International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.2232

(01/2008)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

Next Generation Networks – Service aspects: Service capabilities and service architecture

NGN convergence service model and scenario using web services

Recommendation ITU-T Y.2232

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	**Y.2200–Y.2249**
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.2232

NGN convergence service model and scenario using web services

Summary

The 'convergence service' in NGN implies the integration of services in NGN in a unified manner to access each service in order to interwork with each service. Recommendation ITU-T Y.2232 defines the convergence model for NGN based on web services and provides a detailed scenario of each convergence model in the form of web services.

Source

Recommendation ITU-T Y.2232 was approved on 25 January 2008 by ITU-T Study Group 13 (2005-2008) under Recommendation ITU-T A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Introduction

The NGN service architecture has the three main functional characteristics of (i) agnosticism, (ii) support for legacy capabilities and features, and (iii) support for an open service interface (Recommendation ITU-T Y.2012). Among these, the third characteristic implies that the NGN service platform is required to provide an open service interface, which provides an abstraction of the network capabilities.

NGN users can create and provide enhanced services, which enables application of NGN capabilities using the "applications" functional group known as the application-to-network interface (ANI) (Recommendation ITU-T Y.2012). ANI provides a channel for interactions and exchanges between applications and NGN elements. ANI offers the capabilities and resources needed for the realization of applications. Also, NGN provides open service environment (Recommendation ITU-T Y.2201) for application developers.

This Recommendation proposes the NGN convergence models and scenario of (i) interactions among web services-enabled NGN services, and (ii) interaction with web services-enabled NGN services, along with NGN services, which does not have a web services feature. The overall value proposition of this Recommendation lies in extending the space of application developers for the NGN services interface to include members of IT communities and others who are in skills areas other than programming language developers, such as web developers. The various perspectives on the value proposition are as follows:

- The end user is the consumer of the services. The end user is provided with more services in a timelier manner, and services may be delivered that are more personalized to their unique market segments.
- The application developer is the person programming the application who makes use of web services to deliver application functionality to the end user. The developer benefits from (i) access to NGN capabilities using an intuitive function, (ii) the ability to use a common application framework that supports web services to build and deploy their applications.
- A service provider is an entity that operates the NGN services. The value proposition of web services for the service provider is (i) to offer a wide range of services rapidly and inexpensively, (ii) to differentiate itself by means of offering specialized services and serving strategic niche markets, (iii) to reach customers who are only interested in niche applications, and possibly cross-sell them, and (iv) to build customer loyalty by providing a means to customize services and assistance with this.
- The network operator is the entity that supports the network resources that support the web services. The value proposition for the network operator is the increased use of network resources and hence, increased revenue.

Recommendation ITU-T Y.2232

NGN convergence service model and scenario using web services

1 Scope

The objective of this Recommendation is to describe the NGN convergence service model and scenario using web services. This Recommendation defines the convergence model for NGN based on web services and provides a detailed scenario of each convergence model in the form of web services.

The term 'convergence' in NGN has focused mainly on the convergence of media, such as voice, data and video. However, to realize the ultimate convergence of services in NGN, it is necessary to develop detailed requirements to allow for the convergence of services. It is also important to clearly identify the value being added by a convergence service. The 'convergence services' in NGN imply the integration of services in NGN in a unified manner to access each service in order to interwork with each service.

Implementing the convergence service in current networks may be restricted or impossible due to the capabilities of the installed equipment. Moreover, service provisioning of the convergence service to implement new functionalities is essentially restricted to equipment vendors, as the APIs are typically proprietary [ITU-T Y.2012]. Thus, for the open service interface of NGN, NGN is required to support standard ANI API and a unified manner for accessing the NGN services from the IT side.

Currently, the capabilities and requirements [b-ITU-T Y.2000-Sup.1] [ITU-T Y.2201] for open services environments have been described. Web services is one of the technologies of open services environments. The use of web services is expanding rapidly as the need for application-to-application communication and interoperability grows. In NGN, web services provide a standard means of communication among different software applications of NGN services to the users. To address these capabilities among these applications in NGN environments, and to allow them to be combined for more complex services, a standard reference convergence model is needed in an NGN environment.

The scope of this Recommendation is to address a convergence service scenario in NGN using web services. This Recommendation covers:

- Requirements to support convergence services model and scenario.
- A web services deployment model for NGN.
- NGN convergence services model and its scenarios using web services.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.2012]	Recommendation ITU-T Y.2012 (2006), *Functional requirements and architecture of the NGN release 1.*
[ITU-T Y.2013]	Recommendation ITU-T Y.2013 (2006), *Converged services framework functional requirements and architecture.*

[ITU-T Y.2091] Recommendation ITU-T Y.2091 (2008), *Terms and definitions for Next Generation Networks.*

[ITU-T Y.2201] Recommendation ITU-T Y.2201 (2007), *NGN release 1 requirements.*

[ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1.*

[W3C SOAP 0] W3C Recommendation (2007), *SOAP Version 1.2 Part 0: Primer (Second Edition).* <http://www.w3.org/TR/soap12-part0/>

[W3C SOAP 1] W3C Recommendation (2007), *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition).* <http://www.w3.org/TR/soap12-part1/>

[W3C WSDL 0] W3C Recommendation (2007), *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer.* <http://www.w3.org/TR/wsdl20-primer/>

[W3C WSDL 1] W3C Recommendation (2007), *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language.* <http://www.w3.org/TR/wsdl20/>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 access [b-ITU-T X.1142]: Performing an action.

3.1.2 access control [b-ITU-T X.1142]: Controlling access in accordance with a policy.

3.1.3 access control information [b-ITU-T X.812]: Any information used for access control purposes, including contextual information.

3.1.4 gateway [b-ITU-T H.310]: A function that converts transmission formats and/or protocols between different network environments.

3.1.5 principal [b-ITU-T X.811]: An entity whose identity can be authenticated.

3.1.6 service [ITU-T Y.2091]: A set of functions and facilities offered to a user by a provider.

3.1.7 service convergence [ITU-T Y.2013]: The coordination of a set of services such that the end user's view is that of a single service. The component services may have different providers.

3.1.8 SOAP [W3C SOAP 1]: The formal set of conventions governing the format and processing rules of a SOAP message. These conventions include the interactions among SOAP nodes generating and accepting SOAP messages for the purpose of exchanging information along a SOAP message path.

3.1.9 SOAP intermediary [W3C SOAP 1]: A SOAP intermediary is both a SOAP receiver and a SOAP sender and is targetable from within a SOAP message. It processes the SOAP header blocks targeted at it and acts to forward a SOAP message towards an ultimate SOAP receiver.

3.1.10 WSDL [W3C WSDL 1]: Web services description language Version 2.0 (WSDL 2.0) provides a model and an XML format for describing Web services. WSDL 2.0 enables one to separate the description of the abstract functionality offered by a service from concrete details of a service description such as "how" and "where" that functionality is offered.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 convergence service: A service resulting from service convergence.

NOTE – In this Recommendation, service convergence is realized using web services technologies.

3.2.2 service composition: Service composition is the capability of creating a new NGN service from other existing NGN services.

3.2.3 service provider: An entity that provides services.

3.2.4 service substitution: Service substitution is a capability for replacing a service when an original requested service is not available.

3.2.5 web services: Web services is a service provided using web services systems.

3.2.6 Web services gateway (WSG): A gateway which handles the web services message between the WSP and WSR.

3.2.7 Web services provider (WSP): A service provider that exposes a capability for use to create web services.

3.2.8 Web services requester (WSR): Client software that makes use of the services provided by a WSP.

3.2.9 Web services registry: An entity that stores web services information (e.g., WSDL).

3.2.10 Web services system: A web services system is a software system designed to support interoperable machine-to-machine interaction over a network using web services standards.

NOTE – It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the web services 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.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ANI	Application-to-Network Interface
API	Application Program Interface
AS	Application Server
ASP	Application Service Provider
BP	Basic Profile
BSP	Basic Security Profile
MEP	Message Exchange Pattern
NGN	Next Generation Network
NWSP	Non-Web Services Provider
NWSR	Non-Web Services Requester
PEP	Policy Enforcement Point
SIP	Session Initiation Protocol
SOAP	Simple Object Access Protocol
SSP	Service Support Provider
TLS	Transport Layer Security
UE	User Equipment
WSDL	Web Services Description Language
WSG	Web Services Gateway
WS-I	Web Services Interoperability Organization

WSP	Web Services Provider
WSR	Web Services Requester
WSS	Web Services Security
XML	eXtensible Markup Language

5 Conventions

This Recommendation uses the following assumptions:

- The convergence service is required to utilize various kinds of NGN capabilities.
- If an NGN does not support web services interfaces for the service, the web services system can take into account adaptation mechanisms in WSG.
- The target services of convergence are not only the NGN service but also non-NGN services.
- Convergence services are required to enable timely service creation.
- SOAP is assumed for the binding mechanism, although other binding mechanisms can be used for web services.

In this Recommendation, the term "web services" is not used as plural, but as a proper noun developed by W3C.

6 Requirements to support convergence services model and scenario

This clause describes the requirements to support the convergence services model and scenario using web services. For the convergence service scenario using web services, the following requirements are necessary.

- NGN is required to provide an open interface for application providers to support web services.
- A unified service discovery mechanism is required to be able to find the service, which is the target service for convergence.
- It is required to support the WSDL for the convergence service. In the WSDL, a set of information that describes the interface to and semantics of a service are described.
- It is required to provide service level interoperability underlying various different networks, operating systems and programming languages.
- It is required to provide general security capability for the service, which is opened to make a convergence service.
- Service discovery and invocation is required to support service provider and consumer policies.

7 Web services deployment model in NGN

Web services systems enable business entities and applications to intercommunicate openly with each other over a network. Web services systems have program-language-independent properties, use message-driven communication, and are easily bound to different transport protocols. In NGN services, it is possible to make common interfaces for service integration using web services technology.

7.1 Characteristics of web services

The characteristics of web services for the convergence scenarios in NGN are as follows:

- Loosely coupled connections between services

Web services systems communicate by passing XML messages via a service externalization layer. The service abstraction layer facilitates the task of developing services that consumes NGN exposed services.

- Interoperability across open and proprietary platforms

  Web services systems are of high value to the enterprise as they provide, at their core, interoperability among hardware platforms, operating systems, databases, middleware and applications. These allow large organizations to benefit from a flexible IT architecture built on well-established standards without limiting the flexibility of individual lines of business to implement technologies that provide the business capabilities they need.

- A new approach to programming and application development

  SOAP is a programming protocol that can be used to expose services and contents, thereby creating a dynamic medium (the web, intranet, extranets, and others) for programmable information exchanges. This translates into faster development cycles and lower costs enabled by component reuse, increased developer collaboration, component sharing, and business agility that allow firms to respond to changes in the business environment without being held back by long implementation and deployment cycles on the IT side.

- An open and flexible technique to deliver services to users anywhere

  Using web services, a company can provide access to inventory transactions, financial data, insurance claim processing, or any type of information, transaction, or process in a reusable and open fashion.

7.2 Web services deployment model

The basic web services defines an interaction [W3C SOAP 0] [W3C SOAP 1] between service requesters and service providers as an exchange of messages, as shown in Figure 1.

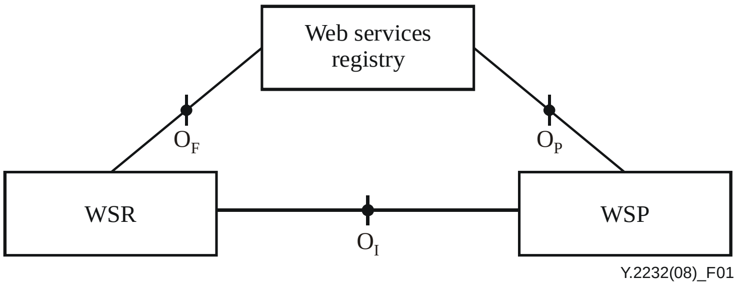

Figure 1 – General architecture of web services

In the WSDL, a service and services descriptions are described [W3C WSDL 0]. A service is enabled by a software module deployed on network-accessible platforms provided by the service provider. The service description contains the details of the interface and implementation of the service including the data types, operations, binding information and network location. The service description is supported by WSP.

In order for an application to take advantage of web services, three behaviours must take place: the publication of service descriptions, the finding and retrieval of service descriptions, and the binding or invoking of services based on the service description. These behaviours can occur singly or iteratively, with any cardinality between the roles. In detail, these operations are:

O_I execution (binding or invoking) of services based on the service description

O_F finding and retrieval of service descriptions

O_P publication of service descriptions

The basic web services architecture models the interactions between three roles (WSP, web services registry and WSR). The entire procedure of web services usage can be divided into publish, find (or discovery) and bind (or interaction) phases.

In the publish phase of web services, the WSP develops its applications and creates a service description, publishes that service description (WSDL) to one or more web services registry to allow discovery with an O_P operation, as shown in Figure 2, and the web services ready to receive messages from the WSR.

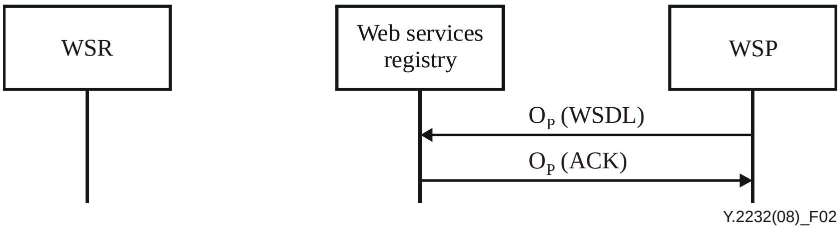

Figure 2 – Information flow of O_P in web services

In the find phase of web services, the WSR finds the service description of interest and uses this service description with the O_F operation.

The web services registry advertises the web services descriptions published to it by WSPs with the O_P operation to helps WSRs search through its registry to find a service description of interest with the O_F operation.

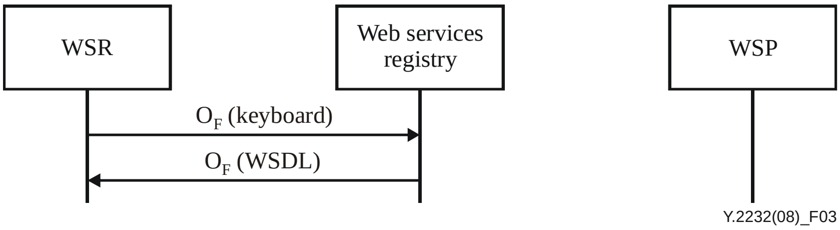

Figure 3 – Information flow of O_F in web services

In the bind phase of web services, the WSR interacts with the services provided by the WSP with the O_I operation.

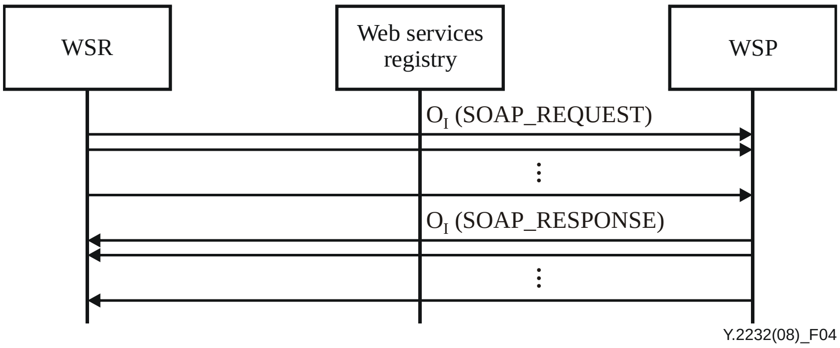

Figure 4 – Information flow of O_I in web services

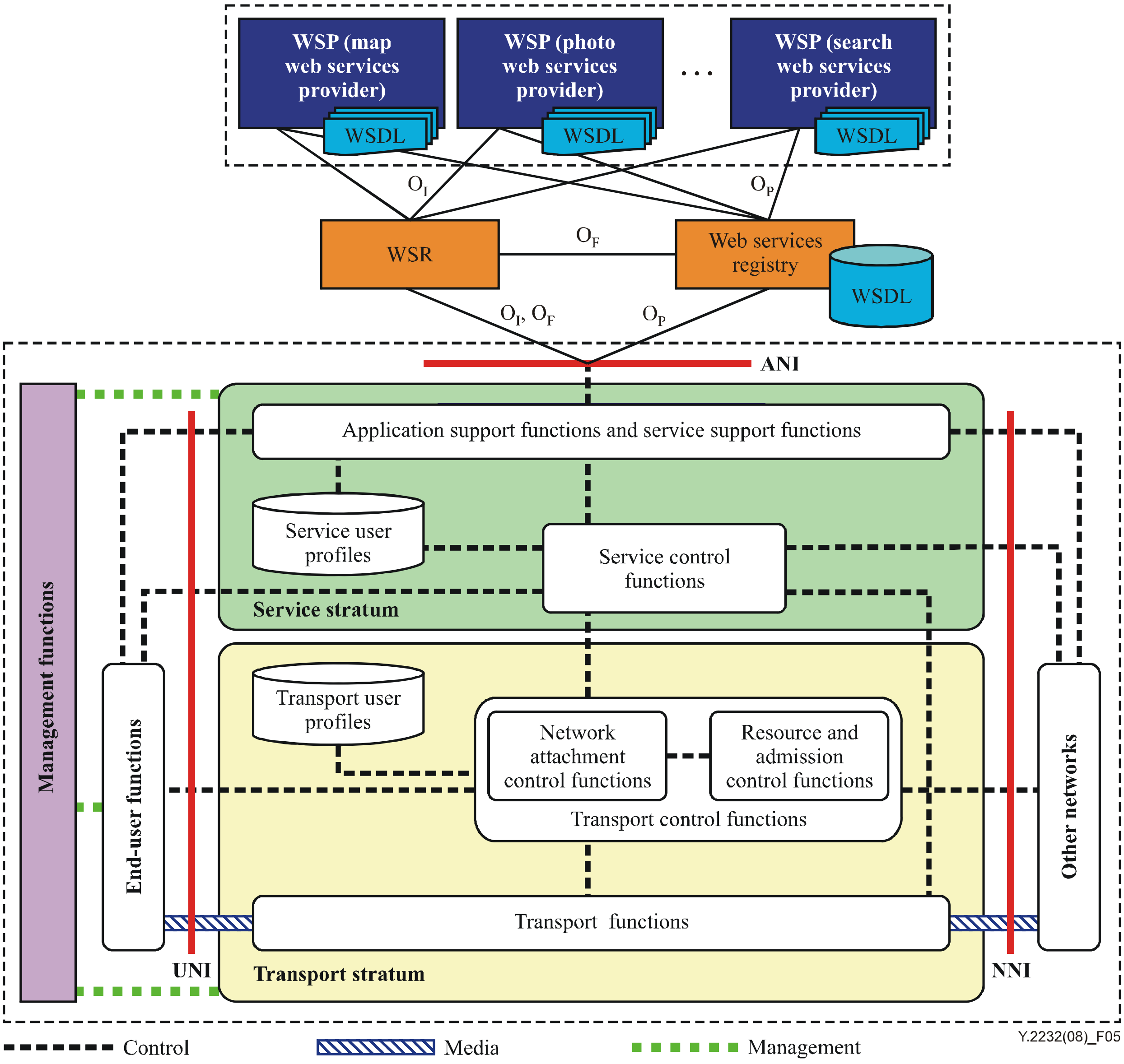
WSP (map web services provider)
WSP (photo web services provider)
WSP (search web services provider)
WSDL
WSDL
WSDL
. . .
O_I
O_P
WSR
O_F
Web services registry
WSDL
O_I, O_F
O_P
ANI
Management functions
Application support functions and service support functions
Service user profiles
Service control functions
Service stratum
End-user functions
Transport user profiles
Network attachment control functions
Resource and admission control functions
Transport control functions
Transport functions
UNI
Transport stratum
NNI
Other networks
Control
Media
Management
Y.2232(08)_F05

Figure 5 – Conceptual diagram of the NGN architecture with an extension of web services

NOTE – Figure 5 assumes that the WSP for NGN is located inside NGN.

For an NGN architecture extension with web services, as shown in Figure 5, ANI can be used as the interface of WSPs, and registers their interfaces (WSDL) to the web services registry, as shown in Figure 5. The WSR can find the interfaces (WSDL) of an NGN service from the web services registry, and invokes the interface of the ANI with a web services access method such as SOAP. Before the interaction, the interface of an NGN service can be registered in the web services registry to access their open interfaces (WSDL) by web services. In WSDL, the location of service, API name, the parameter and its type, and other information for the service are described. If web services are deployed in NGN, various types of services in NGN could combine the services of non-NGN environments such as maps, photos and searches related to web services in a unified manner, implying that additional value-added services are created for NGN end-users.

7.3 Web services gateway (WSG)

The WSG is a component that has functionalities of not only enabling web services features for non-web services, but also performs intermediate work between the WSR and WSP. Therefore, the main roles of the WSG are to transform messages in a value-added manner among the WSR, WSP, NWSP and NWSR, and to perform certain functions associated with the messages such as security management, QoS control, service composition or other operations in a message path among the WSR, WSP, NWSP and NWSR. This implies that the WSG has WSP capabilities, and that its outgoing messages are equivalent to its incoming messages in an application-defined sense.

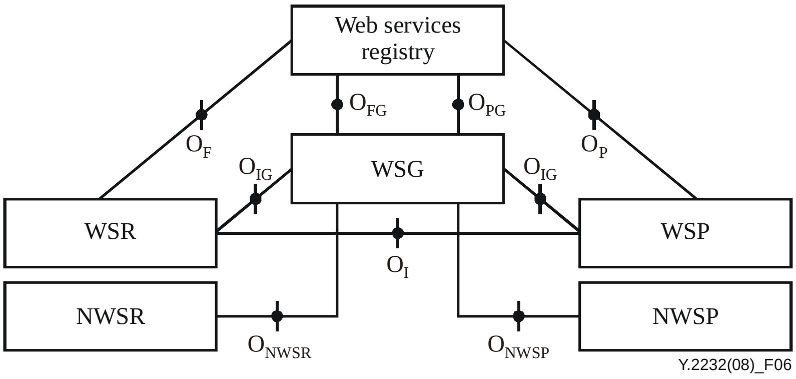

Figure 6 – Conceptual diagram of the web services gateway

The differences between the general architecture of web services in Figure 1 and the WSG in Figure 6 are the relationships of the WSG among WSR, WSP, NWSP and NWSR. The NWSP is another type of service provider that has no web services feature, and the NWSR is a service requester that does not include the features of web services. In terms of web services, the WSG consists of a set of request/response messages between the WSP and WSR, such as O_{FG}, O_{PG}, and O_{IG}. Additionally, the WSG has an adopter capability for NWSP and NWSR as in O_{NWSP} and O_{NWSR}. In detail, these operations are:

O_{FG} WSG operation for the finding and retrieval of service descriptions

O_{PG} WSG operation publication of service descriptions

O_{IG} WSG operation SOAP execution of services based on the service description

O_{NWSP} WSG operation for proprietary operation with NWSP

O_{NWSR} WSG operation for proprietary operation with NWSR

The extended web services model for WSG is similar to the traditional web services model in clause 7.2, and the phases are divided into the publish, find and bind phases. Figures 7, 8 and 9 show the publish phase, find phase and bind phase, respectively.

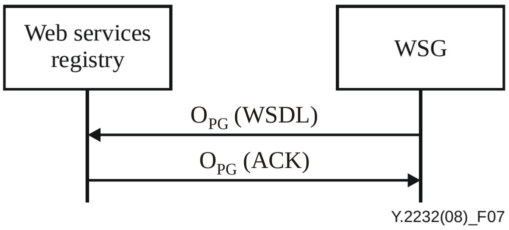

Figure 7 – Information flow of O_{PG}

In the publish phase for WSG, the WSG sends a service description of WSG in forms of WSDL using O_{PG}. The web services registry then registers the service description into the database. The candidate service description, as published in the web services registry, is the basic adapter for NWSP, as well as the security, QoS control, substitution, or other operations interface in a message path between the WSR and WSP.

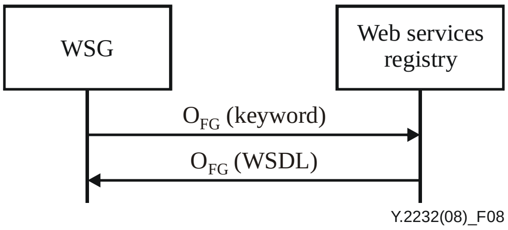

Figure 8 – Information flow of O_{FG}

The O_{FG}, which finds the interface of the WSG, is also similar with O_F. Using O_{FG}, the WSG can retrieve the service description in the web services registry. It can fetch the WSDL from the web services registry.

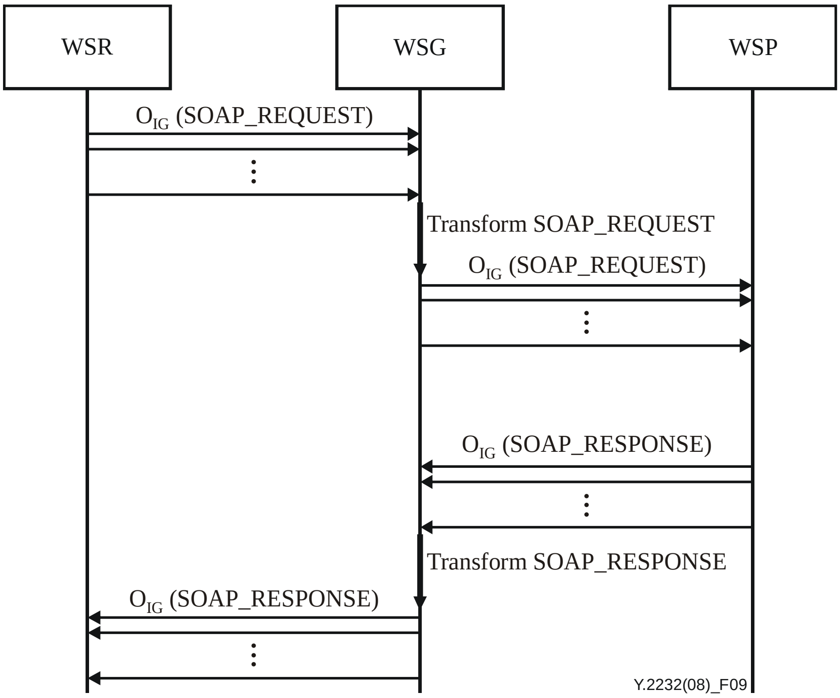

Figure 9 – Information flow of O_{IG}

In the bind phase of web services using the WSG, the WSR interacts with the services provided by the WSP with the O_{IG} operation, and the WSG transforms the request message and response message. In terms of SOAP messaging, the WSR consists of a set of request/response messages between the WSP and WSR.

7.4 Web services gateway interaction patterns

In the context of web services, the interaction pattern is a sequence of interactions between the WSR and one or more WSP/WSG; this achieves measurable results for the services requestor. The basic interaction pattern in web services is "request and response" as shown in Figure 10-a. If the WSR has parameters to be sent to the WSP, the parameters are serialized into a message for transmission to the WSP. The WSP then processes the message content and responds to the WSR.

It is also possible that the WSR delivers the message to multiple WSPs, as shown in Figure 10-b. The delivery of the messages can be implemented using multicast distribution technology if the underlying transport layer supports this. An alternative implementation may use repetition of a distribution list of intended recipients.

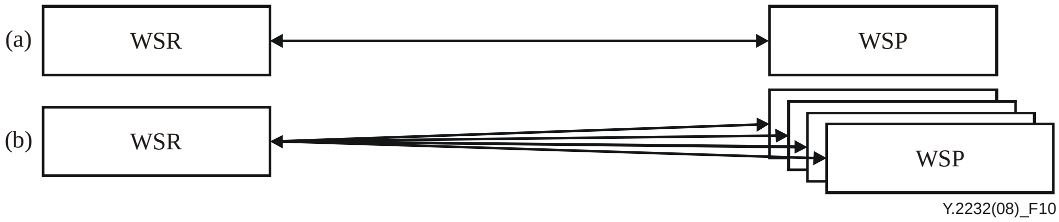

Figure 10 – Request and response interaction pattern in web services

However, in some cases, web services interaction patterns could be complex due to different capabilities between the WSR and WSP. In this case, various types of WSG interaction patterns can be applied. Moreover, by combining and adopting different interaction patterns, it is possible to produce different paths for the same usage case.

In general, the WSG is located between the WSR and WSP, as shown in Figure 11. By use of the WSG, the WSR is able to interact with the WSP. Also, according to the request path of web services, the WSG can be used several times as shown in Figure 12.

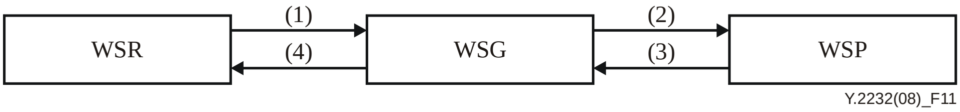

Figure 11 – Basic WSG interaction pattern

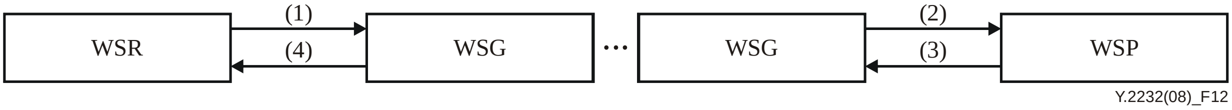

Figure 12 – Combination of WSG interaction pattern

A WSG forwards a message to the ultimate WSP on behalf of the initial WSR. The initial WSR wishes to enforce the non-repudiation property of the route. Any WSG message service handler that appends a routing message must log the routing header information. Signed routing headers and the message readers must be logged at the message handler which passes the message to the ultimate WSP to provide evidence of non-repudiation.

According to the functionalities of the WSG, the WSG has two functions. One is enabling web services features for non-web services; the other is performing certain functions associated with the message. The following subclauses are various WSG interaction patterns for the WSG. The following interaction pattern can work singly, and several interaction patterns can be mixed up if necessary.

7.4.1 Adapter pattern

The adoption of non-web services into web services is one of main roles of the WSG. This assumes that the WSG has the capabilities for translation using a proprietary format for NGN services and interworking with the heterogeneous protocol of NGN services. The information flows for WSG with NWSP, as shown in Figure 13, are similar to the conventional WSG shown in Figure 8. However, all NGN services that cannot support web services features cannot be used by the WSG. Thus, before using the WSG for non-web services, an adaptor for non-web services must be procured from the service provider.

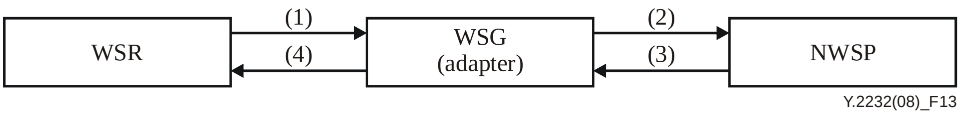

Figure 13 – WSG adapter pattern

7.4.2 Proxy pattern

The WSG acts as a proxy from non-web NGN services requests to web services. The WSG has the capability of translating using a proprietary format from the NGN service protocol (e.g., SIP, HTTP) to a web protocol. In terms of the interfaces, as (third-party) web services are much broader and change more dynamically compared to the (operator) NGN services, the proprietary interface between the non-web NGN service and the WSG may be limited in terms of network service exposure (when NGN is NWSP) and in terms of using new third-party web services (when NGN is NWSR).

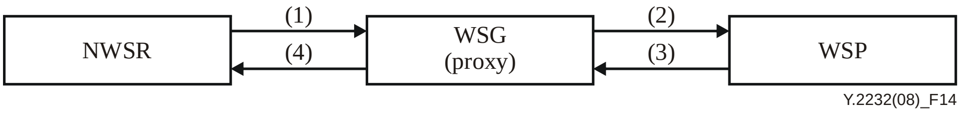

Figure 14 – WSG proxy pattern

In order to use or expose non-web NGN services through the WSG, as the proprietary interfaces are dependent on the NGN service provider, the NGN service provider and web services provider may need to be contacted beforehand.

7.4.3 Security pattern

The security pattern enables the WSG to do the security protection (e.g., authentication, authorization and decryption) for the network. When a web services is deployed in the network, it needs to be registered to the network on a WSG. The WSG and the web services can authenticate each other during the registration process. Transport layer security can be used to secure the data.

The WSG also provides the security protection of the service layer, and it can protect the web services by each web service's own security mechanism or in a single sign on manner. The WSR is authenticated by the WSG, or it can be authenticated by both the WSG and WSP and makes a synthesized authentication result.

A more detailed description of security patterns can be found in Appendix III.

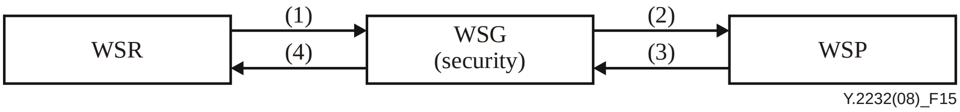

Figure 15 – WSG security pattern

7.4.4 QoS control pattern

The QoS control pattern enables the WSG to control the QoS for the web services. The WSG provides the capability to monitor the QoS of the WSPs, which provide the similar service, and chooses the most suitable WSP to provide the service. When the WSG receives a service request from a WSR, it processes the request according to the WSR's QoS requirement and the WSP's resource status which it monitors or gets it from other WSGs and return the process result to the

WSR. The request information includes service type information, the QoS requirement parameter information, etc.

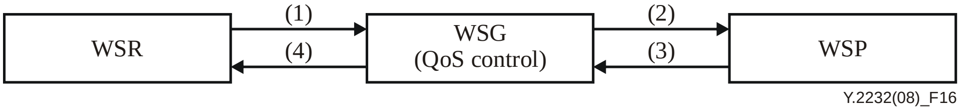

Figure 16 – WSG QoS control pattern

According to the service type information, the WSG queries the WSP's current resource status to get the corresponding WSP's addresses and their current resource status. According to the WSG's control policy, the user's QoS requirement, the obtained WSP's addresses, and their resource status, the WSG may reject the request or generate the addresses list of WSPs and send the result to the WSR. Then the WSR initiates the request to WSP based on the result and gets the response from the WSP.

The control policy of the WSG may include shortest response time, choosing lowest application payload, or the minimum QoS difference between the selected WSP and the required QoS parameter, or choosing WSPs randomly if there are several WSPs satisfying the user's QoS requirement. The QoS parameters in the QoS requirement may include load rate, bandwidth, response time and/or decode rate, etc.

7.4.5 Screening pattern

The screening pattern enables the WSG to do the screening function. Screening implies that the provider can offer different levels or types of service functions or service contents to a requester according to rules, such as those that specify special functions that can be forbidden to some users or at some time, or only some functions can be used, or content-forbidden policy. Cooperating with a WSP, a WSG processes the function or content screening of the request.

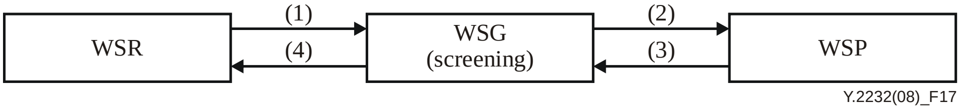

Figure 17 – WSG screening pattern

The WSR initiates the request which includes at least one type of the following information: the service information, the service functions information or the input parameters information. The service functions in this request can be screened before the service result is formed. The request is directly sent to the WSG (the WSR-originated function screening) or firstly sent to the WSP and then is forwarded to the WSG (the WSP-originated function screening). According to the pre-defined screening information, the WSG screens off the invalid service function of the request and forms a new service request. Then the WSP processes this new request and forms the result. Finally, the result of the request is sent to the WSR.

In the situation of content screening, the WSR (the WSR-originated content screening)/WSP (the WSP-originated content screening) sends the contents to the WSG to make them screened: it sends a screening request message which specifies the contents to be screened to the WSG, and the message can include the contents as a direct attachment or a URL pointing to the contents on a remote storage entity. The WSG then screens the contents according to the screening rules, which are defined by the WSP or WSR or another third party, such as virus scanning, content categorizing, content filtering or other screening operations. The WSG returns the screening results to the WSR or the WSP. If the screening results are returned to the WSP, the WSP sends the screened contents to the WSR.

7.4.6 Substitution pattern

The substitution pattern enables the WSG to replace the requested service with another service when the requested service does not work. A more detailed description of the substitution pattern can be found in Appendix III.

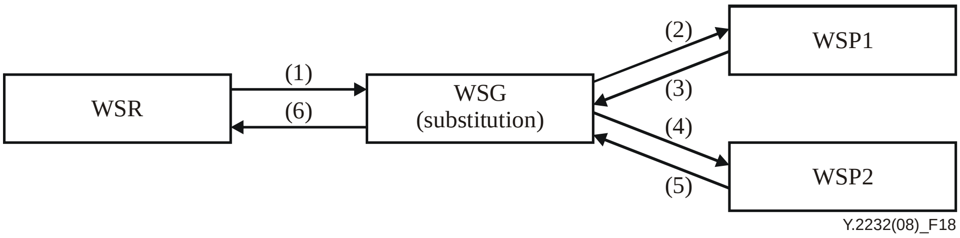

Figure 18 – WSG substitution pattern

Service substitution is used after the service is requested. The WSR invokes a web services through a WSG but the service which is provided by one WSP does not work and the WSG does not receive any response until the deadline. The WSG discovers a web services to replace the original one and invokes the new web services. The WSG is required to provide the capability to obtain the service template information based on the service information (e.g., service name). The service template is an abstract service which maps to more than one exact service component and the service template can be obtained through the mapping relationship. Service template information may include service type, operation (function provided by service), parameter (related to every operation, including input parameter and output parameter), etc. The WSG can select another web services which provides the same function to replace the original based on the service template information, and invoke the selected one.

7.4.7 Composition pattern

The composition pattern enables the WSG to provide a type of service-providing mechanism in which multiple services are invoked in a particular order under the control of service logic which describes the order of the invoking of services and the related parameters. The detailed flow refers to clause III.3.

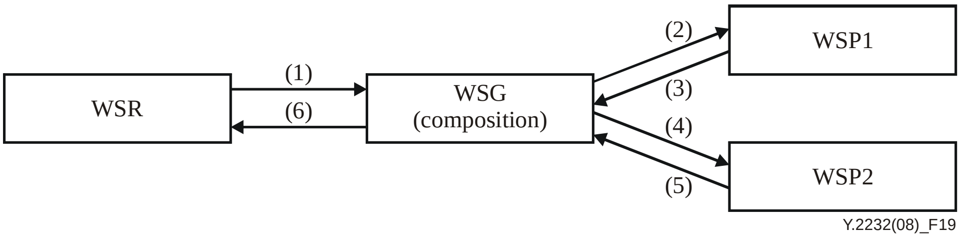

Figure 19 – WSG composition pattern

There are two different kinds of service composition: static service composition and dynamic service composition. For static service composition, it uses concrete service logic specifying concrete services, interfaces invoking information, data flow (services input/output parameters) and control flow (services invoking order) of these services. The WSG invokes these concrete services according to the data flow and control flow and gets the results of these services. The WSG then produces the final result and returns it to the WSR. For dynamic service composition, it uses abstract service logic specifying service classes (different services which provide the same service function belong to the same service class), data flow and control flow of these services. The WSG transfers abstract service logic to concrete service logic first to find the concrete services that can fulfil the requirements to replace the service classes and create invoking information for these services. Then the WSG executes the concrete service logic and gets the results of these concrete services. The WSG then produces the final result and returns it to WSR.

A more detailed description of composition pattern can be found in Appendix III.

8 Web services-based convergence model and convergence service scenarios

In this clause, the basic convergence model, extended convergence model, and the convergence service scenario based on each convergence model are described. For the convergence model, this Recommendation assumes that the NGN service can be located inside and outside of the ANI such as WSP (NGN service) and WSP (third party NGN service).

8.1 Basic convergence model

The basic convergence model indicates the interaction model between the services requester and service provider; this is web services-enabled NGN services. Figure 20 shows the basic convergence model based on web services for Internet and NGN services. The WSP in the Internet and NGN publish their interfaces to the web services registry and the application support functions and service support functions components, respectively. The WSR is able to find the web services it requires through the web services registry, and the application support functions and service support functions are able to create web services.

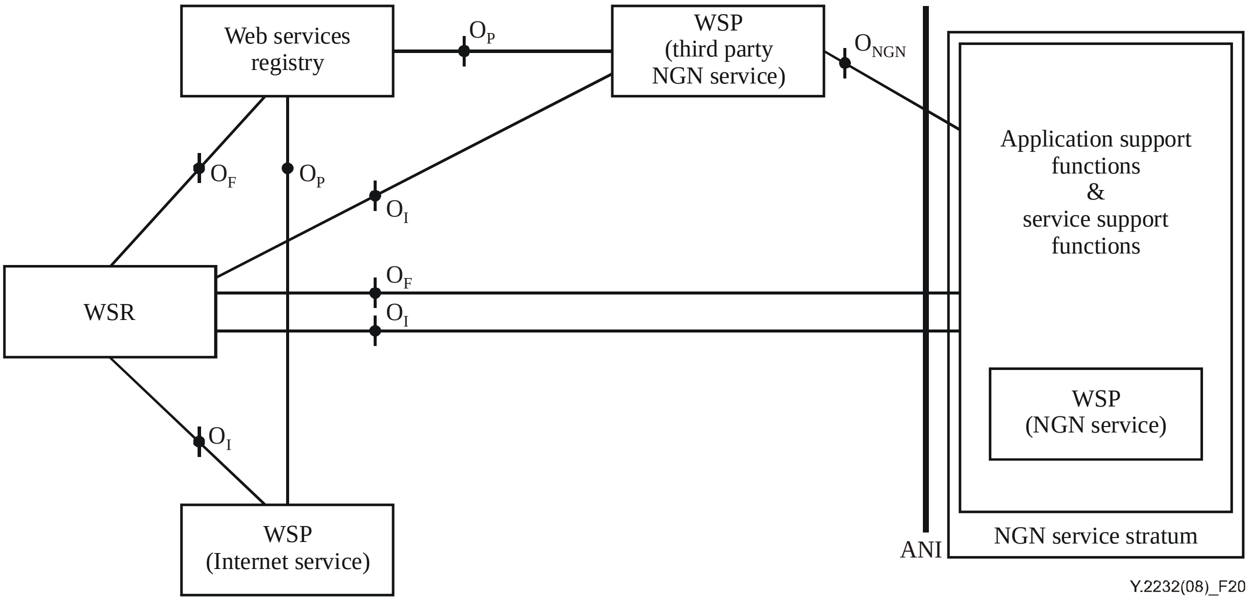

Figure 20 – The components and operations for the basic convergence model

Components

- WSR: Web services request component used for calling the WSP.
- WSP (third party NGN services): The WSP component for third party NGN services.
- WSP (Internet service): WSP component for Internet services.
- Web services registry: Searchable set of service descriptions in which service providers publish their service descriptions.

- Application support functions & service support functions: Providing numerous NGN services based on NGN capabilities for third party service providers and applications. The WSP in application support functions & service support functions is not the FE for NGN, and it is one of the services of NGN, which provides the web services interface.

Key to operations

O_I Binding operation (O_I) between the WSR and third party WSP for NGN services or NGN operation (O_I) for discovering, interaction with the WSP in NGN or binding operation (O_I) between the WSR and WSP for Internet services.

O_{NGN} Proprietary operation (O_{NGN}) between third party web services provider and NGN services.

O_P Publishing operation (O_P) for the service description of NGN services or publishing operation (O_P) for the service description of Internet services.

O_F Finding operation (O_F) for WSP.

Information flows for basic convergence model

NOTE – Figure 21 shows an example of the information flows for the basic convergence model.

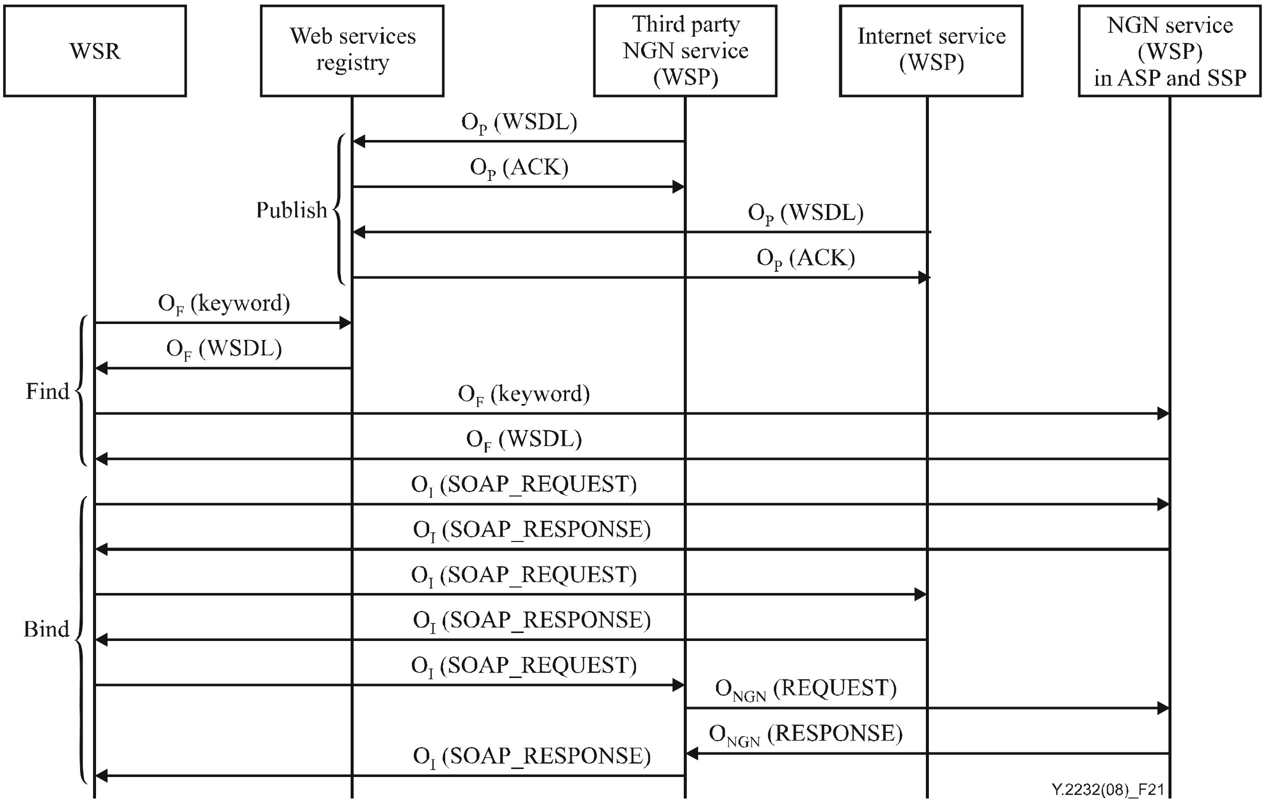

Figure 21 – Information flows for basic convergence model

8.2 Extended convergence model

The extended convergence model is another interaction model between the WSP and NWSP. In essence, all of the interaction based on web services is covered by the basic convergence model. However, although a service in NGN is not web services, the services can be considered in the convergence model. For the extended convergence model, the WSG is required to adopt the NWSP.

Figure 22 shows the extended convergence model that uses the NWSP and NWSR in NGN through the WSG. The WSG is able to support more advanced features such as service screening and service compositions. Therefore, this model can provide the advanced convergence model with services in NGN/Internet depending on the capability of the WSG.

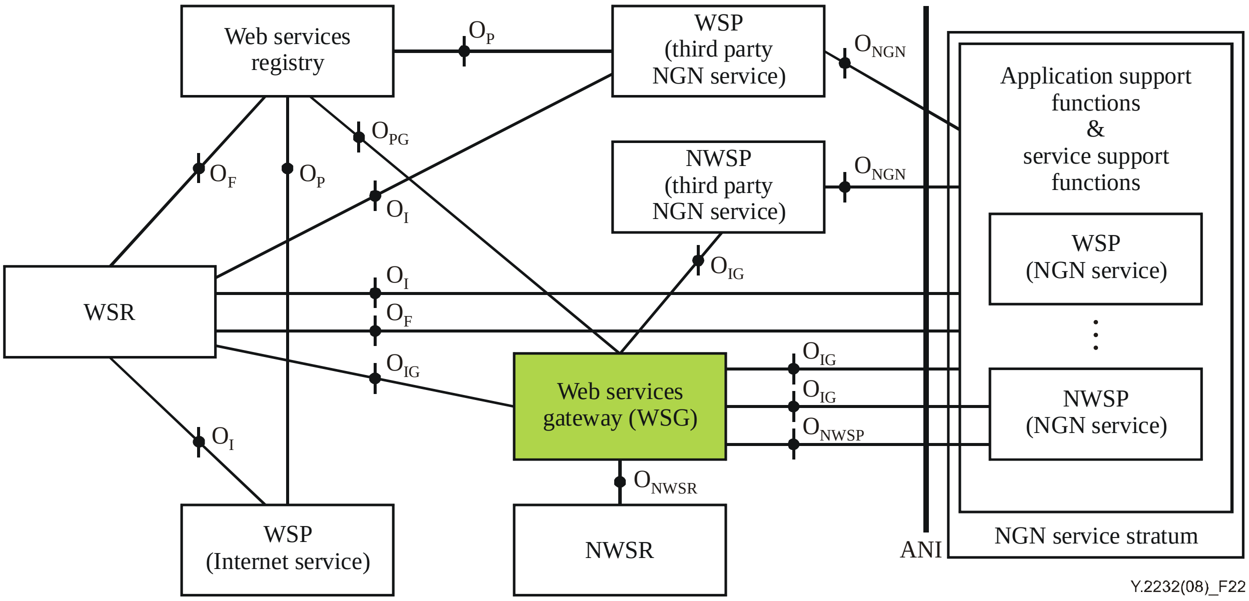

Figure 22 – The components and operations for extended convergence model

Components

- WSR: Web services request component used for calling the WSP.
- WSP (third party NGN services): NGN services to the called WSR.
- NWSP (third party NGN services): NGN services to the called WSG.
- WSP (Internet service): WSP component for Internet services.
- Web services registry: Searchable set of service descriptions in which service providers publish their service descriptions.
- WSG: Gateway for interoperability between web services and NGN services.
- Application support functions & service support functions: Providing numerous NGN services based on NGN capabilities for third party service providers and applications. The WSP and NWSP in application support functions & service support functions are not the FEs for NGN, and are one of the services of NGN with web services interface, or other propriety interface.
- NWSR: Services request component, which has no web services feature.

Key to operations

O_I Bind operation (O_I) between the WSR and third party WSP for NGN or bind operation (O_I) between the WSR and WSP on NGN or Internet.

O_{NGN} Proprietary operation (O_{NGN}) between third party WSP and NGN services.

O_{IG} WSG operation (O_{IG}) between the WSR and WSG and the WSG and NGN service.

O_{PG} Publishing and finding operation (O_{PG}) for the service description of the WSG and WSP in Internet, respectively.

O_P Publishing operation (O_P) for the service description of third party NGN services or publishing operation (O_P) for the service description of Internet services.

O_F Finding operation (O_F) for the WSR.

O_{NWSP} Interaction operation (O_{NWSP}) between the WSG and NWSP in NGN.

O_{NWSR} Interaction operation (O_{NWSR}) between the WSG and NWSR in NGN.

Information flows for extended convergence model

NOTE – Figure 23 shows an example of the information flows for the extended convergence model.

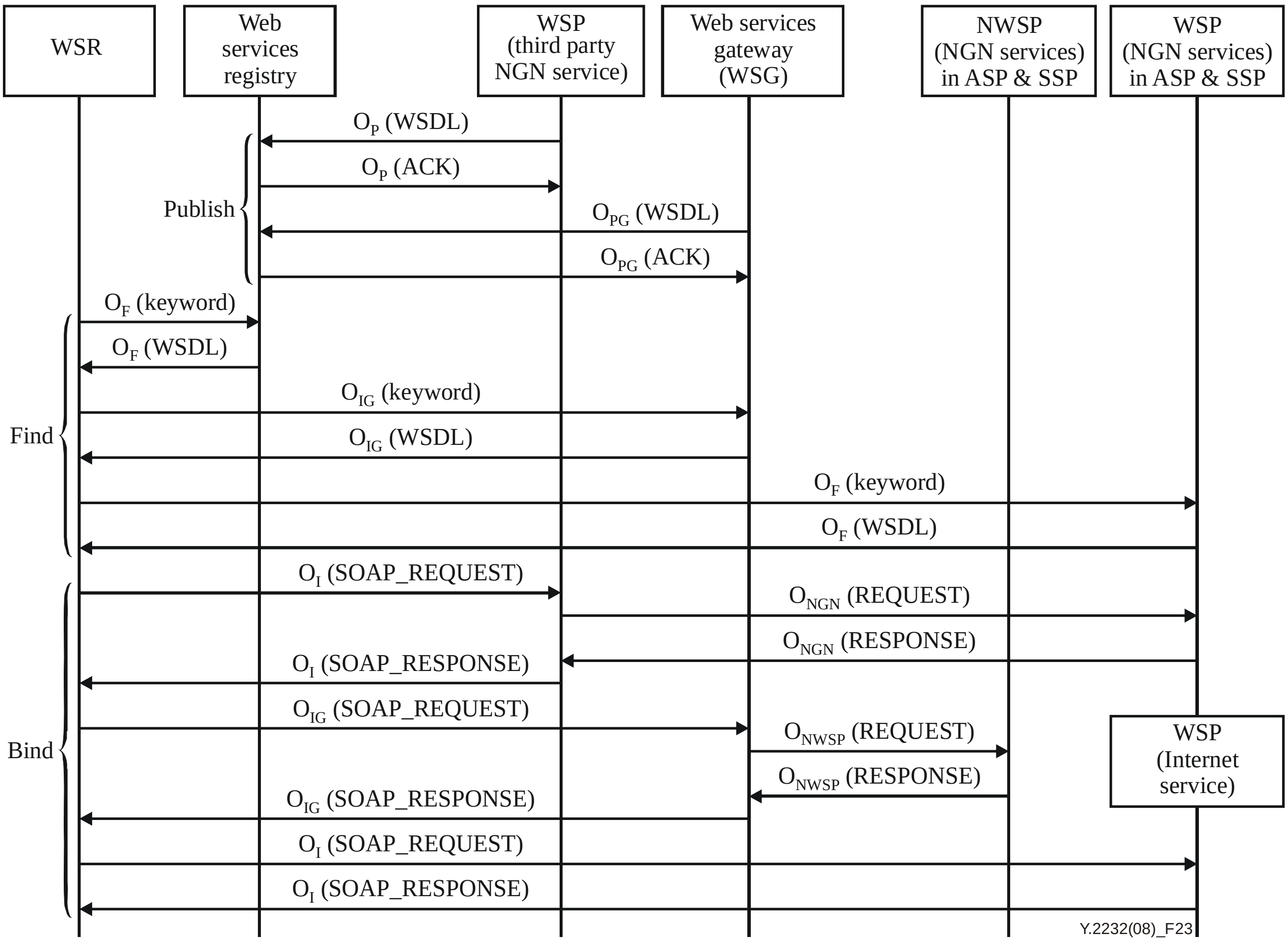

Figure 23 – Introduction flows for extended convergence model

8.3 Convergence service scenarios

8.3.1 Web-based call disposition scenario

The web-based call disposition service provides web pages for a called party to select response menu items such as answer, reject or forwarding to voice mail when a call arrives to the called party in parallel to normal incoming call ringing.

The call disposition menu is provided based on web services, by which a personalized user interface and customized call response menu is possible.

For this service, it is assumed that:

- The terminal equipment of the user is able to support both SIP communications and web services functions.
- The application server providing NGN communication services is able to access web services directly when the AS has web services interfaces, or through the WSG when the AS does not have web services interfaces.

8.3.1.1 Scenario description

The web call disposition service is a web services and it provides a personalized menu to the subscriber. The NGN terminating call application service in the service stratum routes the call based on the response selection of the called party. The web services-enabled NGN AS (WSR) queries the web services registry and requests the web call disposition service based on the returned WSDL. The non-web NGN AS (NWSR) requests the WSG to translate the request and response from the web call disposition services.

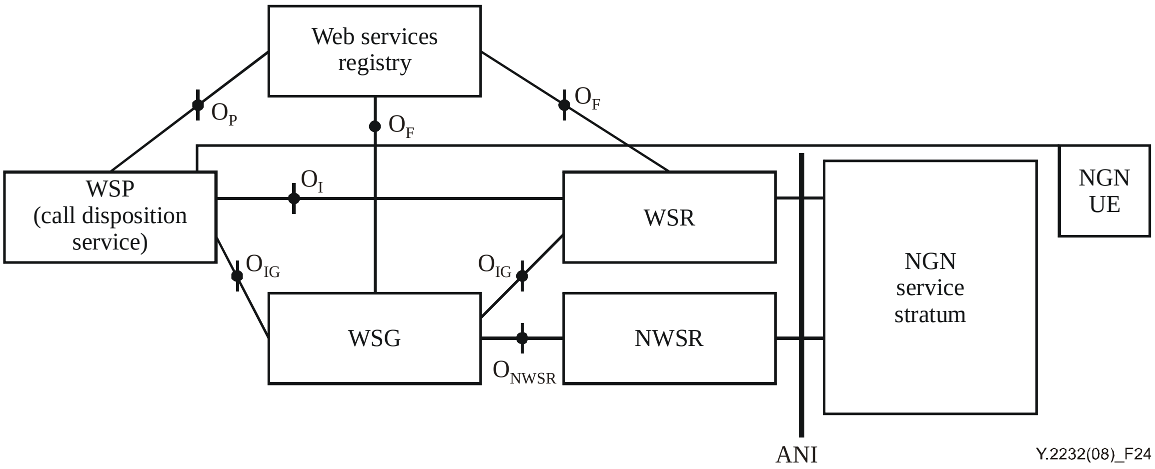

Figure 24 – Web call disposition service

8.3.1.2 Use cases for web-based call disposition

Figures 25 and 26 show possible scenarios for the use cases of the web call disposition service.

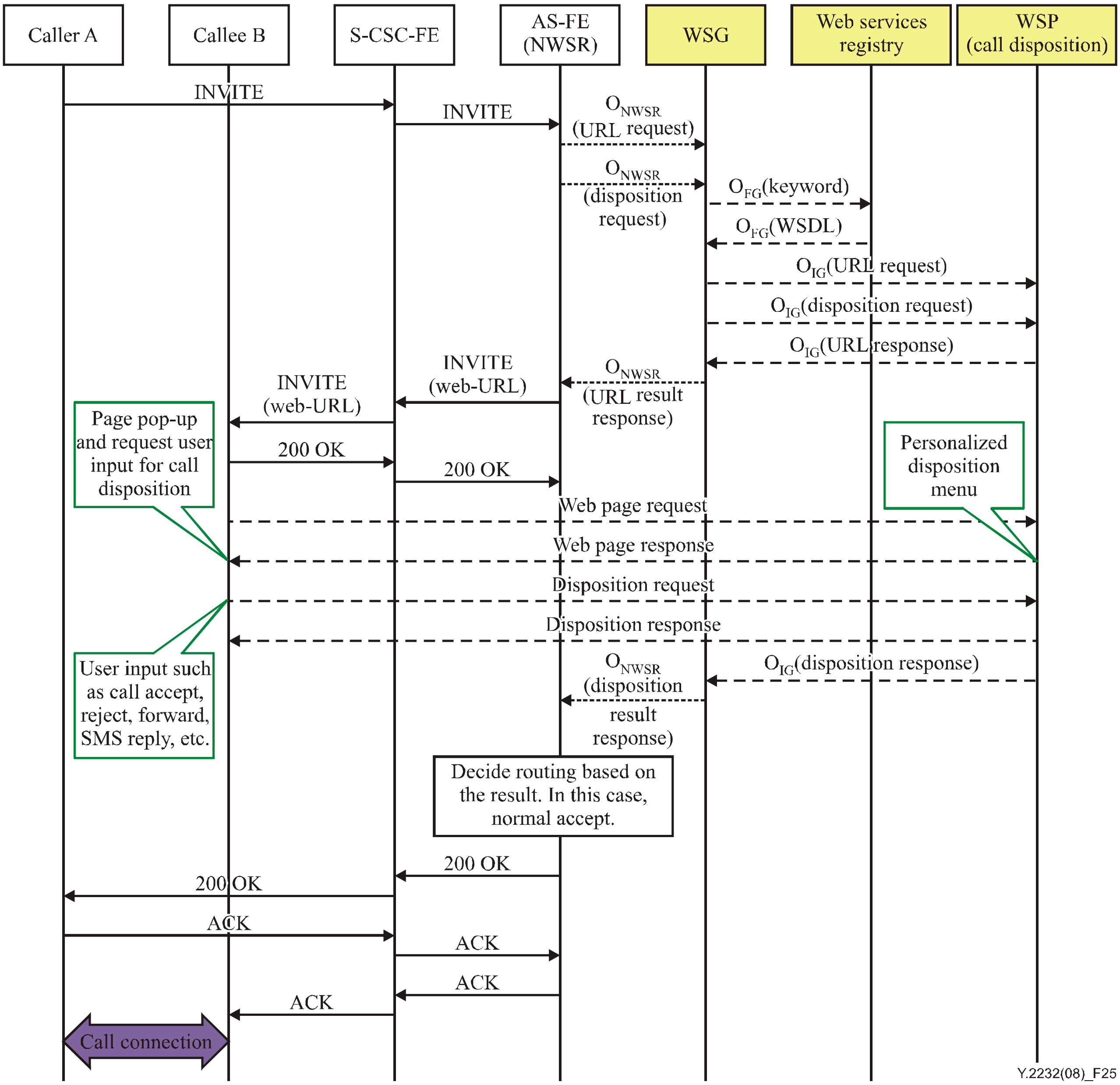
Caller A
Callee B
S-CSC-FE
AS-FE (NWSR)
WSG
Web services registry
WSP (call disposition)
INVITE
INVITE
O_NWSR (URL request)
O_NWSR (disposition request)
O_FG(keyword)
O_FG(WSDL)
O_IG(URL request)
O_IG(disposition request)
O_IG(URL response)
O_NWSR (URL result response)
INVITE (web-URL)
INVITE (web-URL)
Page pop-up and request user input for call disposition
200 OK
200 OK
Personalized disposition menu
Web page request
Web page response
Disposition request
Disposition response
User input such as call accept, reject, forward, SMS reply, etc.
O_IG(disposition response)
O_NWSR (disposition result response)
Decide routing based on the result. In this case, normal accept.
200 OK
200 OK
ACK
ACK
ACK
ACK
Call connection
Y.2232(08)_F25

Figure 25 – Non-web NGN service requesting web disposition service (call disposition web services scenarios)

1) Caller A originates a call to callee B. The terminating S-CSC-FE delivers SIP INVITE to trigger the NGN communication service on application server FE (AS-FE).

2) AS-FE sends a message to the WSG to request the URL for the callee's call disposition WSP and to request a disposition response.

3) WSG retrieves WSDL from the web services registry and then sends the URL and disposition request to the WSP.

4) WSP returns the URL, and AS-FE then returns that URL in the call-info field of INVITE to callee B. Callee B sends normal response 200 OK, to complete the transaction (normal answer case).

5) Using the web URL, callee B's terminal connects to the web server to obtain his personalized web call disposition menu. The web page provides and prompts menus such as call accept, reject or forward to voice mail.

6) Callee B answers via an input response on the menu page. This result is sent to the web call disposition service, WSG and then the AS-FE.

7) Based on the input result, AS-FE determines further routing. In this case, it is normal call accept, and the AS-FE connects the call to the called party.

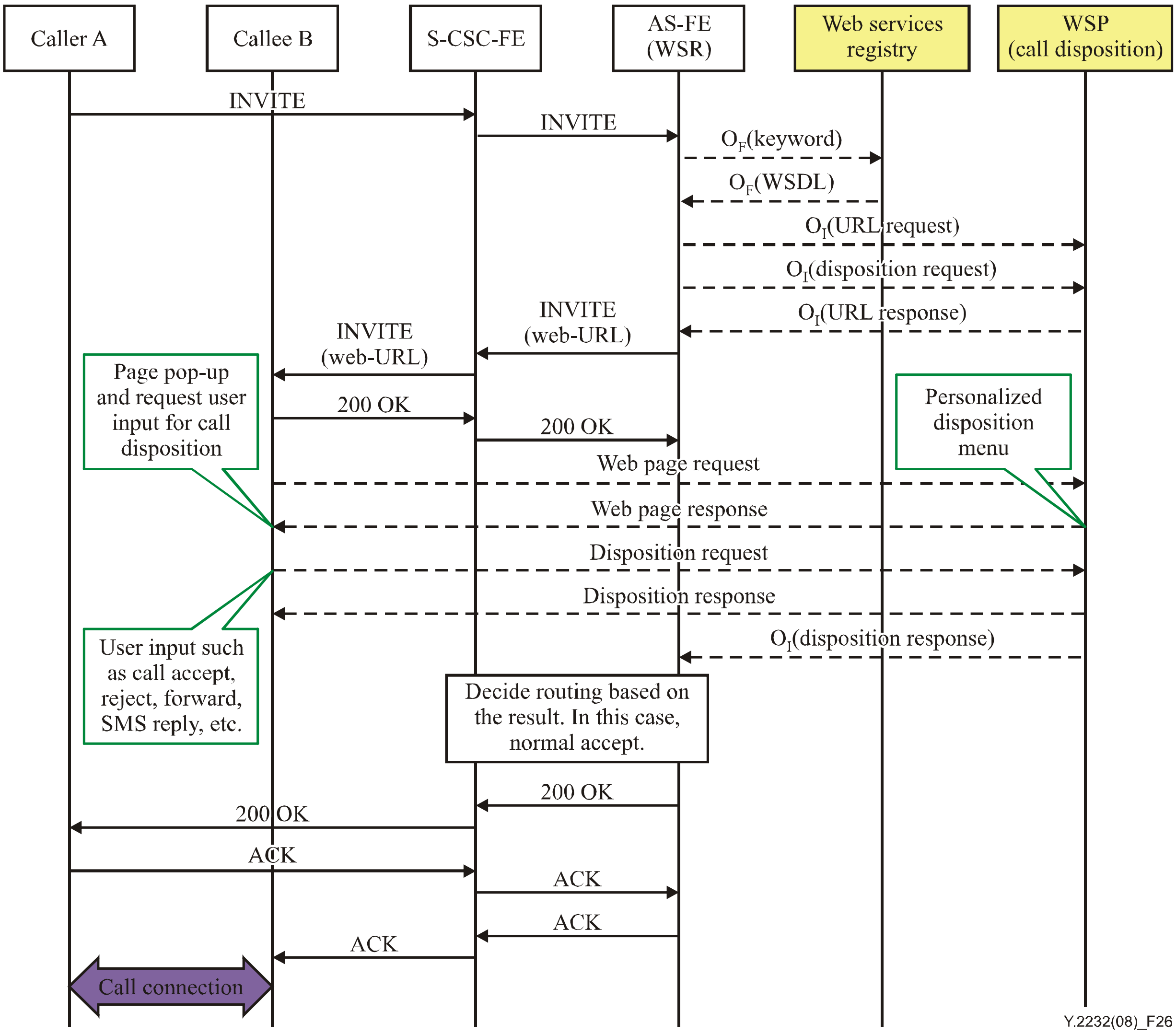

Figure 26 – Web NGN service requesting web disposition service (call disposition web services scenarios)

1) Caller A originates a call to callee B. The terminating S-CSC-FE delivers SIP INVITE to trigger the NGN communication service on application server FE (AS-FE).

2) AS-FE queries and retrieves WSDL from the web services registry and then sends the URL and disposition request to the WSP.

3) When AS-FE receives the URL, it sends the URL in the call-info field of INVITE to callee B. Callee B sends a normal response 200 OK to complete the transaction.

4) Using the web URL, callee B's terminal connects to the WSP to obtain his personalized web call disposition menu. The web page provides and prompts menu items such as call accept, reject or forward to voice mail.

5) Callee B answers by input response on the menu page. This result is sent to the WSP and then to the AS-FE.

6) Based on the input result, the AS-FE determines further routing. In this case, it is normal call accept, and the AS-FE connects the call to the called party.

9 Security considerations

This Recommendation does not identify additional NGN security requirements beyond those contained in [ITU-T Y.2701].

Appendix I provide some technical considerations for web services security.

9.1 Threats to web services

This clause provides a brief description of the most common threats associated with the deployment of web services.

1) Message alteration

   Message information is altered by inserting, removing or modifying information created by the originator of the information and mistaken by the receiver as being the originator's intention.

2) Confidentiality

   Information within the message is accessible to unintended and unauthorized participants.

3) Falsified messages

   Messages are constructed and sent to a receiver who believes them to have originated from a party other than the sender.

4) Man in the middle

   Man in the middle type of attack occurs when a party poses as the other participant to the real sender and receiver in order to fool both participants.

5) Principal spoofing

   A message is sent which appears to be from another principal.

6) Forged claims

   A message is sent in which the security claims are forged with the purpose of gaining access to unauthorized information.

7) Replay of message parts

   A message is sent which includes portions of another message in an effort to gain access to unauthorized information.

8) Replay

   A whole message is resent by an attacker.

9) Denial of service

   This is an attack in which the attacker does a small amount of work and forces the system under attack to do a large amount of work.

9.2 Security requirements for web services

Security can be a key inhibitor to the widespread implementation and adoption of web services. There are many security challenges for adopting web services. At the highest level, the objective is to create an environment where message level transactions and business processes can be conducted securely in an end-to-end fashion. The requirements for providing end-to-end security for web services are:

- Authentication mechanisms: This is needed in order to allow the mutual authentication of service provider and service invoker to verify their identities. This also includes data origin authentication, whereby the receiver can be sure that the data came from the sender without modification.

- Authorization to access resources: Once authenticated, authorization mechanisms control invoker access to appropriate system resources. There is controlled access to systems and their components.
- Data integrity and confidentiality: This is to ensure that information has not been modified during transmission and is only accessible to intended parties. Encryption technology and digital signature techniques can be used for this purpose.
- Integrity of transactions and communications: This is needed to ensure that the business process was done properly and the flow of operations was executed in a correct manner.
- Non-repudiation: A party to a transaction cannot deny the occurrence of the transaction.
- End-to-end integrity and confidentiality of messages: The integrity and confidentiality of messages must be ensured even in the presence of intermediaries.
- Provide security and audit trails: This is needed in order to trace user access, behaviour and system integrity verification.
- Distributed enforcement of security policy: Implementers must be able to define a security policy [b-IETF 2828] and enforce it across various platforms with varying privileges.

Appendix I

Security considerations for web services

(This appendix does not form an integral part of this Recommendation)

Web services can be protected using security techniques that are designed to reduce risks associated with security threats. The choice of security mechanism in many cases is dependent on the application and the related threats to that application. Security can be implemented at the WSG level. The WSG can serve as a policy enforcement point (PEP) to enforce message level security.

I.1 Security technologies for web services-based NGN convergence services

This clause provides a summary of security technologies that can be used with web services.

I.1.1 Transport layer security

This clause takes a look at using the transport layer to secure SOAP messages that are sent from a sender to a receiver. This approach is limited when there are intermediaries in the message path, since termination of transport layer security at an endpoint may allow that intermediary to modify or examine messages. For HTTP-based bindings of SOAP, TLS [b-IETF 4346] provides point-to-point security (see Figure I.1). However, for web services there is a need for end-to-end security in which multiple intermediaries can exist between the original service requester and the service provider. Transport layer TLS does not support XML-based element signing or encryption such as signing or encrypting part of a large XML document. Transport layer security mechanisms may be used to secure messages between two adjacent SOAP nodes whereas message layer security mechanisms can be used in the presence of intermediaries or when data origin authentication is required.

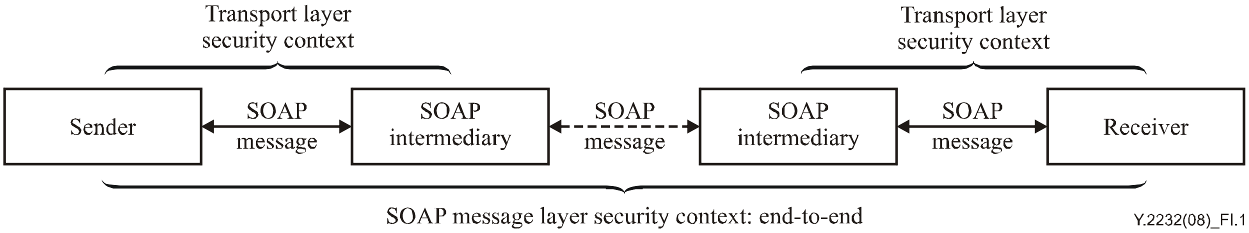
Transport layer security context
Transport layer security context
Sender
SOAP message
SOAP intermediary
SOAP message
SOAP intermediary
SOAP message
Receiver
SOAP message layer security context: end-to-end
Y.2232(08)_FI.1

Figure I.1 – Transport and message level security

I.1.2 Message level security

W3C SOAP specifications do not specify how to deal with message related security issues as such. However, W3C specifications allows for the ability to build extensions to the original SOAP standard. In this regard, the OASIS WS-Security versions specifications (WSS) 1.0 and 1.1 [b-OASIS WSS 1.0] [b-OASIS WSS 1.1] use these extensibility options to provide security functions to SOAP. WSS specifications secure the SOAP foundation layer by leveraging core technologies such as W3C XML Signature, W3C XML Encryption, W3C XML Canonicalization, and TLS. WSS adds security to SOAP messages by specifying how the header part of the message can be used to carry along security information.

In WSS 1.0 [b-OASIS WSS 1.0], the OASIS specification provides the underlying foundation for SOAP message level security. [b-OASIS WSS 1.0] develops mechanisms for identifying the origin of a message and verifying tampering through the use of signatures. [b-OASIS WSS 1.0] provides mechanisms for message integrity by ensuring that only the intended recipient is able to see the message through the use of encryption. [b-OASIS WSS 1.0] introduces a security header to a SOAP message and three key elements:

– Tokens: SOAP messages can contain security tokens with authentication information such as Username tokens, X.509 tokens, SAML tokens, among others. These tokens can be part of security headers and can vouch for security claims to recipients.

– Signature elements: Security headers can contain signature elements that can sign any part of the message. The recipient can use the signature to verify that the sender's request has not been changed and that the message really originated from the sender.

– Encryption elements: Some parts of the SOAP message can be encrypted to protect sensitive information from unauthorized entities.

[b-OASIS WSS 1.1] enhances [b-OASIS WSS 1.0] with additional mechanisms to convey token information (e.g., sending a thumbprint of an X.509, or a SHA1 of an encrypted key previously available with cooperation between parties). [b-OASIS WSS 1.1] introduces the concept of `SignatureConfirmation` that enables a communication sender to confirm that the received message was generated in response to a message it initiated in its unaltered form. In this technique, the recipient sends back the signature values received from the sender in the `SignatureConfirmation` element. This technique helps to prevent certain forms of reply and message alteration attacks.

[b-OASIS WSS 1.1] became an OASIS standard as of February 1, 2006. The series of web services standards consists of the following set of specifications:

1) WS-Security Core Specification 1.1

2) Username Token Profile 1.1

3) X.509 Token Profile 1.1

4) SAML Token profile 1.1

5) Kerberos Token Profile 1.1

6) Rights Expression Language (REL) Token Profile 1.1

7) SOAP with Attachments (SWA) Profile 1.1

This Recommendation requires security solutions of convergence services to build [b-OASIS WSS 1.1] onto in order to implement security for SOAP messages. Developers can use [b-OASIS WSS 1.1] specification in conjunction with other web services extensions and higher-level application-specific protocols to accommodate a wide variety of security models and security technologies.

Appendix II

Interoperability considerations for web services

(This appendix does not form an integral part of this Recommendation)

This appendix provides an overview of interoperability technologies for web services.

II.1 Interoperability at the SOAP layer

In order for web services to work in a uniform fashion, interoperability at the SOAP, WSDL, UDDI and SOAP message security levels must be ensured.

SOAP and WSDL specifications include a number of ambiguities, inconsistencies and potential errors. Ambiguities in these specifications allow adopters to interpret the intention of the standard. As a result, not all developers interpret the standards in the same way.

Interoperability at the SOAP level is addressed by the WS-I Basic Profile family of specifications. WS-I has developed [b-WS-I BP 1.1] that develops a set of constraints and best practice guidelines that can help developers write interoperable web services. [b-WS-I BP 1.1] provides guidelines for how SOAP; Web Services Description Language; Universal Description, Discovery and Integration 2.0; XML 1.0; and XML Schema [b-W3C XML] should interoperate.

In this appendix, the term BP means both [b-WS-I BP 1.0] and [b-WS-I BP 1.1].

II.1.1 BP usage scenarios

The BP [b-WS-I BP 1.0] [b-WS-I BSP 1.0] discuss three basic message exchange patterns (MEPs).

– One-way: A SOAP message is sent, either directly or through intermediaries, to a SOAP receiver. No response message is returned.

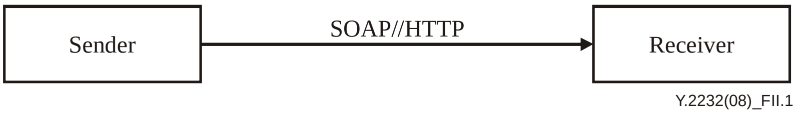

Figure II.1 – One-way exchange pattern

– Synchronous request/response: A SOAP message (the request) is sent, either directly or through intermediaries, to an ultimate SOAP receiver. A SOAP message (the response) is sent by the request's ultimate SOAP receiver through the reverse path followed by the request to the request's initial SOAP sender.

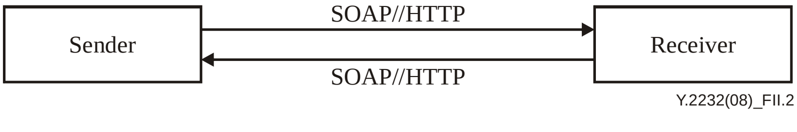

Figure II.2 – Synchronous request/response exchange pattern

– Basic callback: A SOAP message (the request) is sent, either directly or through intermediaries, to an ultimate SOAP receiver, and an acknowledgement message is returned in the form of a synchronous request/response. The request contains information that indicates an endpoint for the SOAP node where the response should be sent. The request's ultimate SOAP receiver sends the response to that SOAP node, which returns an acknowledgement message in the form of synchronous request/response.

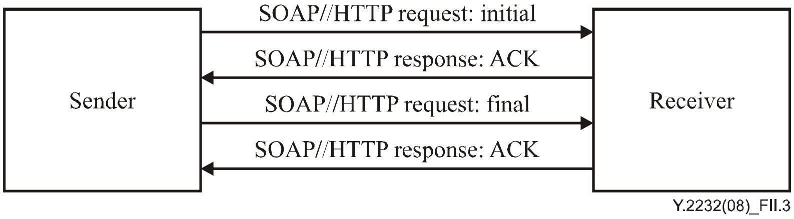

Figure II.3 – Basic callback exchange pattern

II.1.2 BP scope conformance

Conformance to the BP means adherence to the profile's scope and requirements. Requirements are the rules that web services need to adhere to in order to meet the BP standards for interoperability. The basic type is conformance at the level of an artefact. Artefacts are the primary elements of any web service. In general, the BP has rules about the following types of artefacts:

- Messages, which are protocol elements that are exchanged to affect a web service. The scope of conformance is the entire message.
- Descriptions, which in the BP implies descriptions of types, messages, interfaces and their protocol and data format bindings, and the network access points associated with web services. The scope of conformance is `wsdl:port`, or parts of a port.
- REGDATA, which are statements (in UDDI for instance) about web services that are used to discover their capabilities. The scope of conformance is bindingTemplate or tModels.

II.2 Interoperability at the SOAP message security layer

SOAP messages are composed of XML elements. Using WS-Security techniques, these elements may be signed, encrypted, or signed and encrypted. These elements can be referenced from other elements. Each element within a SOAP message may be processed by an intermediary that can add more data and sign and encrypt the incremental data and/or or the combined data.

WS-Security (WSS) provides a framework for securing SOAP massages. The framework is flexible and is designed to accommodate a wide range of security scenarios. As such, it is difficult for various implementations of WS-Security to interoperate properly. For this reason, WS-I has developed the basic security profile (BSP) to provide clarifications and constraints in order to enhance the interoperability of WS-Security implementations. The BSP extends the profiles created by the WS-I Basic Profile (BP) by adding interoperability guidelines for security. [b-WS-I BSP 1.0] and [b-WS-I BSP 1.1] are available on the WS-I public site.

The BSP adds security to the same three basic message exchange patterns (MEPs) that were adopted from the scenarios defined for the basic profile.

II.2.1 BSP strength of requirements

The BSP focuses on improving interoperability by strengthening requirements when possible and constraining flexibility and extensibility when appropriate. The BSP limits the set of common functionality that vendors must implement and thus enhances the chances for interoperability.

The guiding principles enumerated in the BSP declare that there is no guarantee of interoperability, that the profile should "do no harm" and to make statements that are testable when possible. The profile does not address issues in the profiled standards that do not affect interoperability. The BSP committee worked so that enhancing interoperability does not create new security threats.

It is not the intent of the profile to define security best practices. However, when multiple options exist, the profile considers known security weaknesses and makes choices that reduce the risks and that act as a means of reducing choice and thus enhancing interoperability. The profile assumes that interoperability of lower-layer protocols and technologies (e.g., encryption and signature algorithms) are adequate and well-understood. The basic security profile restates selected requirements from the WS-Security errata rather than including the entire errata by reference, preferring interoperability over strict conformance.

The profile includes requirement statements about two kinds of artefacts; SECURE_ENVELOPE and SECURE_MESSAGE. A SECURE_ENVELOPE is a SOAP envelope that has been subjected to integrity and/or confidentiality protection. A SECURE_MESSAGE expands the scope of the SECURE_ENVELOPE to include protocol elements transmitted with the SECURE_ENVELOPE that have been subject to integrity and/or confidentiality protection.

II.2.2 BSP conformance

In order to conform to the BSP, any artefact that contains a construct that is addressed in the profile must conform to any statements that constrain its use. Conformant receivers are not required to accept all possible conformant messages. Conformance applies to deployed instances of services. Since major portions of the BSP may or may not apply in certain circumstances, individual URIs may be used to indicate conformance to parts of the BSP including the core profile or additional sections of the BSP for Username token, X.509 token, and SOAP attachments.

The BSP includes statements that are interoperability requirements as well as statements that are security considerations. The normative requirement statements are identified by numbers prefixed with the letter 'R', for example Raaaa where aaaa is the statement number. These statements contain one requirement level keyword (i.e., "MUST") and one conformance target. The following conformance targets are used in the BSP:

- SECURE_ENVELOPE: A SOAP envelope that contains sub-elements that have been subject to integrity and/or confidentiality protection. A message is considered conformant when all of its contained artefacts are conformant with all statements in the BSP that are related to them. Use of artefacts for which there are no statements in the basic security profile does not affect conformance.
- SECURE_MESSAGE: Protocol elements that have WS-Security applied to them. Protocol elements include a primary SOAP envelope and optionally associated SOAP attachments.
- SENDER: Software that generates a message according to the protocol(s) associated with it. A sender is considered conformant when all of the messages it produces are conformant and its behaviour is conformant with all statements related to SENDER in the BSP.
- RECEIVER: Software that consumes a message according to the protocol(s) associated with it. A receiver is considered conformant when it is capable of consuming conformant messages containing the artefacts that it supports and its behaviour is conformant with all statements related to RECEIVER in the BSP.
- INSTANCE: Software that implements a `wsdl:port` or a `uddi:bindingTemplate`.
- SECURITY_HEADER: An element included as a child of `soap:Envelope/soap:Header` and named `wsse:Security`.
- REFERENCE: A SIGNATURE `ds:Reference` element.
- SIGNATURE: An element included as a child of a SECURITY_HEADER and named `ds:Signature`.
- ENCRYPTED_KEY: An element included as a child of a SECURITY_HEADER and named `xenc:EncryptedKey`.

- ENCRYPTION_REFERENCE_LIST: An element which is included as a child of a SECURITY_HEADER and named `xenc:ReferenceList`.
- ENCRYPTED_KEY_REFERENCE_LIST: An element which is included as a child of an ENCRYPTED_KEY and named `xenc:ReferenceList`.
- ENCRYPTED_DATA: An element named `xenc:EncryptedData` which is referenced by either an ENCRYPTED_KEY_REFERENCE_LIST or an ENCRYPTION_REFERENCE_LIST.
- SECURITY_TOKEN_REFERENCE: An element included as a descendant of a SECURITY_HEADER or an ENCRYPTED_DATA and which is named `wsse:SecurityTokenReference`.
- INTERNAL_SECURITY_TOKEN: A security token defined in a security token profile and that is either a child of a SECURITY_HEADER or a child of a `wsse:Embedded` element in a SECURITY_TOKEN_REFERENCE.
- EXTERNAL_SECURITY_TOKEN: A security token defined in a security token profile that is external to a SECURE_ENVELOPE.
- SECURITY_TOKEN: Either an INTERNAL_SECURITY_TOKEN or an EXTERNAL_SECURITY_TOKEN (e.g., Username token, X.509 certificate token, REL token, or SAML token).
- TIMESTAMP: An element included as a child of a SECURITY_HEADER and named `wsu:Timestamp`.
- MIME_PART: The MIME-defined header fields and contents of one of the parts in the body of a multipart entity in a SECURE_MESSAGE.

In BSP, certain statements are considered clarifying statements. The intent of these statements is to eliminate confusion about the intended interpretation of a requirement from an underlying specification. Clarifying statements are identified by adding a suffix of a lower case letter 'c', i.e., Rxxxxc, where xxxx is the requirement number. Additional clarifying statements are also identified by numbers prefixed by the letter 'C', i.e., Cyyyy, where yyyy is the statement number. These statements are non-normative and are used to provide clarification in order to eliminate confusion.

Appendix III

Information flows for WSG

(This appendix does not form an integral part of this Recommendation)

This appendix provides information flows for the WSG from clause 7.4 in detail.

III.1 Case 1: Information flows for security pattern

There are two models for security management about the web services gateway: redirect model and proxy model.

When the WSR initiates a request without the security information to a WSP, this request which includes the address of the WSP can be directly sent to the WSP or first sent to the WSG and then be forwarded to a WSP. The WSP then instructs the WSR to be authenticated by the WSG and/or instructs the WSG to prepare to the security mechanism to the request.

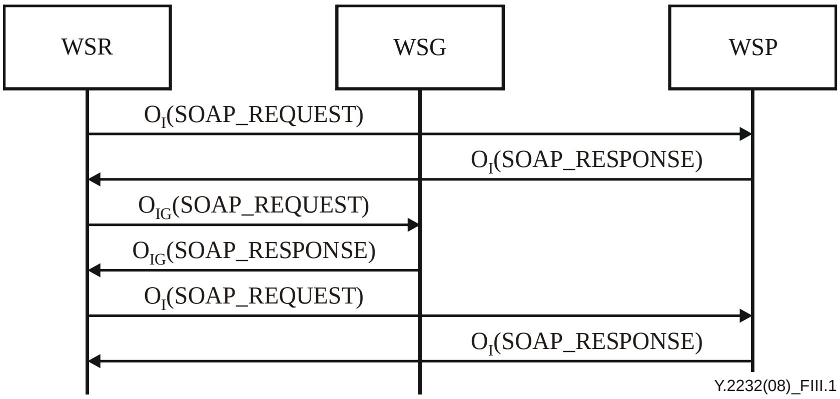

Figure III.1 – Information flow of redirect model for security pattern

In a redirect model, the WSR can send the authentication information to the WSG and can request the WSG to do the authentication. The WSG can authenticate the WSR according to the WSP's own security mechanism or in a single sign on manner. The authentication result is returned to the WSR, then the WSR sends the request carrying the authentication result to the WSP, and this request may be encrypted by the WSR according to the WSP's security mechanism. If the request is encrypted, the WSP can first decrypt the request. The WSP applies the security process to the request, that is checks the authentication result based on its own security mechanism and forms the final authorization result. Then, the WSP can process the request and returns a response to the WSR, and this response may also be encrypted by the WSP.

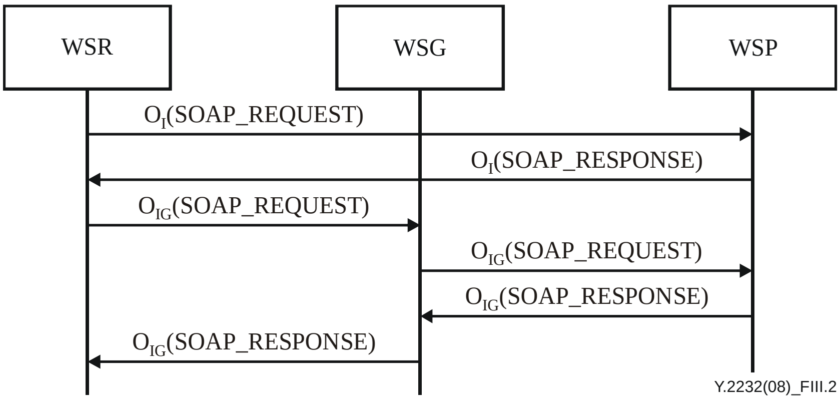

Figure III.2 – Information flow of proxy model for security pattern

In a proxy model, the WSR initiates a request carrying the authentication information and sends it to the WSG. This request may also be encrypted by the WSR according to the WSP's security mechanism or the WSG's security mechanism. If the request is encrypted, the WSG can firstly decrypt the request. The WSG authenticates the request according to the WSP's own security mechanism or in a single sign on manner and then the request is forwarded to the WSP through the transport layer secure link. The WSP processes the request and returns a response, and this response is first sent to the WSG. The WSG may apply some security process to the response, such as encrypting the response, and then forwards the response to the WSR.

III.2 Case 2: Information flows for substitution pattern

The information flow for the substitution pattern is as follows:

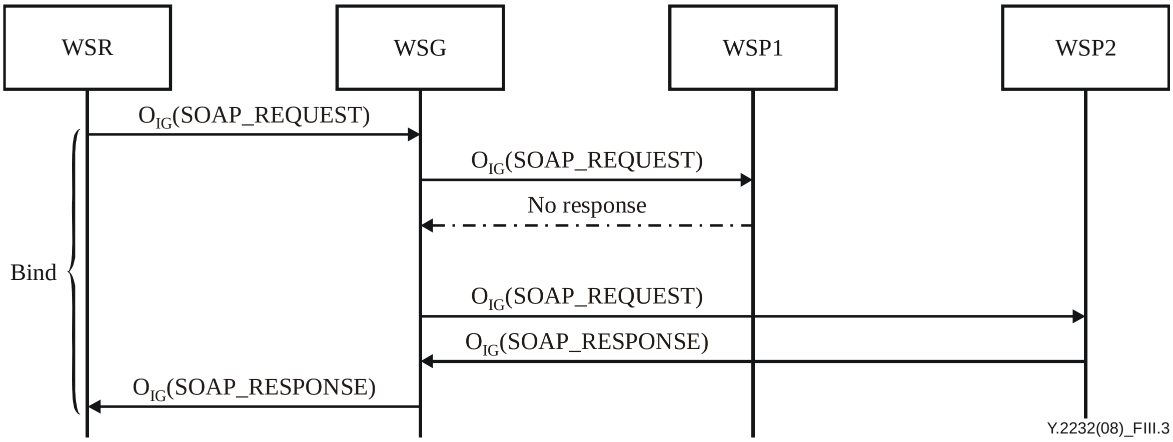

Figure III.3 – Information flow of substitution pattern

The WSR sends a service request via O_{IG} to the WSG and the WSG relays the service request to the WSP1. The WSG does not receive any response or receives a failure response, then the WSG obtains the service template, which maps to more than one service providing the same service function, based on the original WSP1's service information. Based on the obtained service template, the WSG gets another service provided by WSP2 with the same service function. The WSG then uses the new service of WSP2 to replace the original service of WSP1. The WSG sends the service request to WSP2, the WSP then processes the service request and produces the service response. Finally, WSP2 sends the service response to the WSG and the WSG relays this response back to the WSR.

III.3 Case 3: Information flows for composition pattern

There are two models for services composition about the web services gateway: static service composition and dynamic service composition.

In both static service composition and dynamic service composition, the WSG publishes the service description of composition services into the web services registry via an O_{PG} operation. Then the WSR can find the service description via an O_F operation and get the composition services information according to the service descriptions.

The Information flow of concrete service logic of composition pattern is as follows:

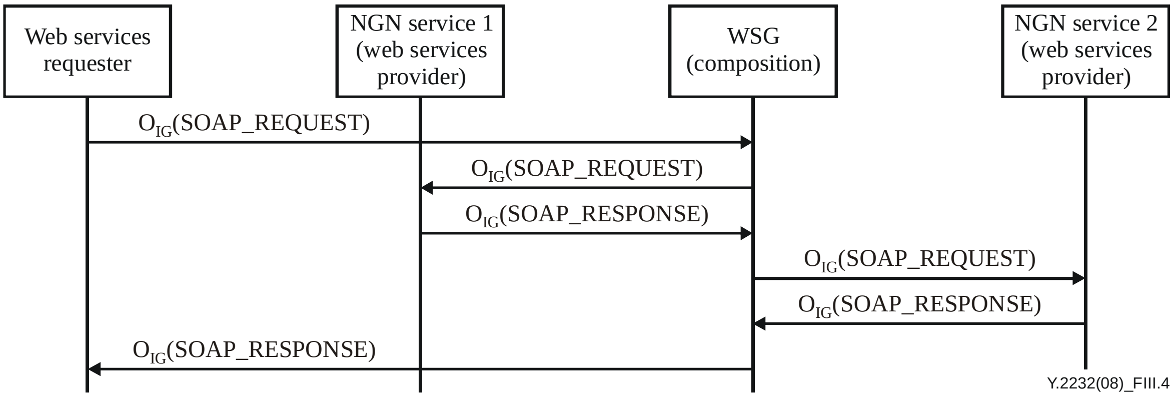

Figure III.4 – Information flow of static service composition pattern

For static service composition, the WSG executes the concrete service logic to invoke the corresponding services and gets the results from them. Finally, the WSG produces the execution result of the composition services and returns it to the WSR.

The information flow of dynamic service composition pattern is as follows:

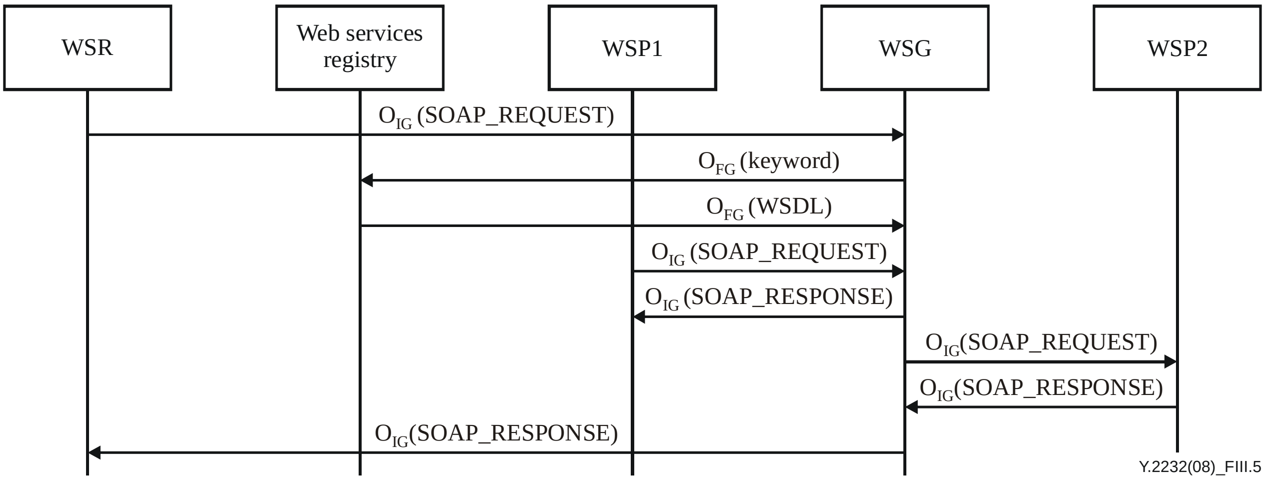

Figure III.5 – Information flow of abstract service logic of composition pattern

For dynamic service composition, according to the abstract service logic of the composition services, the WSG transfers abstract service logic to concrete service logic by finding out the services that can fulfil the requirement to replace the service classes and creates invoking information for these services. Then, the WSG executes the concrete service logic to invoke the corresponding concrete services and returns the final execution result of the composition service to the WSR.

Bibliography

[b-ITU-T H.310] Recommendation ITU-T H.310 (1998), *Broadband audiovisual communication systems and terminals.*

[ITU-T X.509] Recommendation ITU-T X.509 (2000) | ISO/IEC 9594-8:2001, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.*

[b-ITU-T X.811] Recommendation ITU-T X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework.*

[b-ITU-T X.812] Recommendation ITU-T X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework.*

[b-ITU-T X.1141] Recommendation ITU-T X.1141 (2006), *Security Assertion Markup Language (SAML 2.0).*

[b-ITU-T X.1142] Recommendation ITU-T X.1142 (2006), *eXtensible Access Control Markup Language (XACML 2.0).*

[b-ITU-T Y.110] Recommendation ITU-T Y.110 (1998), *Global Information Infrastructure principles and framework architecture.*

[b-ITU-T Y.120] Recommendation ITU-T Y.120 (1998), *Global Information Infrastructure scenario methodology.*

[b-ITU-T Y.1251] Recommendation ITU-T Y.1251 (2002), *General architectural model for interworking.*

[b-ITU-T Y.2001] Recommendation ITU-T Y.2001 (2004), *General overview of NGN.*

[b-ITU-T Y.2011] Recommendation ITU-T Y. 2011 (2004), *General principles and general reference model for Next Generation Networks.*

[b-ITU-T Z.100] Recommendation ITU-T Z.100 (2002), *Specification and Description Language (SDL).*

[b-ITU-T Y.2000-Sup.1] ITU-T Y.2000-series Recommendations – Supplement 1 (2006), *ITU-T Y.2000 series – Supplement on NGN release 1 scope.*

[b-IETF 2828] IETF RFC 2828 (2000), *Internet Security Glossary.* <http://www.ietf.org/rfc/rfc2828.txt>

[b-IETF 4346] IETF RFC 4346 (2006), *The Transport Layer Security (TLS) Protocol Version 1.1.* <http://www.ietf.org/rfc/rfc4346.txt>

[b-OASIS WSS 1.0] OASIS Web Services Security v1.0 (2004), *Web Services Security: SOAP Message Security 1.0, (WS-Security 2004).* <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>

[b-OASIS WSS 1.1] OASIS Web Services Security v1.1 (2006), *Web Services Security: SOAP Message Security 1.1 (WS-Security 2004).* <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>

[b-W3C XML] W3C Recommendation (2004), *XML Schema Part 0: Primer Second Edition.* <http://www.w3.org/TR/xmlschema-0/>

[b-WS-I BP 1.0]	WS-I Basic Profile Version 1.0 (2004), *WS-I Basic Profile Version 1.0.* <http://www.ws-i.org/Profiles/BasicProfile-1.0.html>
[b-WS-I BP 1.1]	WS-I Basic Profile Version 1.1 (2006), *WS-I Basic Profile Version 1.1.* <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>
[b-WS-I BSP 1.0]	WS-I Basic Security Profile Version 1.0 (2007), *Basic Security Profile Version 1.0.* <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects and next-generation networks**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2008