

# SOAP Notification Interface Description

Ericsson Service-Aware Policy Controller

## INTERWORK DESCRIPTION

## **Copyright**

© Ericsson España, S.A. 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

## **Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

## **Trademark List**

**Ericsson** is the trademark or registered trademark of Telefonaktiebolaget LM Ericsson.

All other product or service names mentioned in this document are trademarks of their respective companies.



# Contents

<b>1</b>	<b>SOAP Notification Interface Overview</b>	<b>1</b>
<b>2</b>	<b>SOAP Protocol</b>	<b>1</b>
2.1	SOAP Message Structure	1
<b>3</b>	<b>Notification WSDL Data Structures</b>	<b>2</b>
3.1	Envelope XML Elements	2
3.2	Header XML Elements	2
3.3	Body XML Elements	2
<b>4</b>	<b>SOAP Bindings</b>	<b>7</b>
4.1	SOAP HTTP Request	7
4.2	SOAP HTTP Answer	7
<b>5</b>	<b>SOAP Notification Error Handling</b>	<b>8</b>
<b>6</b>	<b>XML Schemas and WSDL</b>	<b>9</b>
<b>7</b>	<b>SOAP Notification Message Examples</b>	<b>10</b>
	<b>Reference List</b>	<b>13</b>





# 1 SOAP Notification Interface Overview

The SOAP interface offered by the SAPC to other network elements (such as an external database) is used to receive notifications about data changes when the SAPC subscriber data is stored only in an external database.

External database notifies the SAPC about changes of the SAPC subscriber-related data held in the external database by using the SOAP notification interface provided by the SAPC.

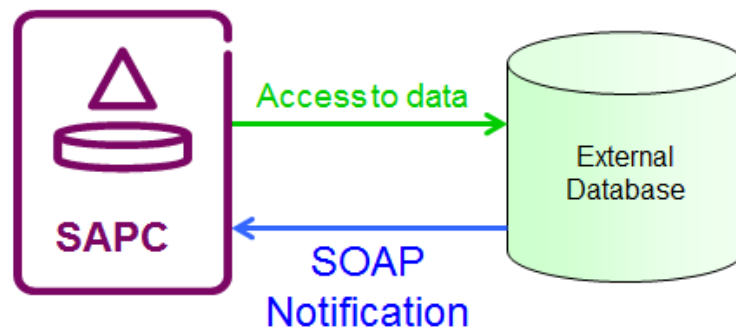


Figure 1 The SAPC SOAP Notification from External Database

The notification interface offered by the SAPC is based on Simple Object Access Protocol (SOAP), Reference [1].

The transport protocol used to carry the SOAP notification messages is the Hypertext Transfer Protocol (HTTP), Reference [2].

## 2 SOAP Protocol

SOAP is a communication protocol between applications that provides a simple and lightweight mechanism for exchanging structured and typed information in a decentralized and distributed environment based on XML.

### 2.1 SOAP Message Structure

A SOAP message is an XML document that consists of an optional SOAP header and a mandatory SOAP body, within an overall mandatory SOAP envelope.



- SOAP Envelope: Root element of the message. It identifies the XML document as a SOAP message.
- SOAP Header: Mechanism used to add features to a SOAP message in a decentralized manner without prior agreement between the communicating parties.
- SOAP Body: It is a container for mandatory information intended for the ultimate recipient of the message. Inside the SOAP Body, a SOAP Fault element can be included to carry error and status information.

## 3 Notification WSDL Data Structures

This section describes the XML data structures used.

**Note:** XML elements and attributes are case-sensitive.

### 3.1 Envelope XML Elements

There is no specific information to be included in the SOAP Envelope but the specified in Reference [1].

### 3.2 Header XML Elements

No specific treatment of the SOAP Header element is defined for notifications towards the SAPC. If received, the SAPC shall behave as specified in Reference [1].

### 3.3 Body XML Elements

#### 3.3.1 Notification Request Messages

The root of the SOAP Body for Notification Request messages is a notify XML element, composed of the following XML elements:

- notificationEvent
- additionalInformation
- modificationInformation

**Note:** There is also an Extension XML element for future purposes, but the SAPC does not use it. It is ignored if received.



Next figure shows the main information elements:

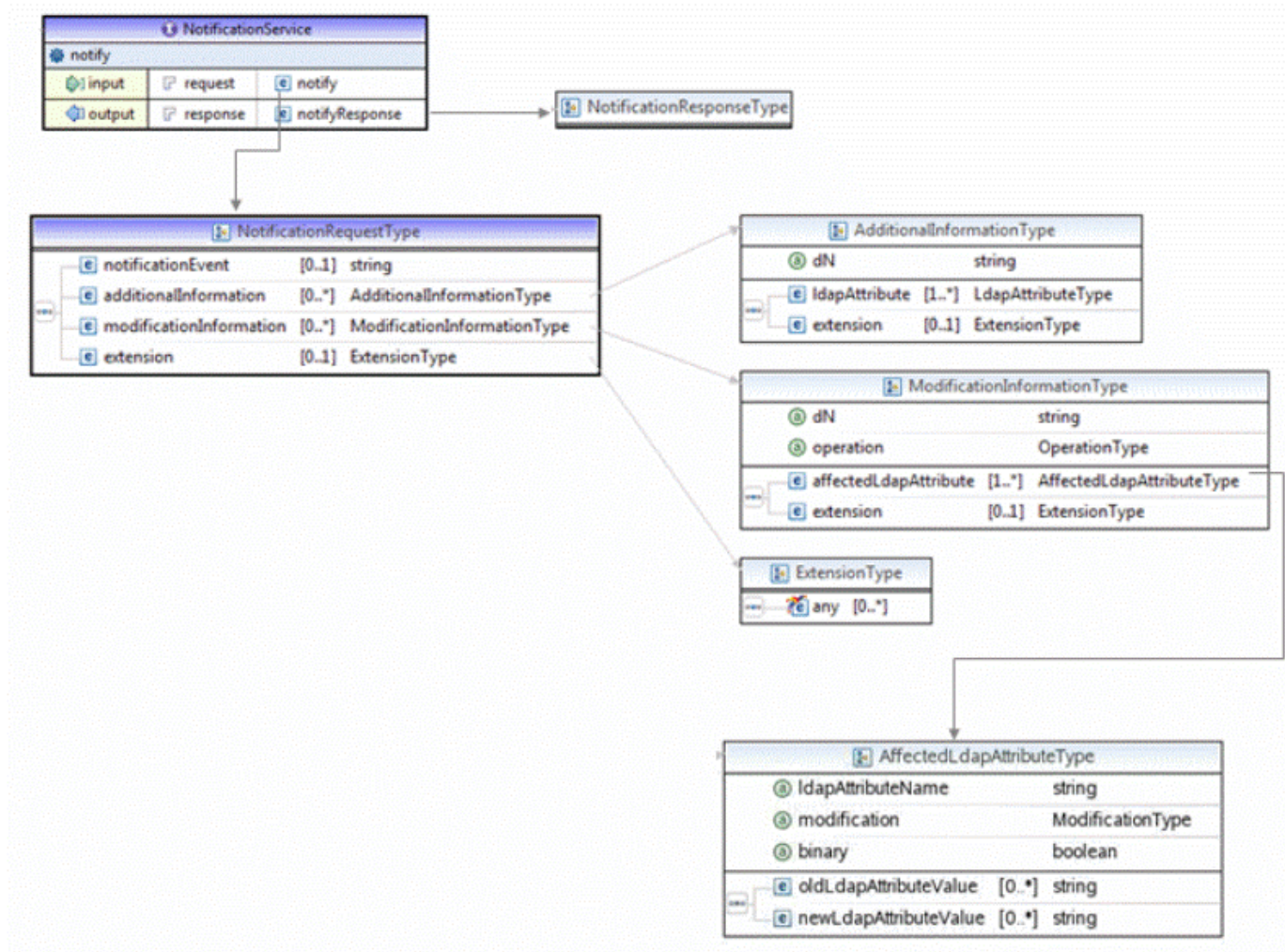


Figure 2 NotificationService Data Elements

### 3.3.1.1 notificationEvent

It is an XML string that indicates the purpose of the notification. It is mandatory in SOAP notifications towards the SAPC.

The SAPC supports the following values:

- `provisioningEvent` to indicate that the SOAP notification is triggered by a provisioning event

### 3.3.1.2 additionalInformation

It is a structure providing relevant information for the handling of the notification. It is optional and can be used to convey for example identifiers and location



information of the subscriber or users subject of the notification event. There may be multiple instances of this XML element present in the notification request with the restriction that all the instances refer to the same subscriber.

It is composed of series of `ldapAttribute` XML elements and the XML attributes shown in the following table:

Table 1 `additionalInformation` XML Attributes

XMLAttributes	Type	Required	Description
<b>dN</b>	String	Mandatory	It contains the provisioned Distinguished Name of the LDAP entry

#### 3.3.1.2.1

#### `ldapAttribute`

It is a structure representing an LDAP attribute related to the subscriber or users subject of the notification event.

The following table contains the XML attributes that can be set in it:

Table 2 `ldapAttribute` XML Attributes

XMLAttributes	Type	Required	Description
<b>ldapAttributeName</b>	String	Mandatory	It contains the name of the LDAP attribute.
<b>binary</b>	Boolean	Optional	For binary LDAP attributes (the attribute contains binary values coded in base64), its value is "true" or "yes"  Default value (when it is not present) is "false" or "no"

`ldapAttribute` is composed of a series of `ldapAttributeValue` XML elements.

#### `ldapAttributeValue`

XML string containing the value of the attribute. For multivalued LDAP attributes, there can be a series of `ldapAttributeValue`.

#### 3.3.1.3

#### `modificationInformation`

It is a structure representing the LDAP object instance whose modification triggered the notification event.

It is mandatory for `ProvisioningEvent` notifications.





If the SAPC receives multiple instances of this element, the SAPC processes them in the received order. If the processing of all them works fine, the SAPC returns success. In case one of them results in a failure, the SAPC returns an error (Section 5 on page 8), and the SAPC does not continue processing the rest of instances.

It consists of a series of affectedLdapAttributes XML elements and the XML attributes shown in the following table:

Table 3 modificationInformation XML Attributes

XML Attributes	Type	Required	Description
<b>dn</b>	String	Mandatory	It stores the Distinguished Name of the LDAP object  If the DN is not associated to a subscriber, the SAPC returns an error (see Section 5 on page 8).
<b>operation</b>	String	Optional	It identifies the LDAP operation triggering the notification.  Values: <ul style="list-style-type: none"> <li>• create</li> <li>• modify</li> <li>• delete</li> </ul> If this attribute is not present, the SAPC assumes modify value.  In Ericsson UDC solution, the SAPC decides the value for this attribute using the procedure explained in <i>Integration in User Data Consolidation</i> .

#### 3.3.1.3.1 affectedLdapAttribute

It is a structure representing the LDAP attribute whose modification triggered the event. The following table contains the XML attributes that can be set in it:



Table 4 affectedLdapAttribute XML Attributes

XML Attributes	Type	Required	Description
<b>ldapAttributeName</b>	String	Mandatory	It stores the name of the LDAP attribute
<b>modification</b>	String	Optional	It identifies the type of modification.  Values: <ul style="list-style-type: none"><li>• add</li><li>• remove</li><li>• replace</li></ul>

affectedLdapAttribute consists of the ordered sequence of:

- a series of oldLdapAttributeValue XML elements
- a series of newLdapAttributeValue XML elements

For multivalued LDAP attributes, a series of oldLdapAttributeValue and/or newLdapAttributeValue may be present including all the attribute values before and after the event which triggered the notification.

Table 5 affectedLdapAttribute XML Elements

XML Element	Description
<b>oldLdapAttributeValue</b>	XML string storing the attribute value before the modification took place. It is only applicable for an LDAP modify or an LDAP delete.
<b>newLdapAttributeValue</b>	XML string storing the attribute value after the modification took place. It is only applicable for an LDAP modify or an LDAP add.

### 3.3.2 SOAP Notification Response Messages

For successful SOAP Notification Response messages, the root of the SOAP body is an empty notificationResponse XML element.

For unsuccessful SOAP Notification Response messages, the root of the SOAP body is a SOAP Fault XML element with the corresponding SOAP Fault Code as defined in Reference [1].



## 4 SOAP Bindings

The transport protocol used to carry SOAP messages is the Hypertext Transfer Protocol (HTTP). The HTTP request/response message model is followed as defined in Reference [2], providing SOAP request parameters in an HTTP request and SOAP response parameters in HTTP response. The only supported binding to carry the SOAP messages is the HTTP POST message request and its answer.

The HTTP content-type entity header must be always set to “text/xml”.

The SOAPAction HTTP request header field is received as part of the HTTP request.

SOAP HTTP follows the semantics of the HTTP Status codes defined in Reference [2] for communicating status information in HTTP.

### 4.1 SOAP HTTP Request

The general format of the HTTP request is the following:

```
Request=      Request-Method= POST
              Request URI: requested-web-service
              Request Version: HTTP/1.1
              Content-Length: <length>
              Content-Type: text/xml
              < message-body>
```

requested-web-service contains the Uniform Resource Locator (URL)-path of the SOAP server managing the notifications request. It is the following:

```
<VIP for Ext-DB>:8080/sapc/v1/notifications
```

For more information about the possible HTTP headers, see Reference [2].

The information to include in the different HTTP headers to carry a SOAP message in the HTTP request is specified in Reference [1]. The SOAPAction HTTP request header field (specific for SOAP) is defined as mandatory in Reference [1], however the SAPC ignores its value.

### 4.2 SOAP HTTP Answer

The general format of the HTTP answer is the following:



<Response>::= HTTP/1.1  
Response Code: <Result>  
Content-Type: text/xml  
< message-body >

For more information about the possible HTTP headers to be defined in the request, see Reference [2].

The message body contains a SOAP message including the envelope and the body. If a header was present in the request, the SOAP message of the SOAP HTTP answer also contains a header.

The following HTTP Status codes defined in Reference [2] with their corresponding Reason-Phrase can be sent by the SAPC:

- **200 OK**, when the operation was successful.
- **500 Internal Server Error** when the operation was not successful (even for client faults at application level). After sending this response the SAPC closes the HTTP connection. See Section 5 on page 8.

Other Status Codes related to HTTP protocol errors could be also sent by the SAPC, for more information see Reference [2]. In these cases, no SOAP message is included in the response.

## 5 SOAP Notification Error Handling

The SAPC can return the following errors within SOAP Responses:

Table 6 Application Errors

SOAP faultcode	SOAP faultstring	SOAP detail	Description
soap:Client	Error from client	-	An invalid SOAP Request is received.
soap:Client	Error from client	NotificationEventType '<event>' not supported	The value received in notificationEvent is not supported.
soap:Client	Error from client	Unknown Object in dN='<dn>'	Unknown object received within dN.



SOAP faultcode	SOAP faultstring	SOAP detail	Description
soap:Server	Error from server	Notification request is received, but reauthorizations are disabled.	The sending of subscriber reauthorizations to the traffic gateway is disabled in the SAPC.
soap:Server	Error from server	Server overloaded	The SAPC is overloaded. The SOAP notification request does not proceed.

## 6 XML Schemas and WSDL

The WSDL including the XML schemas defined for the SOAP notification service, can be get in the following URL:

<VIP for Ext-DB>:8080/sapc/v1/notifications?wsdl



## 7

### Example of Notification SOAP Request

[illegible]

### Example of Successful SOAP Response returned by the SAPC

[illegible]



Example of Erroneous SOAP Response returned by the SAPC

<?xml version='1.0' encoding='UTF-8'>  
  
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'>  
  
 <SOAP-ENV:Header/>  
  
 <SOAP-ENV:Body/>  
  
 <ns1:Fault xmlns:ns1='http://schemas.xmlsoap.org/soap/envelope/'>  
  
 <faultcode>SOAP-ENV:Client</faultcode>  
  
 <faultstring>SOAP-ENV:Client</faultstring>  
  
 <detail/>  
  
 </ns1:Fault>  
  
 </SOAP-ENV:Body>  
  
</SOAP-ENV:Envelope>

SOAP HTTP Answer Example

<?xml version='1.0' encoding='UTF-8'>  
  
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'>  
  
 <SOAP-ENV:Header/>  
  
 <SOAP-ENV:Body/>  
  
 <ns1:Fault xmlns:ns1='http://schemas.xmlsoap.org/soap/envelope/'>  
  
 <faultcode>SOAP-ENV:Client</faultcode>  
  
 <faultstring>SOAP-ENV:Client</faultstring>  
  
 <detail/>  
  
 </ns1:Fault>  
  
 </SOAP-ENV:Body>  
  
</SOAP-ENV:Envelope>







## Reference List

### Standards

- [1] Simple Object Access Protocol (SOAP) 1.1, 2000–08, W3C SOAP/1.1 Note
- [2] Hypertext Transfer Protocol -- HTTP/1.1, 1999-06, RFC 2616
- [3] UTF-8, a transformation format of ISO 10646, 1998-01, RFC 2279
- [4] Lightweight Directory Access Protocol (LDAP): Directory Information Models, 2006-06, RFC 4512
- [5] The Base16, Base32, and Base64 Data Encodings, 2006-10, RFC 4648
- [6] W3C XML Schema Definition Language (XSDL) 1.1 Part 1: Structures, 2007-30, W3C Working Draft