Open Geospatial Consortium Inc.

Date: 2005-11-17

Reference number of this document: OGC 05-042r2

Version: 0.1.0

Category: OpenGIS® Best Practices Paper

Editor: Arliss Whiteside

OpenGIS® web services architecture description

Warning

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OpenGIS® Best Practices Paper
Document subtype: (none)
Document stage: Approved
Document language: English

License Agreement

Contents

Page

Figures

Page

Tables

Page

i. Preface

This document describes many aspects of the OGC web services architecture, which the OGC is currently developing. This architecture is a service-oriented architecture, with all components providing one or more services to other services or to clients. Since that architecture is being developed largely informally, broad descriptions of that architecture have not been previously written. Because that architecture is not yet completed, some aspects are not described here, and other aspects may change in the future.

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

ii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Arliss Whiteside	BAE Systems E&IS

iii. Revision history

Date	Release	Editor	Primary clauses modified	Description
2005-05-23	0.0.0	Arliss Whiteside	All	Initial draft
2005-10-17	0.0.1	Arliss Whiteside	All	Minor editing
2005-11-14	0.1.0	Arliss Whiteside	All	Minor editing as approved paper

iv. Future work

Improvements in this document are desirable to add items to tables and add text to further describe many subjects.

Foreword

This document is related to several previous OGC documents, partially superseding those documents. Among those documents is the OGC Interoperability Program White Paper titled “Introduction to OGC Web Services” [OGC 2001]. Another such document is OGC Discussion Paper OGC 03-025 “OpenGIS Web Services Architecture”.

This document includes three informative annexes.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The OGC shall not be held responsible for identifying any or all such patent rights.

Introduction

This document summarizes the most significant aspects of the Open Geospatial Consortium (OGC) web services architecture, which the OGC is currently developing. This architecture is a service-oriented architecture, with all components providing one or more services to other services or to clients.

OpenGIS® web services architecture description

1 Scope

This document summarizes the most significant aspects of the OGC (Open Geospatial Consortium) web services (OWS) architecture, which the OGC is currently developing. This architecture is a service-oriented architecture, with all components providing one or more services to other services or to clients. Because that architecture is not yet completed, some aspects are not described here, and other aspects may change in the future.

NOTE This document currently contains little information on access control and accounting, because those subjects are just beginning to be explored by the OGC.

This document (currently) just describes this architecture. It does not attempt to provide all the guidance needed by an author or editor of an OGC Web Service specification.

2 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

2.1
client
software component that can invoke an **operation** from a **server**

2.2
coordinate reference system
coordinate system which is related to the real world by a datum [ISO 19111]

2.3
coverage
feature that acts as a **function** to return values from its **range** for any **direct position** within its spatial or **spatiotemporal domain**

EXAMPLE Examples include a **raster** image, polygon overlay, or digital elevation matrix.

2.4
feature
abstraction of real world phenomena [ISO 19101]

2.5
function
rule that associates each element from a **domain** (source, or domain of the function) to a unique element in another domain (target, co-domain, or **range**) [ISO 19107]

2.6
geographic information
information concerning phenomena implicitly or explicitly associated with a location relative to the Earth [ISO 19128]

2.7
interface
named set of operations that characterize the behaviour of an entity [ISO 19119]

2.8
opaque chaining
aggregate service
chained services appear as a single service which handles all coordination of individual services behind the aggregate service [paraphrased from ISO 19119]

NOTE The user has no awareness that there is a set of services behind the aggregate service.

2.9
operation
specification of a transformation or query that an object may be called to execute [ISO 19119]

2.10
parameter
variable whose name and value are included in an operation **request** or **response**

2.11
request
invocation of an **operation** by a **client**

2.12
response
result of an **operation,** returned from a **server** to a **client**

2.13
server
service instance
a particular instance of a **service** [ISO 19119 edited]

2.14
service
distinct part of the functionality that is provided by an entity through interfaces [ISO 19119]

capability which a service provider entity makes available to a service user entity at the interface between those entities [ISO 19104 terms repository]

2.15
service chain
sequence of **services** where, for each adjacent pair of **services**, occurrence of the first action is necessary for the occurrence of the second action [ISO 19119]

2.16
service metadata
metadata describing the **operations** and **geographic information** available at a **server** [ISO 19128]

NOTE Most of this service metadata is specific to one server implementing a service type.

2.17
translucent chaining
workflow-managed chaining
execution of the chain is managed by a workflow service (or multiple workflow services) [paraphrased from ISO 19119]

NOTE The user's involvement in the steps of the chain is mostly watching the chain execute the individual services that are visible to the user. The defined chain exists prior to the user executing the pattern.

2.18
transparent chaining
user defined chaining
user defines and controls the order of execution of the individual services [paraphrased from ISO 19119]

NOTE Details of the services are not hidden from the user.

3 Abbreviated terms

BPEL	Business Process Execution Language
COTS	Commercial Off The Shelf
CRS	Coordinate Reference System
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
KVP	Keyword Value Pair
MIME	Multipurpose Internet Mail Extensions
OGC	Open Geospatial Consortium
OWS	OGC Web Service, or Open Web Service
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TBD	To Be Determined
TBR	To Be Reviewed
UDDI	Universal Description, Discovery and Integration Service

UML	Unified Modeling Language
URI	Universal Resource Identifier
URL	Uniform Resource Locator
URN	Universal Resource Name
WCS	Web Coverage Service
WCTS	Web Coordinate Transformation Service
WFS	Web Feature Service
WICS	Web Image Classification Service
WMS	Web Map Service
WSDL	Web Services Description Language
WTS	Web Terrain Service
XML	Extensible Markup Language

4 OGC web services architecture overview

This Service Oriented Architecture is based on the fundamental roles of service provider and service consumer within a distributed computing system. This pattern emphasizes that desired computing can be realized by combining multiple services, for each of which only the service types (e.g., interfaces and abilities) and server data holdings (e.g. content) need be known. It focuses component definition on providing and/or consuming a defined service. A Service Oriented Architecture also focuses on interactions between components implementing defined services, in the form of service requests, service responses, and service exceptions.

The significant properties of the OGC web services (OWS) architecture, which the OGC is currently developing, include:

a) Service components are organized into multiple tiers.

    1) All components provide services, to clients and/or other components, and each component is usually called a service (with multiple implementations) or a server (each implementation).

    2) Services (or components) are loosely arranged in four tiers, from Clients to Application Services to Processing Services to Information Management Services, but un-needed tiers can be bypassed.

    3) Services can use other services within the same tier, and this is common in the Processing Services tier.

    4) Each tier of services has a general purpose, which is independent of geographic data and services.

    5) Each tier of services includes multiple specific types of services, many of which are tailored to geographic data and services.

6) Servers can operate on (tightly bound) data stored in that server and/or on (loosely bound) data retrieved from another server.

b) Services use is often chained.

1) Services can be chained with other services and often are chained, either transparently (defined and controlled by the client), translucently (predefined but visible to the client), and opaquely (predefined and not visible to client), see Subclause 7.3.5 of [OGC 02-006 and ISO 19119]

2) Services are defined to support defining and executing chains of services.

3) Some service interfaces support server storage of operation results until requested by the next service in a chain.

c) Service interfaces use open standards and are relatively simple.

1) All services support open standard interfaces from their clients, often OGC-specified service interfaces.

2) OGC web service interfaces are coarse-grained, providing only a few static operations per service.

3) Service operations are normally stateless, not requiring servers to retain interface state between operations.

4) One server can implement multiple service interfaces whenever useful.

5) Service interfaces share common parts whenever practical.

6) Service interfaces can have multiple specified levels of functional compliance, and multiple specialized subset and/or superset profiles.

7) Standard XML-based data encoding languages are specified for use in data transfers.

8) Geographic data and service concepts are closely based on the ISO 191XX series of standards.

9) Standard specifications are used for defining and referencing well-known coordinate reference systems (CRSs).

d) Services communication uses open Internet standards.

1) Communication between components uses standard World Wide Web (WWW) protocols, namely HTTP GET, HTTP POST, and SOAP.

2) Specific server operations are addressed using Uniform Resource Locators (URLs).

3) Multipurpose Internet Mail Extensions (MIME) types are used to identify data transfer formats.

4) Data transferred is often encoded using the Extensible Markup Language (XML), with the contents and format specified using XML Schemas.

e) Server and client implementations are not constrained.

1) Services are implemented by software executing on general purpose computers connected to the Internet. The architecture is hardware and software vendor neutral.

2) The same and cooperating services can be implemented by servers that are owned and operated by independent organizations.

3) Many services are implemented by standards-based Commercial Off The Shelf (COTS) software.

4) All services are self-describing, supporting dynamic (just-in-time) connection binding of services supporting publish-find-bind.

5 Service tiers

5.1 Overview

Except for clients, all OWS architecture components provide services, to clients and/or to other components. Each such component is usually called a service when multiple implementations are expected, and each implementation is called a server (or service instance). These components are thus usually called services or servers in this document.

Clients are software packages that provide access to a human user, or operate as agents on behalf of other software. Software that provides access to a human user can be thin (e.g., a web browser), thick (a large application), or “chubby” (in between).

All services (or components) are loosely organized in four tiers, as shown in Figure 1. This organization is loose in that clients and services can bypass un-needed tiers, as indicated by some arrows. Services can use other services within the same tier, and this is common especially in the Processing Services tier. Also, some services perform functions of more than one tier, when those functions are often used together and combined implementation is more efficient. Assignment of such combined services to tiers is somewhat arbitrary.

NOTE Complete separation of services into tiers is not required and is rare, especially when separation would be inefficient.

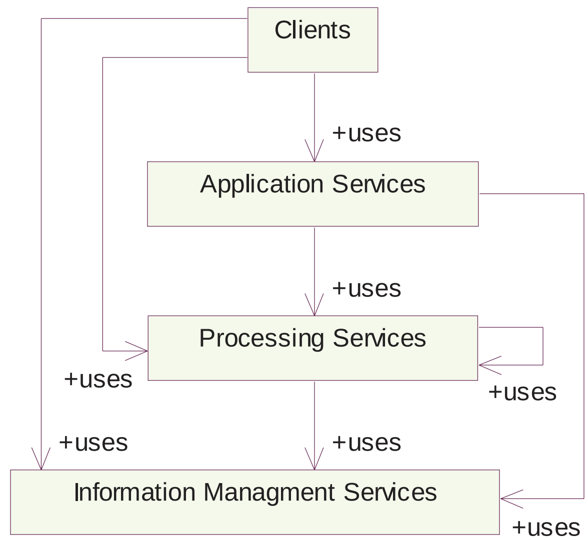

Figure 1 —Service tiers in OWS architecture

This OWS architecture is designed for use where data is important and often voluminous. Servers can operate on (tightly bound) data stored in that server and/or on (loosely bound) data retrieved from another server. Most data is stored by the servers in the Information Management Services tier, but some data (can be and often) is stored in other services and servers.

Each tier of services has a general purpose, as indicated by the names in Figure 1. That tier name is independent of geographic data and services, since some tier services are not specific to geographic data or services. Each tier of services includes multiple specific types of services, many of which are tailored to geographic data and services. Some of the services included in each tier are discussed in the following subclauses.

5.2 Information Management Services tier

The Information Management Services tier contains services designed to store and provide access to data, with each server normally handling multiple separate datasets. In addition, metadata describing multiple datasets can be stored and searched. Access is usually to retrieve a client-specified subset of a stored dataset, or to retrieve selected metadata for all datasets whose metadata meets client-specified query constraints.

The services in the Information Management tier are used by clients and by services in the Application Services and Processing Services tiers. These services can use other services in the Information Management Services tier. The services included in this tier include (but are not limited to) the services listed in Table 1.

Table 1 — Some specific Information Management Services

Service name [a]	Service description
Web Map Service (WMS) [b]	Dynamically produces spatially referenced map of client-specified ground rectangle from one or more client-selected geographic datasets, returning pre-defined pictorial renderings of maps in an image or graphics format
Web Feature Service (WFS)	Retrieves features and feature collections stored that meet client-specified selection criteria
Web Coverage Service (WCS)	Retrieves client-specified subset of client-specified coverage (or image) dataset
Catalog Service for the WEB (CSW) [c]	Retrieves object metadata stored that meets client-specified query criteria
Gazetteer Service	Retrieves location geometries for client-specified geographic names
Universal Description, Discovery and Integration (UDDI) Service	Allows a client to find a web-based service
Standing order services	Allows a user to request that data over a geographic area be disseminated when it becomes available, including reformat, compress, decompress, prioritize, and transmit information requested through standing queries or profiles
Order handling services	Allows clients to order products from a provider, including: selection of geographic processing options, obtaining quotes on orders, submission of order, statusing of orders, billing, and accounting

a Names ending in "Service" are currently specified specific services. Names ending in "services" are types of services that are not yet specified.

b Can store and access both feature and image (coverage) data.

c Many specific profiles of the CSW are expected to be specified and implemented, for metadata for many different types of datasets, and for also storing and accessing small whole datasets.

The services in this Information Management Services tier usually include some processing of the data retrieved. For example, WMS, WCS, and WFS can perform coordinate transformation and format conversion.

One Catalog Service for the WEB (CSW) server can store metadata (and perhaps also datasets) for one or more types of datasets, including (but are not limited to) the types listed in Table 2.

Table 2 — Some specific types of datasets

Type name	Type description
Service	Definition of a service or server
Feature	Geographic features and feature collections (including composite features)
Coverage (including image)	Geographic coverage or image, can be a gridded coverage or another type of coverage, can be georectified or not
Styled Layer Descriptor (SLD) document	Specifies client-controlled styling for map portrayal of features and images (coverages)
Map Symbol	Defines map display symbols
Web Map Context document	Specifies a composite, symbolized map view that can be saved, restored, and transmitted to other viewers, using WMS only
OWS Map Context document	Specifies a composite, symbolized map view that can be saved, restored, and transmitted to other viewers, using WFS, WCS, WMS, etc.
Query template	Template for OGC Common Catalog Query Language queries
Filter Encoding (FE) template	Template for queries for features or other data meeting specified constraints
URN Definitions XML document	Encoding in XML of definitions of URNs defined by OGC or other organization
GML Application Schema	Geography Markup Language (GML) application schema and/or profile for specific application
OWS XML Schema	XML Schema used by an OWS or other service
General XML Schema	Any XML Schema
UML model	Any UML model
Web Service Description Language (WSDL) document	Specifies web service interface
Business Process Execution Language (BPEL) document	Specifies process sequences for specific purpose
Semantics document	Defines semantics (e.g., ontologies coming out of GSW IE)
Annotation document	Specifies annotation of coverage (or image) (e.g., coming out of GMLJP2 IE?)
Accounting record	Records usage of servers and other resources, for billing and other purposes

5.3 Processing Services tier

The Processing Services tier contains services designed to process data, sometimes both feature and image (coverage) data. The services in the Processing Services tier are used by clients and by services in the Application Services tier. These services can use other services in the Processing Services and Information Management Services tiers. The services included in this tier include (but are not limited to) the services listed in Table 3.

Table 3 — Some specific Processing Services

Service name [a]	Service description
SLD Web Map Service (WMS) [b]	Dynamically produces spatially referenced maps from geographic feature and/or coverage data, returning client-specified pictorial renderings of maps in an image format (not actual feature data or coverage data)
Web Terrain Service (WTS) [b]	Dynamically produces client-specified perspective views from geographic feature and/or coverage data, returning client-specified pictorial renderings of data in an image or graphics format
Web 3D Service (W3DS)	Dynamically produces client-specified perspective views from geographic feature data, returning perspectives of feature data in a graphical format
Web Coordinate Transformation Service (WCTS) [b]	Transforms the coordinates of feature or coverage data from one coordinate reference system (CRS) to another, including “transformations”, “conversions”, rectification, and orthorectification
Web Image Classification Service (WICS)	Performs classification of digital images, using client-selected supervised or unsupervised image classification method
Feature Portrayal Service (FPS)	Dynamically produces client-specified pictorial renderings in an image or graphics format of features and feature collections usually dynamically retrieved from a Web Feature Server (WFS)
Coverage Portrayal Service (CPS)	Dynamically produces client-specified pictorial renderings in an image or graphics format of a coverage subset dynamically retrieved from a Web Coverage Service (WCS)
Geoparser Service	Service to scan text documents for location-based references, such as a place names, addresses, postal codes, etc., for passage to a geocoding service.
Geocoder Service	Service to augment location-based text references with position coordinates
Geolinking Service (GLS) [b]	Service that links geospatial data
Geolinked Data Access Service (GDAS)	Service that uses linked geospatial data
Geographic data extraction services	Services supporting extraction of feature and terrain information from images
Dimension measurement services	Services that compute dimensions of objects visible in an image or other geospatial data
Route determination services	Determine optimal path between two specified points based on input parameters and properties contained in a Feature Collection; may also determine distance between points and/or time to follow path
Proximity analysis services	Given a position or geographic feature, finds all objects with a specified set of properties that are located within a user-specified distance of the position or feature.
Change detection services	Services to find differences between two data sets that represent the same geographical area at different times
Data alignment services	Service that adjusts sensor geometry models to improve the match of a coverage (or image) with other coverages and/or known ground positions
Feature generalization services	Service that reduces spatial variation in a feature collection to counteract the undesirable effects of scale reduction

Service name [a]	Service description
Coverage generalization services	Service that reduces spatial variation in a coverage to counteract the undesirable effects of scale reduction
Format conversion services	Service that converts data from one format to another, including data compression and decompression
Semantic translation services [a]	Service that converts data from one set of semantics to another
a Names ending in "Service" are currently specified specific services. Names ending in "services" are types of services that are not yet specified. b Can process both feature and image (coverage) data.	

5.4 Application Services tier

The Application Services tier contains services designed to support Clients, especially thin client software such as web browsers. That is, these Application Services are designed for use by clients instead of each client directly performing these often-needed support functions. The services in the Application Services tier are used by Clients, and can use other services in the Application Services, Processing Services, and Information Management Services tiers. The services included in this tier include (but are not limited to) the services listed in Table 4.

Table 4 — Some specific Application Services

Service name [a]	Service description
Web portal services	Services that allow a user to interact with multiple application services for different data types and purposes
WMS application services	Services that allow a user to interact with a Web Map Service (WMS) to find, style, and get data of interest
Gazetteer application services	Services that allow a user to interact with a Gazetteer service
Geographic data discovery services	Services that allow a user to locate and browse metadata about geographic data, interacting with a catalog
Geographic data extraction services	Services that allow a user to extract and edit feature data, interacting with images and feature data
Geographic data management services	Services that allow a user to manage geospatial data input and retirement, interacting with Information Management Services
Map style management services	Services that allow a user to create, edit, and manage map styles, interacting with Information Management Services
Map symbol management services	Services that allow a user to create, edit, and manage map symbols, interacting with Information Management Services
Data structure management services	Services that allow a user to create, edit, and manage data structures, interacting with Information Management Services
Feature generalization application services	Services that allow a user to interact with Processing services for feature generalization
Coverage general-ization services	Services that allow a user to interact with Processing services for coverage generalization
Other application services	Services that allow a user to interact with other Processing and Information Management services
Chain definition services	Services to define a service chain and enable it to be executed by the workflow enactment service; may also provide a chain validation service
Workflow enactment services	Services to interpret chain definitions and control instantiation of servers and sequencing of activities, maintaining internal state information associated with various services being executed
Access control services	Services that control access to other servers, for privacy, intellectual property, and other reasons
Usage accounting services	Services that keep track of the usage of other servers, for billing and other purposes
a Names ending in "Service" are currently specified specific services. Names ending in "services" are types of services that are not yet specified.	

6 Services chaining

In many cases, multiple services must be used together to perform a useful function. The OWS architecture thus supports "chaining" together of multiple servers, and such chaining is often used. This chaining is not limited to a linear chain; a network of services can also be "chained". Within such a chain, most servers input the data that is output from the previous server in the chain. Services can be chained transparently (defined and controlled by the client), translucently (predefined but visible to the client), and opaquely (predefined and not visible to client), see Subclause 7.3.5 of [OGC 02-006 and ISO 19119].

To facilitate service chaining, some services are defined to support defining and executing chains of services. Also, some Processing Service interfaces are designed to support retrieving the data to be processed from another service, which can be an Information Management Service or another Processing Service. To allow more efficient execution of server chains, some service interfaces support server storage of operation results until requested by next service in a chain.

7 Service interfaces

The basic components of a service oriented architecture are services. There are many definitions of “service”, from the abstract to the concrete and practical. One definition is “useful/meaningful combination of operational interfaces and accessible content”. This definition adds “content” to the ISO/OGC definitions:

a) Service: distinct part of the functionality that is provided by an entity through interfaces.

b) Interface: named set of operations that characterize the behavior of an entity.

c) Operation: specification of a transformation or query that an object may be called to execute. It has a name and lists of input and output parameters.

OGC web service interfaces use open standards and are relatively simple. All services support open standard interfaces from their clients, often OGC-specified service interfaces. In addition to being well-specified and interoperable tested, the OGC-specified service interfaces are coarse-grained, providing only a few static operations per service. For many services, only three service operations are specified.

NOTE 1 OGC web service interfaces are not fine-grained object-oriented, providing tens of operations per service to be implemented and exercised, with some interface objects being dynamically created and destroyed.

In most cases, one service listed in Clause 5 includes just one interface. However, it is possible for one service to include more than one identified interface, normally limited to a few such interfaces. Also, one server can implement multiple interfaces whenever useful, without those interfaces being combined in a specified service.

NOTE 2 One service interface is not required to support all the abilities of one server. For example, some WCS, WMS, and WFS data servers are expected to also implement CSW service interfaces cataloging the datasets available from that server.

The OGC web service interfaces are usually stateless, so session information is not passed between a client and server. Clients (or Application Services) retain any needed interface state between operations.

NOTE 3 Client-service sessions are not used, not requiring servers to retain interface state between operations. This also simplifies use in a dynamic network, where a server can stop operation or fail.

The OGC web service interfaces share common parts whenever practical, allowing those parts to be specified and implemented only once. For example, all OWSs have a mandatory GetCapabilities operation to retrieve server metadata. That server metadata includes four required sections, with the contents and format of three sections common to all services, and part of the fourth section common to most services. In addition, many service interfaces have multiple specified levels of functional compliance, or multiple specialized subset and/or superset profiles.

NOTE 4 The interface parts are NOT separately and independently specified and developed. When more parts are common among interfaces, less work is needed to implement services and define new ones. In addition to the GetCapabilities operation, all OWSs have one mandatory operation to get a data subset, and most have one optional operation to get the description of a dataset or object. Interface compliance levels and profiles inherently share interface parts wherever practical.

Standard XML-based data encoding formats and languages are used in many server-to-client and client-to-server data transfers. The formats and languages specified include (but are not limited to) those listed in Table 5. In these formats and languages and elsewhere, the geographic data and service concepts are closely based on the ISO 191XX series of standards.

NOTE 5 The ISO 191XX data and service concepts were developed by international groups of experts, so are carefully formulated and are being widely used. The concepts or semantics used by different services are not independently developed and specified.

Table 5 — Some standardized encoding formats and languages

Specification name	Description
Filter Encoding (FE)	Encodes queries for features or other data meeting specified constraints
OGC Common Catalog Query Language	Encodes catalog queries for objects meeting specified constraints
Styled Layer Descriptor (SLD)	Encodes client-controlled styling for map portrayal of features and coverages (images)
Geography Markup Language (GML)	Language defined using XML Schemas based on the ISO 191XX series of standards, to be used to specify application-specific XML Schemas
Coordinate Reference Systems (part of GML)	Encodes definitions of coordinate reference systems, coordinate systems, datums, and coordinate transformations (and conversions)
Web Map Context	Encodes WMS application display context
OWS Context	Encodes multiple OWS application display context
URNs using ogc URN namespace	Standardized Universal Resource Identifiers (URNs) referencing most well-known coordinate reference systems (CRSs) and grid CRSs
Sensor Model Language (SensorML)	Encodes descriptions of remote and in-situ sensors, including imaging and environmental sensors
Web Service Description Language (WSDL)	Encodes web service interfaces
Business Process Execution Language (BPEL)	Encodes process sequences for specific purposes

8 Services communication

8.1 Overview

Communication between clients and services, and between services, uses only open non-proprietary Internet standards. That is, the OWS architecture uses the Internet or equivalent as its distributed computing platform (DCP). More specifically, communication between components uses standard World Wide Web (WWW) protocols, namely HTTP GET, HTTP POST, and Simple Object Access Protocol (SOAP). Specific operations of specific servers are addressed using Uniform Resource Locators (URLs). Multipurpose Internet Mail Extensions (MIME) types are used to identify data transfer formats. The data transferred is often encoded using the Extensible Markup Language (XML), with the contents and format carefully specified using XML Schemas.

NOTE Communication between components does not use CORBA, DCOM, .NET, or SQL. However, those protocols can be used within the implementation of a server.

8.2 HTTP GET operation requests

In many cases, a request to perform an operation by a service is transferred as a Hypertext Transfer Protocol (HTTP) GET message. That GET message is addressed to a HTTP Uniform Resource Locator (URL), where that URL locates a specific operation of a specific server. A URL for a HTTP GET request is in fact only a URL prefix, to which additional parameters are appended to construct a valid operation request. The prefix defines the network address to which operation request messages are sent, and may also identify a configuration of that server.

A query is appended to the URL prefix to form a complete request message. Each OWS operation request has mandatory and usually optional request parameters. Each parameter has a defined name, and has multiple allowed values. To formulate the query part of the URL, the mandatory request parameters, and any desired optional parameters, are appended as name/value pairs in the form "name=value&" (parameter name, equals sign, parameter value, ampersand). In the OGC, this parameter encoding is often referred to as keyword value pair (KVP) encoding.

8.3 HTTP POST operation requests

Less frequently, a request to perform an operation by a service is transferred as a Hypertext Transfer Protocol (HTTP) POST message. That POST message is addressed to a (possibly different) HTTP Uniform Resource Locator (URL), where that URL locates a specific operation of a specific server. A URL for a HTTP POST request is a complete URL (not merely a prefix as in the HTTP GET case).

Clients transmit request parameters to the URL in the body of the HTTP POST message. An OWS does not require additional parameters to be appended to the URL in order to construct a valid target for the operation request. When HTTP POST is used, the operation request message is normally encoded as an XML document, formatted as specified by one or more XML Schemas. The operation request message can alternately be KVP encoded, in the body of the HTTP POST message.

8.4 HTTP operation responses

After receiving an operation request, a server replies with a response message corresponding exactly to the request, or sends an exception report if unable to respond correctly. Responses to operation requests are the same whether the request is transferred by HTTP GET or POST. In most cases, the operation response is encoded in XML, using XML Schemas to specify the correct response contents and format. These statements apply to both normal and exception operation responses, which are separately specified for each OWS service.

All XML Schemas used contain documentation of the meaning of each specified element, attribute, and type. All of these documentation elements are considered normative, unless labeled “informative”. Almost all of the concrete XML elements defined in these OWS Schemas can be used without separate XML Schemas, whenever no content extensions or restrictions are needed. An additional XML Schema is used whenever element contents extension is required, and should be used in some other cases to specify needed restrictions.

NOTE XML documents are not required to be validated against their respective schemas for normal service use. However, XML parsing and validation should follow normal XML syntax rules, for example regarding XML Namespaces.

A server may send an HTTP Redirect message (using HTTP response codes as defined in [IETF RFC 2616]) to an absolute URL that is different from the valid request URL that was sent by the client. HTTP Redirect causes the client to issue a new HTTP request for the new URL. Several redirects could in theory occur. Practically speaking, the redirect sequence ends when the server responds with an operation response. The final response shall be an OWS operation response that corresponds exactly to the original operation request, or an exception report.

8.5 MIME types use

Response messages are accompanied by the appropriate Multipurpose Internet Mail Extensions (MIME) type for that message. A list of MIME types in common use on the internet is maintained by the Internet Assigned Numbers Authority [IANA]. Servers can support parameterized MIME types, and this is common to more completely identify the specific format. In addition to parameterized MIME types, servers usually offer the basic un-parameterized version of the format, for clients that do no understand the parameterized MIME type.

Response messages are accompanied by other HTTP entity headers as appropriate. In particular, the Expires and Last-Modified headers provide important information for data caching. Content-Length may be used by clients to know when data transmission is complete and to efficiently allocate space for results, and Content-Encoding or Content-Transfer-Encoding may be necessary for proper interpretation of the results. When returning a large XML document, some form of data compression should be supported; since client-server communication transfer speeds will be considerably faster if the document is compressed.

8.6 SOAP operation requests and responses

In some cases, a request to perform a specific operation by a specific server can be transferred in a Simple Object Access Protocol (SOAP) operation request message. That SOAP request message is addressed to a HTTP Uniform Resource Locator (URL), where that URL locates a specific operation of a specific server (TBR). In this case, the operation request parameters are encoded in XML just like they can be for HTTP POST. When SOAP is used, the response from that operation is transferred in a SOAP operation reply message, and is XML encoded just like responses to HTTP POST requests.

9 Server implementation

Servers and client implementations are not constrained except for supporting the specified service interfaces. Each can be implemented by software executing on any general purpose computer connected to the Internet or equivalent. The architecture is hardware and software vendor neutral. The same and cooperating services can be implemented by servers that are owned and operated by independent organizations.

NOTE Cooperating servers and clients need not be owned and operated by one organization or by formally cooperating organizations.

All OWS services and clients are implemented by available standards-based Commercial Off The Shelf (COTS) software. This commercial software can sometimes be used without requiring major software development, or can be adapted to specific needs with limited software development.

All services are self-describing, supporting dynamic (just-in-time) connection binding of services supporting publish-find-bind. That is, each server is able to “publish itself”, by providing a single authoritative source of its own server description upon request.

Annex A
(informative)

Service trading (publish – find –bind)

Service trading supports the offering and discovery of interfaces which provide services of particular types. A trader implementation records service offers and matches requests for advertised services. Publishing a capability or offering a service is called “export”. Matching a service request against published offers or discovering services is called “import”. This can also be described in an equivalent manner as the “Publish – Find – Bind” (PFB) pattern of service interaction. The fundamental roles are:

a) Trader (Registry) - registers service offers from exporter objects and returns service offers to importer objects upon request according to some criteria.

b) Exporter (Service) - registers service offers with the trader object

c) Importer (Client) - obtains service offers, satisfying some criteria, from the trader object.

NOTE In the OWS architecture, a Registry can be implemented using the Catalog Service for the WEB (CSW) service interface.

The ODP Trading function is elaborated in document [ISO/IEC 13235-1] and refined somewhat in the OMG Trading specification [TBD], which is technically aligned with the computational view of the ODP trading function. Most importantly, a trader supports dynamic (i.e. run-time) binding between service providers and requesters, since sites and applications are frequently changing in large distributed systems. The fundamental roles and interactions are depicted in Figure 2.[1] The equivalent PFB terminology is shown as well (*blue italic text*). A trader registers service offers from exporter objects and returns service offers upon request to importer objects according to some criteria.

[1] Many readers will recognize Figure 2: many of the recent web services white papers include similar diagrams that map onto it directly. In many cases ‘find’, ‘bind’, and ‘publish’ substitute for ‘import’, ‘service interaction’, and ‘export’, respectively.

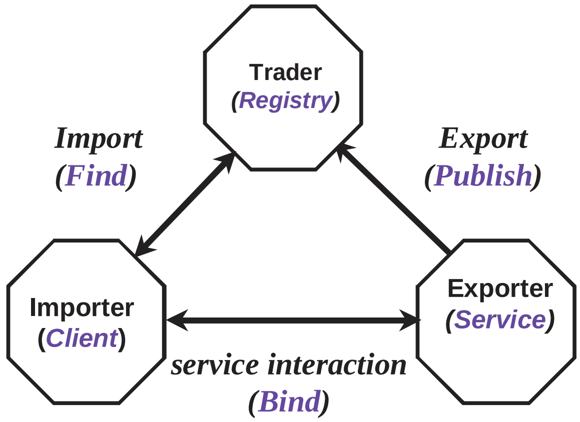

Figure 2 — Service trading objects

A Trader plays the role of “matchmaker” in a service-based architecture. The interaction pattern is:

a) To publish a service offer, an Exporter gives a Trader a description of a server, including a description of the interface at which that service instance is available.

b) To find suitable server offers, an Importer asks a Trader for a server having certain characteristics. The trader checks the previously registered descriptions of servers and responds to the importer with the information required to bind with a service instance. Preferences may be applied to the set of offers matched according to service type, constraint expressions, and various policies. Use of the preferences can determine the order used to return matched offers to the importer.

c) To bind a service, an Importer applies information received from the Trader to bind to a server. The Client then proceeds to use that server.

Annex B
(informative)

Architecture tiers relationship to ISO 19119

The four tiers of the OWS architecture are loosely related to the geographic services categories specified in Subclause 8.3 of ISO 19119 Geographic information — Services (copied in OGC Abstract Specification Topic 12). This relationship is:

a) The Application Services and the Clients together provide the “human interaction services” described in Subclause 8.3.2.

b) The Processing Services provide the “processing services” described in Subclause 8.3.5. It would be possible to divide these services into the “spatial”, “thematic”, “temporal” and “metadata” categories described in Subclause 8.3.5.

c) The Information Management Services provide the “model/information management services” described in Subclause 8.3.3.

In this description of the OWS architecture, the storage of datasets is not separated from catalogues or registries that store and search metadata, because they are not separated in ISO 19119. Also, these services are not separated because that distinction is frequently fuzzy. Furthermore, one server can implement a dataset storage and retrieval interface plus a catalogue interface for storing and searching the metadata for those datasets.

Some services perform functions of more than one tier, when those functions are usually used together and combined implementation is more efficient. Assignment of such combined services to tiers is somewhat arbitrary. For example, all services in the Information Management Services tier can include some processing of the data retrieved. Specifically, WMS, WCS, and WFS can perform coordinate transformation and format conversion.

Annex C
(informative)

OWS architecture relationship to RM-ODP

This architecture description is organized for ease-of-understanding, not according to the five viewpoints specified in the Reference Model for Open Distributed Processing (RM-ODP) [ISO 10746]. The concept of distributing computing functions across a network in a dynamic fashion has been addressed by this RM-ODP, which is a conceptual framework for such systems, and implementation patterns or rules. Five standard viewpoints are defined; these viewpoints address different aspects of the system and enable the 'separation of concerns':

a) Enterprise viewpoint: articulates a "business model" that should be understandable by all stakeholders; focuses on purpose, operational objectives, policies, enterprise objects, etc.

b) Information viewpoint: focuses on information content and system behavior (i.e. data models, semantics, schemas)

c) Computational viewpoint: captures component and interface details without regard to distribution

d) Engineering viewpoint: exposes the distributed nature of the system and provides standard definitions to describe engineering constraints

e) Technology viewpoint: describes where to apply the technologies/products of choice and allows for conformance testing against the architectural specification

This architecture description currently says little directly about the Enterprise and Information viewpoints. We think this OWS architecture is applicable to a broad range of purposes, using a broad range of information. However, Tables 1 and 2 suggest some of the types of information handled.

Many aspects of the Computation viewpoint are discussed in Clauses 5-7 of this document. Clauses 8 and 9 discuss some aspects of the Technology and Engineering viewpoints.

Bibliography

[1] CGI, *The Common Gateway Interface*, National Center for Supercomputing Applications, http://hoohoo.ncsa.uiuc.edu/cgi/

[2] IETF RFC 2045 (November 1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Freed, N. and Borenstein N., eds., http://www.ietf.org/rfc/rfc2045.txt

[3] IETF RFC 2141 (May 1997), *URN Syntax*, R. Moats http://www.ietf.org/rfc/rfc2141.txt

[4] IETF RFC 2396 (August 1998), *Uniform Resource Identifiers (URI): Generic Syntax*, Berners-Lee, T., Fielding, N., and Masinter, L., eds., http://www.ietf.org/rfc/rfc2396.txt

[5] IETF RFC 2616 (June 1999), *Hypertext Transfer Protocol – HTTP/1.1*, Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., eds., http://www.ietf.org/rfc/rfc2616.txt

[6] IANA, Internet Assigned Numbers Authority, *MIME Media Types*, http://www.iana.org/assignments/media-types/

[7] ISO/IEC 10746-1:1998, *Reference Model for Open Distributed Processing (RM-ODP)*

[8] OGC 2001, *Introduction to OGC Web Services*

[9] OGC 02-006, *OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture*

[10] OGC 03-025, *OpenGIS Web Services Architecture*

[11] OGC 05-008, *OGC Web Services Common Specification*, version 1.0.0

[12] OGC 05-010, *URNs of definitions in ogc namespace*

[13] SIDP (Spatial Interoperability Demonstration Project) 2004-2005, *Notional Architecture*, http://www.sidp.com.au/

[14] W3C Recommendation 04 February 2004, *Extensible Markup Language (XML) 1.0 (Third Edition)*, http://www.w3.org/TR/REC-xml

[15] W3C Recommendation 2 May 2001: *XML Schema Part 0: Primer*, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

[16] W3C Recommendation 2 May 2001: *XML Schema Part 1: Structures*, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[17] W3C Recommendation 2 May 2001: *XML Schema Part 2: Datatypes*, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/