs when the server detects that the client violates the fair use rules as described in [EPO-FAIR]
all	The request was invalid.	Occurs when request message is not valid in scope of a requested service. Often indicates invalid numeric values.
all	The request was invalid. Please make sure the request is valid according to the provided schemas.	Occurs when request message is not valid XML document with respect to the underlying schema.
Bibliography Search	A provided query is invalid, ${message}, position ${start}-${end}	Occurs when CQL query is syntactically incorrect, the problem is identified by message and position.
Bibliography Search	Too many terms encountered within the scope ${index};	Occurs when CQL query contains too many terms. The current search limitations are described in

	allowed ${allowedAmount}; encountered ${actualAmount}.	SmartSearch Limitation help topic in esp@cenet portal.
Bibliography Search	Found an invalid term in the query. Scope: ${scope}; term ${invalidTerm}.	Occurs when CQL query contains a terms which is not recognizable. The currently acceptable terms are described in SmartSearch Field Identifiers help topic in esp@cenet portal.
Bibliography Search	The requested result range was invalid. The first element that can be requested is at position 1.	Occurs when begin attribute was 0 in range element.
INPADOC family services	Please request bibliographic data in smaller chunks.	Occurs when INPADOC family is bigger than 250 members and bibliography has been requested
INPADOC services	An application reference in EPODOC format is not supported within this service. Use a publication reference or DocDB format.	Occurs when INPADOC service is called with an application reference in epodoc format.
INPADOC services	A priority reference in EPODOC format is not supported within this service. Use a publication reference or DocDB format.	Occurs when INPADOC service is called with a priority reference in epodoc format.
all except Bibliography Request service	The requested data could not be found.	Occurs when a document reference cannot be resolved, often indicates the document/priority number problem.
Full text services	Unknown or missing value for format or version. Please perform an inquiry to learn about the available formats.	Occurs when a requested format is not supported. In OPS v.2 only one value is acceptable: TEXT_ONLY
Full text services	Country code ${code} restricted for <u>fulltext</u> operation. Valid country codes are: ${codes}.	Occurs when a requested country code is not allowed for full text retrieval. The list of allowed country codes is provided.

Additionally, there are several others, more detailed, exceptional messages which can occur using Bibliography Search service or Number Service.

4.6.2 Document with the status attribute

As it was mentioned almost all services return an error when document is not found, except for Bibliography Request service (see 5.5.2). This service returns a non-error response with the attribute status="not found" as shown below in an example request for an imaginary publication with a number XX123456 in epodoc format.

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="4"/>
  <exchange-documents>
    <exchange-document country="XX" doc-number="123456" kind=""
status="not found" system="ops.epo.org">
      <bibliographic-data>
        <publication-reference data-format="epodoc" status="not found">
          <document-id>
            <country/>
            <doc-number>XX123456</doc-number>
            <kind/>
          </document-id>
        </publication-reference>
        <parties/>
      </bibliographic-data>
    </exchange-document>
  </exchange-documents>
</ops:world-patent-data>
```

The same service returns also status="data-format docdb not supported" when either application or priority reference type is used together with docdb data format:

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="0"/>
  <exchange-documents>
    <exchange-document country="" doc-number="" status="data-format docdb
not supported" system="ops.epo.org">
      <bibliographic-data>
        <publication-reference data-format="epodoc" status="data-format
docdb not supported">
          <document-id>
            <doc-number>data-format docdb not supported</doc-number>
          </document-id>
        </publication-reference>
        <priority-claims>
          <priority-claim data-format="docdb" status="data-format docdb
not supported">
            <document-id>
<country/>
<doc-number>XX123456</doc-number>
<kind/>
            </document-id>
          </priority-claim>
        </priority-claims>
        <parties/>
      </bibliographic-data>
    </exchange-document>
  </exchange-documents>
</ops:world-patent-data>
```

The reason why this service behaves differently lays in its special working mode called "batch processing", described in 5.5.2.1, which allows to request for several documents in a one call. If the server had thrown an error because one document was not found it would have not been possible to retrieve the response for the rest.

4.6.3   Ambiguity message example

```
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Client</faultcode>
```

```
        <faultstring>The requested reference was ambiguous.</faultstring>
        <detail>
          <ambiguous-input>
            <publication-reference data-format="epodoc"
xmlns="http://www.epo.org/exchange" xmlns:ops="http://ops.epo.org">
              <document-id>
                <country/>
                 <doc-number>EP1000000</doc-number>
                 <kind/>
              </document-id>
            </publication-reference>
          </ambiguous-input>
          <resolution>
            <publication-reference data-format="docdb"
xmlns="http://www.epo.org/exchange" xmlns:ops="http://ops.epo.org">
              <document-id>
                <country>EP</country>
                <doc-number>1000000</doc-number>
                <kind>A1</kind>
              </document-id>
            </publication-reference>
          </resolution>
          <resolution>
            <publication-reference data-format="docdb"
xmlns="http://www.epo.org/exchange" xmlns:ops="http://ops.epo.org">
              <document-id>
                <country>EP</country>
                <doc-number>1000000</doc-number>
                <kind>B1</kind>
              </document-id>
            </publication-reference>
          </resolution>
        </detail>
      </SOAP-ENV:Fault>
```

4.6.4   INPADOC family is too big

In situation where the requested family is very large (the sum of family members and their priorities is larger than 1500), all members cannot be returned, due to the technical reasons. The patent-family structure is returned with attribute `truncatedFamily="true"` as shown below in a request for INPADOC family for publication `US2003082717`. All family members have been removed from the example.

```
        <ops:world-patent-data xmlns="http://www.epo.org/exchange"
        xmlns:ops="http://ops.epo.org">
          <ops:meta name="elapsed-time" value="7681"/>
          <ops:patent-family legal="false" truncatedFamily="true">
            <publication-reference data-format="epodoc">
              <document-id>
                <country/>
                <doc-number>US2003082717</doc-number>
                <kind/>
              </document-id>
            </publication-reference>
            <!-- all family members removed -->
          </ops:patent-family>
        </ops:world-patent-data>
```

4.7 Common Query Language

The language itself is formally defined as a list of BNF clauses in [CQL-SPC]. It allows to query within any structured piece of information. Currently, OPS v.2 is using CQL for searching publications which match the given criteria

4.7.1 Index catalogue

Since the information about publication is structured (see the document exchange structure, described in 4.5.1), the following CQL indexes have been designed, with a remark of those which correspond to a single piece of information, meaning, to exactly one element from a document exchange structure.

Index names	Is single?	Description of a related information
`title, ti`	yes	the publication title in English
`abstract, ab`	yes	the publication abstract in English
`titleandabstract, ta`	no	the publication title or abstract in English
`inventor, in`	yes	an inventor name
`applicant, pa`	yes	an applicant name
`inventorandapplicant, ia`	no	a name of an inventor or an applicant
`publicationnumber, pn`	yes	the publication number in any format
`spn`	yes	the publication number in `epodoc` format
`applicantnumber, ap`	yes	the application number in any format
`sap`	yes	the application number in `epodoc` format
`prioritynumber, pr`	yes	the priority number
`spr`	yes	the priority number in `epodoc` format
`num`	no	the publication, application or a priority number in any format
`publicationdate, pd`	yes	the publication date
`citation, ct`	yes	a cited document number during the search
`ex`	yes	a cited document number during the examination
`op`	yes	a cited document number during the opposition
`rf`	yes	a cited document number provided by the applicant
`oc`	yes	an other cited document number
`famn`	yes	a simple family identifier
`ecla, ec`	yes	an ECLA class
`ipc, ic`	no	any IPC1-8 class

ci	yes	IPC8 core invention class
cn	yes	IPC8 core additional class (non-invention)
ai	yes	IPC8 advanced invention class
an	yes	IPC8 advanced additional class (non-invention)
a	no	any IPC8 advanced class
c	no	any IPC8 core class
cl	no	an ECLA or IPC1-8 class
txt	no	the publication title or abstract, or inventor/applicant name

4.7.2 OPS v.2 realization of CQL

The implementation of CQL in OPS v.2 can be summarized in the following statements:

1. a CQL query is either a search clause or multiple search clauses connected by a Boolean operator,

2. a search clause consists of an index, relation and search term or a search term alone, where both might be enclosed within parenthesis

3. if the index is missing then the search clause must not have the relation; it is either both of them or none of them,

4. if the index and a relation is missing then an equality relation is assumed and the index is determined based on the following rules:

   a. if a search term is 2 letter ISO country code, the num index is assumed,

   b. if a search term matches one of the following date formats: yyyy, yyyyMM, yyyyMMdd or dd/MM/yyyy then the pd index is assumed. The 4-digits year (yyyy) is assumed to be within the range 1800-2999, both a month number (MM) and a day number (dd) are having leading zero if necessary,

   c. if a search term matches one of the following patterns: x, xdd, xddw, xddwd, xddwdd, xddwdd/h, the cl index is assumed: x refers to one letter (either upper or lower case) of a classification group within the range a-h or y, d is a digit, w refers to any alphanumeric character and h is a hexadecimal number up to 6 digit long,

   d. if a search term matches \\w{2,4}\\d{1,}[a-zA-Z]?\\d? or is composed of digits only, then the num index is assumed,

   e. if a search term is composed of letters only, then the ia index is assumed,

   f. txt index is assumed,

5. a relation means either:

   a. an equality relation which exact definition varies depend on a kind of a search term and index:

      • when index values fall within the range (like publication date) and search term is a list of exactly 2 words it is assumed that equality relation is within relation,

      • otherwise, equality means that index value and search term are identical,

    b. an order relation for comparable terms: less (<), greater (>), equal (=), less or equal (<=), greater or equal (>=), not equal (<>),

    c. an occurrence relation on search term which is a list of words:

- `within` relation for a terms which are list of exactly 2 words with values that fall within the range, like dates or numerical values, where words are respectfully the beginning and the end of a range,

- `any` or `all` relation for a terms which are list of words, where `any` means any of these words and `all` means all of these words,

6. a search term is either a single word or a list of words enclosed within double quotas assumed to be separated by comma or whitespace if `within` operator is used,

7. a word might contain truncation characters:

    a. unlimited truncation (*) which represents a string of any length including any character,

    b. limited truncation (?) which represents any character or no character,

    c. masking truncation (#) which represents any character which is mandatory present,

    d. it is possible to use truncation at the beginning of a word

8. a word might contain a backslash character (\) used to escape itself (\\), parenthesises "\(" or "\)", hyphens (\-), quotas (\"), commas (\,) or truncation characters.

9. a Boolean operator is one of logical operators: `and`, `or`, `not` or a proximity operator; mathematically, operator `not` is not an unary operator and should be understood like "and not" in a common sense,

10. a proximity operator `prox` is either checking a co-occurrence of two search terms in scope of a defined unit of text (`word`, `sentence` or `paragraph`) or checking the distance between those two search terms with or without the respect of its order. Both terms must be in scope of the same index.

4.7.3 Limitations

There are several restrictions and final comments regarding truncations, number of operators and relations:

- truncation is not allowed on numerical indexes,

- left truncation is allowed only for title and abstract related indexes,

- an unlimited truncation in classification related indexes is allowed only for subclasses,

- at least 3 characters must be mentioned before unlimited truncation (*), unless it is placed at the beginning of a word,

- up to 3 unlimited/limited/masking truncations may be used together in one word,

- only 4 search clauses on one index are allowed,

- only 1 `not` operator is allowed

- when occurrence relation is used (`within`, `all`, `any`) together with a non-single index (like `ti`, `ia` or `txt`) the result might have publications with words found in different elements of a document exchange structure. To avoid this situation a proximity operator with `word` unit modifier should be used instead,

- when search term is 3 letter word and either pn, ap, pr , num index is used then first two letter are treated like country code and the last one as a kind code

- if distance is mentioned in `prox` operator it must use either <= or = relation,

4.7.4 Examples

The following examples illustrates the usage of CQL in OPS v.2.

Please mind, that some special characters must be replaced when placed inside XML document, especially < must be replaced with < and > must become > (both ended with semicolon)

Query	Description of result	Rule
`ti all "green, energy"`	All publications where "green" and "energy" are found among all titles. *Since the ti index checks title in all languages it is possible, but unlikely, that one word will be found in English title and another in German title.*	5c
`ti=green prox/unit=word ti=energy`	All publications where one title contains "green" and "energy" together.	10
`pd within "20051212 20051214"`	All publications published 12th, 13th or 14th December 2005	5c, 6
`pd="20051212 20051214"`	the same like above	5a
`ia any "Smith, John"`	All publications having either John or Smith as a part of the applicant/inventor name	5c
`pn=EP and pr=GB`	All publications published by EPO (EP country code in the publication number) having a priority document from United Kingdom (GB country code in the priority number)	1, 9
`ta=green prox/distance<=3 ta=energy`	All publications having words "green" and "energy" in either title or abstract separated by at most 3 words from each other,	10
`ta=green prox/distance<=2/ordered= true ta=energy`	All publications having a word "green" followed by a word "energy" in either title or abstract separated by at most 2 words,	10
`(ta=green prox/distance<=3 ta=energy) or (ta=renewable prox/distance<=3 ta=energy)`	All publications having either "green" or "renewable" separated by at most 3 words from a word "energy" in either title or abstract, in any order	1, 9, 10
`pa all "federal, bureau, investigation" and US`	All publications with a publication/application/priority document issued by United States (having a US country code in a document number) having "federal", "bureau", "investigation" words in an applicant name (see 4a)	1, 4, 5c, 9
`pa all "federal, bureau, investigation" and US and pd>2000`	like above, published after 2000.	1, 4, 5b, 5c, 9

`pd < 18000101`	All publications published in 18th century.	5b
`ta=synchroni#ed`	Allows to look for "synchronized" or "synchronised" word in abstract/title	7
`EP and 2009 and Smith`	All publications with application/publication/priority document issued by EPO (having EP country code), published in 2009, having Smith as an applicant or inventor	1, 4, 9

5. Open Patent Services catalogue

The following chapter describes 14 web services available in OPS v.2 which have been grouped based on the kind of information they provide.

For the information how to call the service, please refer to 4.2.

5.1 INPADOC family services

There are 4 services related to INPADOC family which provides the list of family members (as a publication reference) with legal status data or/and bibliography if needed.
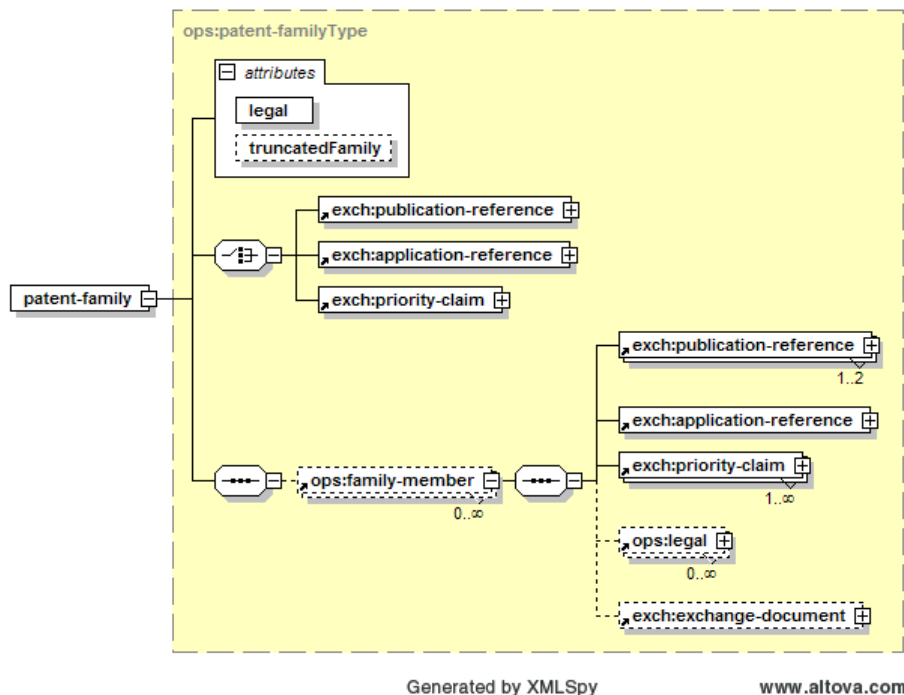

Figure 15 - The patent family structure

All services return a patent-family element but it context slightly differs for each service.

5.1.1 INPADOC Family Request service

Possible requests:	a publication, an application or a priority reference, described in 4.4.1
Possible responses:	**Upon successful completion:** A `patent-family` element which contains a copy of a request reference as a first sub-element and several `family-member` elements, described in 4.5.4. Attribute `legal` of a `patent-family` element is always `false` and there is no `ops:legal` nor `exch:exchange-document` elements in a `family-member` element. When service is called with the `with-biblio` attribute set to `"true"` it provides the `exchange-document` element for each `family-member`, acting like INPADOC Family And Bibliography Request service. **In any exception situation, described in 4.6** `SOAPFault` response, or

	a `patent-family` element with `truncatedFamily="true"` when INPADOC family is too big to be properly calculated
Additional information:	Requests using application or priority number references are currently only supported in docdb format.

5.1.2   INPADOC Family Request With Legal Status Data service

Possible requests:	a publication, an application or a priority reference, described in 4.4.1
Possible responses:	**Upon successful completion:** A `patent-family` element which contains a copy of a request reference as a first sub-element and several `family-member` elements, described in 4.5.4. Attribute `legal` of a `patent-family` element is always `true` and there might be several `ops:legal` elements in a `family-member` element. The element `exchange-document` is not present. When service is called with the `with-biblio` attribute set to `"true"` it provides the `exchange-document` element for each `family-member`, acting like INPADOC Family With Legal Status And Bibliography Request service. **In any exception situation, described in 4.6** `SOAPFault` response, or a `patent-family` element with `truncatedFamily="true"` when INPADOC family is too big to be properly calculated
Additional information:	Requests using application or priority number references are currently only supported in docdb format.

5.1.3   INPADOC Family And Bibliography Request service

Possible requests:	a publication, an application or a priority reference, described in 4.4.1
Possible responses:	**Upon successful completion:** A `patent-family` element which contains a copy of a request reference as a first sub-element and several `family-member` elements, described in 4.5.4. Attribute `legal` of a `patent-family` element is always `true` and there is an `exchange-document` element having all bibliographic information. The element `ops:legal` is not present. **In any exception situation, described in 4.6** `SOAPFault` response, or a `patent-family` element with `truncatedFamily="true"` when INPADOC family is too big to be properly calculated
Additional information:	Requests using application or priority number references are currently only supported in docdb format.

5.1.4 INPADOC Family With Legal Status Data And Bibliography Request service

Possible requests:	a publication, an application or a priority reference, described in 4.4.1
Possible responses:	**Upon successful completion:** The `patent-family` element which contains a copy of a request reference as a first sub-element and several `family-member` elements, described in 4.5.4. Attribute `legal` of a `patent-family` element is always `true` and there is an `exchange-document` element having all bibliographic information. There might be several `ops:legal` elements present. **In any exception situation, described in 4.6** `SOAPFault` response, or a `patent-family` element with `truncatedFamily="true"` when INPADOC family is too big to be properly calculated
Additional information:	Requests using application or priority number references are currently only supported in docdb format.

5.2   **Legal status services**

These 2 services are responding with `patent-family` element having exactly one `family-member`, corresponding to the requested reference.

5.2.1 *Legal Status Request service*

Possible requests:	a publication, an application or a priority reference, described in 4.4.1
Possible responses:	**Upon successful completion:** The `patent-family` element which contains a copy of a request reference as a first sub-element and exactly one `family-member` element (described in 4.5.4) corresponding to the requested reference. Attribute `legal` of a `patent-family` element is always `true` and there might be several `ops:legal` elements in a `family-member` element. The element `exchange-document` is not present. When service is called with the `with-biblio` attribute set to `"true"` it provides the `exchange-document` element for each `family-member`, acting like Legal Status And Bibliography Request service. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	Requests using application or priority number references are currently only supported in docdb format.

5.2.2 *Legal Status And Bibliography Request service*

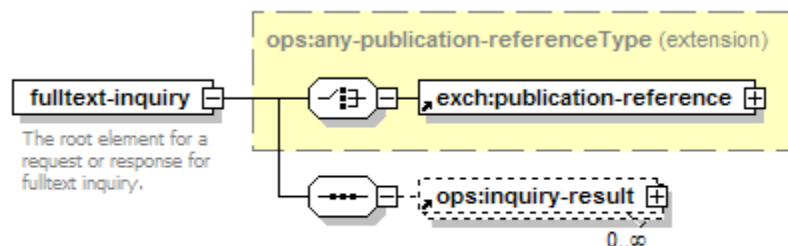| Possible requests:	a publication, an application or a priority reference, described in 4.4.1
Possible	**Upon successful completion:**

responses:	The `patent-family` element which contains a copy of a request reference as a first sub-element and exactly one `family-member` element (described in 4.5.4) corresponding to the requested reference. Attribute `legal` of a `patent-family` element is always `true` and there might be several `ops:legal` elements in a `family-member` element. The element `exchange-document` is present. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	Requests using application or priority number references are currently only supported in docdb format.

5.3 Full text services

In OPS v.2 there are separate services for obtaining claims and description of a publication. Inquiry service is used to learn what kind of formats are available for a requested text of the publication.

All of those services allow to use publication reference only.

5.3.1 Full Text Inquiry Request service

Generated by XMLSpy   www.altova.com

Figure 16 - The Full text inquiry structure

Possible requests:	a publication reference, described in 4.4.1
Possible responses:	**Upon successful completion:** The `fulltext-inquiry` element is provided, having a requested reference and `inquiry-result` element (described in 4.5.6) containing several `fulltext-instance` elements, described in 4.5.9. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	none

5.3.2 Claims Request service

Possible requests:	a publication reference, described in 4.4.1

Possible responses:	**Upon successful completion:** The `fulltext-document` element is provided (described in 4.5.10), enclosed within `fulltext-documents` element having a bibliographic-data element with a requested publication reference and one `claims` element containing all claims. The element `description` is not present. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	In OPS v.2 there is only one `format-version` attribute value allowed: `1.0`. Available formats must be learnt first by calling Full Text Inquiry Request service, described in 5.3.1

5.3.3 Description Request service

Possible requests:	a publication reference, described in 4.4.1
Possible responses:	**Upon successful completion:** A `fulltext-document` element is provided (described in 4.5.10), enclosed within `fulltext-documents` element, having a bibliographic-data element with a requested publication reference and one `description` element containing several text paragraphs in `p` elements. The element `claims` is not present. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	In OPS v.2 there is only one `format-version` attribute value allowed: `1.0` Available formats must be learnt first by calling Full Text Inquiry Request service, described in 5.3.1

5.4 Document services

Currently, there is no way to get a page of a publication document without knowing what is the identifier of a full document instance this page belongs to. Two services have been designed.

Firstly, the document structure must be learnt by calling Document Inquiry Request service. Secondly, the full document instance identifier must be learnt, together with its available data formats. Finally, the Document Retrieval Request service is used to obtain the required page in a selected format.

Only the publication reference is allowed for both services.

5.4.1 Document Inquiry Request service

Generated by XMLSpy   www.altova.com
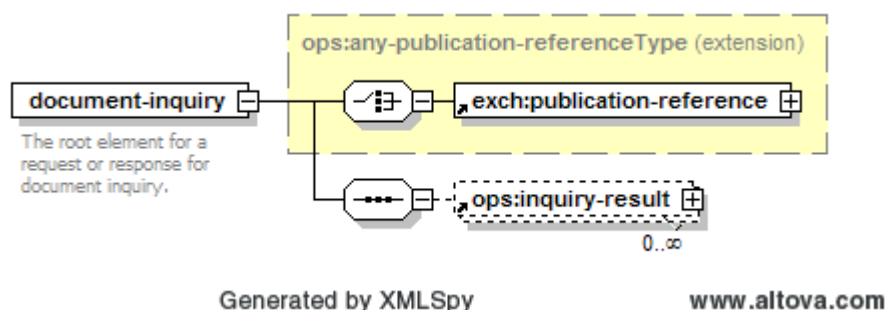
Figure 17 - The document inquiry structure

Possible requests:	a publication reference, described in 4.4.1
Possible responses:	**Upon successful completion:** A `document-inquiry` element is provided, having a requested reference and `inquiry-result` element (described in 4.5.6), which contains several `document-instance` elements, described in 4.5.7. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	none

5.4.1.1 Batch processing

It possible to request for several document-inquiry elements providing several publication references. The response in that case contains all found document instances in the order they were requested.

In OPS v.2 it is possible to request for a maximum of 30 inquiry results in one batch query.

5.4.2 Document Retrieval Request service

The same `document-retrieval` structure is used for a request and a response, which is described in 4.4.2. A page itself is provided as a binary SOAP attachment.

Possible requests:	a `document-retrieval` element, described in 4.4.2
Possible responses:	**Upon successful completion:** The same as requested `document-retrieval` element is provided with binary attachment containing the requested page in a desired format. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	The document instance identifier must be learnt by calling Document Inquiry Request service, described in 5.4.1

5.5 Other services

The following services are related to the publication bibliography data and a simple family.
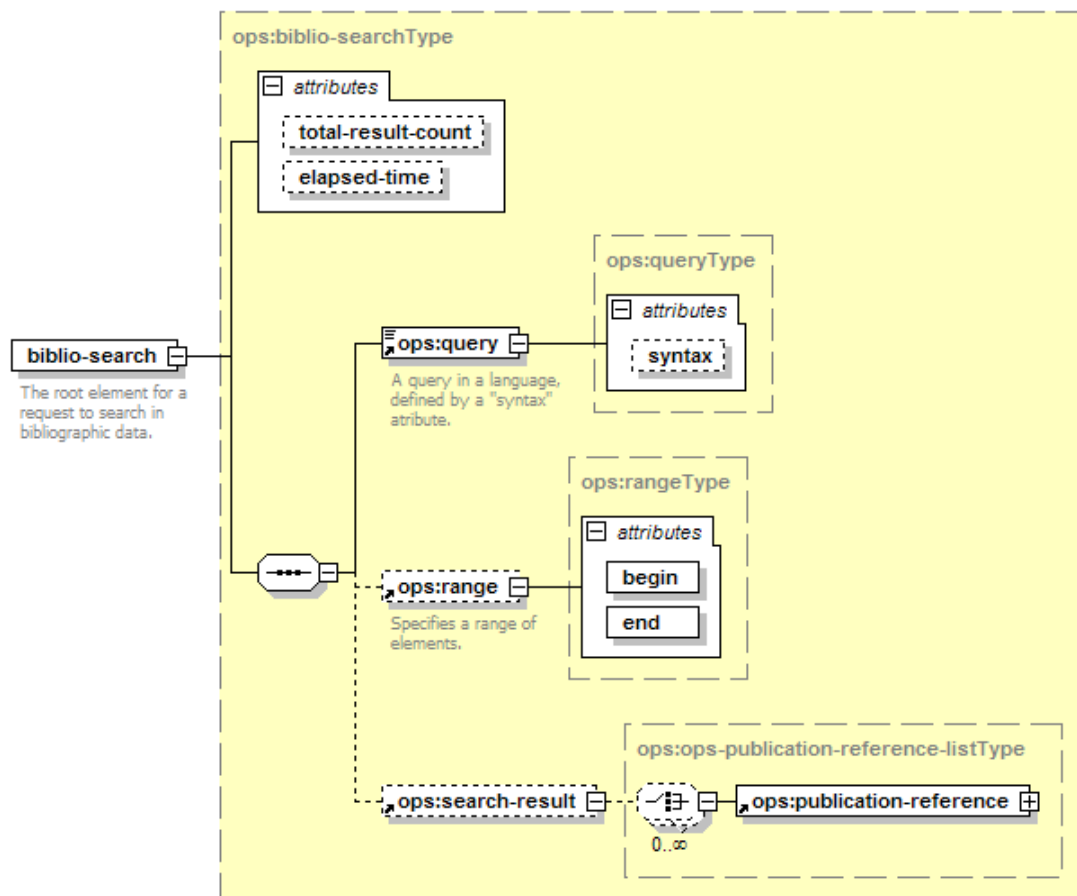
5.5.1 Bibliography Search service

Bibliography Search service uses CQL query to find publications that match provided conditions. All the information related to CQL are provided in 4.7, especially the limitations and examples.

The same structure `biblio-search` is used for request and response, with a slightly different content.

Possible requests:	a `biblio-search` element, described in 4.4.4
Possible responses:	**Upon successful completion:** The same as requested `biblio-search` element is provided having a list of publication references within a requested range grouped in the `search-result` element. The publication reference is provided together with a family identifier in `family-id` attribute. The attribute `syntax` is set to `"CQL"`. The number of all publications is provided in `total-result-count` attribute. When service is called with the `with-biblio` attribute set to `"true"` it provides the `exchange-document` element for each `search-result`, acting like simultaneous call of Bibliography Request service. **In any exception situation, described in 4.6** `SOAPFault` response
Additional information:	A maximum of 2000 results per search can be retrieved by batches up to 40 identifiers.

Figure 18 - The biblio search structure

5.5.2 Bibliography Request service

Possible requests:	SOAP : a publication, an application or a priority references, described in 4.4.1 REST : see below
Possible responses:	**Upon successful completion:** The `exchange-documents` element is provided which contains a list of publication documents found for a given reference. Every publication document is represented by `exchange-document` element, described in 4.5.1. When service is called with the `full-publication-cycle` attribute set to `"true"` all kinds of publication documents are returned, every one in a separate `exchange-document` element. **In any exception situation, described in 4.6** `SOAPFault` response, or the `exchange-documents` element with some `exchange-document` element having attribute `status`, which were not found or problematic.
Additional information:	This service is not responding with a `SOAPFault` response when a requested publication is not found

5.5.2.1 Batch processing

It possible to request for several publications providing several document references. The response in that case contains all found documents in the order they were requested. If some references cannot be found, the corresponding `exchange-documents` element will have an attribute `status`. When some references points to the same publication, there will be only one corresponding `exchange-documents` element provided.

In OPS v.2 it is possible to request for a maximum of 100 documents in one batch query.

5.5.2.2 RESTful interface

Biblio request service is available for RESTful access. Resource can be accessed by the URL format displayed below. RESTful Biblio request service generates response in a "**application/xml**" format.

http://ops.epo.org/rest-services/biblio-retrieval/publication-reference/epodoc/EP/1000000/A/20010101/?fullPublicationCycle=true

URL part	Description
`protocol`	Usually "http" or "https"
`authority`	aka server name. Production authority is ops.epo.org
`prefix`	always `rest-services` in order to distinguish RESTful services from the SOAP services.
`service`	always `biblio-retrieval`
`reference type`	Can be one of "application-reference", "publication-reference" or "priority-claim" depending on the number type is used for conversion.
`CC`	Country code.
`number`	Number in an appropriate input format
`KC`	Kind code. **Optional can be replaced by '*' wildcard.**
`date`	Date in a YYYYMMDD format. **Optional can be replaced by '*' wildcard.**
Query parameters :	
`fullPublicationCycle`	Indicates

5.5.3 Equivalents Inquiry Request service

This service provides the simple family for a publication reference. It responds with the `inquiry-result` elements, described in 4.5.6.

Possible requests:	a publication reference, described in 4.4.1
Possible	**Upon successful completion:**

responses:	The `equivalents-inquiry` element is provided which contains a list of `inquiry-result` elements, each one having a publication reference. When service is called with the `with-biblio` attribute set to `"true"` it provides the `exchange-document` element for each `inquiry-result`, acting like simultaneous call of Bibliography Request service. **In any exception situation, described in 4.6** `SOAPFault` response,
Additional information:	none

5.5.4   Number service

This service provides access to the number conversion facilities. It responds with a `standardization` element, described in a section 4.5.11.

Possible requests:	SOAP: a publication, application or priority reference, described in **Error! Reference source not found.** , in one of the supported formats. The reference must be wrapped into `input` element to distinguish it from the `output` standardized reference. REST: see below
Possible responses:	**Upon successful completion:** The `standardization` element is provided which contains a converted into output format reference wrapper into `output` element. For more details on response format see 4.5.11 chapter. **In any exception situation, described in 4.6** `SOAPFault` response,

Additional information:	To increase level of traceability, Number Service responses contains version of number transformation strategy and in some cases various information messages wrapped into meta elements of the response.
	Example of response meta elements:
	`<ops:world-patent-data>`
	`  <ops:meta name="`**version**`" value="3.00.01"/>`
	`  <ops:meta name="`**info**`" value="[Warning: document date is not defined,`
	`Warning: No Kindcode or generation format defined.]"/>`
	`  <ops:meta name="elapsed-time" value="63"/>`
	`  <ops:standardization inputFormat="original" outputFormat="docdb">`
	`        .......`
	`  </ops:standardization>`
	`</ops:world-patent-data>`

5.5.4.1 RESTful interface

Number service is available for RESTful access. Number Service resource can be accessed by the URL format displayed below. Number service generates response in a "**application/xml**" format.
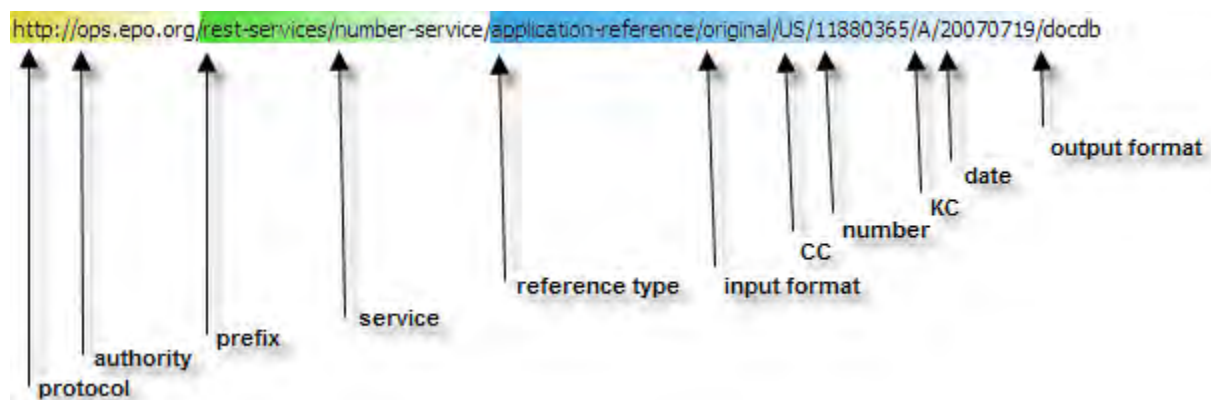
Figure 19 - RESTful standardization request structure

URL part	Description
protocol	Usually "http" or "https"
authority	aka server name. Production authority is ops.epo.org
prefix	always `rest-services` in order to distinguish RESTful services from the SOAP services.
service	always `number-service` for Number Service
reference type	Can be one of "application-reference", "publication-reference" or "priority-claim" depending on the number type is used for conversion.
input format	Input reference format. For possible and supported values see **Error! Reference source not found.**
CC	Country code.
number	Number in an appropriate input format
KC	Kind code. **Optional can be replaced by '*' wildcard.**
date	Date in a YYYYMMDD format. **Optional can be replaced by '*' wildcard.**
output format	Output reference format. For possible and supported values see **Error! Reference source not found.**

Examples of requests:

http://ops.epo.org/rest-services/number-service/application-reference/original/US/11880365/A/20070719/docdb

http://ops.epo.org /rest-services/number-service/application-reference/original/US/11880365/*/20070719/docdb

http://ops.epo.org /rest-services/number-service/application-reference/original/US/11880365/A/*/docdb

http://ops.epo.org /rest-services/number-service/application-reference/original/US/11880365/*/*/docdb

Frequently Asked Questions

This section provides the list of Frequently Asked Questions corresponding to the usage of web services.

5.6 How to retrieve all bibliographic entries for all equivalents of a simple family?

Two services must be used to obtain this goal, and $1 + N/20$ calls are required, where N is the size of a simple family. Firstly, simple family equivalence publication references must be learnt, then the list of publication references must be split into several batches, at most 20 references in each. Finally, for each batch, corresponding bibliographies must be obtained. To get all family equivalences it is mandatory to know at least one publication reference of any equivalence.

The following example describes the necessary steps to retrieve all bibliographies for every equivalent of the family of publication EP1000000:

Step description	Taking all simple family equivalences
Web service name	Equivalents Inquiry Request

<table>
<tr><td colspan="2" align="center">Request</td></tr>
</table>

```
<ops:equivalents-inquiry xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org" >
  <publication-reference data-format="epodoc">
    <document-id>
      <doc-number>EP1000000</doc-number>
    </document-id>
  </publication-reference>
</ops:equivalents-inquiry>
```

<table>
<tr><td colspan="2" align="center">Response</td></tr>
</table>

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="691"/>
  <ops:equivalents-inquiry>
    <publication-reference data-format="epodoc">
      <document-id>
        <country/>
        <doc-number>EP1000000</doc-number>
        <kind/>
      </document-id>
    </publication-reference>
    <ops:inquiry-result>
      <publication-reference data-format="epodoc">
        <document-id>
          <doc-number>EP1000000</doc-number>
        </document-id>
      </publication-reference>
    </ops:inquiry-result>
    <ops:inquiry-result>
      <publication-reference data-format="epodoc">
        <document-id>
          <doc-number>US6093011</doc-number>
        </document-id>
      </publication-reference>
    </ops:inquiry-result>
    <ops:inquiry-result>
      <publication-reference data-format="epodoc">
        <document-id>
          <doc-number>NL1010536C</doc-number>
        </document-id>
      </publication-reference>
    </ops:inquiry-result>
    <ops:inquiry-result>
      <publication-reference data-format="epodoc">
```

```
          <document-id>
            <doc-number>AT232441T</doc-number>
          </document-id>
        </publication-reference>
      </ops:inquiry-result>
   </ops:equivalents-inquiry>
</ops:world-patent-data>
```

Since, the family size is 4, all equivalences bibliographies can be obtained in a one call using batch mode.
Depend on the use case, either publication main members or all kinds of publications might be requested.
The both responses have been cut from irrelevant bibliography details:

Step description	Taking all simple family equivalences
Web service name	Equivalents Inquiry Request

Request

```
<ops:biblio-retrieval full-publication-cycle="false"
xmlns="http://www.epo.org/exchange" xmlns:ops="http://ops.epo.org" >
   <publication-reference data-format="epodoc">
     <document-id>
       <doc-number>EP1000000</doc-number>
     </document-id>
   </publication-reference>
   <publication-reference data-format="epodoc">
     <document-id>
       <doc-number>US6093011</doc-number>
     </document-id>
   </publication-reference>
   <publication-reference data-format="epodoc">
     <document-id>
       <doc-number>NL1010536C</doc-number>
     </document-id>
   </publication-reference>
   <publication-reference data-format="epodoc">
     <document-id>
       <doc-number>AT232441T </doc-number>
     </document-id>
   </publication-reference>
</ops:biblio-retrieval>
```

Response (only publication main members)

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
   <ops:meta name="elapsed-time" value="19"/>
   <exchange-documents>
     <exchange-document country="EP" doc-number="1000000" family-id="19768124" kind="A1"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
     <exchange-document country="US" doc-number="6093011" family-id="19768124" kind="A"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
     <exchange-document country="NL" doc-number="1010536" family-id="19768124" kind="C2"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
     <exchange-document country="AT" doc-number="232441" family- id="19768124" kind="T"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
   </exchange-documents>
</ops:world-patent-data>
```

Response (all kinds of publications)

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
   <ops:meta name="elapsed-time" value="19"/>
   <exchange-documents>
```

```
    <exchange-document country="EP" doc-number="1000000" family-id="19768124" kind="A1"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
    <exchange-document country="EP" doc-number="1000000" family-id="19768124" kind="B1"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
    <exchange-document country="US" doc-number="6093011" family-id="19768124" kind="A"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
    <exchange-document country="NL" doc-number="1010536" family-id="19768124" kind="C2"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
    <exchange-document country="AT" doc-number="232441" family- id="19768124" kind="T"
system="ops.epo.org"/> <!-- bibliography details have been cut -->
  </exchange-documents>
</ops:world-patent-data>
```

5.7   How to retrieve all bibliographic entries for all kinds of the publication?

To complete this task a Bibliography Request service must be called in a full cycle mode (see 5.5.2). As a response a several exchange document elements shall be provided, each one corresponding to one kind of the publication document available.

5.8   How to retrieve page number X from a publication?

Currently, it is not possible to get a page of a publication document without knowing what is the identifier of a full document instance this page belongs to.

Technically, the document structure must be obtained first (by calling Document Inquiry Request service, described in 5.4.1). Then, the full document instance identifier must be obtained, together with its available data formats. Finally, the Document Retrieval Request service must be called, described in 5.4.1.1 to obtain the required page in a selected format.

The following example describes the necessary steps to obtain page number 3 from the publication EP1000000 A1. The response has been cut from irrelevant document instance elements:

Step description	Learning publication structure
Web service name	Document Inquiry Request
Request	

```
<ops:document-inquiry xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org" >
  <publication-reference data-format="docdb">
    <document-id>
      <country>EP</country>
      <doc-number>1000000</doc-number>
      <kind>A1</kind>
    </document-id>
  </publication-reference>
</ops:document-inquiry>
```

Response	

```
<ops:world-patent-data xmlns="http://www.epo.org/exchange"
xmlns:ops="http://ops.epo.org">
  <ops:meta name="elapsed-time" value="29"/>
  <ops:document-inquiry>
    <publication-reference data-format="docdb">
      <document-id>
        <country>EP</country>
        <doc-number>1000000</doc-number>
        <kind>A1</kind>
      </document-id>
    </publication-reference>
    <ops:inquiry-result>
```

```
        <publication-reference data-format="docdb">
          <document-id>
            <country>EP</country>
            <doc-number>1000000</doc-number>
            <kind>A1</kind>
          </document-id>
        </publication-reference>
        <ops:document-instance desc="FullDocument" id="EP       1000000A1 I " number-of-
pages="12" system="ops.epo.org">
          <ops:document-format-options>
            <ops:document-format>SINGLE_PAGE_PDF</ops:document-format>
            <ops:document-format>SINGLE_PAGE_TIFF</ops:document-format>
          </ops:document-format-options>
          <ops:document-section name="ABSTRACT" start-page="1"/>
          <ops:document-section name="BIBLIOGRAPHY" start-page="1"/>
          <ops:document-section name="CLAIMS" start-page="3"/>
          <ops:document-section name="DESCRIPTION" start-page="2"/>
          <ops:document-section name="DRAWINGS" start-page="5"/>
          <ops:document-section name="SEARCH_REPORT" start-page="11"/>
        </ops:document-instance>
        <!-- other instances have been cut -->
      </ops:inquiry-result>
    </ops:document-inquiry>
  </ops:world-patent-data>
```

The full document identifier "EP   1000000A1 I " must be given exactly as it was obtained (with all whitespaces) as an input to the next web service, Document Retrieval Request service, described in 5.4.1.1, together with one of document formats (SINGLE_PAGE_PDF or SINGLE_PAGE_TIFF) and desired page number.

5.9 How to retrieve a publication PDF file?

Currently, there is no way to get a single PDF file corresponding to the publication document due to the technical reasons. By utilizing, both Document Inquiry Request service (see 5.4.1) and Document Retrieval Request service (see 5.4.1.1) it is possible, however, to get every single page of all available publication document instances (full text, drawings, search report, etc.) with 1+M calls, where M is a total number of pages of all document instances.

Firstly, all document instances must be obtained by calling Document Inquiry Request service, secondly, for each document instance, N pages must be obtained from Document retrieval Request service.

5.10 What are the differences between esp@cenet search mechanism and Bibliography Search service?

Technically, esp@cenet SmartSearch mechanism allows to skip Boolean operators between search clauses, while Bibliography Search service requires fully compliant CQL query. Additionally, the web service is faster, since the bibliographic data are not provided in the response, only publication references.

All information related to CQL in OPS v.2 are provided in 4.7, especially the limitations and examples.

5.11 Why some publications cannot be found in OPS v.2 using a reference in EPODOC format if they are available in esp@cenet?

The problem might be related to so-called kind code attachment rules and how those two system process it. There are several countries for which a kind code must be partially attached to the number. Technically, the rule is composed of 3 conditions:

- for Japanease publications (JP) an entire kind code (a letter and possibly a digit) must be attached to the number provided it is not kind code A,

- for the publications having a country code on the following list a kind code letter must be attached to the number, provided it is not kind code A:
AT, AU, BA, BG, CN, DK, FI, IS, KR, NL, NO, PL, SK, UY, TW, YU
- for all publications for which the first and the second rule do not apply, a kind code letter must be attached to the number, provided it is neither kind codes A, B nor C.

The following example shows the difference between `epodoc` and `docdb` number format on a one Japan publication reference with B2 kind code:

```
<publication-reference data-format="epodoc">
  <document-id>
    <doc-number>JP2995292B2</doc-number>
    <date>19991227</date>
  </document-id>
</publication-reference>
<publication-reference data-format="docdb">
  <document-id>
    <country>JP</country>
    <doc-number>2995292</doc-number>
    <kind>B2</kind>
    <date>19991227</date>
  </document-id>
</publication-reference>
```

While esp@cenet might compensate for the missing attached kind code, OPS v.2 is currently not, therefore, it is important to learn that rule.

5.12 Why for some publications when a full cycle attribute is false the entire publication cycle is still returned in Bibliography Request service?

In cases when `epodoc` format is used and the kind code is not provided system treats the given number as a pattern and seeks for all publications matching a given publication number pattern. This is explained in section (5.5.2),

In such countries where additional publication does not change the base number (EP for example) the system shall provide all publications that matching to that base number, effectively, the entire publication cycle shall be provided. To avoid this situation a kind code value should be additionally provided or `docdb` format should be used.

In cases when kind code is unknown the compensation is not possible.

5.13 Why a "published before" type of CQL query responds with documents not having publication date at all?

Currently, when a publication date is unknown, the system treats this date like virtual value"0000-00-00", due to the technical reasons.

A CQL query should be extended with an additional condition (`pd > 00000000`)

5.14 Why there is no publication having "x-ray" in the title found?

Currently, due to technical reasons some queries with words having special characters return no result. The workaround is to treat such word like 2 words and use the proximity operator.

Instead of (`ti=x\-ray`) query, use the(`ti=x prox/distance=0/ordered=true ti=ray`), which looks for title with "x" immediately followed by "ray".