



MMS Gateway

API Reference

v2.3



MMSGWAPI v2.3

Last modified: February 5, 2007

Copyright © 2007 VeriSign, Inc. All rights reserved.

The information in this document belongs to VeriSign. It may not be used, reproduced or disclosed without the written approval of VeriSign.

DISCLAIMER AND LIMITATION OF LIABILITY

VeriSign, Inc. has made efforts to ensure the accuracy and completeness of the information in this document. However, VeriSign, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. VeriSign, Inc. assumes no liability to any party for any loss or damage (whether direct or indirect) caused by any errors, omissions, or statements of any kind contained in this document.

Further, VeriSign, Inc. assumes no liability arising from the application or use of the product or service described herein and specifically disclaims any representation that the products or services described herein do not infringe upon any existing or future intellectual property rights. Nothing herein grants the reader any license to make, use, or sell equipment or products constructed in accordance with this document. Finally, all rights and privileges related to any intellectual property right described herein are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner. VeriSign Inc. reserves the right to make changes to any information herein without further notice.

TRADEMARKS

VeriSign, the VeriSign logo, and other trademarks, service marks, and logos are registered or unregistered trademarks of VeriSign and its subsidiaries in the United States and in foreign countries. Other trademarks and service marks in this document are the property of their respective owners.

This document supports the named release and all subsequent releases unless otherwise indicated in a new edition or release notes. This document may describe features and/or functionality that are not present in your software or your service agreement. Contact your account representative to learn more about what is available with this VeriSign product. If you need help using this product, contact customer support.



Contents

1	Introduction	1
2	Mobile-Terminated MMS Messages	3
	MMS Gateway Web Service URLs	3
	Sending an MMS Message	3
	sendMessage Parameters	3
	sendMessage Example	5
	sendMessage HTTP Response	5
	MMS Gateway Notifications	5
	Notification Parameters	5
3	Mobile-Originated MMS Messages	9
	Requirements	9
	Message Flow	9
	Sample MM7	10
4	Call Flow Design	13
	Call Flow Elements	13
	User Registration	13
	Handset Discovery	13
	Content Push	13
	Billing	14
	Example Call Flows	14
	A. Web-Initiated, SMS Opt-In, Discovery Push, Single Billing	15
	B. Web-Initiated, PIN Opt-In, Handset Wizard, Single Billing	16
	C. MO-Initiated, SMS Opt-In, Discovery Push, Single Billing	17
	D. Web-Initiated, PIN Opt-In, Discovery Push, Subscription Billing	18
	E. Web-Initiated, PIN Opt-In, Handset Wizard, Subscription Billing	20
	F. MO-Initiated, SMS Opt-In, Discovery Push, Subscription Billing	22
	Appendix A: Carrier MMS Capabilities	25
	Verizon Wireless	25
	Approved Handsets	25
	Handset Notes	27
	Purchase Methods	27
	Billing Methods	28
	Supported Content Types and Restrictions	28
	Additional Video Clip Parameter Information	28
	Other Restrictions	29
	Approved Message Text	29
	Sprint	29
	Handset Information	29
	Message Restrictions	29
	Additional Video Clip Parameter Information	30



Appendix B: Troubleshooting	31
Appendix C: Developer FAQ	33
Appendix D: Lab Escalation Policy	35



1 Introduction

The MMS (Multimedia Messaging Service) standard is designed to deliver multiple formats of content in new ways to the most advanced mobile handsets, including video clips, various types of ringtones, and “slideshow” type messages that integrate several types of content into one delivered message.

VeriSign’s MMS Gateway features an HTTP-based API via which Content Providers can submit MMS content for delivery to mobile handsets on any North American mobile carrier that has opened up its system to allow off-net MMS downloads.

This API Reference is intended for Content Provider partners and describes in detail how to send content through MMS Gateway. It includes information on handset discovery, billing, and approved call flows. The manual is divided into the following sections:

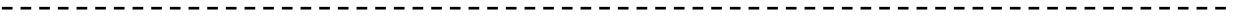
- [Mobile-Terminated MMS Messages](#)
- [Mobile-Originated MMS Messages](#)
- [Call Flow Design](#)
- [Appendix A: Carrier MMS Capabilities](#)
- [Appendix B: Troubleshooting](#)
- [Appendix C: Developer FAQ](#)
- [Appendix D: Lab Escalation Policy](#)

A Note about Carrier MMS Availability

The various North American mobile network carriers have all deployed MMS to different extents, but most only offer in-garden content such as picture messaging and exchange services. Currently, only **Verizon Wireless** and **Sprint** offer full MMS off-net content delivery via MMS Gateway. However, more carriers are planning to connect during 2007. Your Account Manager will notify you as new carriers come online.

Additional References

- MMS Specifications: <http://www.3gpp.org/specs/specs.htm>
- MM7 Specification: http://www.3gpp2.org/Public_html/SC/X.S0016-370-A_v1.0_060124.pdf
- Sample Libraries: http://developer.openwave.com/dvl/tools_and_sdk/mms_sdk/index.htm



2 Mobile-Terminated MMS Messages

The MMS Gateway API consists of a single HTTP-based Web service (`sendMessage`) that allows you to provide a mobile number, indicate the appropriate short code, and specify the location of the MMS content you intend to deliver.

MMS Gateway Web Service URLs

VeriSign hosts two separate MMS Gateway instances:

Lab MMS Gateway	A test environment against which you can develop and verify your MMS applications. The Web service URL is: http://mms-test.m-qube.com:6008/campaignapp/post/sendMessage
Production MMS Gateway	A live production environment for applications that have passed the necessary carrier certification. The Web service URL is: http://mms.mqube.us/campaignapp/post/sendMessage

Basic HTTP authentication is required for both MMS Gateway instances. Please contact your Account Manager for a valid username and password.

Sending an MMS Message

To send either individual or subscription-billed MMS content, call the `sendMessage()` Web service via an HTTP 1.1 POST or GET operation.

`sendMessage` Parameters

The `sendMessage()` call takes the following input parameters:

Parameter	Description
<code>destination</code>	The phone number of the subscriber's handset, in UMDA format. For example: <code>tel:5008885555</code>
<code>serviceCode</code>	The provisioned short code/service code to be used for this message. For example: <code>13012</code>
<code>subject</code>	The MMS subject line you want to appear on the handset (text, URL-encoded, 40 characters maximum). For example: <code>MMS+Content</code>
<code>senderID</code>	The sender ID that is passed through to the handset. Always set this to the same value as the <code>serviceCode</code> parameter. For example: <code>13012</code>



clientTransactionID	Every message submitted to the VeriSign system is assigned a standard 28-character ID that serves to uniquely identify the message in subsequent transactions. However, you can also use the clientTransactionID parameter to pass along any additional message-identifying string your application may require. The value you enter here accompanies the message all the way to the handset and is echoed back in the MMSC Delivery Report. (Text, 64 characters maximum, no spaces.) For example: OURSERV68446
contentFileName	File name used to identify the content on the handset (text, 64 characters maximum). For example: HeyYa.mid Note that some handsets will overwrite the supplied file name with their own default name.
contentURL	The URL of the content you want delivered to the handset (256 characters maximum). MMS Gateway retrieves this content file via an HTTP GET. For example: http://ourserver.com/HeyYa.mid
contentSize	The size of the content file being delivered to the handset, in bytes. For example: 8064
contentType	Indicates the kind of content you're sending to the handset. This is required for content verification purposes. For example: audio/midi
handsetID	The subscriber handset ID, as previously detected by handset discovery or a website handset wizard. For example: audio8900 For discovery messages, leave this parameter blank. NOTE: The Verizon Wireless MMSC uses a proprietary handset ID string (see Appendix A: Carrier MMS Capabilities). Note that if the handset isn't supported, you must not attempt to deliver content and should instead send the user an appropriate unsupported device message.
contentFlock	Determines whether the content is forward locked (protected). Set to true or false.
discovery	Determines if this is a zero-rated handset discovery message that should not be charged to the customer. Set to true or false. See Handset Discovery for more information.
notifyURL	Sets the URL to which MMS Gateway should send its message notifications (256 characters maximum). For example: http://ourserver.com/notify.jsp?cid=OURSERV68446 See MMS Gateway Notifications for more information
interpret	This parameter is required by the VeriSign messaging platform and must always be set to true for MMS.
dialog.op	This parameter is required by the VeriSign messaging platform and must always be set to mms_push for MMS messages.



Please also see the appropriate section in [Appendix A: Carrier MMS Capabilities](#) for any carrier-specific requirements for the `sendMessage()` parameter values.

sendMessage Example

```
http://mms-test.m-qube.com:6008/campaignapp/post/sendMessage?destination=tel%3A5008885555&serviceCode=13012&contentURL=http://ourserver.com/HeyYa.mid&subject=MMS+Content&senderID=13012&clientTransactionID=OURSERV68446&contentFileName=HeyYa.mid&handsetID=audio8900&contentFlock=true&discovery=false&notifyURL=http://ourserver.com/notify.jsp?cid=OURSERV68446&interpret=true&dialog.op=mms_push
```

sendMessage HTTP Response

If the `sendMessage()` call is successful, the body of the HTTP response contains a unique 28-character message ID. For example:

Message Id: 1j0j9u0002bres006s43i3iu9mi0

All subsequent message notifications returned to you by MMS Gateway will reference this message ID.

If the `sendMessage()` call fails, the HTTP response header contains the relevant error code. For example:

HTTP/1.1 500 03573- Invalid or non deliverable UMDA tel:6170000000

MMS Gateway Notifications

For each `sendMessage` call, MMS Gateway sends you one of the following message notifications:

Submit Response Failure Notification

Only sent if the MM7 Submit Response from the carrier MMSC indicates an error.

Delivery Report Notification

Sent as soon as the carrier MMSC has replied with the final message Delivery Report.

These message notifications take the form of an HTTP POST in URL-encoded format (`application/x-www-form-urlencoded`) and are sent to the URL defined in the `sendMessage notifyURL` parameter. It's your responsibility to have a server program running at the specified URL that will accept the HTTP POST. If a receiving program isn't present, or if the POST fails for any reason, MMS Gateway will retry delivery for a set period of time. After that, it places the notifications in a dead letter queue. Upon request, VeriSign will attempt to resend the queued notifications later.

Notification Parameters

Each notification POST contains both the full XML Submit Response or Delivery Report received from the carrier MMSC, along with additional information generated by the VeriSign platform by default.

Parameter	Description
messageID	The 28-character message ID you received from MMS Gateway in response to the original <code>sendMessage</code> HTTP POST operation.
clientTransactionID	The <code>clientTransactionID</code> value you passed in via the <code>sendMessage()</code> call.



outputCode	The delivery status of the message. Successful delivery is indicated by a value of PBS (Premium Billing Succeeded) or an empty string.
errorMessage	A detailed error message describing the failure (if any).
serviceCode	The short code/service code used for the message.
deliveryResponse	The complete Delivery Report or Submit Response XML string that MMS Gateway received from the carrier MMSC. The following elements within the XML are of particular interest for Delivery Report notifications: <MMStatus> Shows the status of the message (Expired, Retrieved, Rejected, Indeterminate, Forwarded). A Delivery Report MMStatus of "Indeterminate" means the handset to which the message was sent does not support MMS. In this case, your application needs to generate the appropriate response. <UACapabilities UAProf> A User Agent Profile string that identifies the make and model of handset to which the message was delivered. See Chapter 4: Call Flow Design for more information about how you can use these elements to validate the target handset for an MMS delivery.
destination	The phone number of the subscriber's handset in UMDA format.
outputState	The overall status of the message. C - Confirmed delivery T - Failure
hs.txid hs.method hs.serviceId hs.senderId hs.timeout hs.url	Additional internal tracking parameters used by the VeriSign messaging platform. These values can be ignored for MMS processing purposes.

Delivery Report Notification Example

The example below shows the complete string posted to the notify URL by MMS Gateway for a successful Delivery Report. The report data is enclosed within the <DeliveryReportReq> tag.

```
messageID=1s0j9u01iuk4am037e4533cg419g&clientTransactionID=12345&hs.method=POST&output  
Code=&errorMessage=&serviceCode=13012&hs.txid=1s0j9u01iuk4am037e4533cg419g  
&deliveryResponse=<?xml version="1.0"?>  
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">  
<env:Header>  
<mm7:TransactionID xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140  
/schema/REL-6-MM7-1-4" env:mustUnderstand="1">033633EBF51800003F00000101  
</mm7:TransactionID>  
</env:Header>  
<env:Body>
```



```
<DeliveryReportReq xmlns="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/
schema/REL-6-MM7-1-4">
<MM7Version>6.8.0</MM7Version>
<MessageID>033633EBF51800003F000001</MessageID>
<Recipient>
<Number>5008885555</Number>
</Recipient>
<Sender>
<Number>2111122221</Number>
</Sender>
<Date>2006-03-27T13:15:21-08:00</Date>
<MMStatus>Retrieved</MMStatus>
<UACapabilities UAProf="sama950" TimeStamp="2006-03-27T13:15:21-08:00" />
</DeliveryReportReq>
</env:Body>
</env:Envelope>&hs.serviceId=1s0j9u00526eo403po44b2j7pb8g&destination=tel:5008885555&h
s.senderId=tel:5008885555&hs.timeout=15000&hs.url=http://ourserver.com&outputState=C
```

If the handset to which the message was delivered isn't MMS-capable, the Delivery Report

<MMStatus> tag has the value Indeterminate:

```
<DeliveryReportReq xmlns="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/
schema/REL-5-MM7-1-2">
<MM7Version>5.3.0</MM7Version>
<MessageID>20060512185549652</MessageID>
<MMStatus>Indeterminate</MMStatus>
<Sender>
<Number>12345</Number>
</Sender>
<Recipient>
<Number>55512345678</Number>
</Recipient>
<Date>2005-03-42T11:31:46-06:00</Date>
</DeliveryReportReq>
```

See [Content Push](#) for more information about handling indeterminate handsets.

Submit Response Failure Notification Example

The example below shows the string posted to the notify URL by MMS Gateway for a failed Submit Response.

```
messageID=1s0j9u008ilarn03tm45i2fk6qtg&clientTransactionID=12345&hs.method=POST&output
Code=DLV&errorMessage=Address Error&serviceCode=2111122221&hs.txid=1s0j9u008ilarn03tm4
5i2fk6qtg&deliveryResponse=<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Header>
<mm7:TransactionID xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/
schema/REL-5-MM7-1-2" env:mustUnderstand="1">20060512185549652@192.168.10.179
</mm7:TransactionID>
</env:Header>
<env:Body>
<SubmitRsp xmlns="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-
MM7-1-2">
```



```
<MM7Version>5.3.0</MM7Version>
<Status>
<StatusCode>2002</StatusCode>
<StatusText>Address Error</StatusText>
</Status>
<MessageID>0339CF5235A4000060000001</MessageID>
</SubmitRsp>
</env:Body>
</env:Envelope>&hs.serviceId=1s0j9u00526eo403po44b2j7pb8g&destination=tel:5008885555&h
s.senderId=tel:5008885555&hs.timeout=15000&hs.url=http://ourserver.com&outputState=T
```

3 Mobile-Originated MMS Messages

This chapter explains the requirements for handling MO-based MMS messages, describes the message flow, and presents a sample MM7 message.

Requirements

To implement MO-based MMS messages, a Content Partner must provide the following:

- An MMS Shortcode configured by each carrier to handle MO-MMS traffic and route said traffic to VeriSign's MMS Gateway.
- A web application that conforms to each carrier's MO-MMS requirements. VeriSign will need to know the URL for this web application in order to provision MMS Gateway.

The web application must meet the following conditions:

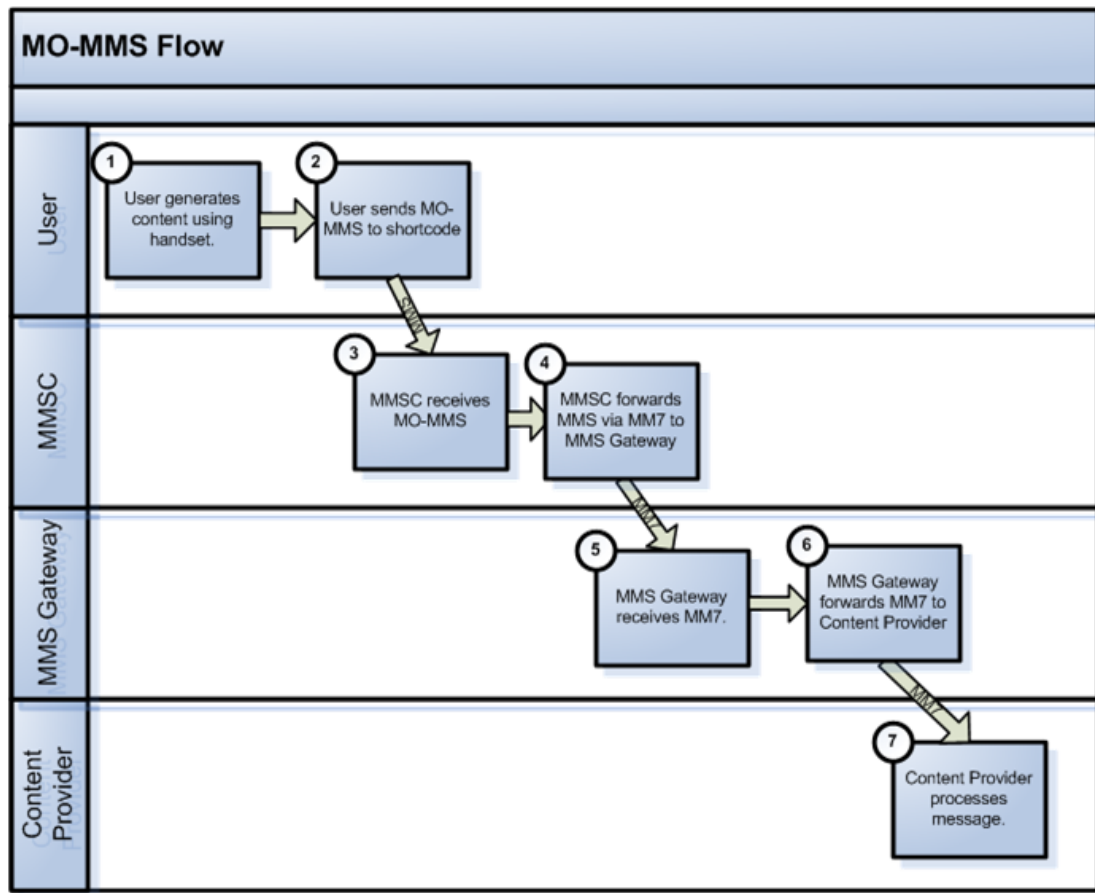
- Upon receiving a message, the application must respond with an HTTP Status of 200 to indicate that it has successfully accepted the message for processing. (MMS Gateway does not require the application to respond with an MM7 Deliver Request Response.) If MMS Gateway receives an HTTP Status other than 200, it will periodically attempt to resubmit the POST for up to 24 hours before marking the message as failed.
- If the application is designed to process keywords in the MMS message, it must parse both the subject and text body fields.

Message Flow

VeriSign MMS Gateway supports a single standard message flow for MO-MMS, although there may be additional application-specific elements allowed within this generic framework (such as the Content Provider responding with an MMS or SMS message). If your application requires a radically different use case, please consult your Account Manager.

The standard message flow is as follows:

- 1 User generates content on a mobile device.
- 2 User sends the content via MMS to a shortcode provisioned in the carrier's MMSC.
- 3 Carrier MMSC receives the MMS.
- 4 Carrier MMSC routes the MMS via an MM7 Deliver Request (MM7_deliver.REQ) to the VeriSign MMS Gateway.
- 5 VeriSign MMS Gateway receives the MM7 Deliver Request.
- 6 VeriSign MMS Gateway forwards the MM7 Deliver Request to the Content Provider's Web application via an HTTP POST.
- 7 Content Provider's application processes the message.



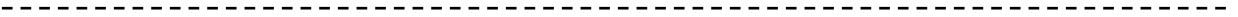
Sample MM7

The sample below shows the MM7 Deliver Request message posted by MMS Gateway to the Content Provider's application URL. Note that this MM7 is configured for Verizon Wireless and that the contents may be slightly different for other carriers.

```
Content-Type: multipart/related; boundary="Part_2_1550013.1161008324771";
type="text/xml"; start="<mm7-deliver>"
--Part_2_1550013.1161008324771
Content-Type: text/xml
Content-Transfer-Encoding: base64
Content-ID: <mm7-deliver>
<?xml version="1.0"?><env:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header>
    <mm7:TransactionID xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/
23.140/schema/REL-6-MM7-1-4">0damij014g208a08124750bt42ch</mm7:TransactionID>
  </env:Header>
  <env:Body>
    <mm7:DeliverReq xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/
23.140/schema/REL-6-MM7-1-4">
      <MM7Version>6.8.0</MM7Version>
```



```
<VASPID>MQUBE_TEST</VASPID><VASID>MQUBE_TEST</VASID>
<LinkedID>0isi28m14g208a08124750bt42cg</LinkedID>
<Sender><Number>8885551212</Number></Sender>
<Recipients><To><ShortCode>12345</ShortCode></To></Recipients>
<TimeStamp>2006-08-20T04:20:00-05:00</TimeStamp>
<Priority>normal</Priority>
<Subject>Deliver Request Test</Subject>
<UACapabilities UAProf="vx6000" />
<Content href="cid:bball.mid"/>
</mm7:DeliverReq>
</env:Body>
</env:Envelope>
--Part_2_1550013.1161008324771
Content-type: audio/midi
Content-Transfer-Encoding: 7bit
Content-ID: <bball.mid>
TVRoZAAAAAYAAQACASxNVHJrAAAA6gD/AxBBc2NlbnRfVG9fQ2hhcmdl
AP9YBAQCGAgA/1EDB6EghAj/UQMHkPt1/1EDB5D7gUv/UQMRB6BPv9RAWCK4oIK/1EDBsgcTP9RAwaiXoE+/
1EDBmb8gSX/UQMG
...
AEd1gRY7AABHAIeWL2WBFi8AADN1gRYZAAA2ZYEWNgAAR2UAO2WBFkCAADSagRY2ZYEWNgAAR2UAP2UAQmUAO
2WFBkCAAD8AAEIA
ADSAAp8vAA==
--Part_2_1550013.1161008324771--
```



4 Call Flow Design

Correct call flow design is essential in order to gain carrier approval of your application. This chapter introduces the basic elements that make up an MMS Gateway call flow and describes several example flows.

Call Flow Elements

A call flow into MMS Gateway consists of four basic phases:

- User Registration
- Handset Discovery
- Content Push
- Billing

User Registration

In most call flows, the subscriber requests an MMS download either from a website (web-initiated) or by sending an SMS to a particular short code (MO-initiated). Both cases require the usual double opt-in. For web-initiated call flows, this is achieved by sending an SMS containing a PIN, which the subscriber must enter on the web site. For MO-initiated flows, the subscriber typically replies to an SMS.

For web-initiated flows, you can also incorporate a handset wizard that allows subscribers to select their handset make and model from a series of menus. If you choose to detect the handset in this way, you'll need to validate the data against the User Agent Profile string returned in the Delivery Report. (See [Content Push](#) below.)

Handset Discovery

Since knowing the display capabilities of the handset is a critical element in any MMS delivery, some carrier MMSCs provide handset discovery mechanisms. Handset discovery is inherently more accurate than asking subscribers to select their phone from a handset wizard, and also makes it possible for you to implement MO-initiated call flows in addition to web-initiated. Because each carrier MMSC may implement the handset discovery process differently, the MMS Gateway API generalizes the process using the discovery parameter.

You initiate handset discovery in your call flow by submitting a free test piece of content (for example, a JPEG image) to the handset for the express purpose of obtaining the handset User-Agent Profile. Setting the `discovery=true` argument in the `sendMessage()` call causes MMS Gateway to submit the message to the MMSC as zero-rated so the subscriber won't incur any costs. When the handset receives the message, the MMSC returns a Delivery Report containing a `<UACapabilities>` tag that you can use to determine the handset.

Note that handset discovery messages are meant to be free to the end user. For additional carrier-specific requirements, please see [Appendix A: Carrier MMS Capabilities](#).

Content Push

In the Content Push phase, your application passes the actual content to MMS Gateway via a `sendMessage()` call. If MMS Gateway receives a Submit Response failure from the carrier MMSC, it



will notify you accordingly. Otherwise, you'll receive a Delivery Report once the message has reached the handset.

If you allowed the subscriber to use a handset wizard to determine the make and model of the target device during the User Registration phase, your application must check the <UACapabilities> tag returned in the Delivery Report to make sure the handsetID parameter was correct.

If the handset isn't MMS-capable, the status of the Delivery Report <MMstatus> tag has the value Indeterminate. In this case, your application should notify the subscriber via a suitable SMS message and (if applicable) cancel any subscription for which the subscriber has signed up.

Billing

MMS Gateway supports both single item and subscription billing. In each case, the Content Provider triggers the billing event by sending the subscriber a premium-rated SMS message via VeriSign's separate billing API. For subscriptions, monthly billing occurs via premium SMS on the anniversary date. For more information, see the *SMS Gateway API Reference* and the *SMS Gateway Subscriptions API Guide*.

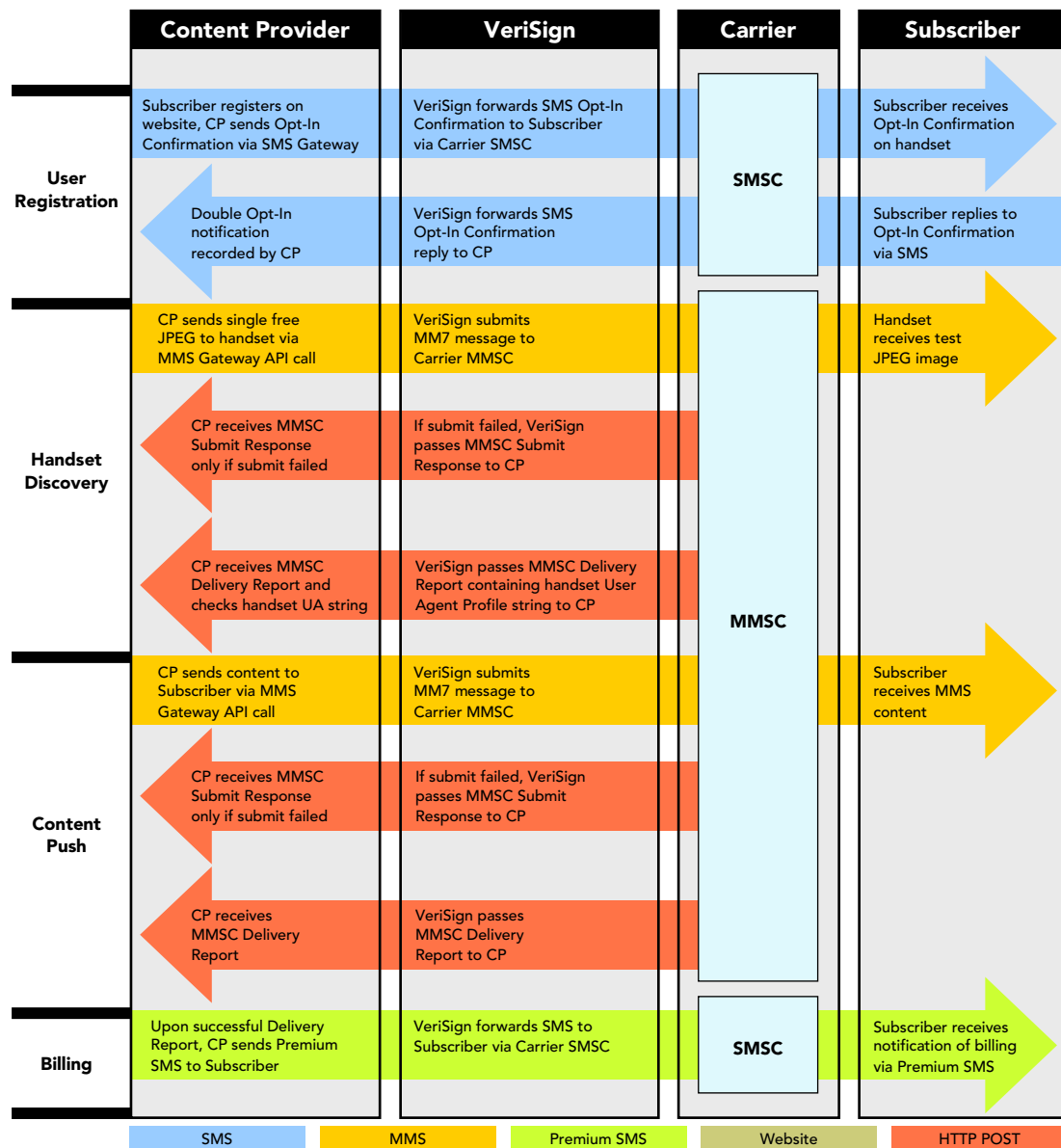
Example Call Flows

The following are call flow examples for MMS off-net content delivery:

- A. Web-Initiated, SMS Opt-In, Discovery Push, Single Billing**
- B. Web-Initiated, PIN Opt-In, Handset Wizard, Single Billing**
- C. MO-Initiated, SMS Opt-In, Discovery Push, Single Billing**
- D. Web-Initiated, PIN Opt-In, Discovery Push, Subscription Billing**
- E. Web-Initiated, PIN Opt-In, Handset Wizard, Subscription Billing**
- F. MO-Initiated, SMS Opt-In, Discovery Push, Subscription Billing**



A. Web-Initiated, SMS Opt-In, Discovery Push, Single Billing

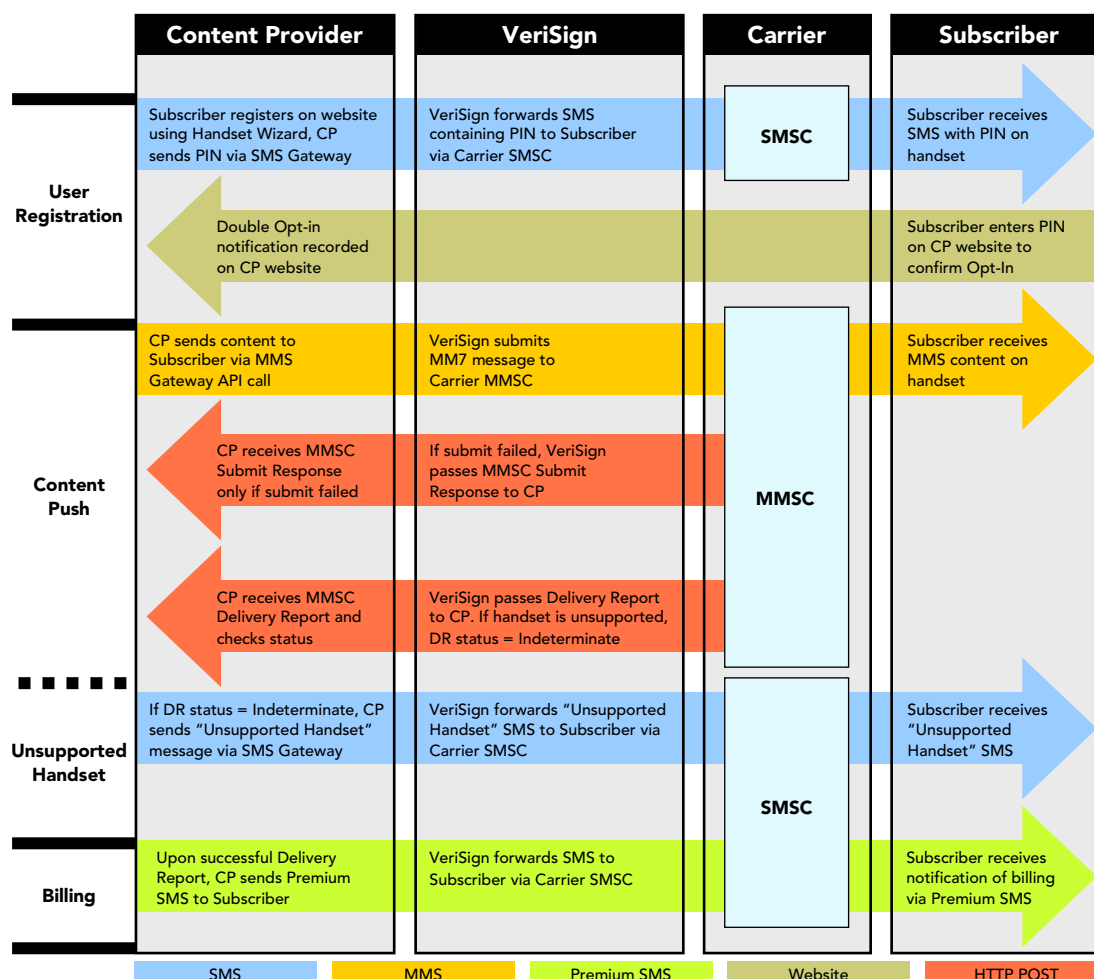


In this example, the subscriber requests content from the web and confirms opt-in via SMS. The Content Provider performs handset discovery by sending a test piece of content via the MMS Gateway API with `discovery=true` and retrieves the User Agent Profile string from the resultant Delivery Report. The User Agent is then used as the `handsetID` parameter in the Content Push.

Upon receiving a successful Delivery Report from the carrier MMSC, the Content Provider triggers a billing event by sending a premium-rated SMS message to the subscriber.



B. Web-Initiated, PIN Opt-In, Handset Wizard, Single Billing



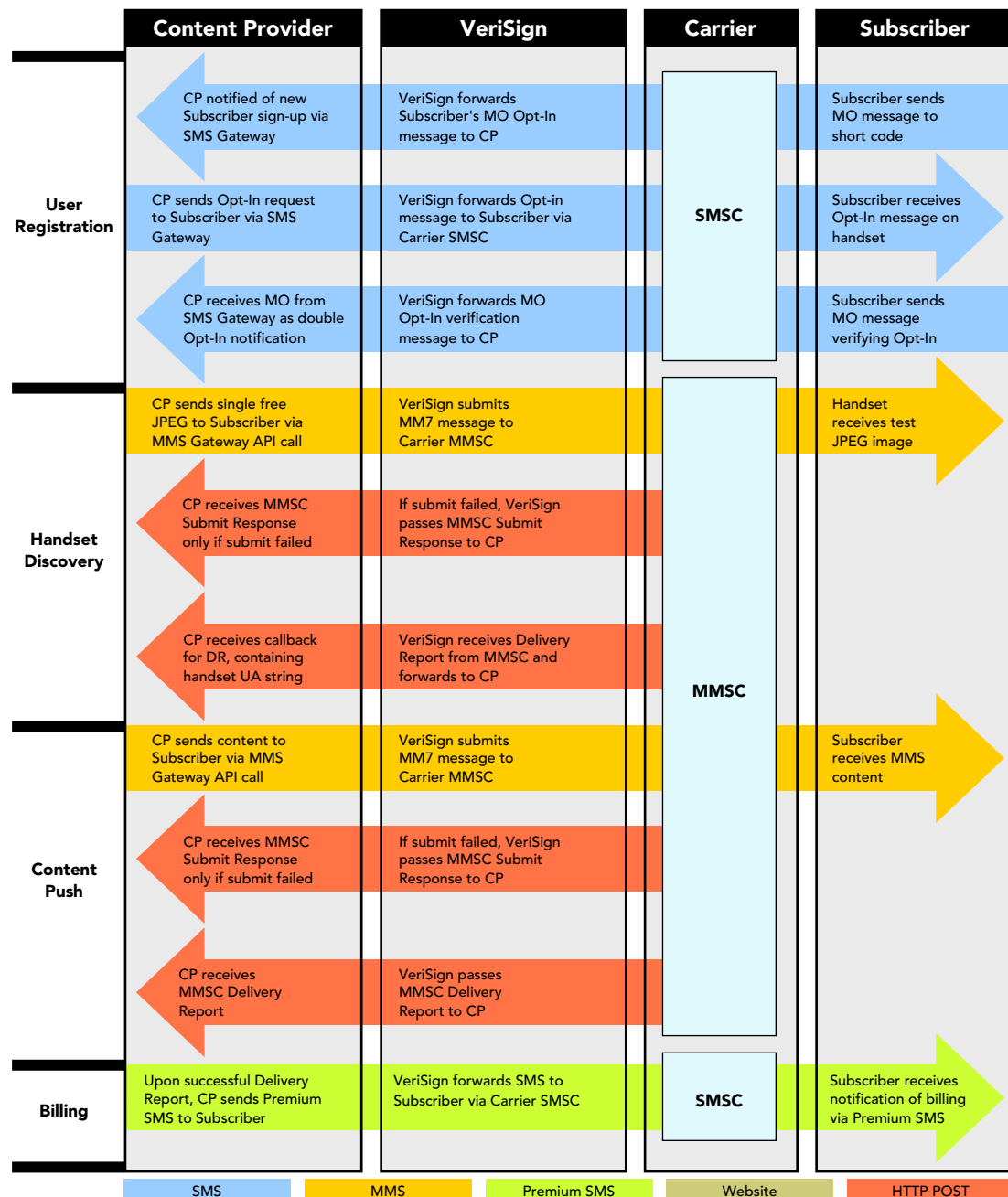
In this example, the subscriber requests content from the web and indicates his or her handset via a handset wizard. The Content Provider sends an SMS containing a PIN, which must be entered on the website to confirm opt-in. The handset wizard result is used as the handsetID parameter in the Content Push.

If the subscriber's handset is unsupported (non-MMS capable), the MMSC responds to the content push by returning a value of "Indeterminate" in the Status field of the Delivery Report. In this case, the Content Provider needs to send an unsupported handset message to the subscriber. The proper text to use for such a message is spelled out in the Program Brief and can vary from carrier to carrier.

Upon receiving a successful Delivery Report from the carrier MMSC, the Content Provider triggers a billing event by sending a premium-rated SMS message to the subscriber.



C. MO-Initiated, SMS Opt-In, Discovery Push, Single Billing

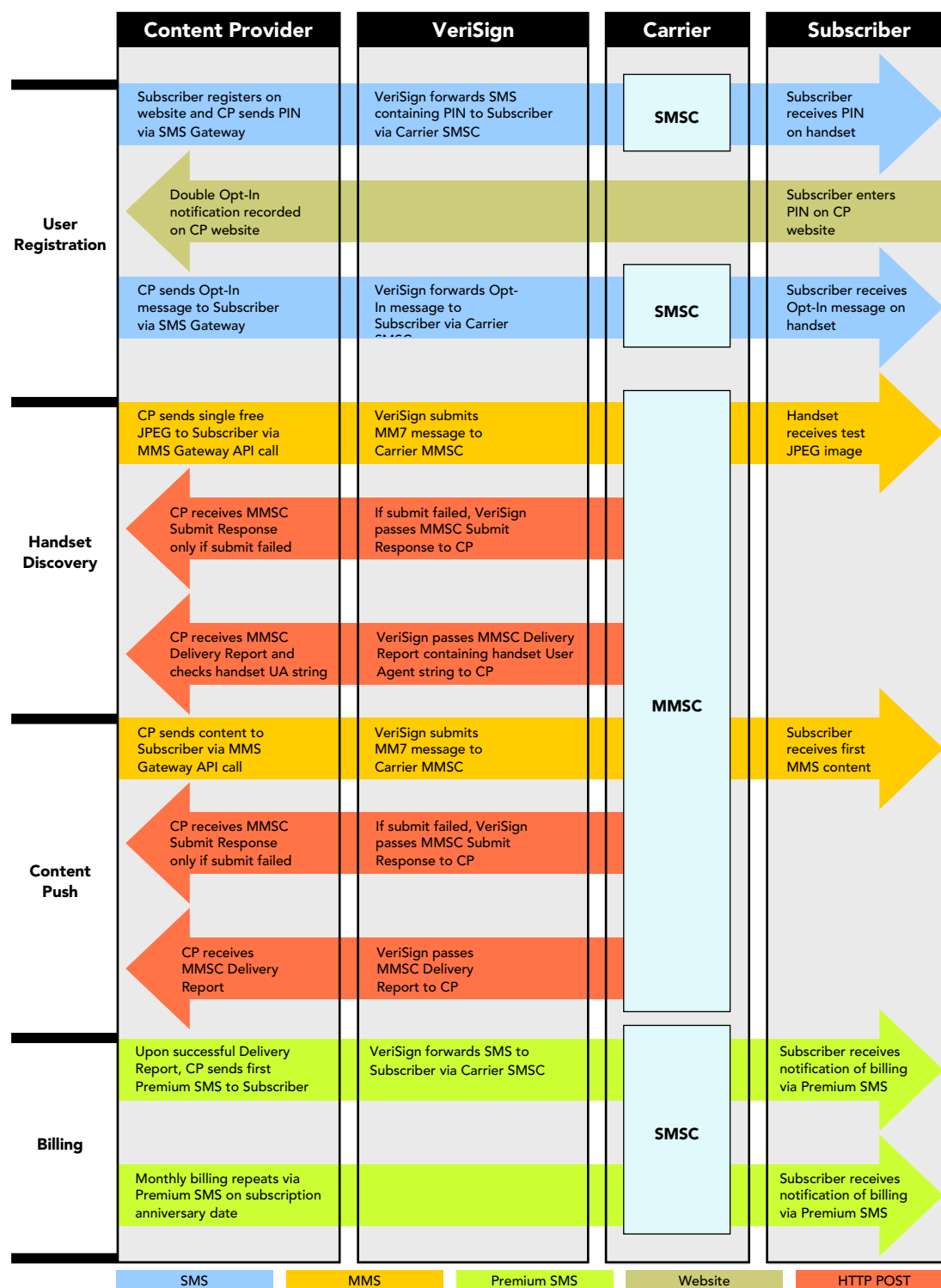


In this example, the subscriber begins the transaction by sending an SMS to a short code. The Content Provider sends back an SMS opt-in message. Once the subscriber confirms, the Content Provider performs handset discovery by sending a test piece of content via the MMS Gateway API with `discovery=true` and retrieves the User Agent Profile string from the resultant Delivery Report. The User Agent is then used in the Content Push.

Upon receiving a successful Delivery Report from the carrier MMSC, the Content Provider triggers a billing event by sending a premium-rated SMS message to the subscriber.



D. Web-Initiated, PIN Opt-In, Discovery Push, Subscription Billing





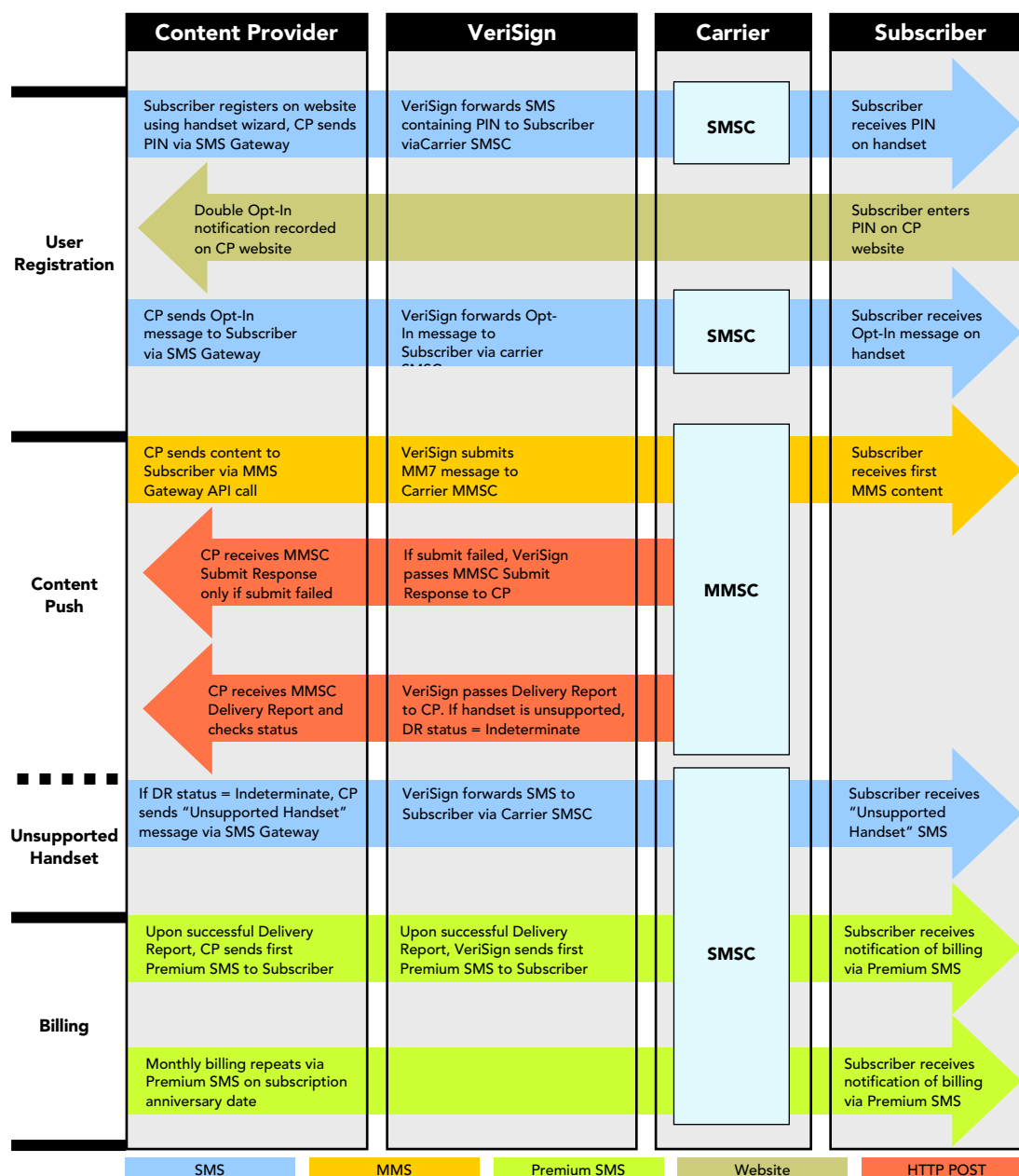
In this example, the subscriber signs up for a subscription service from the web (for example, ring tone of the week). The Content Provider sends an opt-in confirmation message listing the price of the subscribed service and how to opt out or get help. The precise wording of this message has legal implications and is captured in detail in the Program Brief.

The Content Provider then performs handset discovery by sending a test piece of content via the MMS Gateway API with `discovery=true` and retrieves the User Agent Profile string from the resultant Delivery Report. The User Agent is then used in the Content Push.

Upon receiving a successful Delivery Report from the carrier MMSC, the Content Provider triggers a billing event by sending a premium-rated SMS message to the subscriber. Subsequently, monthly billing occurs via premium SMS on the anniversary date. (See the *SMS Gateway Subscriptions API Guide* for more information.)



E. Web-Initiated, PIN Opt-In, Handset Wizard, Subscription Billing



In this example, the subscriber signs up for a subscription service from the web and indicates his or her handset via a handset wizard. The Content Provider sends an opt-in confirmation message containing a PIN, which must be entered on the website to confirm opt-in. Once the subscriber confirms, the Content Partner sends a subscription welcome SMS describing how to opt-out or get help. The precise wording of this message has legal implications and is captured in detail in the Program Brief.

If the subscriber's handset is unsupported (non-MMS capable), the MMSC responds to the content push by returning a value of "Indeterminate" in the Status field of the Delivery Report. In this case,

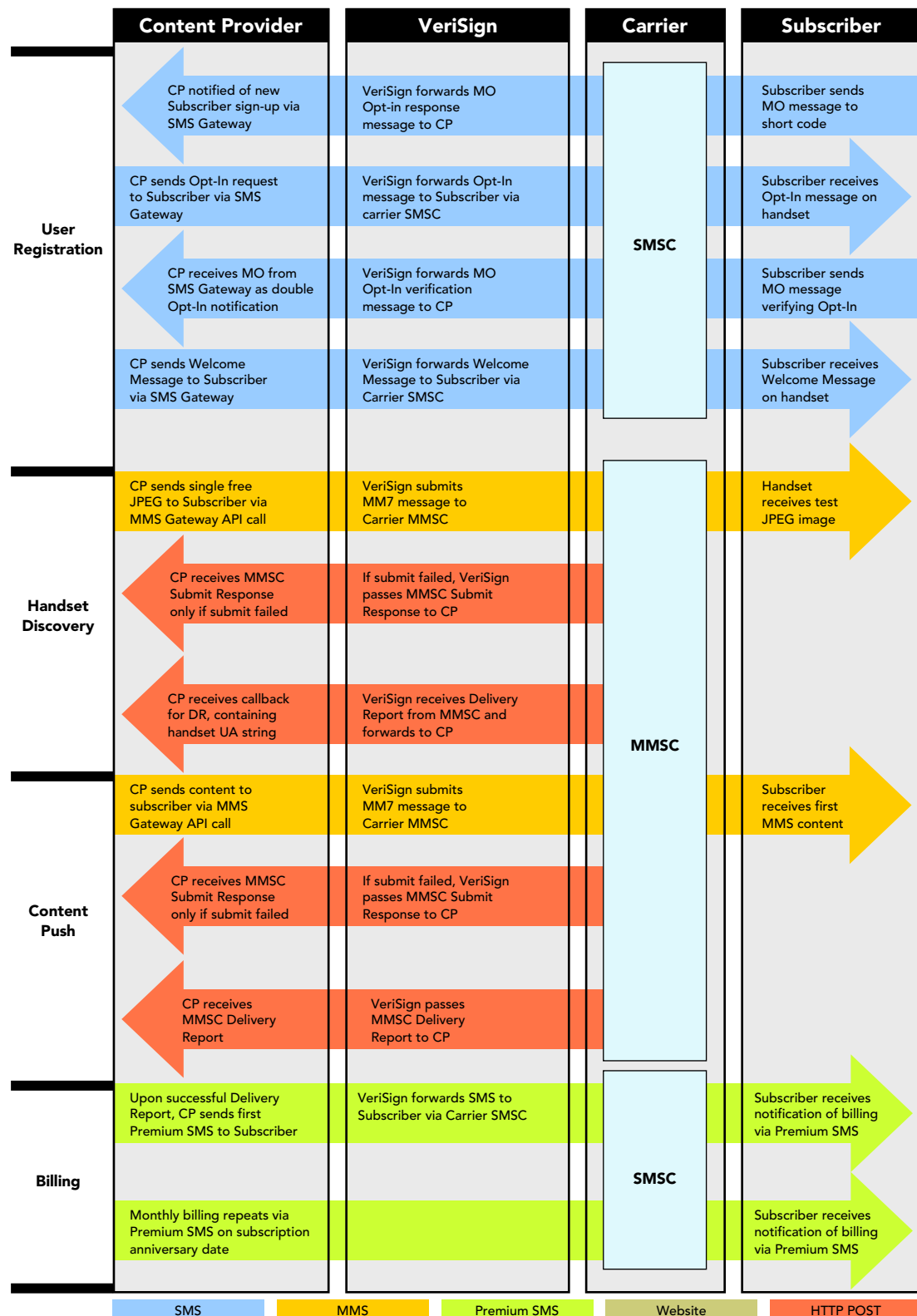


the Content Provider needs to send an unsupported handset message and also immediately opt the subscriber out of the service upon receiving the Delivery Report notification. The proper text to use for such a message is spelled out in the Program Brief and can vary from carrier to carrier.

Upon receiving a successful Delivery Report from the carrier MMSC, the Content Provider triggers a billing event by sending a premium-rated SMS message to the subscriber. Subsequently, monthly billing occurs via premium SMS on the anniversary date. (See the *SMS Gateway Subscriptions API Guide* for more information.)



F. MO-Initiated, SMS Opt-In, Discovery Push, Subscription Billing

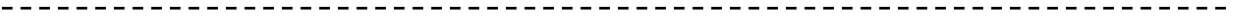




In this example, the subscriber begins the transaction by sending an SMS to a short code to sign up for a subscription. The Content Provider sends back an SMS opt-in message. Once the subscriber confirms, the Content Partner sends a subscription welcome SMS describing how to opt out or get help. The precise wording of this message has legal implications and is captured in detail in the Program Brief.

After the subscriber has confirmed, the Content Provider performs handset discovery by sending a test piece of content via the MMS Gateway API with `discovery=true` and retrieves the User Agent Profile string from the resultant Delivery Report. The User Agent is then used in the Content Push.

Upon receiving a successful Delivery Report from the carrier MMSC, the Content Provider triggers a billing event by sending a premium-rated SMS message to the subscriber. Subsequently, monthly billing occurs via premium SMS on the anniversary date. (See the *SMS Gateway Subscriptions API Guide* for more information.)





Appendix A: Carrier MMS Capabilities

MMS content delivery is inherently different from WAP-based content delivery, and each carrier has its own requirements with respect to permissible call flows, billing requirements, and handset restrictions. This section describes each available carrier's MMS capabilities in detail and includes a list of handsets approved for MMS content delivery, the type of content allowed for each handset, and the maximum file sizes supported.

Please review these capabilities and restrictions carefully when developing an MMS application. If you have any questions about the information included here, please contact your Account Manager.

Verizon Wireless

Approved Handsets

The table below shows the handsets approved for MMS content delivery on Verizon Wireless. Since these details are subject to change, please contact your Account Manager to ensure you have the latest handset information.

	Handset ID	Max File	JPEG	MIDI	GIF	MP3	QCELP	3GP2	Pixels
Audiovox									
CDM180	audio180	128000K	Y	Y	Y	Y	Y	Y	160x99
CDM8900	audio8900	102400K	Y	Y	Y	Y	Y	Y	128x160
CDM8910	audio8910	102400K	Y	Y	Y	Y	Y	N	128x160
CDM8940	audio8940	350000K	Y	Y	Y	Y	Y	N	176x220
CDM8945	audio8945	204800K	Y	Y	Y	Y	Y	N	176x184
CDM9900	audio9900	200000K	Y	Y	Y	Y	Y	N	240x320
Kyocera									
KX2	kyokx2	350000K	Y	Y	Y	N	N	Y	128x131
KX323	KWC-K323	350000K	Y	Y	Y	Y	Y	N	128x160
LG									
VX5200	vx5200v1	125000K	Y	N	Y	N	N	Y	128x131
VX5300	vx5300v1	350000K	Y	Y	Y	Y	Y	N	128x160
VX6000	vx6000	102400K	Y	Y	Y	N	N	N	128x160
VX6000V2	vx6000v2	102400K	Y	Y	Y	N	N	N	128x160
VX6100	vx6100v1	125000K	Y	Y	Y	Y	Y	Y	128x160
VX7000	vx7000v1	350000K	Y	Y	Y	Y	Y	Y	176x184



	Handset ID	Max File	JPEG	MIDI	GIF	MP3	QCELP	3GP2	Pixels
VX8000	vx8000v1	350000K	Y	Y	Y	Y	Y	Y	176x184
VX8100	vx8100v1	350000K	Y	N	Y	N	N	Y	176x184
VX8300	vx8300v1	350000K	Y	Y	Y	Y	Y	Y	176x220
VX8500	vx8500v1	350000K	Y	Y	Y	Y	Y	Y	240x320
VX8600	vx8600v1	350000K	Y	Y	Y	Y	Y	Y	176x220
VX8700	vx8700v1	500000K	Y	Y	Y	Y	Y	Y	240x320
VX9400	vx9400v1	350000K	Y	Y	Y	Y	Y	Y	240x320
VX9800	vx9800v1	358400K	Y	N	Y	N	N	Y	320x256
VX9900	vx9900v1	350000K	Y	Y	Y	Y	Y	Y	320x240

Motorola

A840	motoa840	358400K	Y	Y	Y	Y	N	N	176x220
E815	motoe815	350000K	Y	Y	Y	Y	N	Y	176x220
K1c	motok1c	350000K	Y	Y	Y	Y	Y	Y	176x220
L7c	motol7c	350000K	Y	Y	Y	Y	Y	Y	176x220
V3c	motov3	350000K	Y	Y	Y	N	N	N	176x184
V260	motov260	350000K	Y	Y	Y	Y	N	N	128x128
V265	motov265	350000K	Y	Y	Y	Y	N	Y	128x128
V276	motov276	350000K	Y	Y	Y	N	N	N	128x128
V325	motov325	350000K	Y	Y	Y	N	N	N	176x184
V710	motov710	358400K	Y	Y	Y	Y	N	Y	176x220
Ve	motove	350000K	Y	Y	Y	Y	Y	Y	240x320

Nokia

6215i	nok6215v1	350000K	Y	Y	Y	Y	Y	N	128X128
6236i	nok6236v1	307200K	Y	Y	Y	Y	Y	Y	128x128
6256i	nok6256v1	307200K	Y	Y	Y	Y	Y	Y	128x160
6305i	nok6305v1	350000K	Y	Y	Y	Y	Y	N	176x220
6315i	nok6315v1	350000K	Y	Y	Y	Y	Y	Y	176x220

Pantech

PN-215	pant215v1	125000K	Y	N	Y	N	N	N	128x160
--------	-----------	---------	---	---	---	---	---	---	---------



	Handset ID	Max File	JPEG	MIDI	GIF	MP3	QCELP	3GP2	Pixels
Samsung									
A610	sama610	97280K	Y	Y	Y	N	Y	N	128x160
A670	sama670	97280K	Y	Y	Y	N	Y	N	128x160
A790	sama790	102400K	Y	Y	Y	N	Y	N	176x220
A795	sama795	102400K	Y	Y	Y	N	Y	Y	176x220
A850	sama850	125000K	Y	Y	Y	N	Y	N	128x131
A870	sama870	350000K	Y	Y	Y	N	Y	N	128x131
A890	sama890	325000K	Y	Y	Y	N	Y	Y	176x220
A930	sama930	350000K	Y	Y	Y	Y	N	Y	176x220
A950	sama950	325000K	Y	Y	Y	N	N	Y	176x184
A970	sama970	325000K	Y	Y	N	N	Y	Y	176x184
A990	sama990	325000K	Y	Y	Y	Y	Y	Y	240x320
U540	samau540	350000K	Y	Y	Y	Y	N	Y	176x220
UTStarcom									
G'zOne	utstargz1	350000K	Y	N	Y	N	N	Y	240x320
G'zOne Type S	utstargz1s	350000K	Y	Y	Y	Y	Y	Y	

Handset Notes

The Verizon Wireless MMSC uses a proprietary handset ID string. For example, `motok1c` identifies the Motorola K1c. Content providers are expected to pass in only these values to specify a handset, not the traditional mobile browser UA string.

If a Verizon Wireless subscriber attempts to request content – either via a handset wizard or via handset discovery – from an unsupported device, you must send an error message as described in the [Approved Message Text](#) section below. You must not send any content or attempt to bill the subscriber if the handset is not listed as supported.

The handset discovery message (always an MT MMS)

- must always be free to the end user
- must contain either text or a still image of less than 30KB in size
- must state that the message is free but that but airtime charges may apply. For example:
“Thanks for your order, your content will arrive shortly. This message is Free. (Airtime charges still apply)”

Purchase Methods

Verizon Wireless currently supports the following purchase methods for MMS content:



- Web-Initiated, SMS Opt-In, Discovery Push
- Web-Initiated, PIN Opt-In, Handset Wizard
- SMS MO-Initiated, SMS Opt-In, Discovery Push
- Web-Initiated, PIN Opt-In, Discovery Push
- Web-Initiated, PIN Opt-In, Handset Wizard

Please see Chapter 4: [Call Flow Design](#) for a detailed description of each.

Billing Methods

Verizon Wireless allows premium SMS billing for both single purchases and subscriptions, as defined in the call flow models listed above.

Supported Content Types and Restrictions

Type	Content	MIME Type
Image	Joint Photographic Experts Group (JPEG)	image/jpeg
Image	Wireless Bitmap (WBMP)	image/vnd.wap.wbmp
Image	Portable Network Graphics (PNG)	image/png
Image	Windows Bitmap	image/bmp
Animation	Graphics Interchange (GIF)	image/gif
Audio	Adaptive multi-rate (AMR)	audio/amr
Audio	Windows WAV	audio/x-wav
Audio	Advanced Audio Coding (AAC)	audio/aac
Audio	Qualcomm Code Excited Linear Prediction (QCELP)	audio/qcp
Video	3GPP (MPEG4, QCELP)	video/3gpp
Video	3GPP2 (MPEG4, QCELP)	video/3gpp2
Text	UTF-8 text	text/plain; charset=UTF-8
Text	US-ASCII text	text/plain; charset=US-ASCII

Additional Video Clip Parameter Information

Video Parameter	Values
Maximum Frames Per Second	15 FPS (variable)
Quality of Video (Bit Rates)	96 kbps (variable)
Video Encoding Type	MPEG 4 Simple Profile 0, 1



Audio Encoding Type	QCELP, AMR, EVRC
File Extension	3g2, mp4, 3gp, qcp
MIME Type	video/3gpp2, video/3gpp, audio/amr, audio/mp4a-latm, video/mp4v-es, video/h263-2000
Maximum Message Size	200 KB
Maximum Message Text	1000 characters

Other Restrictions

Your Account Manager will supply you with additional documentation outlining specific content guidelines for Verizon Wireless off-net MMS content.

Approved Message Text

Verizon Wireless has advised that certain guidelines must be followed when sending subscribers a message in response to an error condition. The following table shows the approved message text:

Error Condition	Subscriber Message
Error encountered parsing MO-SMS	The request could not be understood. Please retry your request
Device does not support MMS	Your device is not capable of receiving the requested content. Visit www.verizonwireless.com for a list of Pix handsets.
Device does not support DRM	Your device is not capable for receiving rights protected content. Visit www.verizonwireless.com for more details.

Sprint

Handset Information

For detailed handset specifications, see the Sprint Application Developer site (<http://developer.sprint.com/>).

Message Restrictions

NOTE

All MMS content is transcoded by the Sprint Picture Mail system.



Parameter	Values
Maximum MT Message Size	250 KB NOTE: Maximum MO message size is handset-dependent.
Maximum Messages Per Second	1
Maximum Number of Videos/Slides Per Message	1

Additional Video Clip Parameter Information

Video Parameter	Values
File Format	3gpp or 3gpp2
File Extension	3gp, 3gpp, 3g2, 3gpp2
Maximum Number of Frames	225
Maximum Video Duration/Frames Per Second	Up to 30 seconds at 7.5 FPS or 15 seconds at 15 FPS (minimum frame rate above 0.5 FPS)
Maximum Video Bit Rate	<60K
Video Encoding Type	MPEG-4 VSP L0, L1, L2, H.263 Profile 3 Level 10, H.263 Profile 0 Level 10
Audio Encoding Type	AMR-NB, AMR-WB, AAC LC, QCELP - 13k (for 3g2 files)
Audio Bit Rate	QCELP = 14400 AMR-NB = 64000 AMR-WB = 7200 AAC-LC = 29500
Dimensions (Picture Mail)	176x144 (pixels)

Appendix B: Troubleshooting

Message Delivery

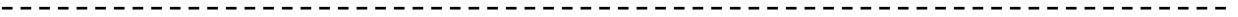
If you believe your application is working correctly but your MMS messages are not reaching the handset, please check the following:

- Are you certain the subscriber handset number is a valid on the carrier in question?
- Are you getting any notification back from VeriSign for failed Submit Responses? If so, see the error for more information.
- Are you getting a failed Delivery Report? If so, see the error for more information.
- Is the handset's inbox full? Certain handsets can run out of room and will not accept additional MMS notifications from an MMSC. Some MMSCs won't send if there are more than X messages (where X = 60 on Verizon Wireless) in a handset's inbox. Try deleting all messages from the handset and sending again.
- Is your `contentURL` parameter pointing to a valid file that has the correct permissions set in your application or web server?
- Does the handset support for the content type you're sending?

Message Forwarding

The MMS Gateway sets content DRM flags for forward locking on a per-carrier and per-handset basis. If content is arriving correctly on the handset but can be forwarded, check the following:

- Have you specified the `contentFlock=true` parameter?
- Are you setting the `handsetID` parameter correctly? MMS Gateway refers to this parameter to determine the correct DRM settings.





Appendix C: Developer FAQ

Q: Why do we have to use VeriSign's API? Why can't I use MM7?

A: VeriSign will support MM7 directly from Content Providers in a future release. For now, this HTTP API is the only way to deliver MMS content via VeriSign.

Q: Do you support the Synchronized Multimedia Integration Language (SMIL)?

A: Only for MO messages. SMIL for MTs will be added in a future release of MMS Gateway.

Q: When will you support feature X?

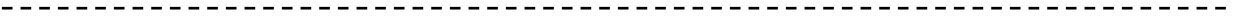
A: All feature requests are evaluated on their merits. If you have a specific feature request, please submit it to your Account Manager, who will ensure that VeriSign's Product Management team reviews it for inclusion in a future release.

Q: How do we bill on MMS?

A: All billing for MMS now occurs via a premium SMS that you (or VeriSign) submit after receiving a successful Delivery Report for a content item.

Q: What is a Program Brief and how do I get one?

A: The Program Brief describes all aspects of your program and how it works, who the proper points of contact in your organization are, etc. You must work with your Account Manager to create one.



Appendix D: Lab Escalation Policy

If you encounter a problem in the Lab MMS Gateway environment that isn't covered by the information in this document, please use the Client Support Ticketing system to escalate the issue to our Developer Support team.

To use the ticketing system, go to <http://www.m-qube.com/clientsupport> and log in with your VeriSign username and password. If you do not have the necessary login credentials, please contact your Account Manager.

Once you've accessed the ticketing system:

- 1 Click on **Log a Case**.
- 2 Under **Incident Level**, select "4. Other Time Sensitive Incident."
- 3 Under **Case Reason**, select "MMS Lab Developer Support."

Please fill out all the other fields as completely as possible. Be sure to include the phone numbers of any test handsets for which you need to trace messages, and also any messageIDs you received in response to message attempts. Steps to reproduce your problem are also essential, so please include anything you think might help VeriSign Developer Support resolve your issue.

You will receive an automated email response, and a Developer Support team member will reply to the case as soon as possible. Please keep all communication about the case in the actual ticket, since this allows us to examine the history of a case more easily than reconstructing e-mail threads.

NOTE

All communication regarding Lab MMS Gateway issues must be sent through the ticketing system and NOT through mmslab-escalation@m-Qube.com. This e-mail address is no longer active.

If you need to escalate a problem in the Production MMS Gateway environment, please consult the VeriSign Operations Point of Contact listed in your Program Brief.