PROJECT NAME: SERVICE DELIVERY PLATFORM (SDP)	**VERSION:** 1.1
DOCUMENT NAME: SAG – DEVELOPING SMS SERVICES	
PRODUCT & SERVICE DEVELOPMENT	**DATE:** 3/7/2012

SAG

Developing SMS Services (Parlay X 2.1)

Contents

PROJECT NAME: SERVICE DELIVERY PLATFORM (SDP)	**VERSION:** 1.1
DOCUMENT NAME: SAG – DEVELOPING SMS SERVICES	
PRODUCT & SERVICE DEVELOPMENT	**DATE:** 3/7/2012

Approvals

This document needs below approvals for implementation.

Author(s)	**Name**	**Signature**	
Principal Service Development Engineer	Eric Mokaya		
Reviewers	**Name**	**Signature**	
Senior Manager Service Development	Reginald Tole		
Service Development Engineer	Eneth Kubai		

Revision Log

Revision Number	Revision Date	Revision	Revision made by
1.0	3rd July, 2012	Initial Draft	Eric Mokaya
1.1	4th July, 2012	• Added Section 2: Releasing an SMS or MMS Service • Added Section 5: Appendix for Section 2	Eneth Kubai
1.2	30th July, 2012	• Added linkid parameter specification • Added getReceivedSms sequence diagram	Eric Mokaya

1.3	16th April 2013	• Added service creation guidelines and billing parameters	Eneth Kubai

Abbreviations

SDP- Service delivery platform

MDSP- Mobile Data Service Platform

CP- Content provider

SP- Service provider

SAG- Service Access Gateway

SMS- Short Message Service

SMSC- Short Message Service Center

NE- Network Elements

1 Overview

Applications using specific protocols to access Short Message Service (SMS) functions provided by network elements (NEs) such as the Short Message Service Center (SMSC) are written to programmatically receive and send SMS messages. This approach requires a high degree of network expertise. Alternatively it is possible to use the Parlay/OSA approach, invoking standard interfaces (for example, user interaction or messaging service interfaces) to gain access to SMS capabilities, but these interfaces are usually perceived to be quite complex by IT application developers. Developers/Service Providers (SPs) must have advanced telecommunication skills to use OSA interfaces.

This section describes a Parlay X 2 Web service for sending and receiving SMS messages. The overall scope of this Web service is to provide primitives for application developers to handle SMS messages in a simple way. In fact, using the SMS Web service, application developers can invoke SMS functions without specific telecommunication knowledge.

Short messaging provides operations for sending an SMS message to the network, a polling mechanism for monitoring the delivery status of a sent SMS message, and an asynchronous notification mechanism for notifying the delivery status.

Short messaging also allows an application to receive SMS messages. Both a polling and an asynchronous notification mechanism are available.

2 Releasing an SMS or MMS Service

2.1 Task Description

SMS services include the common SMS service. After subscribing to a common SMS service product, the user can send and receive common SMS from the CP/SP periodically.

2.2 Application Scenario

To provide latest weather information for users, CP/SP operator **Jerry** needs to release an SMS service. After the SMS service is priced, users can order or subscribe to this service product. After successful order or subscription, users receive the weather SMS messages from the CP/SP.

2.3 Prerequisites

- The CP/SP administrator who has the right to access the **Service** menu can perform this task.
- The CP/SP operator who has the right to access the **Service** menu can perform this task.

2.4 Operation Procedure

1. Access the page for releasing an SMS service.
    a. Choose **Service** > **Service Management** from the menu bar.
    b. Choose **Service** > **Common VAS** from the navigation tree.
    c. Choose **Add service** > **Common SMS Service**. The system displays the **Releasing the Common SMS Service** page.
2. Release a SMS service.

    Configure information about the service to be released, as shown in Figure 1. See appendix for more information about what each field means.

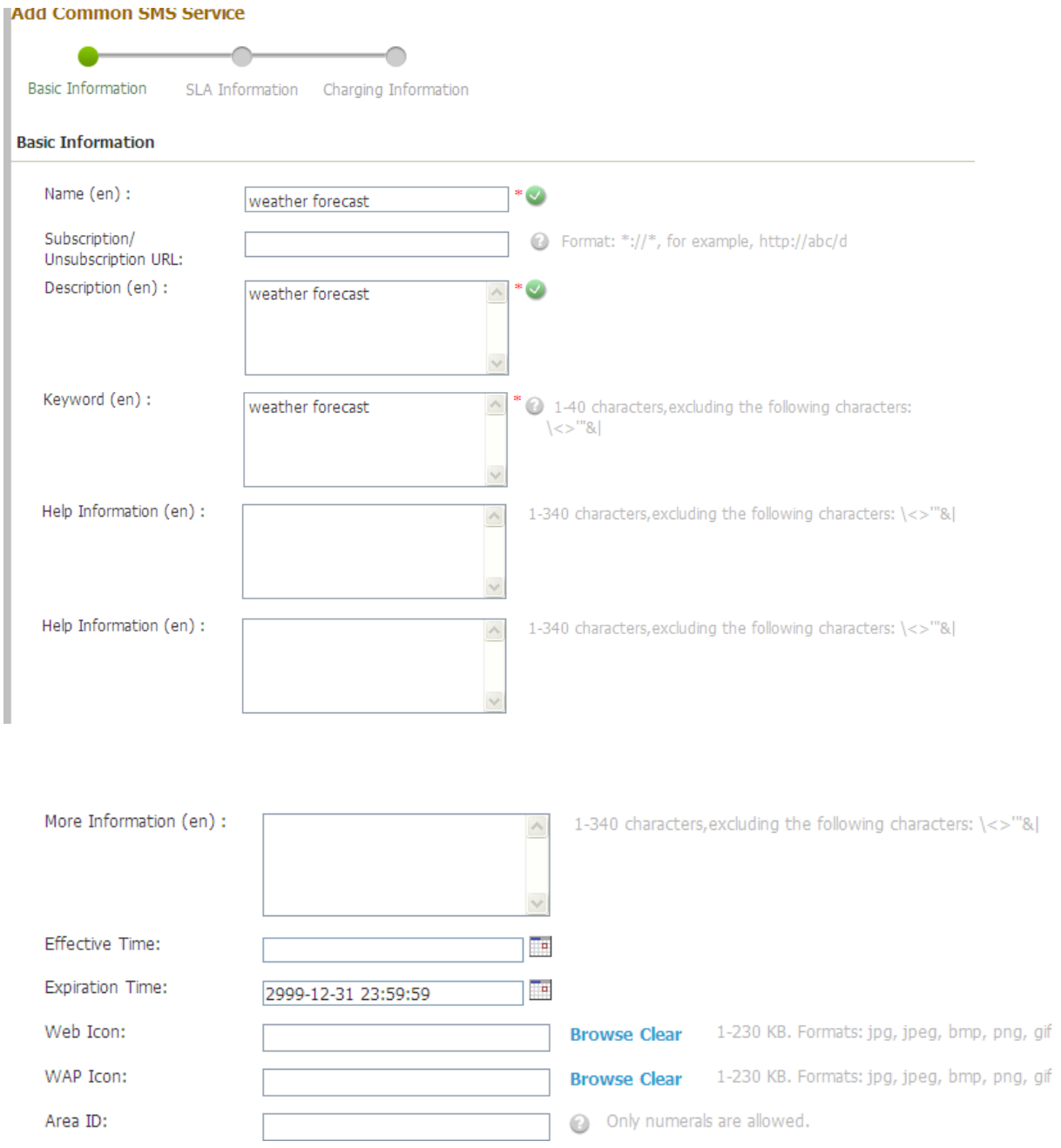

Add Common SMS Service
Basic Information
SLA Information
Charging Information
Basic Information
Name (en) :
weather forecast
Subscription/ Unsubscription URL:
Format: *://*, for example, http://abc/d
Description (en) :
weather forecast
Keyword (en) :
weather forecast
1-40 characters,excluding the following characters: \<>'"&|
Help Information (en) :
1-340 characters,excluding the following characters: \<>'"&|
Help Information (en) :
1-340 characters,excluding the following characters: \<>'"&|
More Information (en) :
1-340 characters,excluding the following characters: \<>'"&|
Effective Time:
Expiration Time:
2999-12-31 23:59:59
Web Icon:
Browse Clear
1-230 KB. Formats: jpg, jpeg, bmp, png, gif
WAP Icon:
Browse Clear
1-230 KB. Formats: jpg, jpeg, bmp, png, gif
Area ID:
Only numerals are allowed.

Figure 1: Release Common SMS Service

NOTE:

Set the fields according to preference with these guidelines.

- **Name (en)** should be in this format **21568_Music** (Shortcode_Nameofservice).
- **Using Mode** to **P2A** for all services that need to receive SMSs from subscribers.
- **Service Credibility** to **No credit** for charged short codes and **Credit** for Bulk SMSs
- Fill all fields with the * mark.
- For Subscription services fill **Subscription/Un-subscription URL** with the URL of the syncOrder service. Fill Command code with the keyword of the service.

3. Click **Next** to configure the service-related SLA information. For details, see Figure 2.

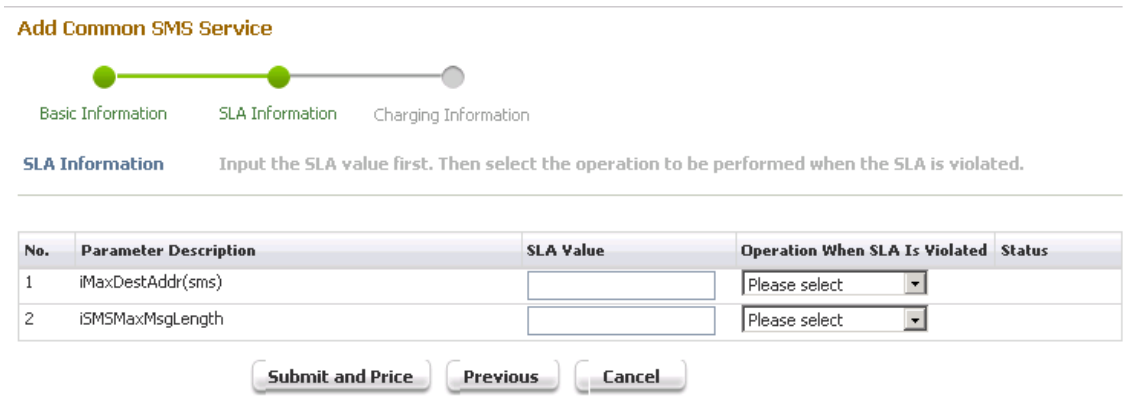
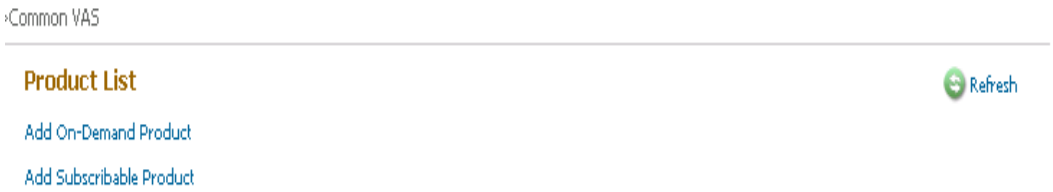


Figure 2: Release Common SMS Service

4. Click **Submit and Price**. The system displays the page for pricing services.



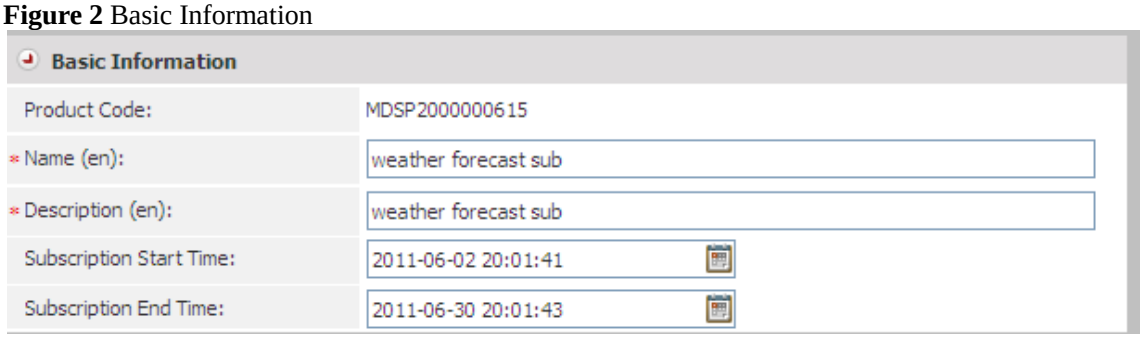
2.4.1 Subscription Services

1. Click **Add Subscription Product** on the page.
2. According to the application scenario of the product, configure the information about service pricing as prompted.
    . Enter the product name and product description, as shown in Figure 2.

NOTE;

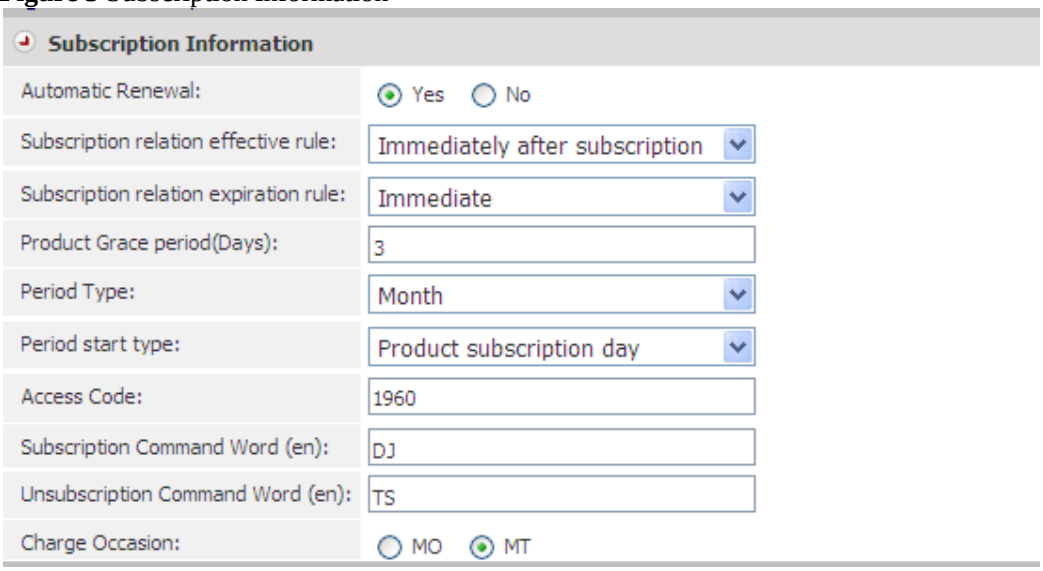
The product name should be in this format; **21568_Music** (Shortcode_Nameofservice). Make sure it's the same as the service name.

Figure 2 Basic Information



a. Configure the tariff information, as shown in Figure 3 and Figure 4.

Figure 3 Subscription Information



NOTE;

In case the SMS service charges a rental fee it is advisable for the Product Grace period to be set to infinity (-1)

Figure 4 Fee Information

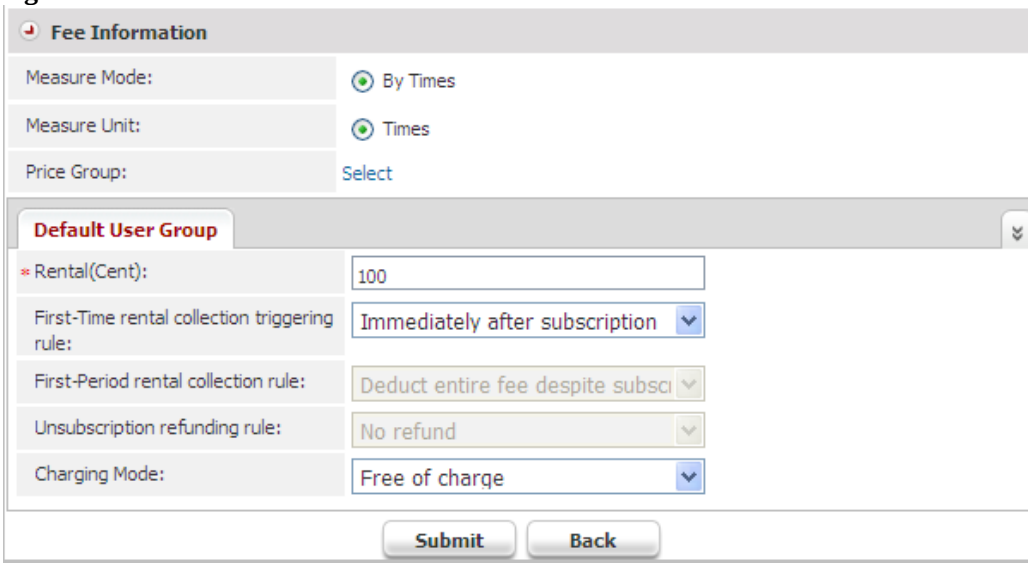
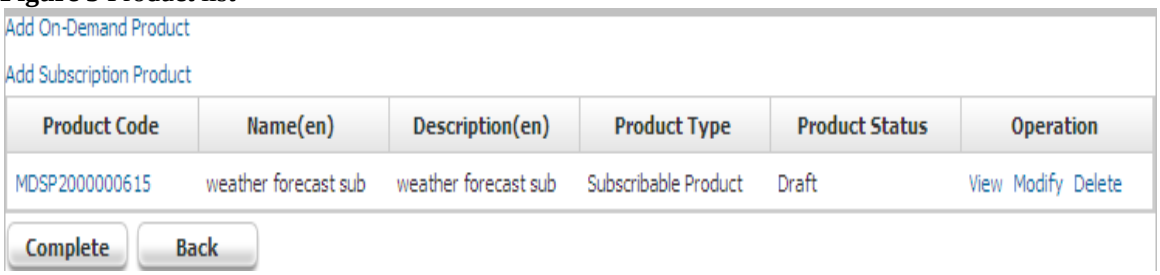


Figure 2, Figure 3 and Figure 4 show that the user sends **DG** to **1960** to subscribe to the SMS service. The monthly rent is 100 cents. The settings take effect immediately and the service supports automatic resuming. The SMS service is free of charge in the MT process.

b. Click **Submit** to submit the pricing information.

After the subscribable product is set, the system displays the product list, as shown in Figure 5.

Figure 5 Product list



NOTE:

You can click **View**, **Modify** or **Delete** in the **Operation** column to view, modify information about the product or delete the product.

3. Click **Complete** to finish setting the product.

   If the CP/SP is a low-credibility CP/SP, after the SMS/MMS service is added and priced, the review process is triggered. After the system reviewer approves the service, the service is in the commercial use state.

2.4.2 On-Demand Services

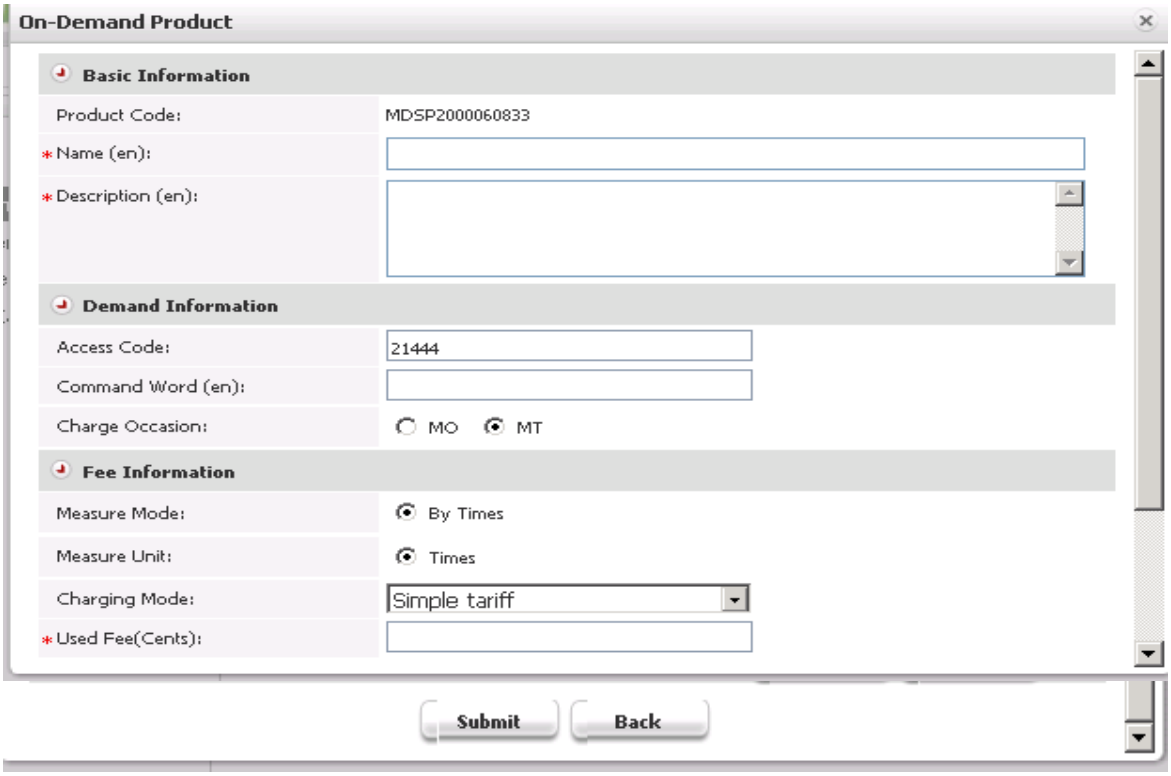
1. Click **Add On-Demand Product** on the page.
2. According to the application scenario of the product, configure the information about service pricing as prompted.

   Enter the product name and product description, as shown in Figure 6.

NOTE;

The product name should be in this format; **21568_Music** (Shortcode_Nameofservice). Make sure it's the same as the service name.

Figure 6 Basic Information



Fill the access code field with the short code assigned.

Leave the command word field Empty for a service used for both sending and receiving SMSs.

. Click **Submit** to submit the pricing information.

After the on-demand product is set, the system displays the product list, as shown in Figure 7.

Figure 7 Product list

NOTE:

You can click **View**, **Modify** or **Delete** in the **Operation** column to view, modify information about the product or delete the product.

4. Click **Complete** to finish setting the product.

If the CP/SP is a low-credibility CP/SP, after the SMS/MMS service is added and priced, the review process is triggered. After the system reviewer approves the service, the service is in the commercial use state.

3 Data Types

3.1 DeliveryStatus Enumeration

Table 3-1 List of delivery status values

Enumeration Value	Description
DeliveredToNetwork	Successful delivery to network.
DeliveryUncertain	Delivery status unknown. The message may be handed off to another network.
DeliveryImpossible	Unsuccessful delivery. The message fails to be delivered before it expires.
MessageWaiting	Queuing for delivery. This is a temporary state, pending transition to one of the preceding states.
DeliveredToTerminal	Successful delivery to a terminal.
DeliveryNotificationNotSupported	Unable to provide delivery receipt notification. The notifySMSDeliveryReceipt operation will return "DeliveryNotificationNotSupported" to indicate that delivery receipt for the specified address in a sendSMS request message is not supported.

3.2 DeliveryStatus Structure

Table 3-2 DeliveryStatus information

Parameter	Type	Optional	Description
address	xsd:anyURI	No	Destination address to which the notification is related.

Parameter	Type	Optional	Description
deliveryStatus	DeliveryStatus	No	Delivery result.

3.3 SmsMessage Structure

The **senderAddress** is the address from which the message was actually sent, which may or may not match the **senderName** value provided in the sendSms operation.

Table 3-3 SmsMessage information

Parameter	Type	Optional	Description
message	xsd:string	No	Text received in an SMS message.
senderAddress	xsd:anyURI	No	Address sending the SMS message.
smsServiceActivationNumber	xsd:anyURI	No	Number associated with the invoked message service; that is, the destination address used to send the message.
dateTime	xsd:dateTime	Yes	Time when the message is received by a carrier.

3.4 SimpleReference Structure

For those services that require a reference to a Web service, the information required to create the endpoint information is contained in this type.

Table 3-4 SimpleReference information

Parameter	Type	Optional	Description
Endpoint	xsd:anyURI	No	Endpoint address.
InterfaceName	xsd:string	No	Name of interface.
Correlator	xsd:string	No	Correlation information. The correlator is the transactionId generated by the SP that is used to match the various requests and responses as detailed in later sections.

3.5 ChargingInformation Structure

For services that include charging as an inline message part, the charging information is provided in this data structure.

Table 3-5 ChargingInformation information

Parameter	Type	Optional	Description
Description	xsd:string	No	Description text to be use for information and billing text.
Currency	xsd:string	Yes	Currency identifier as defined in ISO 4217 [12].
Amount	xsd:decimal	Yes	Amount to be charged.
Code	xsd:string	Yes	Charging code, referencing a contract under which the charge is applied.

3.6 UserID Structure

Table 3-6 UserID

Parameter	Mandatory/ Optional	Type	Description
ID	Mandatory	String	ID of an end user.
type	Mandatory	Integer	Optional values of the ID type. **0**: MSISDN Others: Reserved.

3.7 NamedParameterList Structure

Table 3-7 NamedParameterList

Parameter	Mandatory/ Optional	Type	Description
NamedParameters	Optional	Named Parameter[ ]	List of extended parameters.

3.8 NamedParameter Structure

Table 3-8 NamedParameter

Parameter	Mandatory/ Optional	Type	Description
key	Mandatory	String	Name of the extended parameter.
value	Mandatory	String	Value of the extended parameter.

3.9 RequestSOAPHeader

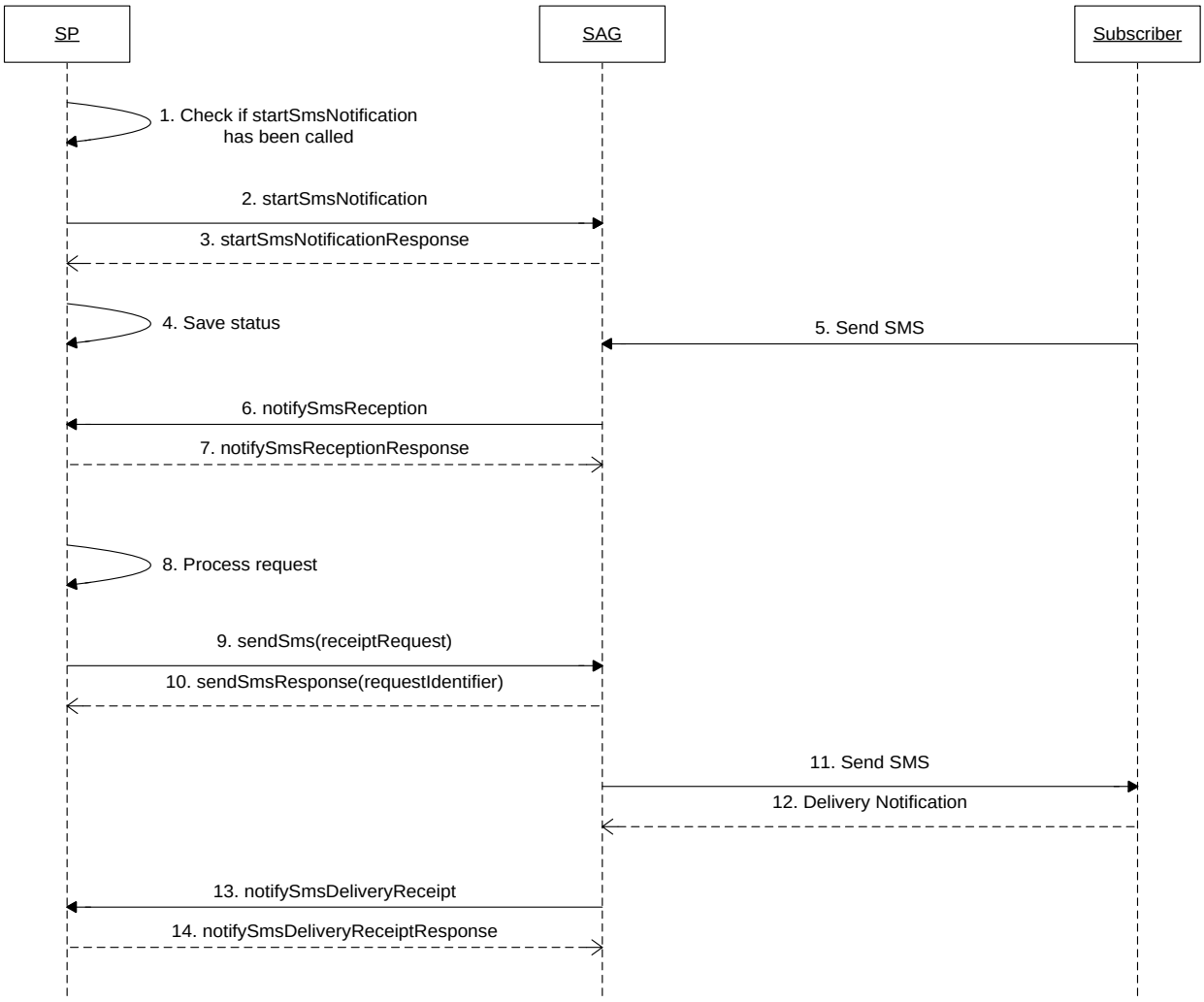
This is an addition to the SOAP header that is primarily used for authentication purposes.

Parameter	Type	Description
spId	xsd: string	SP ID. When an SP issues a service on an SDP, the SDP allocates an SP ID to the SP.
spPassword	xsd: string	This is an encrypted form of the SP password issued to an SP when an account is created on the SDP. The encryption algorithm is an MD5 hash of the spId then spPassword and the timestamp as illustrated below. spPassword = MD5 (spId + spPassword + timeStamp), e.g. spPassword = MD5 (601399myPassword20120703222854)
serviceId	xsd: string	Service ID. When an SP creates a service on an SDP, the SDP allocates a service ID to the SP.
timeStamp	xsd: string	Time stamp of a message/request This parameter is mandatory. Format is YYYYMMDDHH24MISS
OA	xsd: string,	Service subscription address. This parameter is mandatory in the case of single transmission but it is optional in the case of group transmission.
FA	xsd: string	Payment address. This parameter is optional.
linkid	xsd:string	Relational parameter generated in the notifySmsReception request. This parameter is mandatory in the sendSms request in the event of an on-demand service.

4 Interface Definitions

4.1 Typical Service Scenarios

4.1.1 On Demand Service – Online /Asynchronous notifications



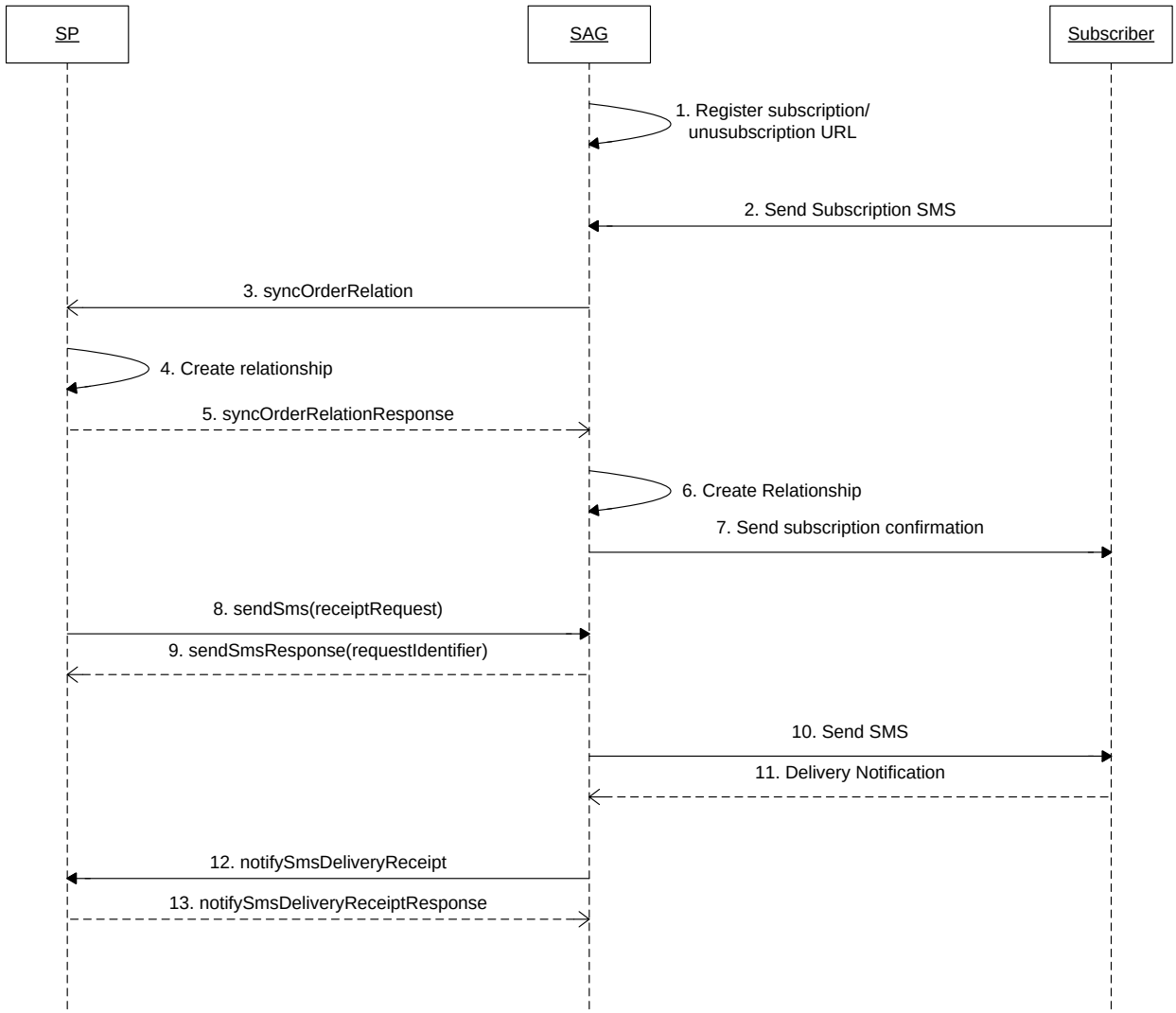
This process flow makes the following assumptions:

- The service is an on-demand service, i.e. the subscriber always initiates the request whenever content is desired and the SP can only respond to this request but can never initiate.
- The service provider already has an account on the SDP
- The service provider has created a service on the SDP that has been approved by a Safaricom administrator.
- The SP has chosen to receive online notifications as opposed to calling the getReceivedSms and the getSmsDeliveryStatus interfaces.
- The service provider has developed the endpoint(s) that support the **notifySmsReception** and the **notifySmsDeliveryReceipt** interfaces to facilitate online notification.

1. The SP checks whether the startSmsNotification interface has been invoked when the system starts. If the startSmsNotification interface has been invoked, and steps 2 to 4 are skipped. If the interface has not been invoked the process goes to step 2.
2. The SP invokes SAG's startSmsNotification interface to send a request to SAG to enable the MO message notification function.
3. SAG sends the startSmsNotificationResponse to the SP.
4. The SP should make a record of the startSmsNotificationResponse
5. The subscriber sends an SMS to a shortcode to request for a particular service or content
6. SAG sends a notifySmsReception request to the SP at the endpoint specified in the startSmsNotification request (the **linkid** parameter required for the sendSms request will be included in the header)
7. The SP sends an empty response to the SAG.
8. The SP processes the request from the subscriber and records the event and generates the appropriate response for the subscriber.
9. The SP will then call the sendSms interface of the SAG
   - As this is an on-demand service, which means that any correspondence between the SP and the subscriber must be subscriber initiated, the **linkid** received in the notifySmsReception request must be included in the sendSmsRequest.
   - If the SP requires to be notified of the delivery status, then the receiptRequest parameter will have to be specified. Failure to specify this optional parameter will require the SP to poll the SAG for the delivery status via the getSmsDeliveryStatus interface.
10. The SAG will respond with requestIdentifier which will be used to identify this request
11. The SDP will send the SMS to the subscriber
12. The subscriber will send the delivery confirmation to the SDP
13. The SAG will call the notifySmsDeliveryReceipt interface at the endpoint specified by the SP in the sendSms call. The SP should make a record of this notification.
14. The SP will issue an empty response to the SAG

N.B: The specifications for all the interfaces mentioned are detailed below from section 3.2.2

4.1.2 Subscription Service – Online /Asynchronous notifications



This process flow makes the following assumptions:

- The service is a subscription service, i.e. the subscriber will initiate the subscription & un-subscription requests. However, apart from these two requests, the SP can initiate a notification to the subscriber
- The service provider already has an account on the SDP
- The service provider has created a service on the SDP that has been approved by a Safaricom administrator.
- The SP has chosen to receive online notifications as opposed to calling the getReceivedSms and the getSmsDeliveryStatus interfaces.
- The service provider has developed the endpoint(s) that support the **syncOrderRelation**, the **notifySmsReception** and the **notifySmsDeliveryReceipt** interfaces to facilitate online notification.

1. When creating the service the SP will register a subscription/un-subscription URL. This is the endpoint at which the syncOrderRelation interface will be called.
   N.B: This will only be done once during service creation.
2. The subscriber sends a subscription request the content of which should be a pre-configured keyword.
3. The SDP verifies the keyword and the SAG calls the syncOrderRelation interface (at the endpoint specified in Step 1) and pass the subscription information to the SP.
4. The SP will create the relationship between the subscriber and the product/service they have ordered and make a record of this.
5. The SP will respond to the SAG indicating the status of the creation of the subscription.
6. Assuming the subscription relation is created successfully the SDP will also create the subscription relation.
7. The SDP will send the subscriber a notification confirming the successful creation of the subscription.
8. The SP generates an appropriate notification for the subscriber and calls the sendSms interface of the SAG (specifying the receiptRequest)
9. The SAG will respond with requestIdentifier which will be used to identify this request
10. The SDP will send the SMS to the subscriber
11. The subscriber will send the delivery confirmation to the SDP
12. The SAG will call the notifySmsDeliveryReceipt interface at the endpoint specified by the SP in the sendSms call. The SP should make a record of this notification.
13. The SP will issue an empty response to the SAG

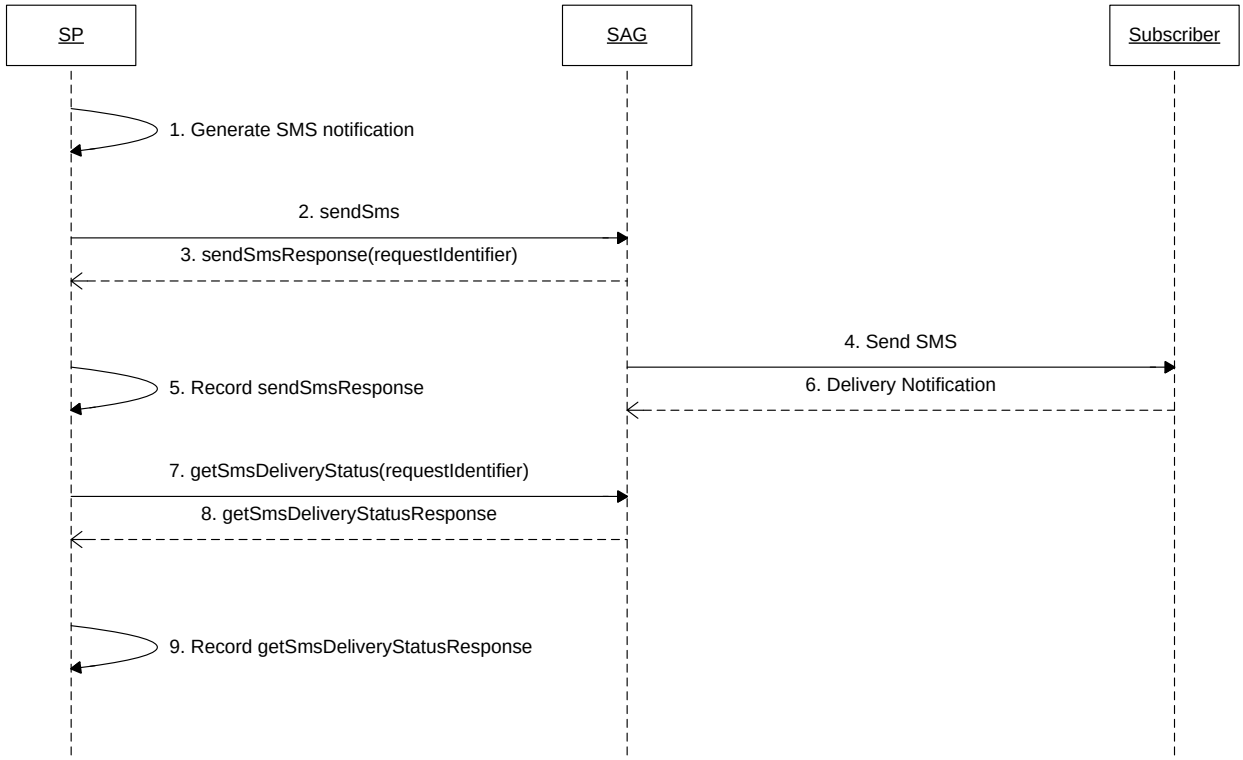
N.B: The specifications for all the interfaces mentioned are detailed below from section 3.2.2

4.2 Interface: SendSms

This interface defines operations to send various types of SMS messages and to subsequently poll for delivery status. The description of the sendSms message is as follows:

- **addresses** specifies the destination address of the short message. It may include group URIs as defined in the **Address List Management** specification. If groups are not supported, a **PolicyException** (POL0006) will be returned to the application.
- **senderName** is optional and specifies the sender name. The value is a string that is displayed on the user's terminal as the originator of the message.
- **charging** specifies the charging information.
- **receiptRequest** is optional and is specified when the application requires receiving notifications of the SMS delivery status. It is a SimpleReference structure that indicates the application endpoint and interface used for notification of delivery receipt, and a correlator that uniquely identifies the sending request:
  - If the notification mechanism is not supported by a network, a service exception (SVC0283) will be returned to the application and the message will not be sent to the addresses specified.
  - The **correlator** provided in the **receiptRequest** must be unique for this Web service and application at the time the notification is initiated; otherwise a service exception (SVC0005) will be returned to the application.
  - Notification to the application is done by invoking the notifySmsDeliveryReceipt operation at the endpoint specified in the receiptRequest.
- **requestIdentifier** is specified in the response message associated with each send operation. The application can use it to invoke the getSmsDeliveryStatus operation to poll for the delivery status.

Sequence Diagram: Send SMS – Offline/Polling Delivery Notification



This process flow makes the following assumptions:

- The service provider already has an account on the SDP
- The service provider has created a service that can be used to send the notifications

1. The SP will generate the content of the message to be sent to the subscriber
2. The SP will then call the sendSms interface of the SAG
3. The SAG will respond with requestIdentifier which will be used to identify this request
4. The SDP will send the SMS to the subscriber
5. The SP should make a record of the sendSms response
6. The subscriber will send the delivery confirmation to the SDP
7. The SP will call the getSmsDeliveryStatus interface of the SAG passing in the requestIdentifier received from the sendSmsResponse
8. The SAG will respond with the DeliveryInformation
9. The SP should make a record of the the DeliveryInformation

Sequence Diagram: Send SMS – Online/Asynchronous Delivery Notification

This process flow makes the following assumptions:

- The service provider already has an account on the SDP
- The service provider has created a service that can be used to send the notifications

1. The SP will generate the content of the message to be sent to the subscriber
2. The SP will then call the sendSms interface of the SAG (specifying the receiptRequest)
3. The SAG will respond with requestIdentifier which will be used to identify this request
4. The SDP will send the SMS to the subscriber
5. The SP should make a record of the sendSms response
6. The subscriber will send the delivery confirmation to the SDP
7. The SAG will call the notifySmsDeliveryReceipt interface at the endpoint specified by the SP in the sendSms call
8. The SP will issue an empty response to the SAG

N.B: For the following operations and the examples given take note of the different namespaces.

4.2.2 Operation: sendSms

The application invokes the sendSms operation to send an SMS message that is specified by the **message** string.

For GSM systems, if **message** contains characters not in the GSM 7-bit character set, the SMS is sent as a Unicode SMS.

If **message** is longer than the maximum supported length (for GSM, 160 GSM 7-bit characters or 70 Unicode characters), the message content will be sent as several concatenated SMS messages.

Input Message: sendSmsRequest

Table 4-1 Description of the sendSmsRequest message

Parameter	Type	Optional	Description
addresses	xsd:anyURI[1..unbounded]	No	Addresses to which the SMS message will be sent.
senderName	xsd:string	Yes	Name of the SMS message sender. It is displayed on the user's terminal as the originator of the message.
message	xsd:string	No	Content of the SMS message.
receiptRequest	common:SimpleReference	Yes	Application endpoint, interface name and correlator that will be used to notify the application when the message has been delivered to the terminal or whether the delivery is impossible.

The following is a sample sendSmsRequest message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://www.huawei.com.cn/schema/common/v2 1"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/send/v2 2/local">
   <soapenv:Header>
         <v2:RequestSOAPHeader>
         <v2:spId>35000001</v2:spId>
         <v2:spPassword>ZhlEXrrAmfosSFpYPfdPCA==</v2:spPassword>
         <v2:serviceId>35000001000001</v2:serviceId>
         <v2:timeStamp>20120721112622</v2:timeStamp>
         <!--mandatory if service is on-demand-->
         <v2:linkid>072013123900000000006</v2:linkid>
         <v2:OA>tel:254722123456</v2:OA>
         <v2:FA>tel:254722123456</v2:FA>
      </v2:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:sendSms>
         <!--1 or more repetitions:-->
```

```
         <loc:addresses>tel:254722123456</loc:addresses>
         <!--Optional:-->
         <loc:senderName>1234</loc:senderName>
         <loc:message>This is a test message</loc:message>
         <!--Optional:-->
         <loc:receiptRequest>
            <endpoint>http://10.138.30.123:9080/notify</endpoint>
            <interfaceName>SmsNotification</interfaceName>
            <correlator>123</correlator>
         </loc:receiptRequest>
      </loc:sendSms>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Message: sendSmsResponse

Table 4-2 Description of the sendSmsResponse message

Parameter	Type	Optional	Description
result	xsd:string	No	A specific SMS delivery request. This is the **requestIdentifier** that is required for the getSmsDeliveryStatus operation

The following is a sample sendSmsResponse message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Header/>
   <soapenv:Body>
      <ns1:sendSmsResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/sms/send/v2_2/local">
         <ns1:result>10000520040111022506320100041</ns1:result>
      </ns1:sendSmsResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Reference Faults

- Service exceptions
  SVC0001: Service error.
  SVC0002: Invalid input value.
  SVC0280: The length is exceeded.
  SVC0901: Access authentication or authorization error.
  SVC0905: Parameter error.
- Policy exceptions
  POL0003: There are too many addresses.
  POL0006: The function of sending a message to a group is not supported.
  POL0900: The group-send function is not supported.

POL0904: The message sending rate exceeds the limit.

4.2.3 Operation: getSmsDeliveryStatus

The application invokes the getSmsDeliveryStatus operation to obtain the status of a previous SMS delivery request identified by **requestIdentifier**. The information about the status is returned in **deliveryStatus**, which is an array of status related to the request identified by **requestIdentifier**. The status is identified by a couplet indicating a user address and the associated delivery status. This operation can be invoked multiple times by the application even if the status has reached a final value. After the status has reached a final value, status information will be available only for a limited period of time that should be specified in an offline configuration step. The following five different SMS delivery status values have been identified:

- **DeliveredToNetwork**: In case of concatenated messages, the SMS message enters this status only when all the SMS contents have been successfully delivered to the network.
- **DeliveryUncertain:** The SMS message may be handed off to another network.
- **DeliveryImpossible:** Unsuccessful delivery. The message cannot be delivered before it expires.
- **MessageWaiting:** The message is still queuing for delivery.
- **DeliveredToTerminal:** In case of concatenated messages, the SMS message enters this status only when all the SMS contents have been successfully delivered to the terminal.

Input Message: getSmsDeliveryStatusRequest

Table 4-3 Description of the getSmsDeliveryStatusRequest message

Parameter	Type	Optional	Description
requestIdentifier	xsd:string	No	A specific SMS delivery request.

The following is a sample getSmsDeliveryStatusRequest message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://www.huawei.com.cn/schema/common/v2 1"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/send/v2 2/local">
   <soapenv:Header>
      <v2:RequestSOAPHeader>
         <v2:spId>35000001</v2:spId>
         <v2:spPassword>e6434ef249df55c7a21a0b45758a39bb</v2:spPassword>
         <v2:serviceId>35000001000001</v2:serviceId>
         <v2:timeStamp>20120812005752</v2:timeStamp>
         <v2:OA>tel:254722123456</v2:OA>
         <v2:FA>tel:254722123456</v2:FA>
      </v2:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:getSmsDeliveryStatus>
<loc:requestIdentifier>1000052004011102250632010000041</loc:requestIdentifier>
      </loc:getSmsDeliveryStatus>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Message: getSmsDeliveryStatusResponse

Table 4-4 Description of the getSmsDeliveryStatusResponse message

Parameter	Type	Optional	Description
result	DeliveryInformation [0..unbounded]	Yes	Possible delivery statuses of the SMS message are: • DeliveredToNetwork • DeliveryUncertain • DeliveryImpossible • MessageWaiting • DeliveredToTerminal

The following is a sample getSmsDeliveryStatusResponse message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:getSmsDeliveryStatusResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/sms/send/v2 2/local">
         <ns1:result>
            <address>tel:254722123456</address>
            <deliveryStatus>DeliveredToTerminal</deliveryStatus>
         </ns1:result>
      </ns1:getSmsDeliveryStatusResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

- Reference faults
  - Service exceptions

    SVC0001: Service error.

    SVC0002: Invalid input value.

    SVC0901: Access authentication or authorization error.

    SVC0905: Parameter error.
  - Policy exception

    POL0903: The storage duration expires.

    POL0904: The message sending rate exceeds the limit.

4.3 Interface: SmsNotification

SmsNotification is the application side interface to which notifications about SMS messages are delivered.

4.3.1 Operation: notifySmsReception

The notification is used to send an SMS message to the application. The notification will occur only if the SMS message fulfills the criteria that are specified for starting the SMS notification.

The notifySmsReception operation must be implemented by a Web Service on the application side. It will be invoked by the Parlay X 2 server to notify the application of the reception of an SMS message. The notification will occur only if the received SMS message fulfills the criteria specified in a provisioning step, identified by the correlator. The criteria must at least include an smsServiceActivationNumber, that is, the SMS destination address that can be "monitored" by the application. The parameter senderAddress contains the address of the sender. The application can apply the appropriate service logic to process the SMS message. The linkid specified in the NotifySOAPHeader section is required in the sendSmsRequest as detailed in section 4.2.2.

Input Message: notifySmsReceptionRequest

Table 4-5 Description of the notifySmsReceptionRequest message

Parameter	Type	Optional	Description
correlator	xsd:string	No	Correlator provided in the request to set up this notification.
message	SmsMessage	No	Message received.

The following is a sample notifySmsReceptionRequest message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://www.huawei.com.cn/schema/common/v2 1"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/notification/v2 2/local">
   <soapenv:Header>
      <ns1:NotifySOAPHeader
xmlns:ns1="http://www.huawei.com.cn/schema/common/v2 1">
         <ns1:spRevId>ekubai</ns1:spRevId>
         <ns1:spRevpassword>Abc123</ns1:spRevpassword>
         <ns1:spId>601399</ns1:spId>
         <ns1:serviceId>6013992000001491</ns1:serviceId>
         <ns1:linkid>0716172243075800009</ns1:linkid>
         <ns1:traceUniqueID>40409010259120716142243089 2004</ns1:traceUniqueID>
      </ns1:NotifySOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:notifySmsReception>
         <loc:correlator>123</loc:correlator>
         <loc:message>
            <message>This is a test message</message>
            <senderAddress>tel:722123456</senderAddress>
            <smsServiceActivationNumber>1234</smsServiceActivationNumber>
            <dateTime>2012-07-03T00:00:00.000+08:00</dateTime>
         </loc:message>
      </loc:notifySmsReception>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Message: notifySmsReceptionResponse

Table 4-6 Description of the notifySmsReceptionResponse message

Parameter	Type	Optional	Description
None	None	None	None

Sample notifySmsReceptionResponse

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/notification/v2 2/local">
   <soapenv:Header/>
   <soapenv:Body>
      <loc:notifySmsReceptionResponse/>
   </soapenv:Body>
</soapenv:Envelope>
```

Reference Faults

None.

4.3.2 Operation: notifySmsDeliveryReceipt

The **notifySmsDeliveryReceipt** operation must be implemented by a Web Service on the application side if it requires notification of SMS delivery receipt. It will be invoked by the Parlay X 2 server to notify the application when an SMS message sent by an application has been delivered to the terminal of the recipient or whether the delivery is impossible. The notification will occur only if the status of the sent SMS message is **DeliveredToTerminal** or **DeliveryImpossible** and the application has specified the optional **receiptRequest** part when sending an SMS message. The correlator returned corresponds to the identifier specified by the application in **receiptRequest** of the original sendSMS request.

When an SMS message is sent to multiple addresses, the server will send a notification for each terminal when an SMS message is delivered to the terminal.

The following three different SMS delivery status values will be returned in a notifySMSDeliveryReceiptResponse message:

- **DeliveryImpossible**: Unsuccessful delivery. The message cannot be delivered before it expires.
- **DeliveredToTerminal**: In case of concatenated messages, the SMS message enters this status only when all the SMS contents have been successfully delivered to the terminal.
- **DeliveredNotificationNotSupported:** If notification is supported by the network but it does not support delivery receipt for the addresses specified in the **sendSMSRequest** message, the service will send this status for those addresses.

Input Message: notifySmsDeliveryReceiptRequest

Table 4-7 Description of the notifySmsDeliveryReceiptRequest message

Parameter	Type	Optional	Description
correlator	xsd:string	No	ID of the original "Send" Request. This correlator is provided by the application in the sendSMS Request message.
deliveryStatus	DeliveryInformation	No	Possible delivery statuses of the SMS message sent to a terminal are: • DeliveryImpossible • DeliveredToTerminal • DeliveryNotificationNotSupported • DeliveryUncertain

The following is a sample notifySmsDeliveryReceiptRequest message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://www.huawei.com.cn/schema/common/v2 1"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/notification/v2 2/local">
   <soapenv:Header>
      <v2:NotifySOAPHeader>
         <v2:timeStamp>20100904042855</v2:timeStamp>
         <v2:subReqID>11111111111111</v2:subReqID>
         <traceUniqueID>504021503311009040428550001002</traceUniqueID>
      </v2:NotifySOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:notifySmsDeliveryReceipt>
         <loc:correlator>123456789</loc:correlator>
         <loc:deliveryStatus>
            <address>tel:254722123456</address>
            <deliveryStatus>DeliveredToTerminal</deliveryStatus>
         </loc:deliveryStatus>
      </loc:notifySmsDeliveryReceipt>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Message: notifySmsDeliveryReceiptResponse

Table 4-8 Description of the notifySmsDeliveryReceiptResponse message

Parameter	Type	Optional	Description
None	None	None	None

Reference Faults

None

4.4 Interface: ReceiveSms

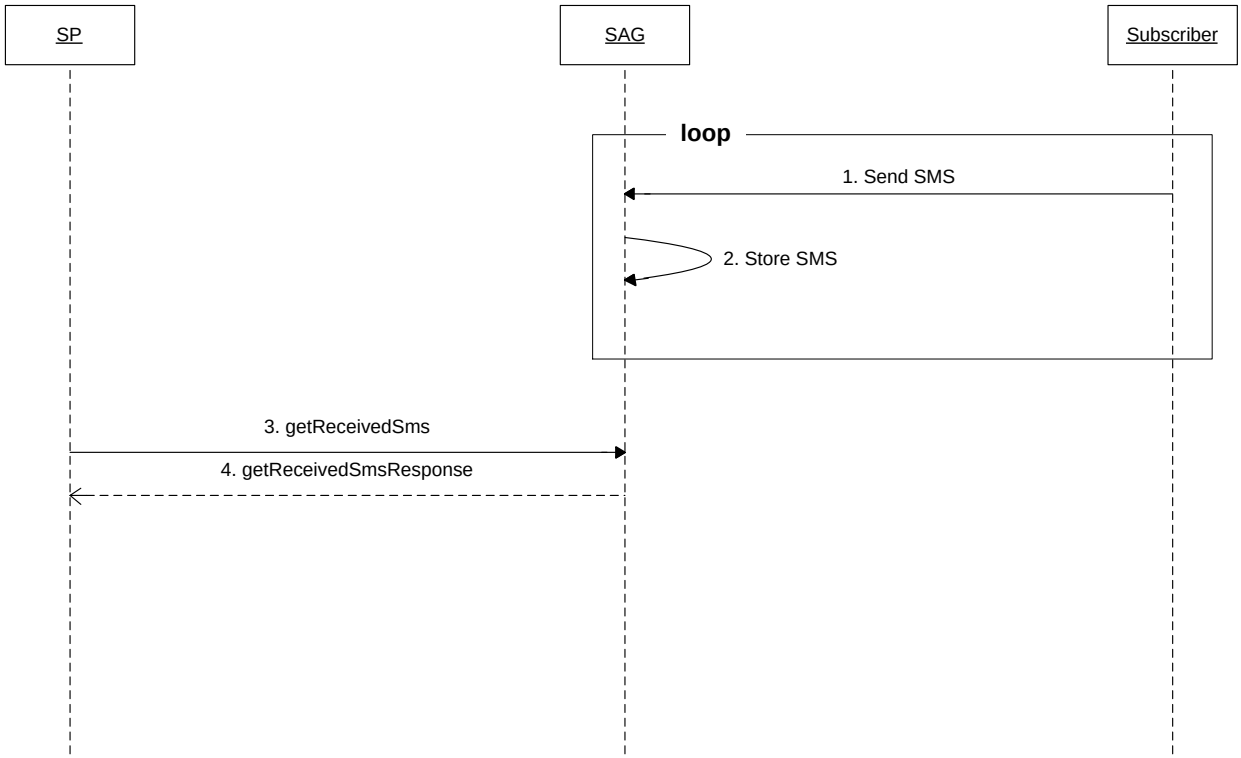
4.4.1 Operation: getReceivedSms

Note that the application may also register offline for the reception of short messages: i.e. without using the Parlay X interface and the startSmsNotification operation. The registration request should at a minimum specify the message destination address. The request may also specify the optional text string criteria. The registration request is assigned a unique registration identifier.

The **getReceivedSms** is invoked to retrieves all the received SMS messages that fulfill the criteria identified by registrationIdentifier. The operation returns only the list of SMS messages that are received since the previous invocation of the same operation. That is, each time the operation is executed, the messages returned are removed from the server. Moreover, each SMS message will be automatically removed from the server after a maximum time interval specified in an offline configuration step.

The received SMS messages are returned in the **receivedSms** field. An SMS message is identified by a structure indicating the sender of the SMS message and the content.

Sequence Diagram: Get Received SMS – Online/Asynchronous Subscriber Requests



This process flow makes the following assumptions:

- The service provider already has an account on the SDP
- The service provider has created a service that can be used to receive the subscriber requests

1. The subscriber sends an SMS to a published service
2. The SDP will store the SMS
   - Please note that steps 1 & 2 above will be repeated until the SP calls the getReceivedSms interface to retrieve all pending requests
3. The SP will call the getReceivedSms interface
4. The SAG will respond with all subscriber requests since the last getReceivedSms call

Input Message: getReceivedSmsRequest

Table 4-9 Description of the getReceivedSmsRequest message

Parameter	Type	Optional	Description
registrationIdentifier	xsd:string	No	Offline provisioning step that enables the application to receive the notification of SMS reception according to the specified criteria. The offline provisioning step is the service creation and configuration on the MDMC and the registrationIdentifier is the service shortcode.

The following is a sample getReceivedSmsRequest message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://www.huawei.com.cn/schema/common/v2_1"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/receive/v2_2/local">
   <soapenv:Header>
      <v2:RequestSOAPHeader>
         <v2:spId>35000001</v2:spId>
         <v2:spPassword>062015113c5a302d41bc8c684b8eef9e</v2:spPassword>
         <v2:serviceId>35000001000001</v2:serviceId>
         <v2:timeStamp>20120721112645</v2:timeStamp>
      </v2:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:getReceivedSms>
         <loc:registrationIdentifier>1111</loc:registrationIdentifier>
      </loc:getReceivedSms>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Message: getReceivedSmsResponse

Table 4-10 Description of the getReceivedSmsResponse message

Parameter	Type	Optional	Description
result	SmsMessage [0..unbounded]	Yes	SMS message received since last invocation.

The following is a sample getReceivedSmsResponse message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <ns1:getReceivedSmsResponse
xmlns:ns1="http://www.csapi.org/schema/parlayx/sms/receive/v2 2/local">
         <ns1:result>
            <message>This is test message 1</message>
            <senderAddress>tel:721214848</senderAddress>
            <smsServiceActivationNumber>tel:1234</smsServiceActivationNumber>
            <dateTime>2012-06-14T07:38:13Z</dateTime>
         </ns1:result>
         <ns1:result>
            <message>This is test message 2</message>
            <senderAddress>tel:721214848</senderAddress>
            <smsServiceActivationNumber>tel:1234</smsServiceActivationNumber>
            <dateTime>2012-06-14T07:38:25Z</dateTime>
         </ns1:result>
      </ns1:getReceivedSmsResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Reference Faults

Service exceptions:

SVC0001: Service error.

SVC0002: Invalid input value.

SVC0901: Access authentication or authorization error.

SVC0905: Parameter error.

4.5 Interface: SmsNotificationManager

The SMS message notification manager enables applications to set up and tear down notifications for SMS messages online.

4.5.1 Operation: startSmsNotification

The application registers for the reception of short messages by invoking startSmsNotification. The request includes event criteria consisting of a value for the short message destination address (the smsServiceActivationNumber part) and an optional text string for matching against the first word of the message body (the criteria part); also a URI for a Web Service implementing the SmsNotification interface on the client application side, and a correlation value for identifying this event registration request.

This operation initiates notifications to the application for a given SMS Service activation number and criteria.

The smsServiceActivationNumber is an address data item, for example, a short code, as defined in ES 202 391-1 [2].

The correlator provided in the reference must be unique for the application Web Service at the time the notification is initiated; otherwise a fault (SVC0005) will be returned to the application.

If specified, criteria will be used to filter messages that are to be delivered to an application. The use of criteria will allow different notification endpoints to receive notifications for the same smsServiceActivationNumber. If criteria are not provided, all messages for the smsServiceActivationNumber will be delivered to the application. If the values of **criteria** overlap, SVC0008 will be returned to the application and the notification will not be set up. The combination of **smsServiceActivationNumber** and **criteria** must be unique, so that a notification will be delivered to only one notification endpoint. If matching combination of **smsServiceActivationNumber** and **criteria** is found, the message will not be delivered to the application.

Input Message: startSmsNotificationRequest

Table 4-11 Description of the startSmsNotificationRequest message

Parameter	Type	Optional	Description
reference	common:SimpleReference	No	Notification endpoint definition
smsServiceActivationNumber	xsd:anyURI	No	Destination address of the SMS message
criteria	xsd:string	Yes	Text to match against to determine the application to receive the notification. This text is matched against the first word in the message, defined as the initial characters after discarding any leading Whitespace and ending with a Whitespace or end of message. The matching shall be case-insensitive.

The following is a sample startSmsNotificationRequest message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://www.huawei.com.cn/schema/common/v2 1"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/notification manager/v2 3/local
">
   <soapenv:Header>
      <RequestSOAPHeader xmlns="http://www.huawei.com.cn/schema/common/v2 1">
         <spId>35000001</spId>
         <spPassword>de96d901b3bad1db2aab76b7b0b202f2</spPassword>
         <serviceId>0003062000001100</serviceId>
         <timeStamp>20120731064245</timeStamp>
      </RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:startSmsNotification>
         <loc:reference>
            <endpoint>>http://10.138.30.153:9080/notify</endpoint>
            <interfaceName>notifySmsReception</interfaceName>
            <correlator>1232</correlator>
         </loc:reference>
         <loc:smsServiceActivationNumber>23424</loc:smsServiceActivationNumber>
         <!--Optional:-->
         <loc:criteria>Love</loc:criteria>
      </loc:startSmsNotification>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Message: startSmsNotificationResponse

Table 4-12 Description of the startSmsNotificationResponse message

Parameter	Type	Optional	Description
None	None	None	None

Reference Faults

Service exceptions:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0005: Duplicate correlator.
- SVC0008: Overlapping Criteria.
- SVC0901: Access authentication or authorization error.
- SVC0905: Parameter error.

4.5.2 Operation: stopSmsNotification

The application may end an SMS notification by using this operation.

Input Message: stopSmsNotificationRequest

Table 4-13 Description of the stopSmsNotificationRequest message

Parameter	Type	Optional	Description
correlator	xsd:string	No	Correlator of the request to end an SMS notification.

The following is a sample stopSmsNotification message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v2="http://www.huawei.com.cn/schema/common/v2 1"
xmlns:loc="http://www.csapi.org/schema/parlayx/sms/notification manager/v2 3/local
">
   <soapenv:Header>
      <v2:RequestSOAPHeader>
         <spId>35000001</spId>
         <spPassword>de96d901b3bad1db2aab76b7b0b202f2</spPassword>
         <serviceId>0003062000001100</serviceId>
         <timeStamp>20120731064245</timeStamp>
      </v2:RequestSOAPHeader>
   </soapenv:Header>
   <soapenv:Body>
      <loc:stopSmsNotification>
         <loc:correlator>1232</loc:correlator>
      </loc:stopSmsNotification>
   </soapenv:Body>
</soapenv:Envelope>
```

Output Message: stopSmsNotificationResponse

Table 4-14 Description of the stopSmsNotificationResponse message

Parameter	Type	Optional	Description
None	None	None	None

Reference Faults

Service exceptions:

SVC0001: Service error.

SVC0002: Invalid input value.

SVC0901: Access authentication or authorization error.

SVC0905: Parameter error

4.6 Interface: DataSync

This interface is used to synchronize the subscription data to third-party applications.

4.6.1 Operation: SyncOrderRelation

This operation is invoked to notify SPs that a subscription has been changed on the SDP so that they can update their local subscription database.

The input message, output message, and reference faults are as follows:

- Input message: SyncOrderRelationRequest

Table 4-15 SyncOrderRelationRequest description

Parameter	Mandatory/ Optional	Type	Description
userID	Mandatory	UserID	ID of a subscriber.
spID	Mandatory	String	ID of an SP.
productID	Mandatory	String	Service ID which may correspond to one or more products, and each of product has a price.
serviceID	Mandatory	String	Service ID.
serviceList	Optional	String	ID list of all services in a service package. For example, "1234567890|1234567891|1234567892". All IDs are separated with "|" characters. If no IDs are specified, the value of this parameter is blank.
updateType	Mandatory	int	Reason type of updating subscription information. Optional values are as follows: 1: Addition 2: Deletion 3: Modification
updateTime	Mandatory	String	Time stamp in the format of YYYYMMDDhhmmss.
updateDesc	Optional	String	Update description.
effectiveTime	Optional	String	Time when the subscription takes effect. The default value is the current time of the MDSP.
expiryTime	Optional	String	Expiry time of the subscription.
extensionInfo	Optional	NamedParameterList	Extended parameter.

Table 4-16 List of extended parameters

Key	Actual Type	Meaning
N/A	-	-

The following is a sample syncOrderRelationRequest message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/data/sync/v1 0/local">
   <soapenv:Header/>
   <soapenv:Body>
      <loc:syncOrderRelation>
         <loc:userID>
            <ID>254721214848</ID>
            <type>0</type>
         </loc:userID>
         <loc:spID>601399</loc:spID>
         <loc:productID>MDSP2000052892</loc:productID>
         <loc:serviceID>6013992000001442</loc:serviceID>
         <loc:serviceList>601399200000144</loc:serviceList>
         <loc:updateType>1</loc:updateType>
         <loc:updateTime>20120621232135</loc:updateTime>
         <loc:updateDesc>Addition</loc:updateDesc>
         <loc:effectiveTime>20120622001311</loc:effectiveTime>
         <loc:expiryTime>20120822001311</loc:expiryTime>
         <loc:extensionInfo>
            <namedParameters>
               <key>TransactionID</key>
               <value>1339730498361</value>
               <key>orderKey</key>
               <value>999000000009508556</value>
               <key>MDSPSUBEXPMODE</key>
               <value>1</value>
               <key>objectType</key>
               <value>1</value>
               <key>TraceUniqueID</key>
               <value>06212031580010010012</value>
               <key>rentSuccess</key>
               <value>false</value>
            </namedParameters>
         </loc:extensionInfo>
      </loc:syncOrderRelation>
   </soapenv:Body>
</soapenv:Envelope>
```

- Output message: SyncOrderRelationResponse

Table 4-17 SyncOrderRelationResponse description

Parameter	Mandatory/ Optional	Type	Description
result	Mandatory	Integer	Return code; **0** means success.
resultDescription	Mandatory	String	Operation result description.
extensionInfo	Optional	NamedParameterList	Extended parameter.

The following is a sample syncOrderRelationResponse message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:loc="http://www.csapi.org/schema/parlayx/data/sync/v1 0/local">
   <soapenv:Header/>
   <soapenv:Body>
      <loc:syncOrderRelationResponse>
         <loc:result>0</loc:result>
         <loc:resultDescription>OK</loc:resultDescription>
         <!--Optional:-->
         <loc:extensionInfo>
            <!--Zero or more repetitions:-->
            <namedParameters>
               <key>ServiceName</key>
               <value>LoveQuotes</value>
            </namedParameters>
         </loc:extensionInfo>
      </loc:syncOrderRelationResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Table 4-18 List of extended parameters

key	Actual Type	Meaning
N/A	-	-

- Reference faults

  Service exceptions
  - 1211: The field format is incorrect or the value is invalid.
  - 2030: The subscription relationship already exists.
  - 2031: The subscription relationship does not exist.
  - 2032: The service does not exist.
  - 2033: The service is unavailable.
  - 2034: The service cannot be subscribed.
  - 2500: An internal system error occurred.

5 Appendix

1. On-Demand Products Pricing Parameters

Table 1 shows the on-demand SMS and MMS service product parameters.

NOTE:

The parameters with the asterisk symbol ***** on MDMC are mandatory.

Table 1 Description of parameters on the page for pricing on-demand SMS and MMS service products

Parameter	Description	Setting
Product Code	Unique identifier of a product.	[Value Range] The value is a string of 1 to 21 characters, consisting of letters (case sensitive), digits, and underscores.
Name (en)	Name of the on-demand SMS or MMS service product.	[Value Range] The value consists of 85 characters at most, excluding the following special characters: \ < > ' " & |
Description (en)	Description of the on-demand SMS or MMS service product.	[Value Range] The value consists of 250 characters at most, excluding the following special characters: \ <> ' " & | This will be visible on the Portal.
Access Code	Access code of a product. Access code of the platform to which SMS messages for product order are sent.	[Value Range] A numeric string.

Table 1 Description of parameters on the page for pricing on-demand SMS and MMS service products

Parameter	Description	Setting
	The access code can be used by: • The MDSP to identify a subscribable product. The combination of **Access Code** and **Order Command** uniquely identifies a subscribable product in the MDSP. • The SAG to route order requests to required SPs.	
Command (en)	Command word for demanding the SMS or MMS service product. A user can edit the command word in an SMS message and send the SMS message to the corresponding gateway to order a product.	[Value Range] The value is a string of digits, uppercase letters, and lowercase letters.
Charge Occasion	Charging occasion. • MO: mobile originating application terminated. • MT: application originating mobile terminated.	[Default Value] MT
Measure Mode	Measuring mode of a product.	[Default Value] By Times
Measure Unit	Measuring unit of a product.	[Default Value] Times
Charging Mode	Charging rate mode. Options are as follows: • Simple Tariff • Free of charge • Graded Tariff	[Default Value] Simple Tariff
Simple Tariff		
Used Fee (Cent)	Real-time fee for using a product (money).	[Value Range] The value is a positive integer.

Table 1 Description of parameters on the page for pricing on-demand SMS and MMS service products

Parameter	Description	Setting
		[Note] The currency unit is configurable and it's in Kenyan cents.
Graded Tariff		
Grading Mode	Type of calculating grading tariffs. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	[Default Value] By Accumulation: indicates that the graded tariff is set based on users' accumulated usage.
No.	Tariff grade ID. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	[Note] Click **Add** to add a grade. Click **Delete** to delete a grade.
Start Point	Start point of the range corresponding to a tariff grade. CPs/SPs specify tariff intervals based on the user usage. An example mapping between the number of order times and fees collected is as follows: 1 to 10 times: 50 cents; 11 to 20 times: 40 cents; equal to or more than 21 times: 20 cents. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	[Note] The system automatically sets a value for this parameter.
End Point	End point of the range corresponding to a tariff grade. CPs/SPs specify tariff intervals based on the user usage. An example mapping between the number of order times and fees collected is as follows: 1 to 10 times: 50 cents; 11 to 20 times: 40	[Value Range] The value is a positive integer and is greater than the value of the start point.

Table 1 Description of parameters on the page for pricing on-demand SMS and MMS service products

Parameter	Description	Setting
	cents; equal to or more than 21 times: 20 cents. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	
Used Fee (Cent)	Fee for an accumulated volume range (money). Users are charged according to different charging rates of traffic volume ranges. An example mapping between the number of order times and fees collected is as follows: 1 to 10 times: 50 cents; 11 to 20 times: 40 cents; equal to or more than 21 times: 20 cents.	[Value Range] The value is a positive integer. [Note] The currency unit is configurable.

2. Subscription Product Pricing Parameters

Table 2. Shows the subscription SMS and MMS service product parameters.

NOTE:

The parameters with the asterisk symbol ***** are mandatory.

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
Product Code	Unique identifier of a product.	[Value Range] The value is a string of 1 to 21 characters, consisting of letters (case sensitive), digits, and underscores.
Name (en)	Name of the subscription SMS or MMS service product.	[Value Range] The value consists of 85 characters at most, excluding the following special characters:

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
		\ < > ' " & |
Description (en)	Description of the subscription SMS or MMS service product.	[Value Range] The value consists of 250 characters at most, excluding the following special characters: \ < > ' " & |
Subscription Start Time	Start time of the validity period of a subscription product. When a CP/SP releases a subscription product, the start time and end time of its validity period are synchronized to the portal. Users can subscribe to the product within the validity period.	[Note] The value format is *YYYY-MM-DD hh*:*mm*:*ss*.
Subscription End Time	End time of the validity period of a subscription product. When a CP/SP releases a subscription product, the start time and end time of its validity period are synchronized to the portal. Users can subscribe to the product within the validity period.	[Note] The value format is *YYYY-MM-DD hh*:*mm*:*ss*.
Automatic Renewal	Whether to automatically renew the subscription for a user when a product expires.	[Default Value] Yes
Subscription relation effective rule	Effective policy of a subscription relationship. Options are as follows: • **Immediately after subscription**: The subscription relationship takes effect immediately after a user subscribes to the product. • **Delay N days**: The subscription relationship takes effect *N* days after the user subscribes to a product. • **Delay N months**: The subscription relationship takes effect *N* months after the user	[Default Value] Immediately after subscription

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
	subscribes to a product. Assume that this parameter is set to **Delay 5 days**. If a user subscribes to a product at 12:53:42 on June 3, 2010, the subscription relationship takes effect from 12:53:42 on June 8, 2010 and then the user can use the product.	
Subscription relation effective rule value	Number of days or months after subscription when a subscription relationship takes effect. NOTE: This option is valid only when **Subscription relation effective rule** is set to **Delay N days** or **Delay N months**.	[Value Range] The value is a positive integer.
Subscription relation expiration rule	Expiration policy of a subscription relationship. • Next cycle The subscription relationship becomes invalid upon the start of the next cycle after the user unsubscribes from a cycle-package product. • Immediate If a user cancels the subscription, the subscription relationship expires immediately.	[Default Value] Immediate
Product Grace period(Days)	Product grace period. When a product expires and rental collection fails, an additional period of time is provided for a user to make payment and the MDSP to collect the rental. When releasing an auto-renewal subscription product on the MDSP management portal, the CP/SP needs to set the grace period for the	[Value Range] • **-1**: The grace period is endless. • An integer ranging from **0** to **2147483647**.

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
	product. When a product expires and the MDSP fails to collect its rental, the MDSP sends an SMS to notify the user of recharge and the grace period is provided for the product. • In the grace period, the MDSP tries to collect the rental at 00:00 every day until the rental collection succeeds. • When the user uses services within the grace period, the MDSP also tries to collect the rental. • When the grace period ends and the MDSP still fails to collect the rental, the MDSP cancels the subscription relationship.	
Period Type	Cycle type of a product subscription relationship. • Day • Month • Multiday • Multimonth • Multiweek • Week • Bill Cycle For example, a CP/SP provides a package with a monthly rental of 50 cents and a multiday package with a rental of 20 cents for 10 days.	[Default Value] Month
Period Length	Length of a cycle. A cycle can be multiple weeks, multiple days, or multiple months. The length is the number of weeks, days, or months. NOTE: This parameter is valid only when **Period Type** is set to **Multiday**, **Multimonth**, or **Multiweek**.	[Value Range] The value is an integer ranging from 2 to 20000.

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
Period start type	Start time when the subscription relationship of a product takes effect. Options are as follows: • Product subscription day The subscribed product takes effect immediately. • Fixed day in a month The subscribed product takes effect on a fixed date in a month.	[Default Value] Product subscription day
Fixed day in a month	Fixed day in a month. After a user subscribes to a product, the product is charged from the next bill cycle. For example, a carrier sets the fixed charging date in a month to 5. After a user subscribes to the product, the cycle is from the 5th day in this month to the 5th day in the next month. NOTE: This parameter is valid only when **Period Type** is set to **Month** or **Multimonth**.	[Value Range] The value is an integer ranging from 1 to 28.
Access Code	Access code of the platform to which SMS messages for product subscription are sent. The access code can be used by: The MDSP to identify a subscription product. The combination of **Access Code** and **Order Command** uniquely identifies a subscription product in the MDSP. The SAG to route subscription requests to required SPs. NOTE: Product access codes must be different from service access codes.	[Value Range] A numeric string.

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
Subscription Command Word (en)	Subscription command word for the SMS and MMS service product. A user can edit the command word in an SMS message and send the message to the corresponding gateway to subscribe to a product.	[Value Range] The value is a string of digits, uppercase letters, and lowercase letters.
Unsubscription Command Word (en)	Unsubscription command word for the SMS and MMS Service Product. A user can edit the command word in an SMS message and send the message to the corresponding gateway to unsubscribe from a product.	[Value Range] The value is a string of digits, uppercase letters, and lowercase letters.
Charge Occasion	Options are as follows: • MO: mobile originating application terminated. • MT: application originating mobile terminated.	[Default Value] MT
Measure Mode	Measuring mode of a product.	[Default Value] By Times
Measure Unit	Measuring unit of a product.	[Default Value] Times
Price Group	Whether to enable the group pricing. The MDSP allows subscription products to be priced based on user groups so that CPs/SPs can customize charging policies based on user groups.	[Default Value] No default value
Rental(Cent)	Rental that the CP/SP is charged for a product (money).	[Value Range] The value is a positive integer. [Note] The currency unit is configurable and it’s in Kenyan cents.
First-Time rental collection triggering rule	Policy of charging rentals for the first time. After a subscription	[Default Value]

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
	relationship takes effect, the CP/SP charges the user the rental for the subscribed product. Options are as follows: • Immediately after subscription • Period free of charge NOTE: Carriers do not charge users any fee within the period free of charge.	Immediately after subscription
First-Time rental collection triggering rule value	Number of periods free of charge in the first-time rental charging policy. NOTE: This parameter is valid only when **First-Time rental collection triggering rule** is set to **Period free of charge**.	[Value Range] The value is an integer ranging from 1 to 999.
First-Time rental collection triggering rule unit	Unit of the fee collected based on the first-time rental charging policy. NOTE: This parameter is valid only when **First-Time rental collection triggering rule** is set to **Period free of charge**.	[Default Value] Days
First-Period rental collection rule	Policy of charging rentals for the first cycle. Charge the monthly rental no matter when the user subscribes to the product.	[Default Value] Deduct entire fee despite subscription day
Charging Mode	Charging rate mode. Options are as follows: Simple Tariff Free of charge Graded Tariff	[Default Value] Simple Tariff

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
	Complex Tariff	
Simple Tariff		
Used Fee (Cent)	Real-time fee for using a product (money).	[Value Range] The value is a positive integer. [Note] The currency unit is configurable and it’s in Kenyan cents.
Graded Tariff		
Grading Mode	Type of calculating grading tariffs. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	[Default Value] By Accumulation: indicates that the graded tariff is set based on users' accumulated usage.
No.	Tariff grade ID. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	[Note] • Click **Add** to add a grade. • Click **Delete** to delete a grade.
Start Point	Start point of the range corresponding to a tariff grade. CPs/SPs specify tariff intervals based on users' usage. For example, the mapping between the number of times products are used and fees collected is as follows: 1 to 10 times: 50 cents; 11 to 20 times: 40 cents; equal to or more than 21 times: 20 cents. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	[Note] The system automatically sets a value for this parameter.
End Point	End point of the range	[Value Range]

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
	corresponding to a tariff grade. CPs/SPs specify tariff intervals based on users' usage. For example, the mapping between the number of times products are used and fees collected is as follows: 1 to 10 times: 50 cents; 11 to 20 times: 40 cents; equal to or more than 21 times: 20 cents. NOTE: This parameter is valid only when **Charging Mode** is set to **Graded Tariff**.	The value is a positive integer and is greater than the value of the start point.
Used Fee (Cent)	Fee for an accumulated volume range (money). Users are charged according to different charging rates of traffic volume ranges. For example, the mapping between the number of times products are used and fees collected is as follows: 1 to 10 times: 50 cents; 11 to 20 times: 40 cents; equal to or more than 21 times: 20 cents.	[Value Range] The value is a positive integer. [Note] The currency unit is configurable and it’s in Kenyan cents.
Complex Tariff		
Access Channel	A user can obtain a service through various access channels. The carrier can charge users based on channels. Options are as follows: • All • CC (CRM) • IVR • PDA • SMS • USSD • WAP (WAP1x) • WEb • XHTML (WAP 2)	[Default Value] WAP (WAP1x)
Charging Mode	Charging rate mode. Options are as follows:	[Default Value] Free

Table 2 Description of parameters on the page for pricing subscription SMS and MMS service products

Parameter	Description	Setting
	Simple Tariff Free	
Used Fee (Cent)	Real-time fee for using a product (money).	[Value Range] The value is a positive integer. [Note] The currency unit is configurable and it's in Kenyan cents.