

# Asynchronous CAI3G Interface Specification 1.2

Ericsson Dynamic Activation 1

---

SYSTEM INTERFACE

**Copyright**

© Ericsson AB 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**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose and Scope	1
1.2	Target Groups	1
1.3	Typographic Conventions	1
1.4	Prerequisites	1
<b>2</b>	<b>Async CAI3G Overview</b>	<b>2</b>
<b>3</b>	<b>Async CAI3G Definition</b>	<b>5</b>
3.1	Authentication	5
3.2	Correlation Requests and Notifications	6
3.3	Asynchronous Parameters	7
3.4	Request Tracing Support	8
3.5	Transactions	9
3.6	Security	9
3.7	Acknowledgement	9
3.8	Generating WSDL File	11
3.8.1	Request WSDL File	11
3.8.2	Notification WSDL File	12
3.9	Operation	14
3.9.1	Create Operation	14
3.9.2	Get Operation	16
3.9.3	Set Operation	17
3.9.4	Delete Operation	18
<b>4</b>	<b>Appendix A Async CAI3G WSDL</b>	<b>20</b>
4.1	Asynchronous Common WSDL	20
4.2	Fault WSDL	21
4.3	Request WSDL Template	22
4.4	Response WSDL Template	24
4.5	Intermediate Notification Template	26
<b>5</b>	<b>Appendix B Async CAI3G Examples</b>	<b>29</b>
5.1	Create Operation	29
5.1.1	Request Phase	29
5.1.2	Notification Phase	31
5.2	Get Operation	33



5.2.1	Request Phase	33
5.2.2	Notification Phase	34
5.3	Set Operation	35
5.3.1	Request Phase	35
5.3.2	Notification Phase	36
5.4	Delete Operation	36
5.4.1	Request Phase	36
5.4.2	Notification Phase	37
<b>Reference List</b>		<b>39</b>



# 1 Introduction

The Asynchronous Customer Administration Interface Third Generation (Async CAI3G) 1.2 defines an asynchronous Web Service interface based on CAI3G which is a synchronous web service interface. This document describes how to provide Asynchronous Web Service for Simple Object Access Protocol (SOAP) over HTTP/HTTPS in Ericsson™ Dynamic Activation (EDA).

## 1.1 Purpose and Scope

This document provides the information of using Async CAI3G 1.2, based on the well-known CAI3G 1.2 specification, to send the asynchronous orders to Dynamic Activation.

## 1.2 Target Groups

The target groups for this document are as follows:

- System Integrator
- Business Logic Developer

## 1.3 Typographic Conventions

Typographic conventions are described in the document *Library Overview*, Reference [1].

## 1.4 Prerequisites

It is assumed that the readers/users of this document:

- Is familiar with Dynamic Activation
- Is familiar with CAI3G



## 2 Async CAI3G Overview

The Async CAI3G interface is based on the CAI3G interface. For information about CAI3G interface, refer to *Generic CAI3G Interface 1.2*, Reference [2]. Figure 1 shows the Async CAI3G Protocol Stack.

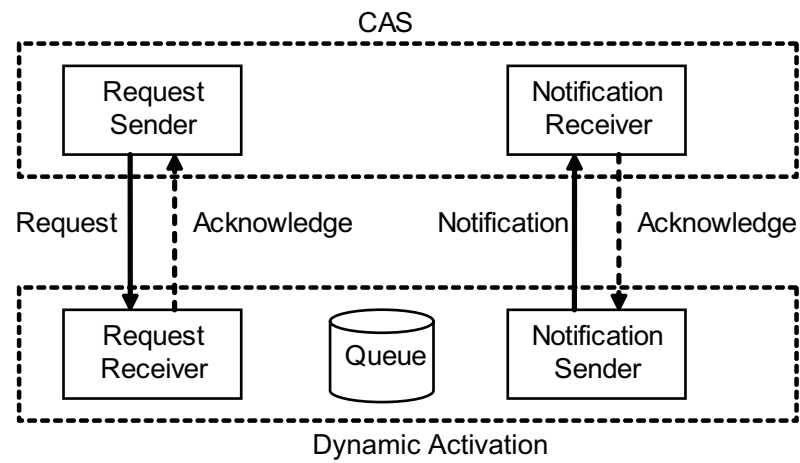
Async Scheduling
WS-Security/WS-Addressing
CAI3G1.2
SOAP
HTTP/HTTPS

Figure 1 Async CAI3G Protocol Stack

Compared to CAI3G, Async CAI3G:

- Is an asynchronous interface while CAI3G is synchronous interface.
- Uses WS-Addressing to correlate request and notification (the execution result).
- Introduces extra optional parameters for asynchronous purpose.
- Use WS-Security as authentication mechanism while CAI3G also supports login-logout authentication.
- Supports SOAP over HTTP/HTTPs while CAI3G supports SOAP over HTTP/HTTPs/JMS.
- Does not support the `Notification` service in CAI3G.
- The Search operation is not supported in the current release.

Figure 2 shows the Async CAI3G interface:



*Figure 2 Async CAI3G Interface*

The two logical parts of CAS: `RequestSender` and `NotificationReceiver`. The two logical parts of Dynamic Activation are `RequestReceiver` and `NotificationSender`.

There are two phases in the provisioning process: request phase and notification phase.

Figure 3 shows the interaction process in request phase between EDA and CAS

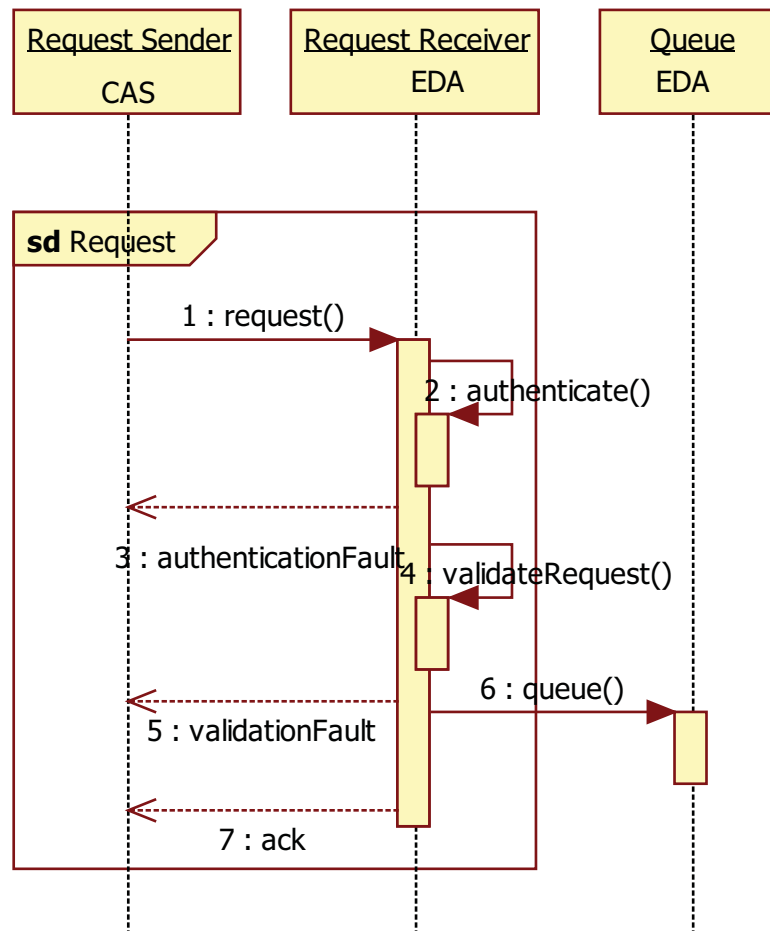


Figure 3 Async CAI3G Request Sequence Diagram

1. CAS constructs SOAP request according to the WSDL provided by Dynamic Activation. CAS sends to the request to Dynamic Activation synchronously.
2. Dynamic Activation authenticates the request.
3. Dynamic Activation returns authentication fault to CAS if the authentication fails.
4. Dynamic Activation validates the request.
5. Dynamic Activation returns validation fault to CAS if the request format is not correct.
6. Dynamic Activation queues the request.
7. Dynamic Activation returns acknowledgement back to the CAS.

Figure 4 shows the interaction process in notification phase between EDA and CAS



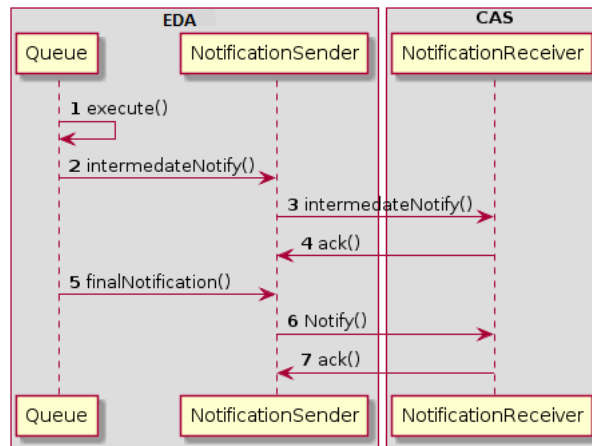


Figure 4 Async CAI3G Notification Sequence Diagram

1. Dynamic Activation starts to process the request according to certain scheduling rule.
2. The queue of Dynamic Activation sends intermediate notification to Notification Sender if matches the notification rule.
3. Notification Sender sends intermediate notification to CAS.
4. CAS returns the acknowledgement back to Dynamic Activation.
5. The queue of Dynamic Activation sends final notification to Notification Sender.
6. Notification Sender sends final notification to CAS.
7. CAS returns the acknowledgement back to Dynamic Activation.

## 3 Async CAI3G Definition

### 3.1 Authentication

Async CAI3G uses WS-Security as authentication mechanism. Only username and password are used to authenticate the request. The attribute `mustUnderstand` defined in WS-Security must not be specified. Other attributes or elements are ignored. Example 1 shows how to use authentications in request.

```
<soapenv:Header>
  <wsse:Security xmlns:wsse=
    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
    <wsse:UsernameToken>
      <wsse:Username>sogadm</wsse:Username>
      <wsse:Password>sogadm</wsse:Password>
    </wsse:UsernameToken>
    </wsse:Security>
</soapenv:Header>
```

### *Example 1 Use Authentications in Request*

The notifications sent from Dynamic Activation to CAS do not support any authentication method.

## 3.2 Correlation Requests and Notifications

Async CAI3G uses WS-Addressing to correlation request with notification.

Only the following SOAP headers are used in Async CAI3G requests. The attribute `mustUnderstand` defined in WS-Security must not be specified. Other SOAP headers defined WS-Addressing are ignored.

- **MessageID:** It is used to identify asynchronous CAI3G request and the corresponding notification. It is provided by CAS. This element is mandatory when `ReplyTo` or `FaultTo` is applied.

**Note:** `messageId` cannot be reused in the short time, it will cause confusion to identify the asynchronous request execution status. CAS must ensure the uniqueness of the `messageId`.

- **ReplyTo:** It is used to specify the address of CAS where the reply is sent to by Dynamic Activation. Only HTTP/HTTPs endpoint is supported by Async CAI3G. No response is sent from Dynamic Activation received if the value is not specified.
- **FaultTo:** It is used to specify address of CAS where the fault is sent to by Dynamic Activation. Only HTTP/HTTPs endpoint is supported by Async CAI3G. No fault is sent from Dynamic Activation if the value is not specified.
- **Timestamp:** It is used to record the time that Dynamic Activation sends the notification back to CAS.

Example 2 shows a part of how to correlation requests and notifications.

```
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <wsa:MessageID>uuid:6B29FC40-CA47-1067-B31D-00DD010662DA</wsa:MessageID>
  <wsa:ReplyTo>
    <wsa:Address>http://business456.example/client1</wsa:Address>
  </wsa:ReplyTo>
</soapenv:Header>
```

### *Example 2 Correlation Requests and Notifications*



Only the following SOAP headers are used in Async CAI3G responses. Other SOAP headers defined WS-Addressing are ignored.

- **RelatesTo:** Correlate request and notification for CAS. The MessageID in the notification is sent as **RelatesTo** in response and fault by Dynamic Activation.

Example 3 shows part of the notification for the previous request.

```
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <wsa:RelatesTo>uuid:6B29FC40-CA47-1067-B31D-00DD010662DA</wsa: RelatesTo>
</soapenv:Header>
```

### Example 3 Notification

## 3.3 Asynchronous Parameters

The following SOAP headers defined in the namespace `http://schemas.ericsson.com/async/` support the asynchronous functions:

**Table 1** Asynchronous Parameters

Parameter	O/M	Description	Value
Priority <sup>(1)</sup>	O	It indicates the priority of the request.	[1, 10]  The default value is 5.  The default value is used when the priority has not been set in the request or the request does not match the configured priority rule.



Parameter	O/M	Description	Value
FireTime	O	<p>It indicates the scheduled time of the request.</p> <p>When using the FireTime parameter, avoid scheduling a large amount requests provisioning in a narrow time range. Because it will lead to backend Cassandra I/O hotspot, and then slow down the overall system performance.</p>	<p><code>&lt;yyyy-mm-dd&gt;T&lt;hh:mm:ss&gt;</code></p> <p>The request is executed immediately if not specified.</p>
GroupId	O	<p>It indicates the GroupId of requests.</p> <p>It aims to guarantee the execution order of the requests having the same GroupId.</p> <p>Using GroupId slows down the concurrent provisioning throughput, as the requests having the same GroupId is executed in a single node in serial order. When the node is rebooted, other nodes will try to take over the rest requests, in such situation the execution order can not be guaranteed</p>	<p>The type of the value is String. If it is not specified, a default GroupId is generated by Dynamic Activation according to the MOId.<sup>(2)</sup></p> <p>For example, msisdN=18688889999:imsi=12345678</p>

(1) It is not supported in Dynamic Activation 7.0.

(2) -1 is reserved by system. If a request has no GroupId, "-1" will be used to mark the non-GroupId request.

Example 4 shows a part of how to use asynchronous parameters.

```
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:async="http://schemas.ericsson.com/async/">
  <async:FireTime>2012-05-30T09:00:00</async:FireTime>
  <async:GroupId>MSISDN=18988889999</async:GroupId>
</soapenv:Header>
```

#### Example 4 Use Asynchronous Parameters

## 3.4 Request Tracing Support

The following SOAP header defined in the namespace `http://schemas.ericsson.com/async/` supports the request tracing function:

- **Context:** This parameter is reserved to store the specified identifier assigned by Customer Administration System (CAS) or Business Support System (BSS). It provides the possibility to trace the provisioning requests from CAS or BSS by the specified identifier. This specified identifier can be the service order ID, the service package ID, the customer ID, the contract ID and so on.

Example 5 shows a part of how to use request tracing support parameter.



```
<soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:async="http://schemas.ericsson.com/async/" xmlns:cai3="http://schemas.ericsson.com/cai3/"
<cai3:Context>SHLTE12865973</cai3:Context >
</soapenv:Header>
```

#### Example 5 Use Request Tracing Support Parameter

## 3.5 Transactions

Transactions defined in CAI3G are not supported in Async CAI3G.

## 3.6 Security

Use HTTPs transportation layer security in Async CAI3G.

## 3.7 Acknowledgement

Acknowledgement messages show that the request/notification messages have been received.

The following example shows an acknowledgement message:

```
<message name="async:AsyncAck">
  <part name="ackCode" element="xs:integer" />
  <part name="ackMessage" element="xs:string" />
</message>
```

The following table lists the Acknowledgement codes. The `ackCode` is also logged as the `ResponseCode` of notification in Processing Log.

Table 2 Acknowledgement Code

ackCode	ackMessage
0	The request was queued successfully  The notification was received successfully
5001	The endpoint for <code>ReplyTo</code> in the SOAP header is not correct.  Only HTTP/HTTPs endpoints are supported. The format is <code>https://CAS.example/notify</code> .



ackCode	ackMessage
5002	<p>The endpoint for <code>FaultTo</code> in SOAP header is not correct.</p> <p>Only HTTP/HTTPS endpoints are supported. The format is <code>http://CAS.errorhandler.example/notify</code>.</p>
5003	<p>The parameter <code>MessageID</code> is mandatory when <code>ReplyTo</code> or <code>FaultTo</code> is used.</p>
5004	<p>The format of the parameter <code>Priority</code> is not correct.</p> <p>The parameter <code>Priority</code> can have the following values: High, Normal or Low.</p>
5005	<p>The format of the parameter <code>FireTime</code> is not correct.</p> <p>The correct format is <code>&lt;yyyy-MM-dd&gt;T&lt;HH:mm:ss&gt;</code>.</p>
5006	<p><code>GroupId</code> is presented, but the value is empty.</p>
5007	<p>The endpoint for <code>ReplyTo</code> in the SOAP header is not correct.</p> <p>URL protocol is unacceptable.</p> <p>Only HTTP/HTTPS endpoints are supported. The format is <code>https://CAS.example/notify</code>.</p>
5008	<p>The endpoint for <code>FaultTo</code> in the SOAP header is not correct.</p> <p>URL protocol is unacceptable.</p> <p>Only HTTP/HTTPS endpoints are supported. The format is <code>https://CAS.example/notify</code>.</p>
6003	<p>The scheduled request is not to be planned to be executed earlier than the current time.</p>
6005	<p>Duplicated <code>MessageID</code> <code>create-sub-1234567-4567890</code>.<sup>(1)</sup></p>
6006	<p>The request failed to be added in the queue.</p>



ackCode	ackMessage
6007	Receiver has stopped. Requests sent during this period is not to be processed.
6008	License is not available.

(1) *create-sub-1234567-4567890* is the input value provided by the customer.

If no acknowledgement message is received, the following codes are logged as the ResponseCode of notification in Processing Log.

**Table 3 Response Code**

Response Code	Response Message
6009	Failed to send notification for northbound request.
6010	Failed to send intermediate notification for sub-request.

## 3.8 Generating WSDL File

This section describes how to generate a specific Async CAI3G WSDL file for specific CAI3G MO based on the specific synchronous CAI3G WSDL file and Async CAI3G WSDL template file.

### 3.8.1 Request WSDL File

The Async CAI3G 1.2 request WSDL template shows in Section 4.3 on page 22.

Do the following to generate the Async CAI3G request WSDL file for specific CAI3G MO:

1. Replace the WSDL location (`location="cai3g1.2_specific_mo.wsdl"` in the example below) with filename of the specific CAI3G MO WSDL (for example, `location="hlr.wsdl"`).

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:cai3g="http://schemas.ericsson.com/cai3g1.2/"
  targetNamespace="http://schemas.ericsson.com/async/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:async="http://schemas.ericsson.com/async/">

  <import namespace="http://schemas.ericsson.com/cai3g1.2/"
    location="cai3g1.2_specific_mo.wsdl" />
</definitions>
```

2. Remove the unsupported operations by the specific CAI3G MO in the port type.



For example, in the template below, if the `Create` operation is not supported, remove `<operation name="Create"><!-- ... --></operation>`.

```
<portType name="AsyncProvisioningRequest">
  <operation name="Create">
    <!-- ... -->
  </operation>
  <operation name="Delete">
    <!-- ... -->
  </operation>
  <operation name="Get">
    <!-- ... -->
  </operation>
  <operation name="Set">
    <!-- ... -->
  </operation>
</portType>
```

3. Remove the unsupported operation by the specific CAI3G MO in the binding type.

For example, in the template below, if the `Create` operation is not supported, remove `<operation name="Create"><!-- ... --></operation>`.

```
<binding name="AsyncProvisioningRequest" type="AsyncProvisioningRequest">
  <operation name="Create">
    <!-- ... -->
  </operation>
  <operation name="Delete">
    <!-- ... -->
  </operation>
  <operation name="Get">
    <!-- ... -->
  </operation>
  <operation name="Set">
    <!-- ... -->
  </operation>
</binding>
```

### 3.8.2 Notification WSDL File

The Async CAI3G 1.2 notification WSDL template shows in Section 4.4 on page 24.

Do the following to generate the Async CAI3G Notification WSDL file for specific CAI3G MO:

1. Replace the WSDL location (`location="cai3g1.2_specific_mo.wsdl"` in the example below) with filename of the specific CAI3G MO WSDL (for example, `location="hlr.wsdl"`).





```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:cai3g="http://schemas.ericsson.com/cai3g1.2/"
  targetNamespace="http://schemas.ericsson.com/async/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:async="http://schemas.ericsson.com/async/">

  <import namespace="http://schemas.ericsson.com/cai3g1.2/"
    location="cai3g1.2_specific_mo.wsdl" />
</definitions>
```

2. Remove the unsupported operations by the specific CAI3G MO in the port type.

For example, in the template below, if the CreateResponse operation is not supported, remove `<operation name="CreateResponse"><!-- ... --></operation>`.

```
<portType name="AsyncProvisioningNotification">
  <operation name="CreateResponse">
    <!-- ... -->
  </operation>
  <operation name="DeleteResponse">
    <!-- ... -->
  </operation>
  <operation name="GetResponse">
    <!-- ... -->
  </operation>
  <operation name="SetResponse">
    <!-- ... -->
  </operation>
  <operation name="FaultResponse">
    <!-- ... -->
  </operation>
</portType>
```

3. Remove the unsupported operation by the specific CAI3G MO in the binding type.

For example, in the template below, if the CreateResponse operation is not supported, remove `<operation name="CreateResponse"><!-- ... --></operation>`.

```
<binding name="AsyncProvisioningNotification" type="AsyncProvisioningNotification">
  <operation name="CreateResponse">
    <!-- ... -->
  </operation>
  <operation name="DeleteResponse">
    <!-- ... -->
  </operation>
  <operation name="GetResponse">
    <!-- ... -->
  </operation>
  <operation name="SetResponse">
    <!-- ... -->
  </operation>
  <operation name="FaultResponse">
    <!-- ... -->
  </operation>
</binding>
```



## 3.9 Operation

### 3.9.1 Create Operation

#### 3.9.1.1 Request Phase

The structure of this pseudo-method looks like:

CREATE(IN MOType, [IN MOId], [IN MOAttributes], [IN extension], OUT AsyncAck)

#### Request

Example 6 shows a part of the request format from CAS to Dynamic Activation:

```
<Create xmlns="http://schemas.ericsson.com/cai3g1.2/">
  <MOType>
    Subscription@http://schemas.ericsson.com/ma/auc/
  </MOType>
  <MOId>
    <imsi>1234567890</imsi>
  </MOId>
  <MOAttributes>
    <CreateSubscription imsi="1234567890"
      xmlns="http://schemas.ericsson.com/ma/auc/">
      <imsi>1234567890</imsi>
      <eki>AB34567890ABCDEF1234567890ABCDEF</eki>
      <kind>511</kind>
      <fsetind>15</fsetind>
      <amf>65535</amf>
      <a4ind>7</a4ind>
    </CreateSubscription>
  </MOAttributes>
</Create>
```

#### *Example 6 Request*

#### Acknowledgement

If the request format follows the CAI3G specification, the request can be authenticated. Example 7 shows a part of the acknowledgement format from Dynamic Activation to CAS:

```
<AsyncAck xmlns="http://schemas.ericsson.com/async/">
  <ackCode>0</ackCode>
  <ackMessage>The request was queued successfully</ackMessage>
</AsyncAck>
```

#### *Example 7 Acknowledgement*

#### Fault

The format follows the CAI3G specification. If the request format does not follow the CAI3G specification, the request cannot be authenticated. Example 8 shows a part of the fault format from Dynamic Activation to CAS:



```
<Cai3gFault xmlns="http://schemas.ericsson.com/cai3g1.2/">
  <faultcode>1103</faultcode>
  <faultreason>
    <reasonText>License error.</reasonText>
  </faultreason>
  <faultrole>MF</faultrole>
  <details/>
</Cai3gFault>
```

#### *Example 8 Fault*

### 3.9.1.2 Notification Phase

The structure of this pseudo-method looks like:

CREATERESPONSE(IN MOType, IN MOId, [IN MOAttributes], [IN extension],  
OUT AsyncAck)

#### **Response**

Example 9 shows a part of the response format from Dynamic Activation to CAS when the request is executed successfully:

```
<CreateResponse xmlns="http://schemas.ericsson.com/cai3g1.2/">
  <MOId>
    <imsi>1234567890</imsi>
  </MOId>
</CreateResponse>
```

#### *Example 9 Response*

#### **Fault**

Example 10 shows a part of the fault format from Dynamic Activation to CAS when the request is not executed successfully:

```
<Cai3gFault xmlns="http://schemas.ericsson.com/cai3g1.2/">
  <faultcode>2001</faultcode>
  <faultreason>
    <reasonText>Invalid managed object type.</reasonText>
  </faultreason>
  <faultrole>MF</faultrole>
  <details>
    <UDCFault xmlns="http://schemas.ericsson.com/ma/auc/">
      <errorcode>1001</errorcode>
      <errormessage>Invalid resource ID.</errormessage>
      <errordetails>Failed to read resource ID. \
        IMSI is not found.
      </errordetails>
    </UDCFault>
  </details>
</Cai3gFault>
```

#### *Example 10 Response*

#### **Acknowledgement**

Example 7 shows a part of the acknowledge format from CAS to Dynamic Activation when the notification is received:

## 3.9.2 Get Operation

### 3.9.2.1 Request Phase

The structure of this pseudo-method looks like:

GET(IN MOType, [IN MOId], [IN MOAttributes], [IN extension], OUT AsyncAck)

#### Request

Example 11 shows a part of the request format from CAS to Dynamic Activation:

```
<Get xmlns="http://schemas.ericsson.com/ma/auc/">
  <MOType>
    Subscription@http://schemas.ericsson.com/ma/auc/
  </MOType>
  <MOId>
    <imsi>1234567890</imsi>
  </MOId>
</Get>
```

*Example 11 Request*

#### Acknowledgement

If the request format follows the CAI3G specification, the request can be authenticated. Example 7 shows a part of the acknowledgement format from Dynamic Activation to CAS.

#### Fault

The format follows the CAI3G specification. If the request format does not follow the CAI3G specification, the request cannot be authenticated. Example 8 shows a part of the acknowledgement format from Dynamic Activation to CAS.

### 3.9.2.2 Notification Phase

The structure of this pseudo-method looks like:

GETRESPONSE(IN MOType, [IN MOId], [IN MOAttributes], [IN extension],  
OUT AsyncAck)

#### Response

Example 12 shows a part of the response format from Dynamic Activation to CAS when the request is executed successfully:



```
<GetResponse xmlns="http://schemas.ericsson.com/cai3g1.2/">
  <MOAttributes>
    <GetResponseSubscription imsi="1234567890"
      xmlns="http://schemas.ericsson.com/ma/auc/">
      <imsi>1234567890</imsi>
      <eki>AB34567890ABCDEF1234567890ABCDEF</eki>
      <kind>511</kind>
      <fsetind>15</fsetind>
      <a4ind>7</a4ind>
      <amf>65535</amf>
      <akatype>1</akatype>
      <akaalgind>1</akaalgind>
    </GetResponseSubscription>
  </MOAttributes>
</GetResponse>
```

### *Example 12 Response*

## **Fault**

Example 10 shows a part of the fault format from Dynamic Activation to CAS when the request is not executed successfully.

## **Acknowledgement**

Example 7 shows a part of the fault format from CAS to Dynamic Activation when the notification is received.

## **3.9.3 Set Operation**

### **3.9.3.1 Request Phase**

The structure of this pseudo-method looks like:

SET(IN MOType, [IN MOId], [IN MOAttributes], [IN extension], OUT AsyncAck)

## **Request**

Example 13 shows a part of the request format from CAS to Dynamic Activation:

```
<Set xmlns="http://schemas.ericsson.com/cai3g1.2/">
  <MOType>
    Subscription@http://schemas.ericsson.com/ma/auc/
  </MOType>
  <MOId>
    <imsi>1234567890</imsi>
  </MOId>
  <MOAttributes>
    <SetSubscription imsi="1234567890"
      xmlns="http://schemas.ericsson.com/ma/auc/">
      <akaalgind>1</akaalgind>
    </SetSubscription>
  </MOAttributes>
</Set>
```

### *Example 13 Request*

### **Acknowledgement**

If the request format follows the CAI3G specification, the request can be authenticated. Example 7 shows a part of the acknowledgement format from Dynamic Activation to CAS.

### **Fault**

The format follows the CAI3G specification. If the request format does not follow the CAI3G specification, the request cannot be authenticated. Example 8 shows a part of the fault format from Dynamic Activation to CAS.

## **3.9.3.2 Notification Phase**

The structure of this pseudo-method looks like:

SETRESPONSE(IN MOType, [IN MOAttributes], [IN extension], OUT AsyncAck)

### **Response**

Example 14 shows a part of the response format from Dynamic Activation to CAS when the request is executed successfully:

```
<SetResponse xmlns="http://schemas.ericsson.com/cai3g1.2/" />
```

*Example 14 Response*

### **Fault**

Example 10 shows a part of the fault format from Dynamic Activation to CAS when the request is not executed successfully.

### **Acknowledgement**

Example 7 shows a part of the fault format from CAS to Dynamic Activation when the notification is received.

## **3.9.4 Delete Operation**

### **3.9.4.1 Request Phase**

The structure of this pseudo-method looks like:

DELETE(IN MOType, IN MOId, [IN MOAttributes], [IN extension], OUT AsyncAck)



## Request

Example 15 shows a part of the request format from CAS to Dynamic Activation:

```
<Delete xmlns="http://schemas.ericsson.com/cai3g1.2/">
  <MOType>
    Subscription@http://schemas.ericsson.com/ma/auc/
  </MOType>
  <MOId>
    <imsi>1234567890</imsi>
  </MOId>
</Delete>
```

*Example 15 Request*

## Acknowledgement

If the request format follows the CAI3G specification, the request can be authenticated. Example 7 shows a part of the acknowledgement format from Dynamic Activation to CAS.

## Fault

The format follows the CAI3G specification. If the request format does not follow the CAI3G specification, the request cannot be authenticated. Example 8 shows a part of the fault format from Dynamic Activation to CAS.

### 3.9.4.2

## Notification Phase

The structure of this pseudo-method looks like:

```
DELETERESPONSE(IN MOType, [IN MOId], [IN MOAttributes], [IN extension],
OUT AsyncAck)
```

## Response

Example 16 shows a part of the response format from Dynamic Activation to CAS when the request is executed successfully:

```
<DeleteResponse xmlns="http://schemas.ericsson.com/cai3g1.2/" />
```

*Example 16 Response*

## Fault

Example 10 shows a part of the fault format from Dynamic Activation to CAS when the request is not executed successfully.

## Acknowledgement

Example 7 shows a part of the fault format from CAS to Dynamic Activation when the notification is received.



## 4 Appendix A Async CAI3G WSDL

### 4.1 Asynchronous Common WSDL

Example 17 defines the Asynchronous Common WSDL that are imported by the Request WSDL and the Notification WSDL.





```

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="http://schemas.ericsson.com/async/"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:async="http://schemas.ericsson.com/async/" xmlns:cai3g="http://schemas.ericsson.com/cai3g1.2/">

  <types>
    <xs:schema>
      <xs:import namespace="http://www.w3.org/2005/08/addressing" schemaLocation="ws-addr.xsd" />
    </xs:schema>
    <xs:schema targetNamespace="http://schemas.ericsson.com/async/" xmlns="http://schemas.ericsson.com/async/">
      <xs:element name="Priority">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="^10|[1-9]$" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="FireTime">
        <xs:simpleType>
          <xs:restriction base="xs:dateTime" />
        </xs:simpleType>
      </xs:element>
      <xs:element name="GroupId">
        <xs:simpleType>
          <xs:restriction base="xs:string" />
        </xs:simpleType>
      </xs:element>
      <xs:element name="AsyncAck">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ackCode" type="xs:integer" />
            <xs:element name="ackMessage" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </types>

  <message name="AsyncAck">
    <part name="ack" element="async:AsyncAck" />
  </message>

  <message name="AsyncRequestHeadInfo">
    <part name="priority" element="async:Priority" />
    <part name="fireTime" element="async:FireTime" />
    <part name="groupId" element="async:GroupId" />
    <part name="context" element="cai3g:Context" />
    <part name="messageId" element="wsa:MessageID" />
    <part name="replyTo" element="wsa:ReplyTo" />
    <part name="faultTo" element="wsa:FaultTo" />
  </message>

  <message name="AsyncNotificationHeadInfo">
    <part name="Timestamp" element="ns1:Timestamp"
      xmlns:ns1="http://schemas.xmlsoap.org/ws/2002/07/utility"/>
    <part name="relatesTo" element="wsa:RelatesTo" />
    <part name="context" element="cai3g:Context" />
  </message>
</definitions>

```

### Example 17 Asynchronous Common WSDL

## 4.2 Fault WSDL

Example 18 defines the Fault message.



```

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:async="http://schemas.ericsson.com/async/"
xmlns:wsa="http://www.w3.org/2005/08/addressing" targetNamespace="http://schemas.xmlsoap.org/soap/envelope/">
  <import namespace="http://schemas.ericsson.com/async/" location="async_common.wsdl"/>
  <types>
    <xs:schema>
      <xs:import namespace="http://schemas.xmlsoap.org/soap/envelope/" schemaLocation="soapenv.xsd"/>
    </xs:schema>
  </types>
  <message name="Fault">
    <part name="fault" element="soapenv:Fault"/>
  </message>
  <portType name="AsyncProvisioningFault">
    <operation name="Fault">
      <input message="soapenv:Fault"/>
      <output message="async:AsyncAck"/>
    </operation>
  </portType>
  <binding name="AsyncProvisioningFault" type="soapenv:AsyncProvisioningFault">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsaw:UsingAddressing wsdl:required="false" xmlns:wsdl="http://www.w3.org/2006/01/wsdl"/>
    <operation name="Fault">
      <soap:operation soapAction="Fault" style="document"/>
      <input>
        <soap:body use="literal"/>
        <soap:header message="async:AsyncNotificationHeadInfo" part="relatesTo" use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="AsyncProvisioningFaultService">
    <port name="AsyncProvisioningFault" binding="soapenv:AsyncProvisioningFault">
      <soap:address location="http://anyema.anyprovisioningprovider.com/cai3g1.2/async"/>
    </port>
  </service>
  <jaxws:bindings xmlns:jaxws="http://java.sun.com/xml/ns/jaxws">
    <!-- disable wrapper style generation -->
    <jaxws:enableWrapperStyle>false</jaxws:enableWrapperStyle>
  </jaxws:bindings>
</definitions>

```

### Example 18 Fault WSDL

## 4.3 Request WSDL Template

### Request WSDL Definition

```

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:cai3g="http://schemas.ericsson.com/cai3g1.2/"
targetNamespace="http://schemas.ericsson.com/cai3g1.2/async/"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:async="http://schemas.ericsson.com/async/"
xmlns:asynccai3g="http://schemas.ericsson.com/cai3g1.2/async/">

  <jaxws:bindings xmlns:jaxws="http://java.sun.com/xml/ns/jaxws">
    <!-- disable wrapper style generation -->
    <jaxws:enableWrapperStyle>false</jaxws:enableWrapperStyle>
  </jaxws:bindings>

  <import namespace="http://schemas.ericsson.com/async/" location="async_common.wsdl" />

  <import namespace="http://schemas.ericsson.com/cai3g1.2/" location="cai3g1.2_.wsdl" />

```



```

<portType name="AsyncProvisioningRequest">
  <!-- supported asynchronous operations -->
  <operation name="Create">
    <input message="cai3g:CreateRequest" />
    <output message="async:AsyncAck" />
    <fault name="Cai3gFault" message="cai3g:Cai3gFault" />
  </operation>
  <operation name="Delete">
    <input message="cai3g:DeleteRequest" />
    <output message="async:AsyncAck" />
    <fault name="Cai3gFault" message="cai3g:Cai3gFault" />
  </operation>
  <operation name="Get">
    <input message="cai3g:GetRequest" />
    <output message="async:AsyncAck" />
    <fault name="Cai3gFault" message="cai3g:Cai3gFault" />
  </operation>
  <operation name="Set">
    <input message="cai3g:SetRequest" />
    <output message="async:AsyncAck" />
    <fault name="Cai3gFault" message="cai3g:Cai3gFault" />
  </operation>
  <operation name="Search">
    <input message="cai3g:SearchRequest" />
    <output message="async:AsyncAck" />
    <fault name="Cai3gFault" message="cai3g:Cai3gFault" />
  </operation>
</portType>

<binding name="AsyncProvisioningRequest" type="asynccai3g:AsyncProvisioningRequest">
  <wsaw:UsingAddressing wsdl:required="false" xmlns:wsdl="http://www.w3.org/2006/01/wsdl" />
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="Create">
    <soap:operation soapAction="CAI3G#Create" style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="async:AsyncRequestHeadInfo" part="priority" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="fireTime" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="groupId" use="literal" minOccurs="0" />
      <soap:header message="cai3g:HeadInfo" part="context" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="messageId" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="replyTo" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="faultTo" use="literal" minOccurs="0" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
    <fault name="Cai3gFault">
      <soap:fault name="Cai3gFault" use="literal" />
    </fault>
  </operation>
  <operation name="Delete">
    <soap:operation soapAction="CAI3G#Delete" style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="async:AsyncRequestHeadInfo" part="priority" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="fireTime" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="groupId" use="literal" minOccurs="0" />
      <soap:header message="cai3g:HeadInfo" part="context" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="messageId" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="replyTo" use="literal" minOccurs="0" />
      <soap:header message="async:AsyncRequestHeadInfo" part="faultTo" use="literal" minOccurs="0" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
    <fault name="Cai3gFault">
      <soap:fault name="Cai3gFault" use="literal" />
    </fault>
  </operation>
  <operation name="Get">
    <soap:operation soapAction="CAI3G#Get" style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="async:AsyncRequestHeadInfo" part="priority" use="literal" minOccurs="0" />

```



```

        <soap:header message="async:AsyncRequestHeadInfo" part="fireTime" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="groupId" use="literal" minOccurs="0" />
        <soap:header message="cai3g:HeadInfo" part="context" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="messageId" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="replyTo" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="faultTo" use="literal" minOccurs="0" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="Cai3gFault">
        <soap:fault name="Cai3gFault" use="literal" />
    </fault>
</operation>
<operation name="Set">
    <soap:operation soapAction="CAI3G#Set" style="document" />
    <input>
        <soap:body use="literal" />
        <soap:header message="async:AsyncRequestHeadInfo" part="priority" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="fireTime" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="groupId" use="literal" minOccurs="0" />
        <soap:header message="cai3g:HeadInfo" part="context" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="messageId" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="replyTo" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="faultTo" use="literal" minOccurs="0" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="Cai3gFault">
        <soap:fault name="Cai3gFault" use="literal" />
    </fault>
</operation>
<operation name="Search">
    <soap:operation soapAction="CAI3G#Search" style="document" />
    <input>
        <soap:body use="literal" />
        <soap:header message="async:AsyncRequestHeadInfo" part="priority" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="fireTime" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="groupId" use="literal" minOccurs="0" />
        <soap:header message="cai3g:HeadInfo" part="context" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="messageId" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="replyTo" use="literal" minOccurs="0" />
        <soap:header message="async:AsyncRequestHeadInfo" part="faultTo" use="literal" minOccurs="0" />
    </input>
    <output>
        <soap:body use="literal" />
    </output>
    <fault name="Cai3gFault">
        <soap:fault name="Cai3gFault" use="literal" />
    </fault>
</operation>
</binding>

<service name="AsyncProvisioningRequestService">
    <port name="AsyncProvisioningRequest" binding="asnccai3g:AsyncProvisioningRequest">
        <soap:address location="http://anyema.anyprovisioningprovider.com/cai3g1.2/async" />
    </port>
</service>
</definitions>

```

### Example 19 Request WSDL

## 4.4 Response WSDL Template

### Response WSDL Definition

```

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:cai3g="http://schemas.ericsson.com/cai3g1.2/" targetNamespace="http://schemas.ericsson.com/async/"

```



```

xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl/" xmlns:wsa="http://www.w3.org/2005/08/addressing/"
xmlns:async="http://schemas.ericsson.com/async/">

<import namespace="http://schemas.ericsson.com/cai3g1.2/"
  location="cai3g1.2_specific_mo.wsdl" />

<import namespace="http://schemas.ericsson.com/async/" location="async_common.wsdl" />

<portType name="AsyncProvisioningResponse">
  <operation name="CreateResponse">
    <input message="cai3g:CreateResponse" />
    <output message="async:AsyncAck" />
  </operation>
  <operation name="DeleteResponse">
    <input message="cai3g:DeleteResponse" />
    <output message="async:AsyncAck" />
  </operation>
  <operation name="SetResponse">
    <input message="cai3g:SetResponse" />
    <output message="async:AsyncAck" />
  </operation>
  <operation name="GetResponse">
    <input message="cai3g:GetResponse" />
    <output message="async:AsyncAck" />
  </operation>
</portType>

<binding name="AsyncProvisioningResponse" type="AsyncProvisioningResponse">
  <wsaw:UsingAddressing wsdl:required="false"
    xmlns:wsdl="http://www.w3.org/2006/01/wsdl" />
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />

  <operation name="CreateResponse">
    <soap:operation soapAction="CAI3G#CreateResponse"
      style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="async:AsyncResponseHeadInfo" part="relatesTo"
        use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="DeleteResponse">
    <soap:operation soapAction="CAI3G#DeleteResponse"
      style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="async:AsyncResponseHeadInfo" part="relatesTo"
        use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="SetResponse">
    <soap:operation soapAction="CAI3G#SetResponse" style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="async:AsyncResponseHeadInfo" part="relatesTo"
        use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="GetResponse">
    <soap:operation soapAction="CAI3G#GetResponse" style="document" />
    <input>
      <soap:body use="literal" />
      <soap:header message="async:AsyncResponseHeadInfo" part="relatesTo"
        use="literal" />
    </input>
  </operation>
</binding>

```



```
        </input>
        <output>
          <soap:body use="literal" />
        </output>
      </operation>
    </binding>

    <service name="AsyncProvisioningResponseService">
      <port name="AsyncProvisioningResponse" binding="AsyncProvisioningResponse">
        <soap:address
          location="http://anyema.anyprovisioningprovider.com/cai3g1.2/async" />
      </port>
    </service>
  </definitions>
```

*Example 20 Response WSDL Definition*

## 4.5 Intermediate Notification Template

### Intermediate Notification WSDL Definition



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net.
RI's version is JAX-WS RI 2.2.6b21 svn-revision#12959. -->
<definitions targetNamespace="http://schemas.ericsson.com/async/intermediate"
name="IntermediateNotificationReceiverService" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:tns="http://schemas.ericsson.com/async/intermediate"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://schemas.xmlsoap.org/ws/2002/07/utility"
        schemaLocation="http://schemas.xmlsoap.org/ws/2002/07/utility"/>
    </xsd:schema>
    <xsd:schema>
      <xsd:import namespace="http://schemas.ericsson.com/async/intermediate"
        schemaLocation="IntermediateNotificationReceiverService_schema2.xsd"/>
    </xsd:schema>
  </types>
  <message name="notify">
    <part name="parameters" element="tns:notify"/>
    <part name="Timestamp" element="ns1:Timestamp"
      xmlns:ns1="http://schemas.xmlsoap.org/ws/2002/07/utility"/>
  </message>
  <message name="notifyResponse">
    <part name="result" element="tns:notifyResponse"/>
  </message>
  <portType name="AsyncIntermediateNotificationService">
    <operation name="notify" parameterOrder="parameters Timestamp">
      <input wsam:Action="IntermediateNotification" message="tns:notify"/>
      <output wsam:Action="NotifyResponse" message="tns:notifyResponse"/>
    </operation>
  </portType>
  <binding name="AsyncIntermediateNotificationBinding"
    type="tns:AsyncIntermediateNotificationService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="notify">
      <soap:operation soapAction="IntermediateNotification"/>
      <input>
        <soap:body use="literal" parts="parameters"/>
        <soap:header message="tns:notify" part="Timestamp" use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="IntermediateNotificationReceiverService">
    <port name="AsyncIntermediateNotification"
      binding="tns:AsyncIntermediateNotificationBinding">
      <soap:address location="REPLACE_WITH_ACTUAL_URL"/>
    </port>
  </service>
</definitions>
```

## Intermediate Notification XSD Definition



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" targetNamespace="http://schemas.ericsson.com/async/intermediate"
xmlns:tns="http://schemas.ericsson.com/async/intermediate" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="intermediateNotification" type="tns:intermediateNotification"/>
  <xs:element name="notify" type="tns:notify"/>
  <xs:element name="notifyResponse" type="tns:notifyResponse"/>

  <xs:complexType name="notify">
    <xs:sequence>
      <xs:element ref="tns:intermediateNotification" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="intermediateNotification">
    <xs:sequence>
      <xs:element name="messageId" type="xs:string" minOccurs="0"/>
      <xs:element name="timestamp" type="xs:string" minOccurs="0"/>
      <xs:element name="latest" type="tns:latest" minOccurs="0"/>
      <xs:element name="status" type="tns:status" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="latest">
    <xs:sequence>
      <xs:element name="expired" type="tns:expiredNE" minOccurs="0"/>
      <xs:element name="failed" type="tns:failedNE" minOccurs="0"/>
      <xs:element name="retrying" type="tns:retryingNE" minOccurs="0"/>
      <xs:element name="successful" type="tns:successfulNE" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="expiredNE">
    <xs:sequence>
      <xs:element name="name" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="failedNE">
    <xs:complexContent>
      <xs:extension base="tns:ne">
        <xs:sequence>
          <xs:element name="errorCode" type="xs:string" minOccurs="0"/>
          <xs:element name="errorMessage" type="xs:string" minOccurs="0"/>
          <xs:element name="errorDetails" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="ne">
    <xs:sequence>
      <xs:element name="name" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="retryingNE">
    <xs:complexContent>
      <xs:extension base="tns:ne">
        <xs:sequence>
          <xs:element name="retryTimes" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="successfulNE">
    <xs:complexContent>
      <xs:extension base="tns:ne">
        <xs:sequence/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="status">
    <xs:sequence>
      <xs:element name="failed" type="tns:failedNE" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="successful" type="tns:successfulNE" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="retried" type="tns:retryingNE" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="waiting" type="tns:ne" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="notifyResponse">
    <xs:sequence>
      <xs:element name="return" type="tns:asyncAck" form="qualified" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="asyncAck">
```





## 5 Appendix B Async CAI3G Examples

### 5.1 Create Operation

#### 5.1.1 Request Phase

##### Request

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:cai3="http://schemas.ericsson.com/cai3gl.2/"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>sogadm</wsse:Username>
        <wsse:Password>sogadm</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:MessageID>6B29FC40-CA47-1067-B31D-00DD010662DA</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:FaultTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:FaultTo>
    <async:FireTime>2012-05-30T09:00:00</async:FireTime>
    <async:GroupId>IMSI=1234567890</async:GroupId>
    <cai3:Context>667890123</cai3:Context>
  </Header>
  <Body>
    <Create xmlns="http://schemas.ericsson.com/cai3gl.2/">
      <MOType>
        Subscription@http://schemas.ericsson.com/ma/auc/
      </MOType>
      <MOId>
        <imsi>1234567890</imsi>
      </MOId>
      <MOAttributes>
        <CreateSubscription imsi="1234567890"
          xmlns="http://schemas.ericsson.com/ma/auc/">
          <imsi>1234567890</imsi>
          <eki>AB34567890ABCDEF1234567890ABCDEF</eki>
          <kind>511</kind>
          <fsetind>15</fsetind>
          <amf>65535</amf>
          <a4ind>7</a4ind>
        </CreateSubscription>
      </MOAttributes>
    </Create>
  </Body>
</Envelope>
```

#### Example 21 Request

## Acknowledgement

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns3:AsyncAck xmlns:ns2="http://schemas.ericsson.com/cai3g1.2/"
      xmlns:ns3="http://schemas.ericsson.com/async/"
      xmlns:ns4="http://www.w3.org/2005/08/addressing">
      <ackCode>0</ackCode>
      <ackMessage>The request was queued successfully</ackMessage>
    </ns3:AsyncAck>
  </S:Body>
</S:Envelope>
```

### Example 22 Acknowledgement

## Fault

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:auc="http://schemas.ericsson.com/ma/auc/">
  <Header />
  <Body>
    <Fault>
      <faultcode>soapenv:Client</faultcode>
      <faultstring>This is a client fault</faultstring>
      <detail>
        <Cai3gFault xmlns="http://schemas.ericsson.com/cai3g1.2/">
          <faultcode>2001</faultcode>
          <faultreason>
            <reasonText>Invalid managed object type.</reasonText>
          </faultreason>
          <faultrole>MF</faultrole>
          <details>
            <UDCFault xmlns="http://schemas.ericsson.com/ma/auc/">
              <errorcode>1001</errorcode>
              <errormessage>Invalid resource ID.</errormessage>
              <errordetails>Failed to read resource ID. \
                IMSI is not found.
            </errordetails>
            </UDCFault>
          </details>
        </Cai3gFault>
      </detail>
    </Fault>
  </Body>
</Envelope>
```

### Example 23 Fault



## 5.1.2 Notification Phase

### Response for Intermediate Notification

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:asyncN="http://schemas.ericsson.com/async/intermediate">
  <SOAP-ENV:Header>
    <ns3:Timestamp xmlns:ns3="http://schemas.xmlsoap.org/ws/2002/07/utility">
      <ns3:Created>2016-05-03T17:04:35+0200</ns3:Created>
    </ns3:Timestamp>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <asyncN:notify>
      <intermediateNotification>
        <messageId>37e76f64-57d9-4814-809c-acd7a79e7b6d</messageId>
        <latest>
          <failed>
            <name>Mock_MO</name>
            <errorCode>10014</errorCode>
            <errorMessage>Not OK</errorMessage>
            <errordetails>Create operation failed: External error. </errordetails>
          </failed>
        </latest>
        <status>
          <successful>
            <name>Mock_MO</name>
          </successful>
        </status>
      </intermediateNotification>
    </asyncN:notify>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

#### Example 24 Response for Intermediate Notification

### Response for Final Notification

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <ns2:Timestamp xmlns:ns2="http://schemas.xmlsoap.org/ws/2002/07/utility">
      <ns2:Created>2016-05-02T15:20:38+0200</ns2:Created>
    </ns2:Timestamp>
    <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
      4fb6e45b-5705-4860-bfa0-4ff157666ba1</wsa:RelatesTo>
    <ns2:Context xmlns:ns2="http://schemas.ericsson.com/cai3g1.2/">667890123</ns2:Context>
  </Header>
  <Body>
    <CreateResponse xmlns="http://schemas.ericsson.com/cai3g1.2/">
      <MOId>
        <imsi>1234567890</imsi>
      </MOId>
    </CreateResponse>
  </Body>
</Envelope>
```

#### Example 25 Successful Response

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <ns2:Timestamp xmlns:ns2="http://schemas.xmlsoap.org/ws/2002/07/utility">
      <ns2:Created>2016-05-02T15:20:38+0200</ns2:Created>
    </ns2:Timestamp>
    <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
      4fb6e45b-5705-4860-bfa0-4ff157666ba1</wsa:RelatesTo>
    </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultstring>This is a server fault</faultstring>
      <faultcode>SOAP-ENV:Server</faultcode>
      <detail>
        <Cai3gFault:Cai3gFault xmlns="http://schemas.ericsson.com/cai3g1.2/"
xmlns:Cai3gFault="http://schemas.ericsson.com/cai3g1.2/">
          <Cai3gFault:faultcode>4006</Cai3gFault:faultcode>
          <Cai3gFault:faultreason>
            <Cai3gFault:reasonText>External error.</Cai3gFault:reasonText>
          </Cai3gFault:faultreason>
          <Cai3gFault:faultrole>MF</Cai3gFault:faultrole>
          <details>
            <PGFault:PGFault xmlns="http://schemas.ericsson.com/pg/1.0"
xmlns:PGFault="http://schemas.ericsson.com/pg/1.0">
              <errorcode>10014</errorcode>
              <errormessage>Not OK</errormessage>
              <errordetails>Create operation failed: External error.</errordetails>
            </PGFault:PGFault>
          </details>
        </Cai3gFault:Cai3gFault>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## Example 26 Fault Response

### Acknowledgement

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wssse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:async="http://schemas.ericsson.com/async/">
  <Header />
  <Body>
    <AsyncAck xmlns="http://schemas.ericsson.com/async/">
      <ackCode>0</ackCode>
      <ackMessage>The notification was accepted successfully</ackMessage>
    </AsyncAck>
  </Body>
</Envelope>
```

## Example 27 Acknowledgement



## 5.2 Get Operation

### 5.2.1 Request Phase

#### Request Phase

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>sogadm</wsse:Username>
        <wsse:Password>sogadm</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:MessageID>6B29FC40-CA47-1067-B31D-00DD010662DA
    </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:FaultTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:FaultTo>
    <async:FireTime>2012-05-30T09:00:00</async:FireTime>
    <async:GroupId>IMSI=1234567890</async:GroupId>
  </Header>
  <Body>
    <Get xmlns="http://schemas.ericsson.com/cai3g1.2/">
      <MOType>
        Subscription@http://schemas.ericsson.com/ma/auc/
      </MOType>
      <MOId>
        <imsi>1234567890</imsi>
      </MOId>
    </Get>
  </Body>
</Envelope>
```

**Example 28** Request

#### Acknowledgement

See Example 22.

#### Fault

See Example 23.



## 5.2.2 Notification Phase

### Response

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <wsa:RelatesTo>6B29FC40-CA47-1067-B31D-00DD010662DA
    </wsa:RelatesTo>
  </Header>
  <Body>
    <GetResponse xmlns="http://schemas.ericsson.com/cai3g1.2/">
      <MOAttributes>
        <GetResponseSubscription imsi="1234567890"
          xmlns="http://schemas.ericsson.com/ma/auc/">
          <imsi>1234567890</imsi>
          <eki>AB34567890ABCDEF1234567890ABCDEF</eki>
          <kind>511</kind>
          <fsetind>15</fsetind>
          <a4ind>7</a4ind>
          <amf>65535</amf>
          <akatype>1</akatype>
          <akaalgind>1</akaalgind>
        </GetResponseSubscription>
      </MOAttributes>
    </GetResponse>
  </Body>
</Envelope>
```

*Example 29 Response*

### Acknowledgement

See Example 27.



## 5.3 Set Operation

### 5.3.1 Request Phase

#### Request

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>sogadm</wsse:Username>
        <wsse:Password>sogadm</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:MessageID>6B29FC40-CA47-1067-B31D-00DD010662DA
    </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:FaultTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:FaultTo>
    <async:FireTime>2012-05-30T09:00:00</async:FireTime>
    <async:GroupId>IMSI=1234567890</async:GroupId>
  </Header>
  <Body>
    <Set xmlns="http://schemas.ericsson.com/cai3g1.2/">
      <MOType>
        Subscription@http://schemas.ericsson.com/ma/auc/
      </MOType>
      <MOId>
        <imsi>1234567890</imsi>
      </MOId>
      <MOAttributes>
        <SetSubscription imsi="1234567890"
          xmlns="http://schemas.ericsson.com/ma/auc">
          <akaalgind>1</akaalgind>
        </SetSubscription>
      </MOAttributes>
    </Set>
  </Body>
</Envelope>
```

#### Example 30 Request

#### Acknowledgement

See Example 22.

#### Fault

See Example 23.

## 5.3.2 Notification Phase

### Response

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <wsa:RelatesTo>6B29FC40-CA47-1067-B31D-00DD010662DA
  </wsa:RelatesTo>
  </Header>
  <Body>
    <SetResponse xmlns="http://schemas.ericsson.com/cai3g1.2/" />
  </Body>
</Envelope>
```

#### Example 31 Response

### Acknowledgement

See Example 27.

## 5.4 Delete Operation

### 5.4.1 Request Phase

#### Request

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>sogadm</wsse:Username>
        <wsse:Password>sogadm</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:MessageID>6B29FC40-CA47-1067-B31D-00DD010662DA
  </wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:FaultTo>
      <wsa:Address>http://business456.example/client1</wsa:Address>
    </wsa:FaultTo>
    <async:FireTime>2012-05-30T09:00:00</async:FireTime>
    <async:GroupId>IMSI=1234567890</async:GroupId>
  </Header>
  <Body>
    <Delete xmlns="http://schemas.ericsson.com/cai3g1.2/">
      <MOType>
        Subscription@http://schemas.ericsson.com/ma/auc/
      </MOType>
      <MOId>
        <imsi>1234567890</imsi>
      </MOId>
    </Delete>
  </Body>
</Envelope>
```

#### Example 32 Request





### **Acknowledgement**

See Example 22.

### **Fault**

See Example 23.

## **5.4.2 Notification Phase**

### **Response**

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:async="http://schemas.ericsson.com/async/">
  <Header>
    <wsa:RelatesTo>6B29FC40-CA47-1067-B31D-00DD010662DA
    </wsa:RelatesTo>
  </Header>
  <Body>
    <DeleteResponse xmlns="http://schemas.ericsson.com/cai3g1.2/" />
  </Body>
</Envelope>
```

*Example 33 Response*

### **Acknowledgement**

See Example 27.





## Reference List

- [1] *Library Overview*, 18/1553-CSH 109 628 Uen
- [2] *Generic CAI3G Interface 1.2*, 2/15519-FAY3020003 Uen