NowSMS MMSC Training: MMS Protocol Overview

NOVEMBER 2015

MMS Overview

MMS Architecture Overview

MMS Delivery Process

Integration With Other Network Components

- SMS - SMPP
- User Authentication - WAPGW / RADIUS / ICAP
- Accounting & Charging - HTTP / DIAMETER
- MNP & Message Routing

MMSC (MMS Centre)

Provides store-and-forward MMS Messaging Service for mobile phone subscribers

Temporarily stores MMS messages awaiting subscriber retrieval

May interconnect with MMSCs for other network operators, either with direct connections, or via an aggregator

May convert MMS messages to other formats (e.g., SMS with web link) for non-subscribers or external recipients

May allow MMS submissions from Value Added Service Providers

May apply content adaptation on message content based upon capabilities of the receiving device

Multimedia Messaging Service (MMS)

Designed to compliment SMS by providing support for multimedia content

- Text
- Images
- Video
- Audio
- Contact Objects
- Calendar Objects
- SMIL Presentation (legacy support)

MMS and Group Messaging

Supports multiple recipients and group conversations

The biggest driver of MMS message traffic growth has been its use in modern smartphones for group messaging

Even when the content is text only, MMS is preferred for group messaging because recipients can reply all, allowing all group members to see all messages.

MMS Message Structure

An MMS Message has a similar structure to an SMTP E-Mail Message

Structure based upon SMTP and Multipart Internet Mail Extensions (MIME)

- Envelope – Recipients for in-transit message instance.
- Headers – Sender, Displayed Recipients, Subject, other attributes
- Content – MIME encoded multipart content. One or more multimedia objects (text, image, video, etc.)

MMS Envelope

Routing Attributes (sender and recipient) for an instance of an in-transit message.

Example: A message is sent to multiple recipients, some of which are subscribers of a different mobile network. The message may be split by the MMSC for delivery to another MMSC. The MMS envelope for the message instance sent to the other MMSC will be only the remote recipients. The envelope list of recipients tells the other MMSC which recipients to deliver the message to.

The MMS Header will still contain a list of all recipients to provide full group messaging support.

In MM4/SMTP these are represented by MAIL FROM: / RCPT TO: commands.

MMS Headers

Sender

Recipients (To/CC/BCC)

Subject

Priority

Message Class (Personal, Advertisement)

Message Type (can be delivery report or read report)

MMS Content

Multipart (MIME) object containing one or more of the following:

- Text
- Images
- Video
- Audio
- Contact Objects
- Calendar Objects
- SMIL Presentation (legacy support)

MMS Protocols

Overall Architecture: 3GPP TS 23.140
http://www.3gpp.org

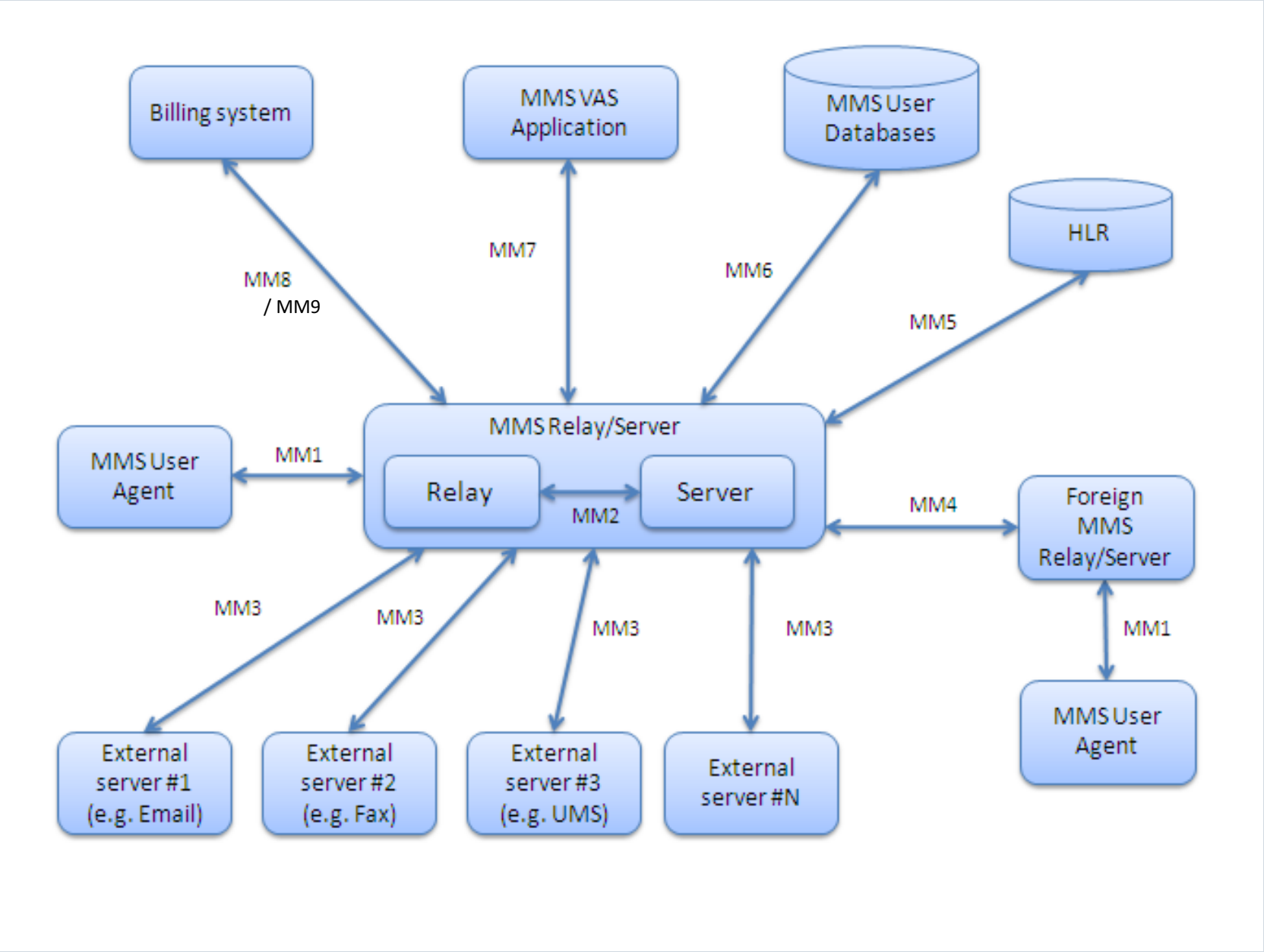
Over-the-Air Phone to MMSC: Open Mobile Alliance (OMA) – MMS Encapsulation Protocol
http://www.openmobilealliance.org

External Connectivity Protocols: 3GPP TS 23.140

- MM4 – Interoperator
- MM7 – Value Added Service Provider

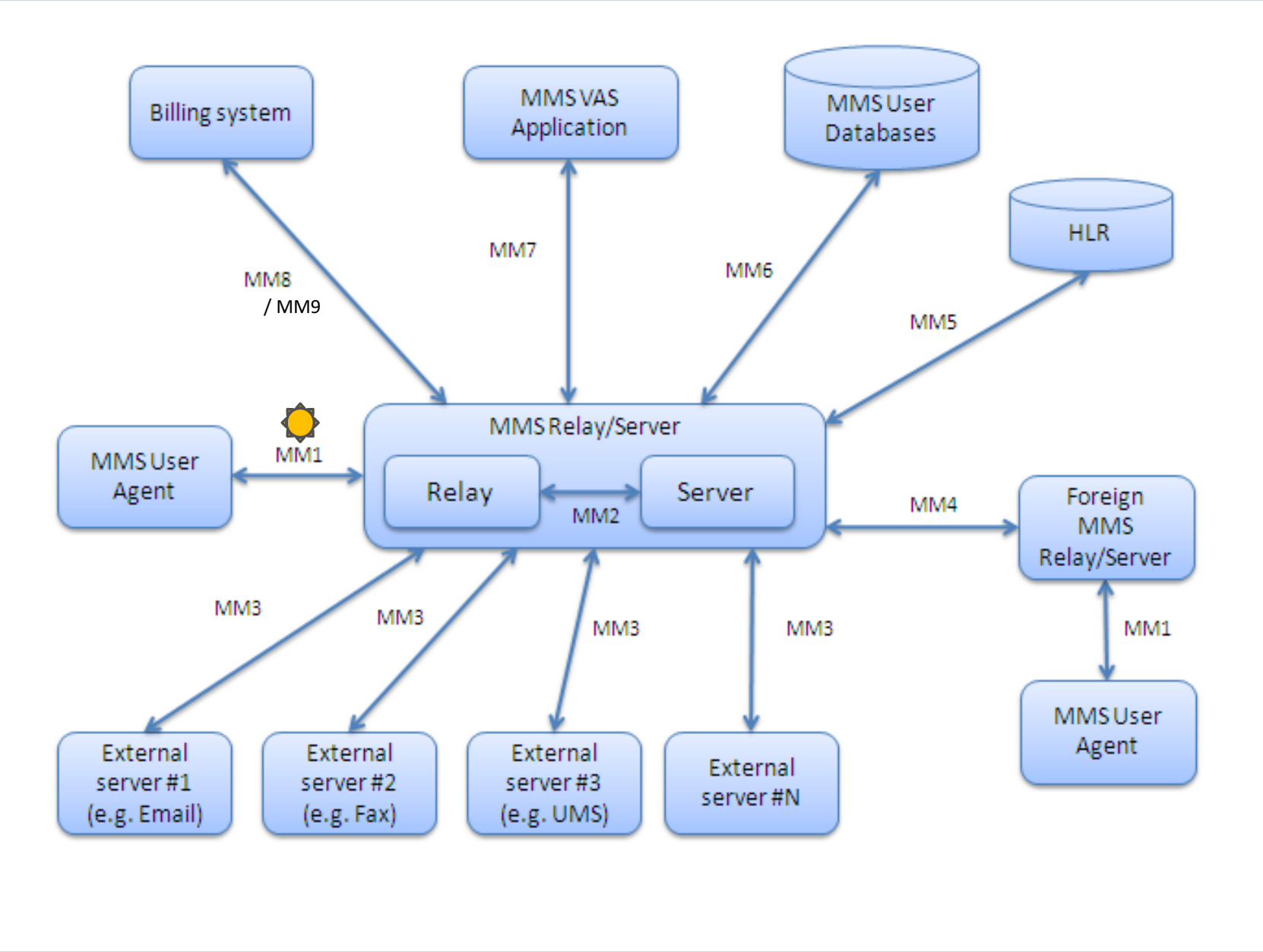
MMS Protocols

- MM1 thru MM11
- Some protocols are fully defined (MM1, MM4, MM7)
- Other protocols are loosely defined, conceptual, and implementation dependent
- Defined by 3GPP TS 23.140



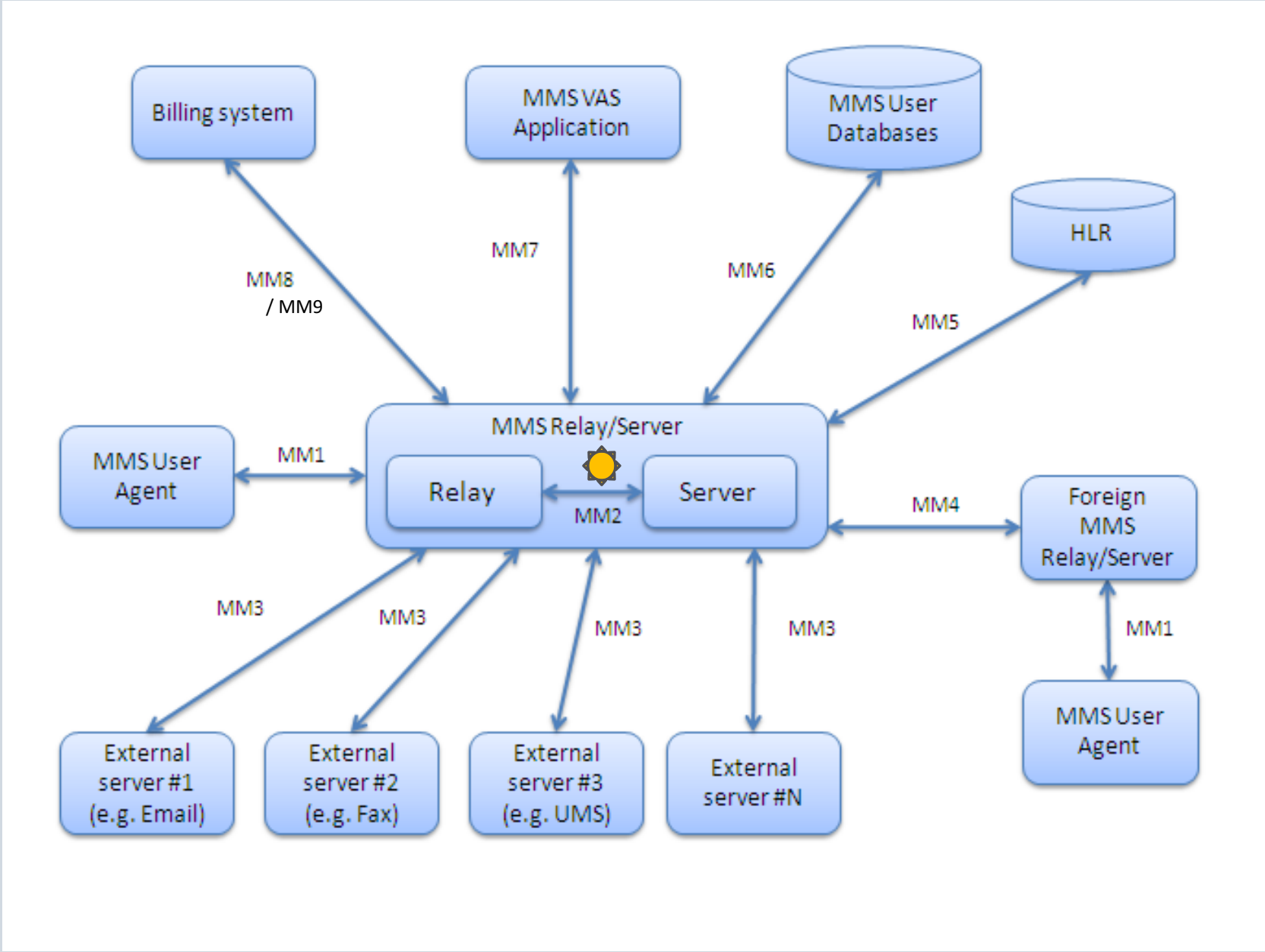
MM1

- Interface between mobile station and MMSC
- Based on HTTP, WAP Push and MIME
- MMS content encoded as MIME payload using WAP/WSP binary encoding
- Defined by Open Mobile Alliance (OMA) – MMS Encapsulation Protocol



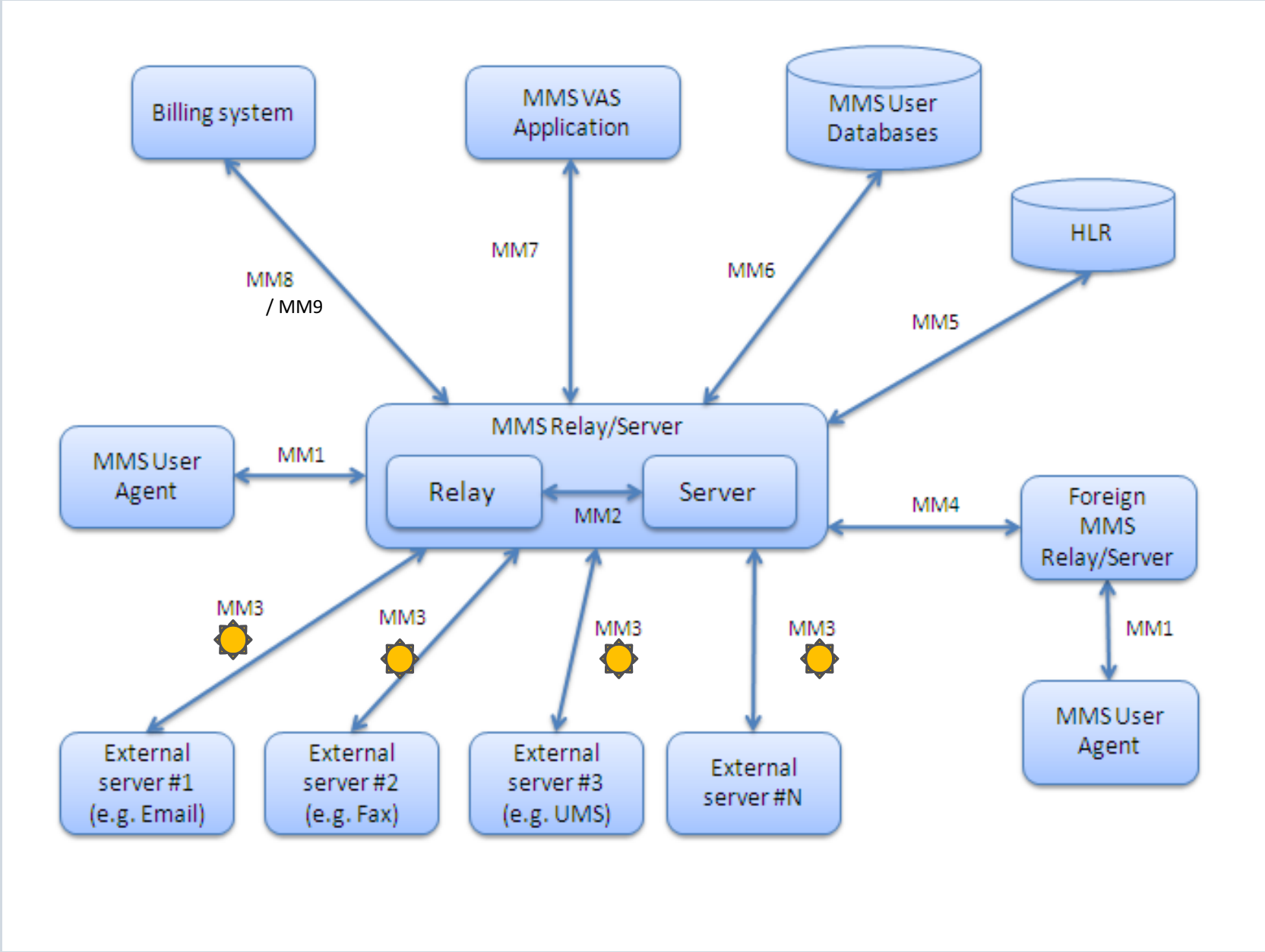
MM2

- Conceptual interface between two internal elements of an MMSC
- No actual API or protocol defined
- MMS Relay provides client interfaces
- MMS Server provides storage services
- Commercial MMSC implementations combine functions



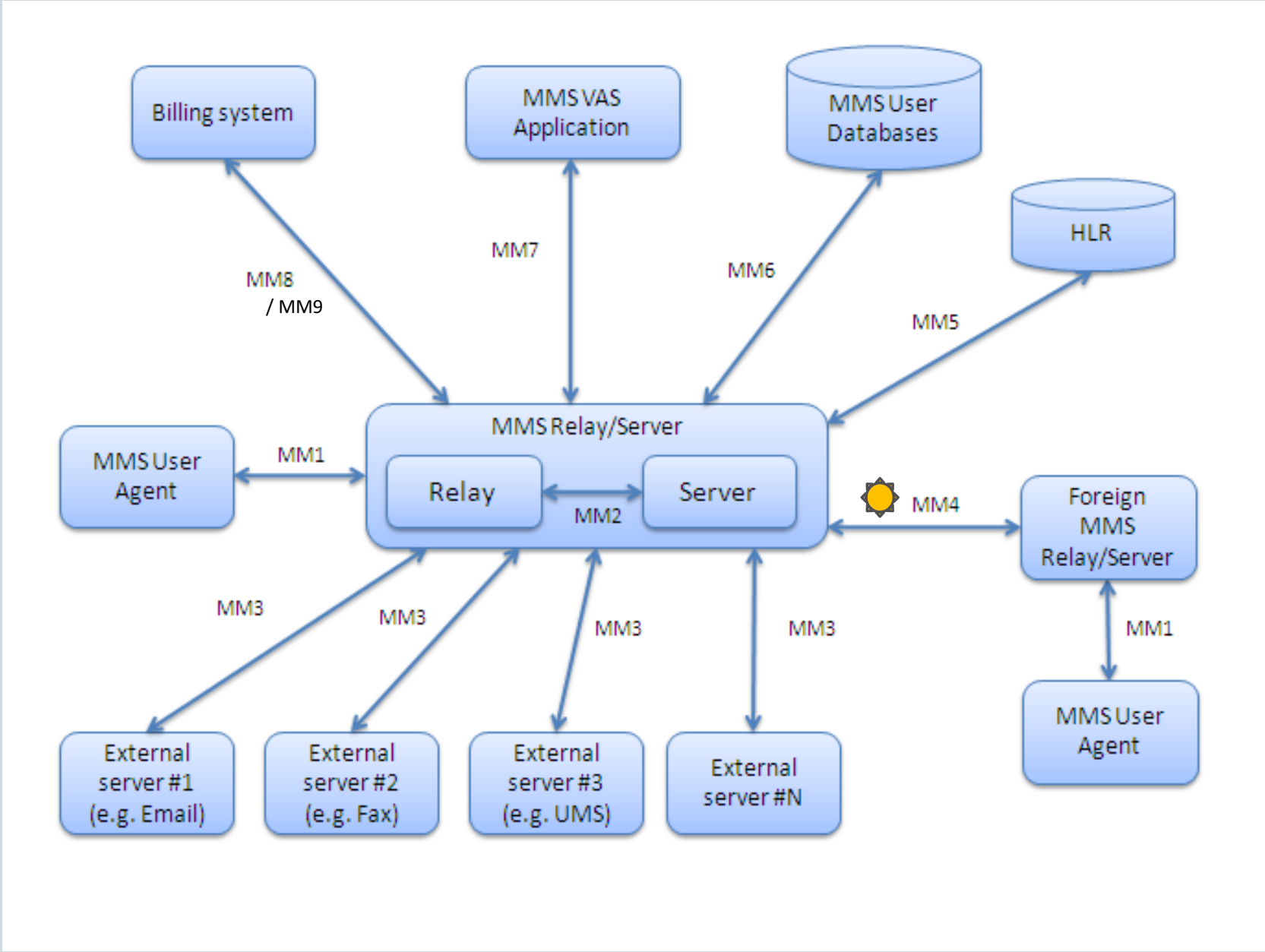
MM3

- Conceptual interface between MMSC and external servers such as E-Mail, FAX and Voice Mail
- No actual API or protocol defined – use of SMTP suggested



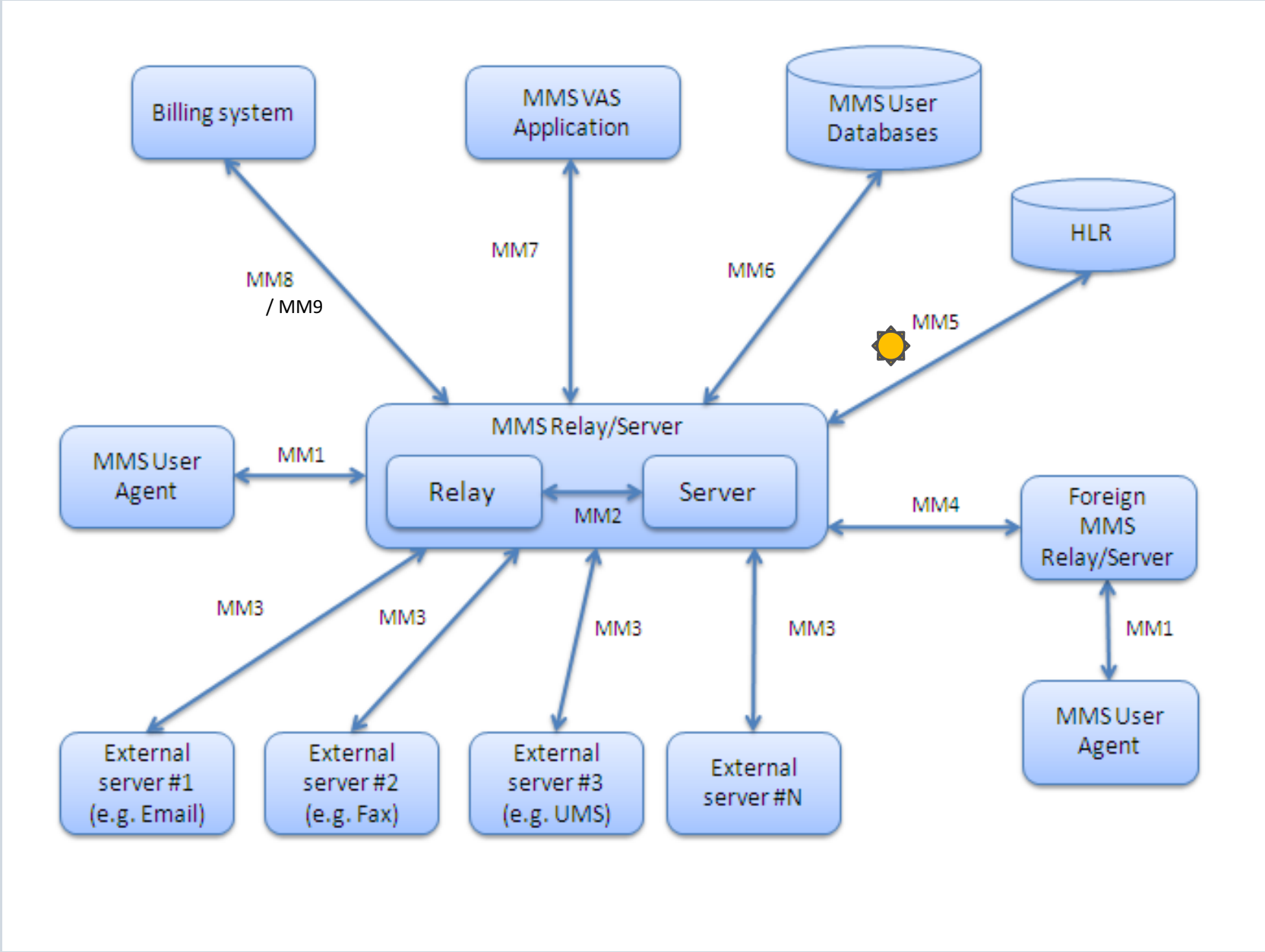
MM4

- Interface between multiple MMSCs
- SMTP based protocol defined by 3GPP TS 23.040
- Additional X-MMS SMTP headers defined
- Different PDUs mapped to SMTP headers



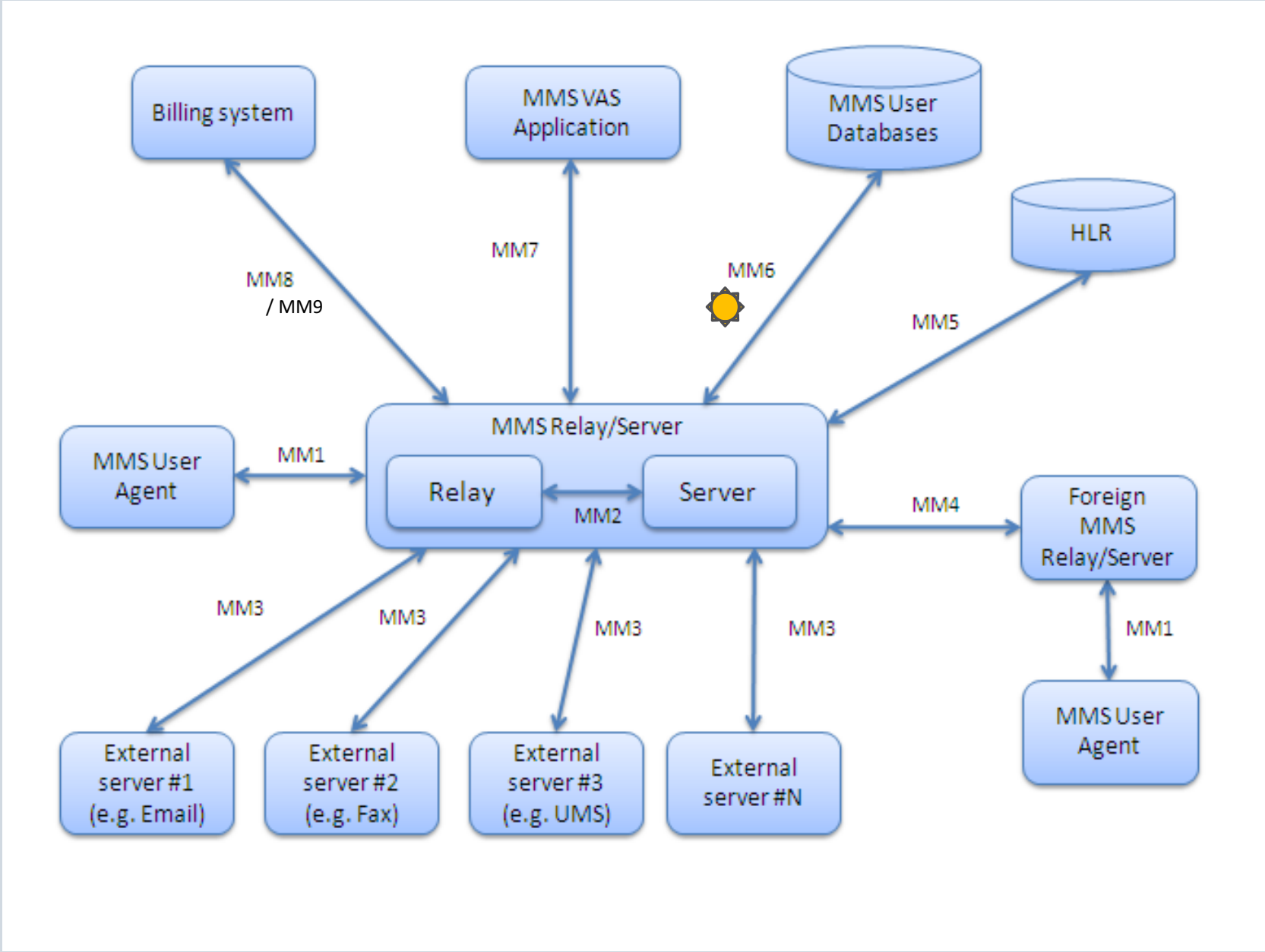
MM5

- Conceptual interface between MMSC and HLR for message routing lookup
- No actual API or protocol defined
- HLR lookup does not provide routing info for subscriber MMSC
- NowSMS MMSC uses HTTP-based routing callbacks



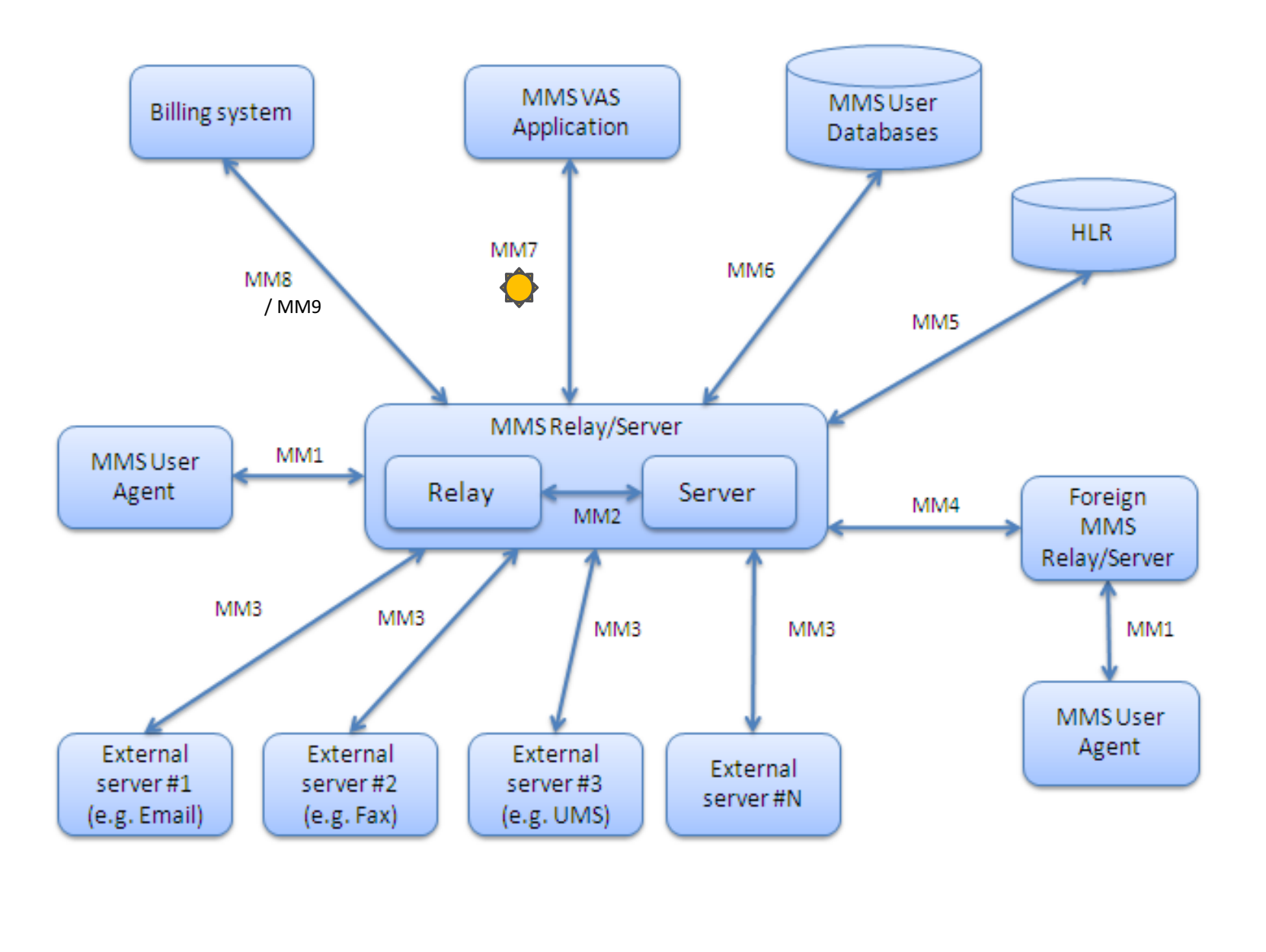
MM6

- Conceptual interface MMSC and subscriber database
- No actual API or protocol defined
- NowSMS MMSC uses HTTP-based accounting callbacks



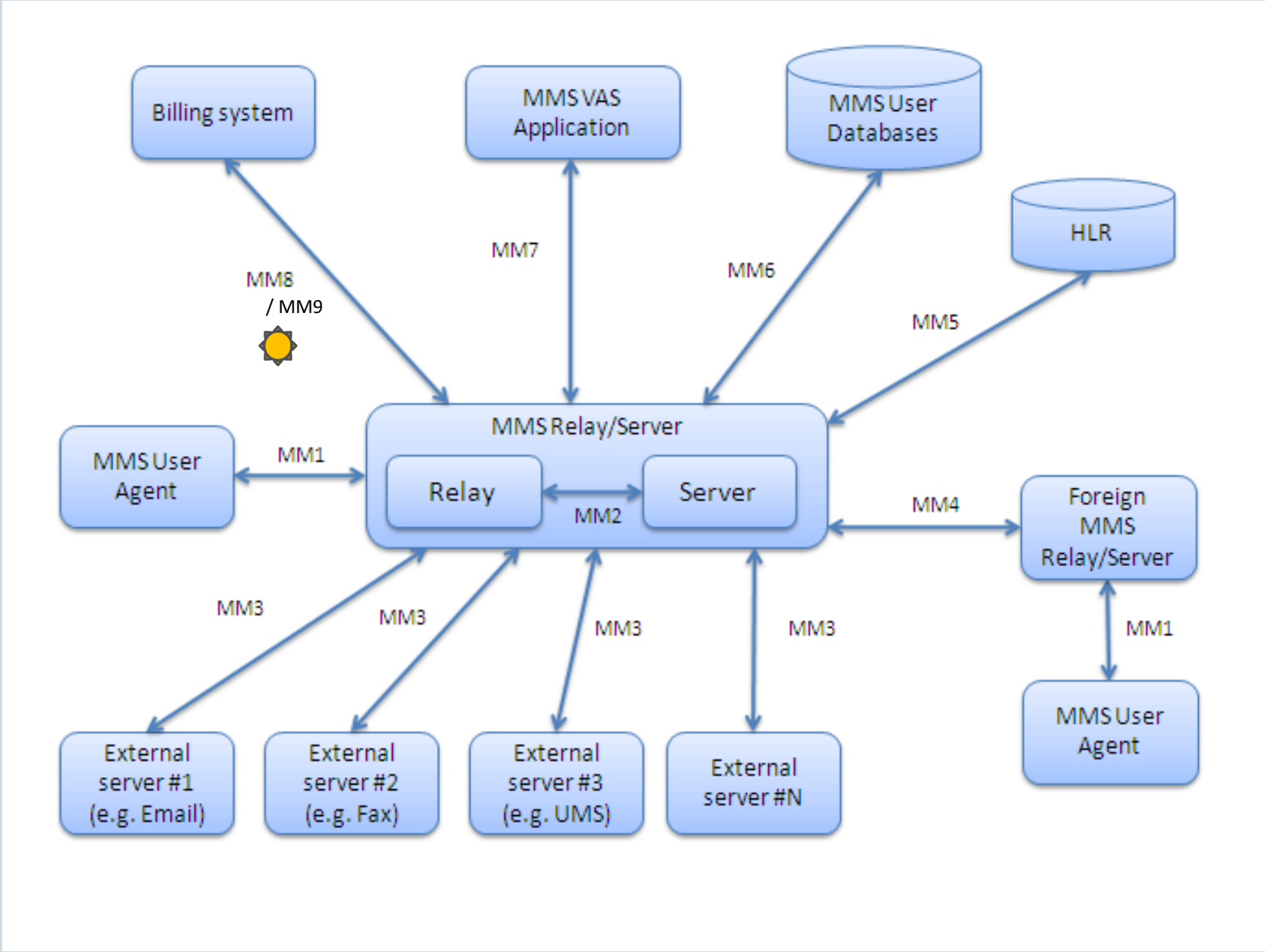
MM7

- Interface between MMSC and Value Added Service Provider (VASP) applications.
- SOAP/XML/HTTP based protocol defined by 3GPP TS 23.040
- MMS content encoded as MIME payload
- Bi-directional HTTP POST – MMSC and VASP can both initiate connections



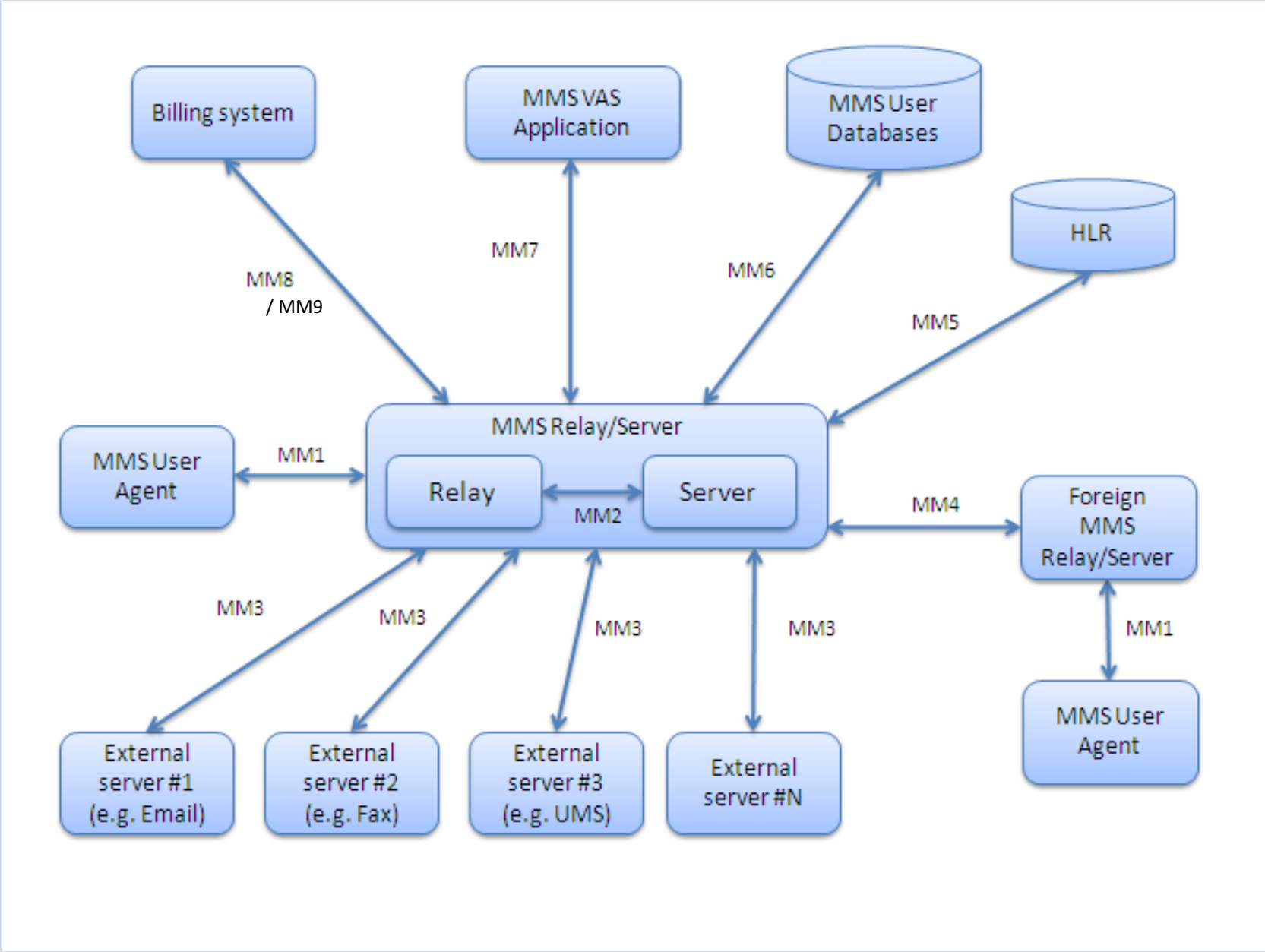
MM8 & MM9

- Interface between MMSC and billing/charging systems
- Billing generally refers to post-paid CDR generation
- Charging generally refers to real time charging required by pre-paid
- NowSMS MMSC supports HTTP based accounting callbacks which some customers use to generate CDRs
- NowSMS MMSC supports DIAMETER Credit Control or Base Accounting



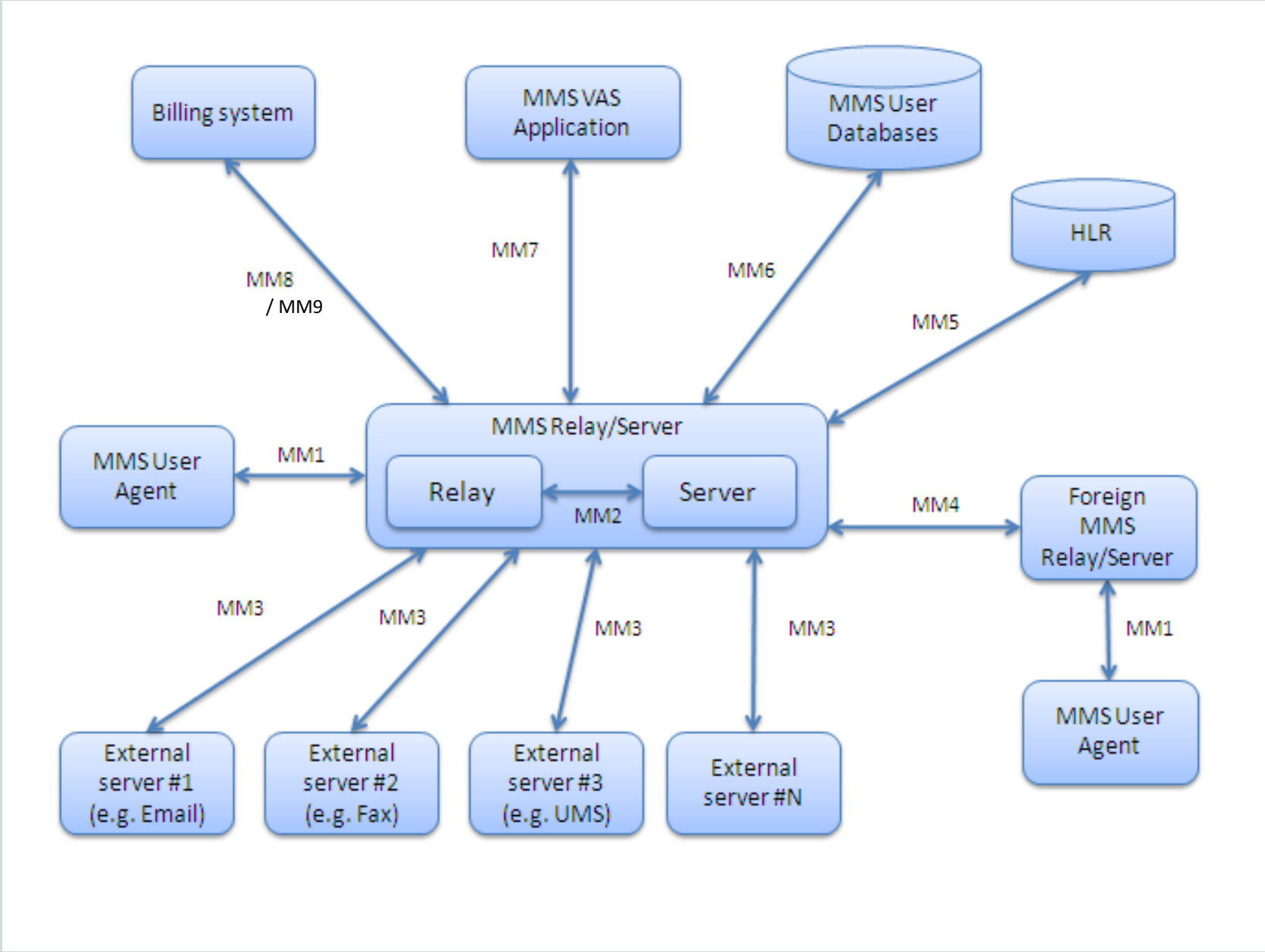
MM10

- Interface for message service control and filtering
- Relatively new, no known commercial implementations
- DIAMETER based protocol
- Not implemented by NowSMS MMSC



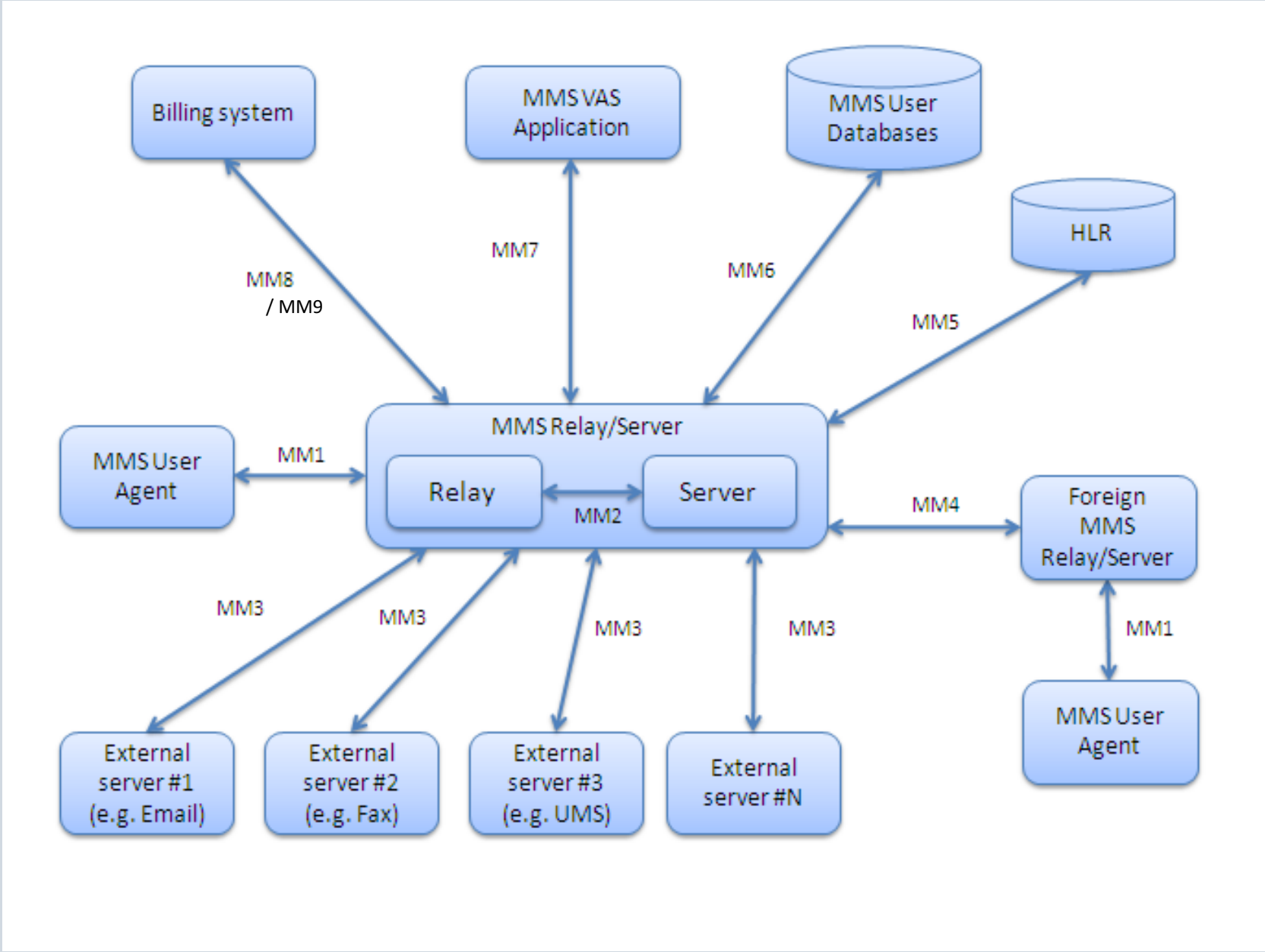
MM11

- Conceptual interface between MMSC and external transcoder



Key Protocols

- MM1 – Phone <-> MMSC
  - HTTP POST
  - Binary WSP MIME
  - WAP Push
- MM4 – MMSC <-> MMSC
  - SMTP / MIME
- MM7 – VASP <-> MMSC
  - HTTP POST
  - SOAP/XML Request
  - MIME MMS Content
- MM9 – Accounting / Billing / Charging
  - HTTP Callbacks
  - DIAMETER



MM1 Overview

MM1 is the over-the-air protocol for MMS message transmission between a phone and the network.

Defined by Open Mobile Alliance (OMA) – MMS Encapsulation Protocol
http://www.openmobilealliance.org

Based on HTTP, WAP Push and MIME technologies

MM1 Protocol Data Units (PDUs)

PDUs are the data elements that are exchanged between a mobile phone and the MMSC

PDUs are encoded using the MIME type application/vnd.wap.mms-message

PDUs are transferred as the payload of HTTP requests, HTTP responses, or WAP push messages

Example (HTTP POST):

```
POST / HTTP/1.1
Content-Type: application/vnd.wap.mms-message
Content-Length: 99999
```

MM1 PDU Kinds

There are three kinds of MM1 PDUs:

1. **Request**: Denoted as type-name.req
2. **Confirmation** (Response): Denoted as type-name.conf
3. **Indication** (notification): Denoted as type-name.ind (not confirmed)

MM1 PDU Transmission

When the client needs to send a PDU (Request or Indication) to the MMSC, it always uses HTTP POST.

- If the MMSC receives a Request PDU via HTTP, it generates an appropriate Confirmation PDU in the HTTP response.
- If the MMSC receives an Indication PDU via HTTP, it generates an empty HTTP OK response (status code 200 or 204).

When the MMSC needs to send a PDU (Indication only) to a client, it only uses WAP Push.

WAP 1.x Client Considerations

Older MMS clients based upon the WAP 1.x protocol use the Wireless Session Protocol (WSP) instead of HTTP.

These clients must use a WAP Gateway which converts between WSP and HTTP.

The MM1 interface at the MMSC is always HTTP based.

MM1 PDUs

Transaction	PDU Type Name	Originated By	Transport
Send Message	m-send.req (.conf)	Client	HTTP POST
New Message Received Notification	m-notification.ind	MMSC	WAP PUSH
Acknowledge New Message Received Notification (optional)	m-notifyresp.ind	Client	HTTP POST
Retrieve Message	HTTP GET / m-retrieve.conf	Client	HTTP GET
Acknowledge Retrieve Message Complete (optional)	m-acknowledge.ind	Client	HTTP POST
Delivery Report	m-delivery.ind	MMSC	WAP PUSH
Read Report from Client	m-read-rec.ind	Client	HTTP POST
Read Report to Client	m-read-orig.ind	MMSC	WAP PUSH
Forward Message	m-forward.req (.conf)	Client	HTTP POST

REVIEW

MMS Message Structure

An MMS Message has a similar structure to an SMTP E-Mail Message

Structure based upon SMTP and Multipart Internet Mail Extensions (MIME)

- Envelope – Recipients for in-transit message instance.
- Headers – Sender, Displayed Recipients, Subject, other attributes
- Content – MIME encoded multipart content. One or more multimedia objects (text, image, video, etc.)

REVIEW

MMS Envelope

Routing Attributes (sender and recipient) for an instance of an in-transit message.

Example: A message is sent to multiple recipients, some of which are subscribers of a different mobile network. The message may be split by the MMSC for delivery to another MMSC. The MMS envelope for the message instance sent to the other MMSC will be only the remote recipients. The envelope list of recipients tells the other MMSC which recipients to deliver the message to.

The MMS Header will still contain a list of all recipients to provide full group messaging support.

In MM4/SMTP these are represented by MAIL FROM: / RCPT TO: commands.

REVIEW

MMS Headers

Sender

Recipients (To/CC/BCC)

Subject

Priority

Message Class (Personal, Advertisement)

Message Type (can be delivery report or read report)

Message ID

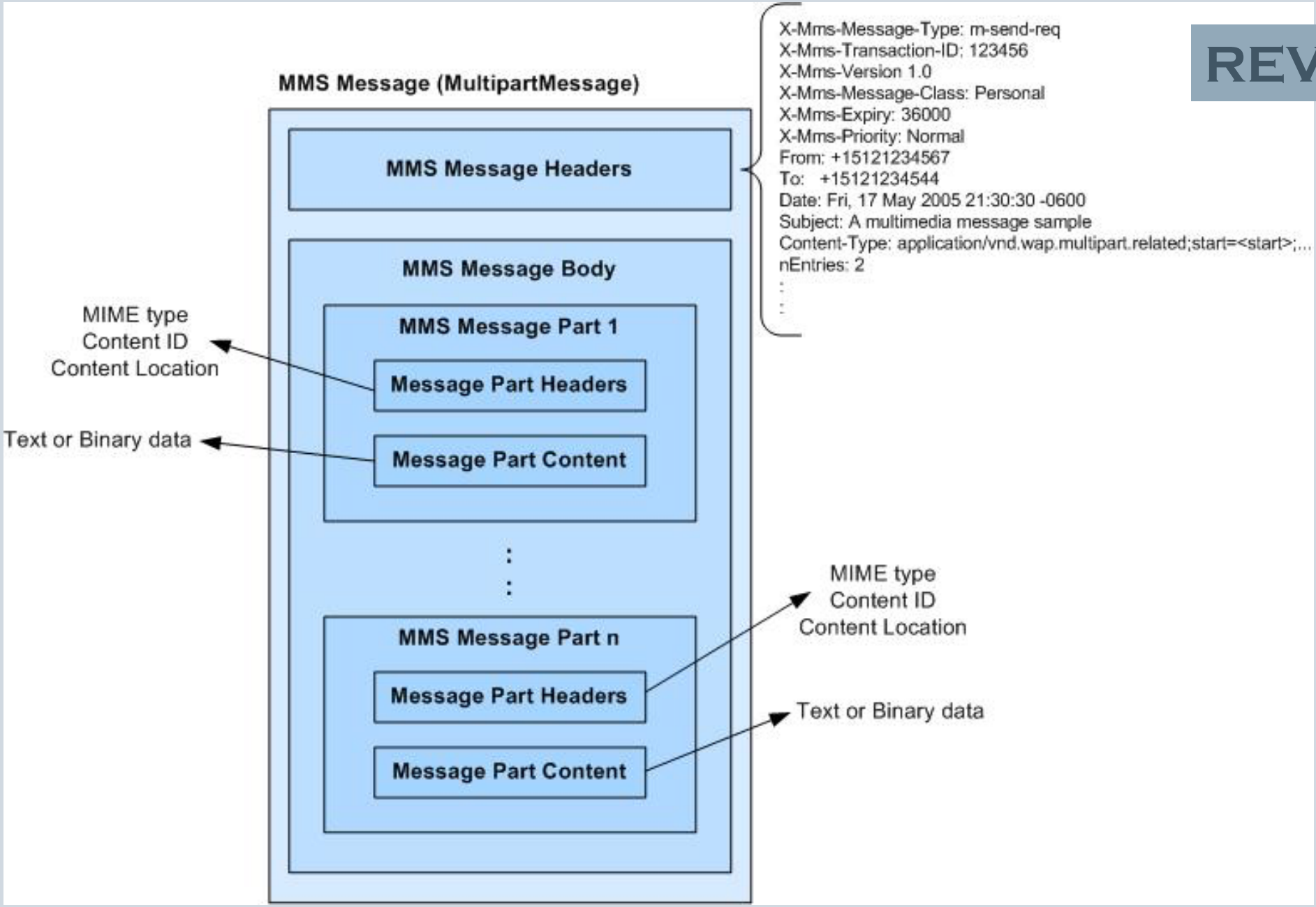
REVIEW

MMS Content

Multipart (MIME) object containing one or more of the following:

- Text
- Images
- Video
- Audio
- Contact Objects
- Calendar Objects
- SMIL Presentation (legacy support)

REVIEW


MMS Message (MultipartMessage)
MMS Message Headers
MMS Message Body
MMS Message Part 1
Message Part Headers
Message Part Content
MMS Message Part n
Message Part Headers
Message Part Content
MIME type
Content ID
Content Location
Text or Binary data
MIME type
Content ID
Content Location
Text or Binary data
X-Mms-Message-Type: m-send-req
X-Mms-Transaction-ID: 123456
X-Mms-Version 1.0
X-Mms-Message-Class: Personal
X-Mms-Expiry: 36000
X-Mms-Priority: Normal
From: +15121234567
To: +15121234544
Date: Fri, 17 May 2005 21:30:30 -0600
Subject: A multimedia message sample
Content-Type: application/vnd.wap.multipart.related;start=<start>;...
nEntries: 2

MM1 Message Encoding

No MMS Envelope: All recipients are in MMS Headers

The MMS Encapsulation Protocol defines the MIME type application/vnd.wap.mms-message, which consists of headers and an optional content body.

MMS Headers are encoded in a binary format, following the WAP Session Protocol (WSP)

- WSP defines a single byte code for common header names and values to reduce message size
- X-MMS-Message-Type: m-retrieve.conf reduces from a 36 character string to two bytes: 8C 84

MM1 Message Encoding

MMS Content is encoded as multipart MIME object.

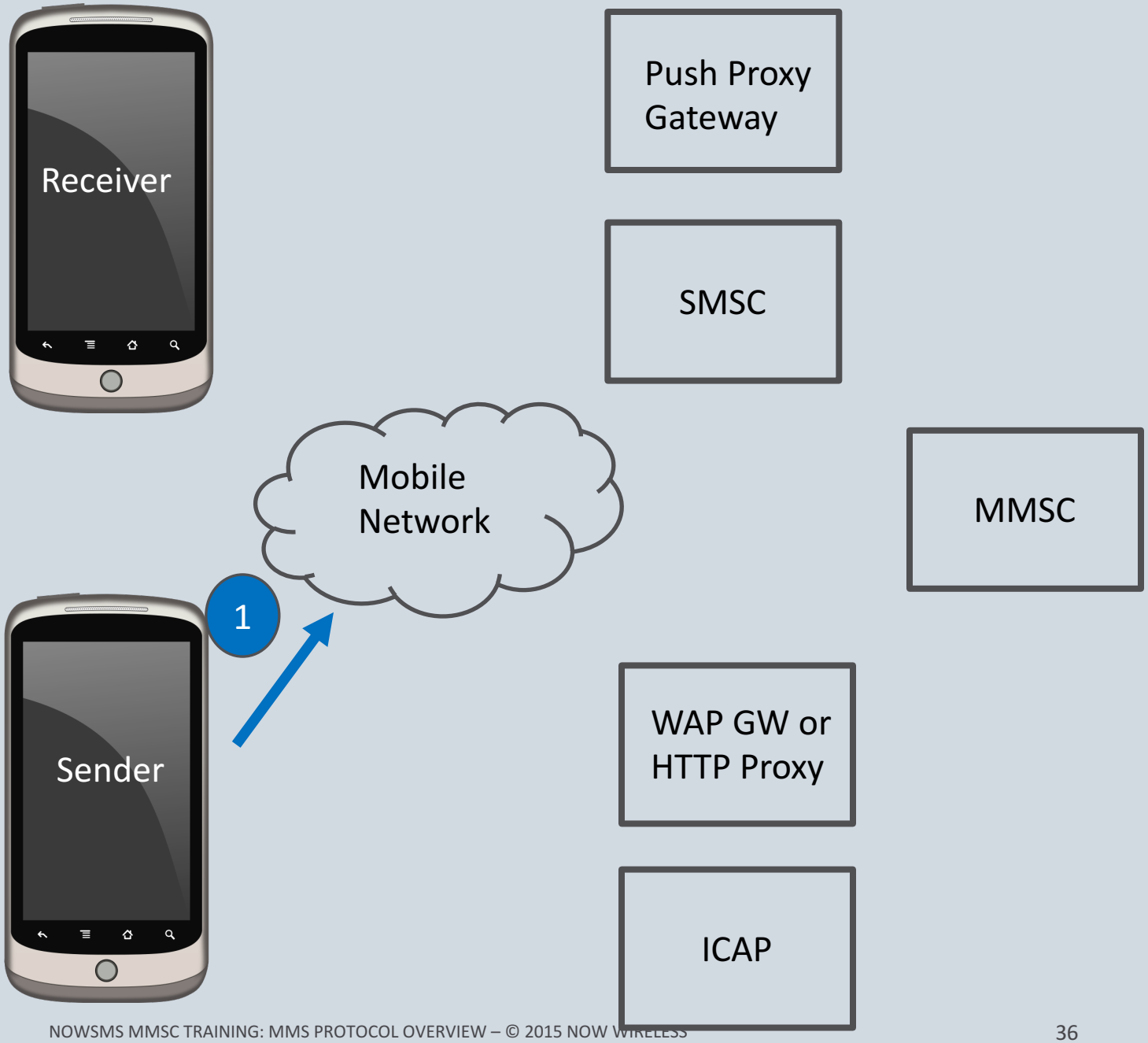
The multipart object is are encoded in a binary format, following the WAP Session Protocol (WSP)

- application/vnd.wap.multipart.related is used in place of multipart/related
- application/vnd.wap.multipart.mixed is used in place of multipart/mixed

MM1 Transaction: Step 1

Sender activates data connection to MMS APN

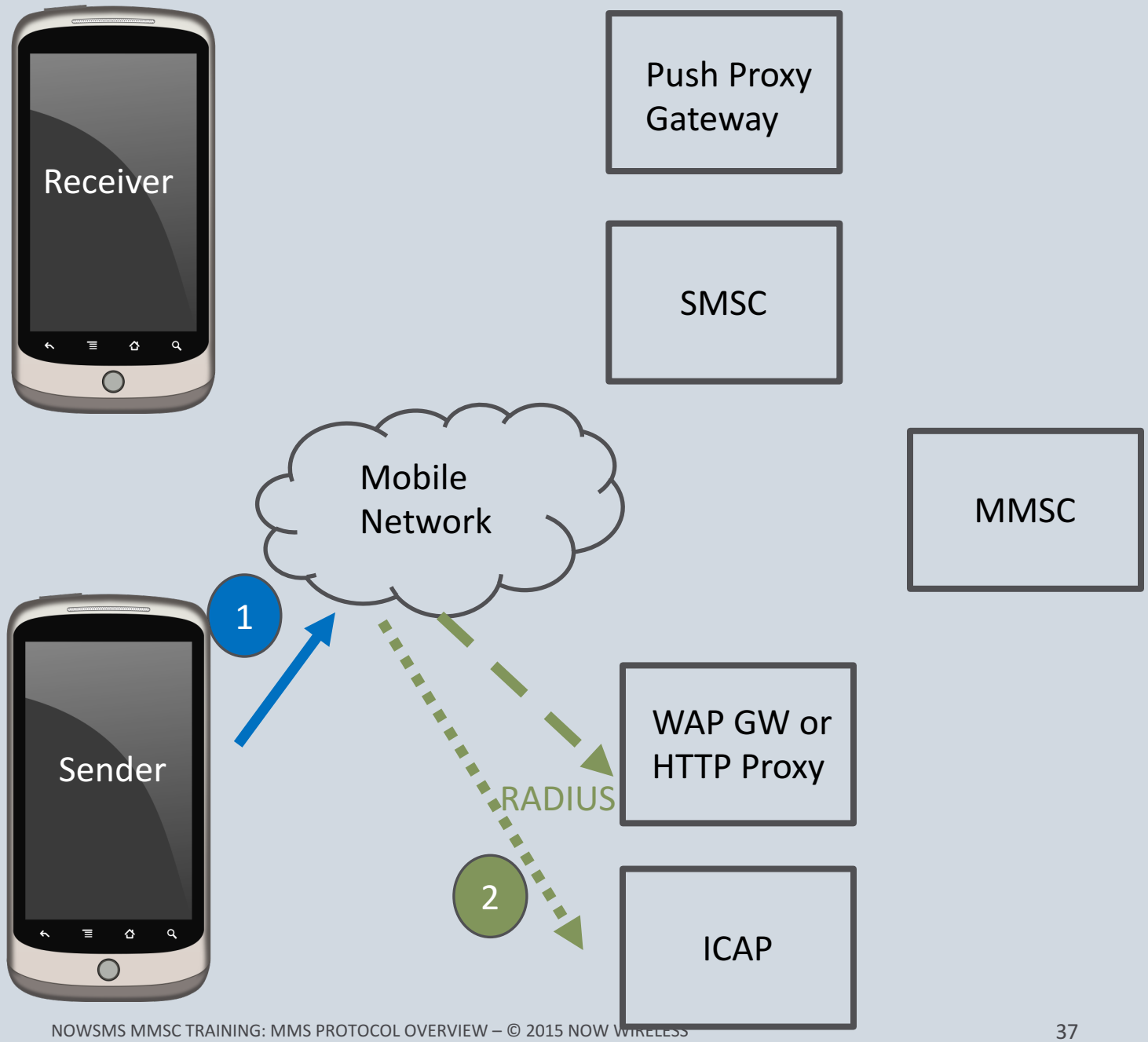
(APN setting configured in the phone)



MM1 Transaction: Step 2

GGSN sends RADIUS accounting message to notify the network about the device IP assignment.

WAP Gateway or ICAP server maintains table of active device IP addresses.



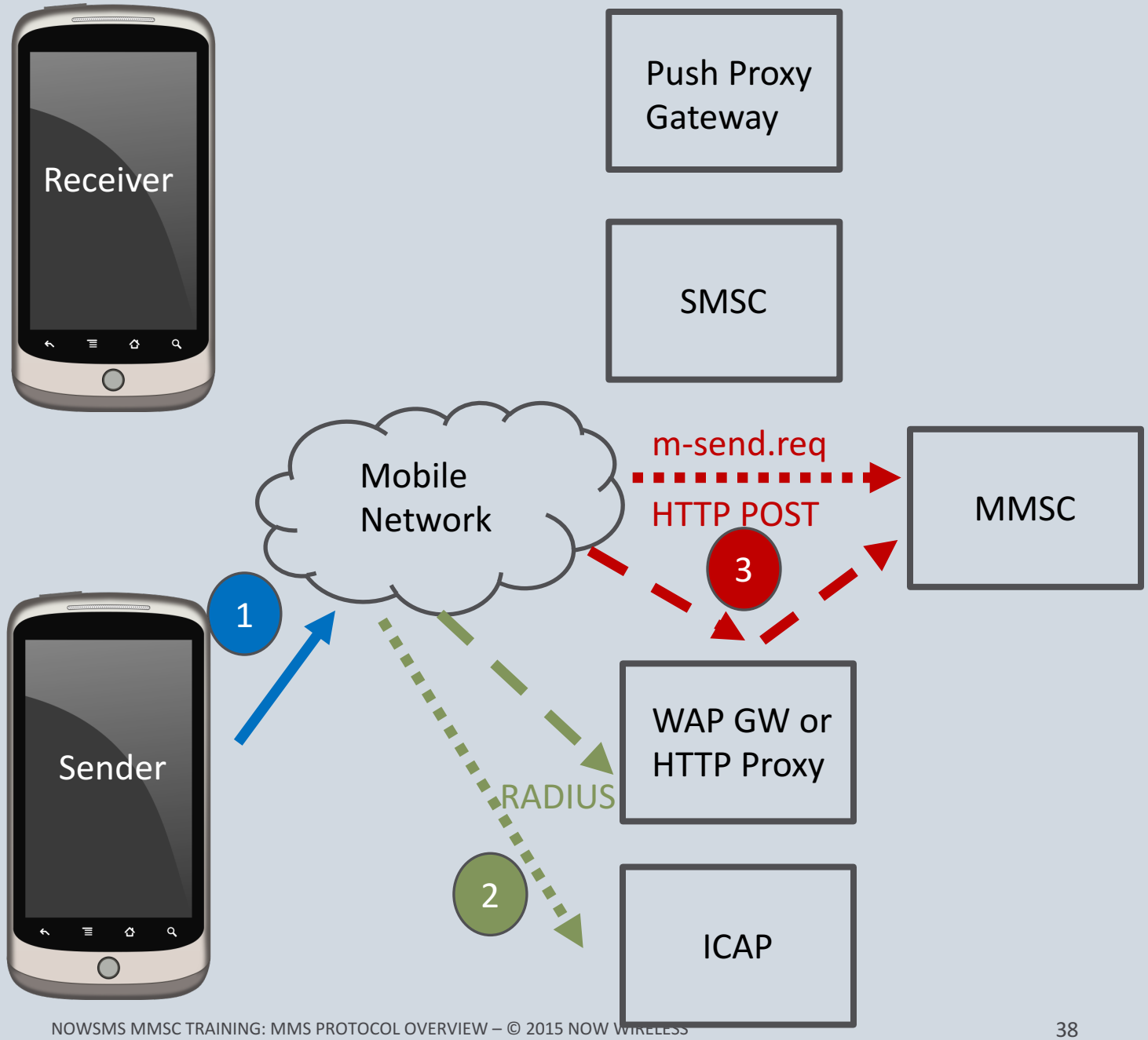
MM1 Transaction: Step 3

Sending phone submits an MM1 m-send.req to the MMSC. (MMSC Server URL configured in phone.)

MMSC receives m-send.req in HTTP POST

Phone may submit:

- WAP1/WSP POST via WAPGW
- WAP2/HTTP POST via Proxy
- HTTP POST direct to MMSC



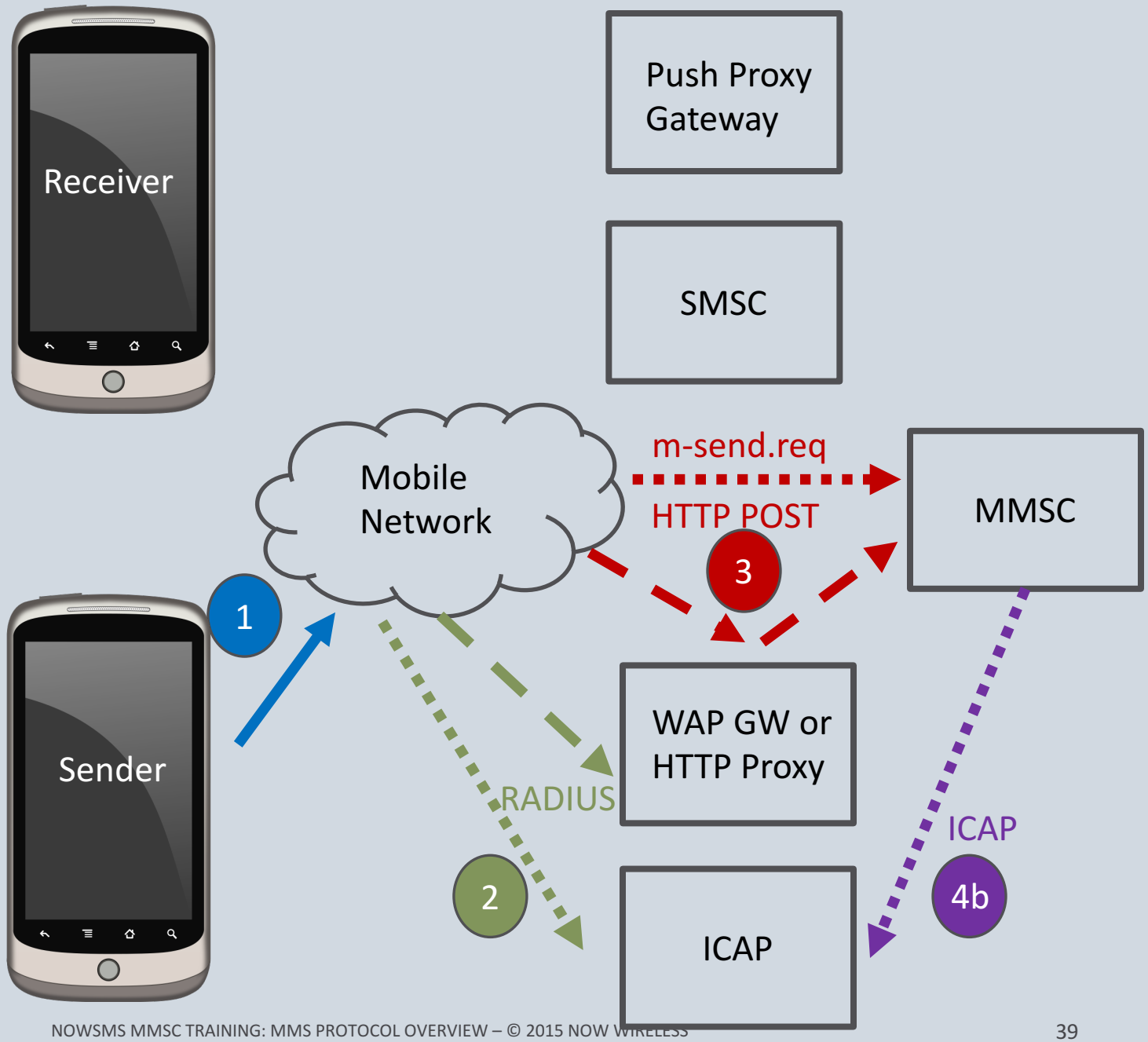
MM1 Transaction: Step 4

MMSC must identify and authenticate sender

MM1 does not define how this is done

Two options:

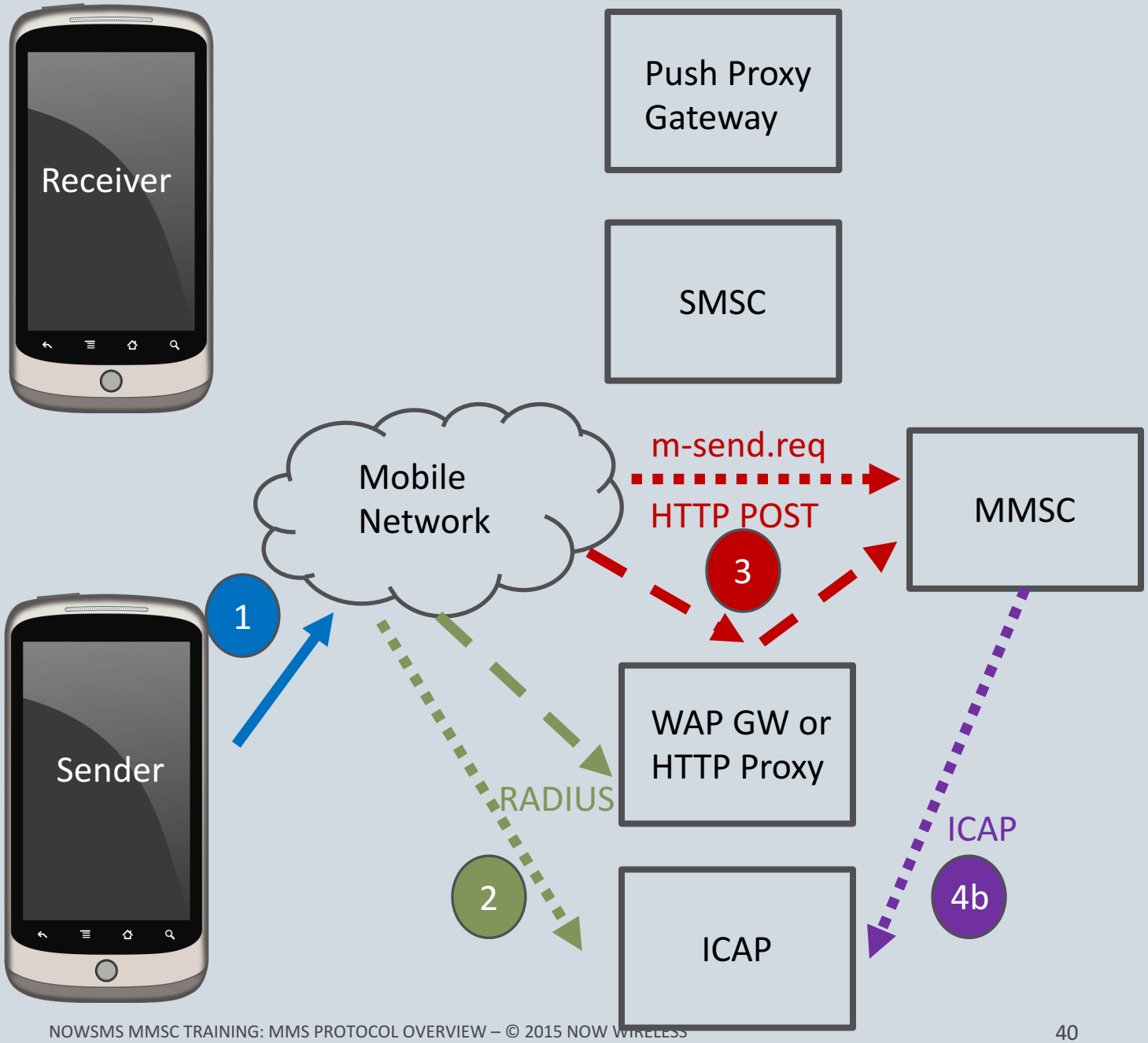
- WAPGW or HTTP Proxy inserts X-MSISDN HTTP header
- MMSC requests MSISDN using ICAP



MM1 Transaction: Step 5

The MMSC accepts and processes the content of the MMS message. If to a local recipient, the MMSC stores and makes it available as a dynamically generated URL link.

The MMSC generates an MMS notification message (m-notification.ind), which is sent via WAP Push over SMS to the recipient(s). This MMS notification message contains a URL pointer to the dynamically generated MMS content.

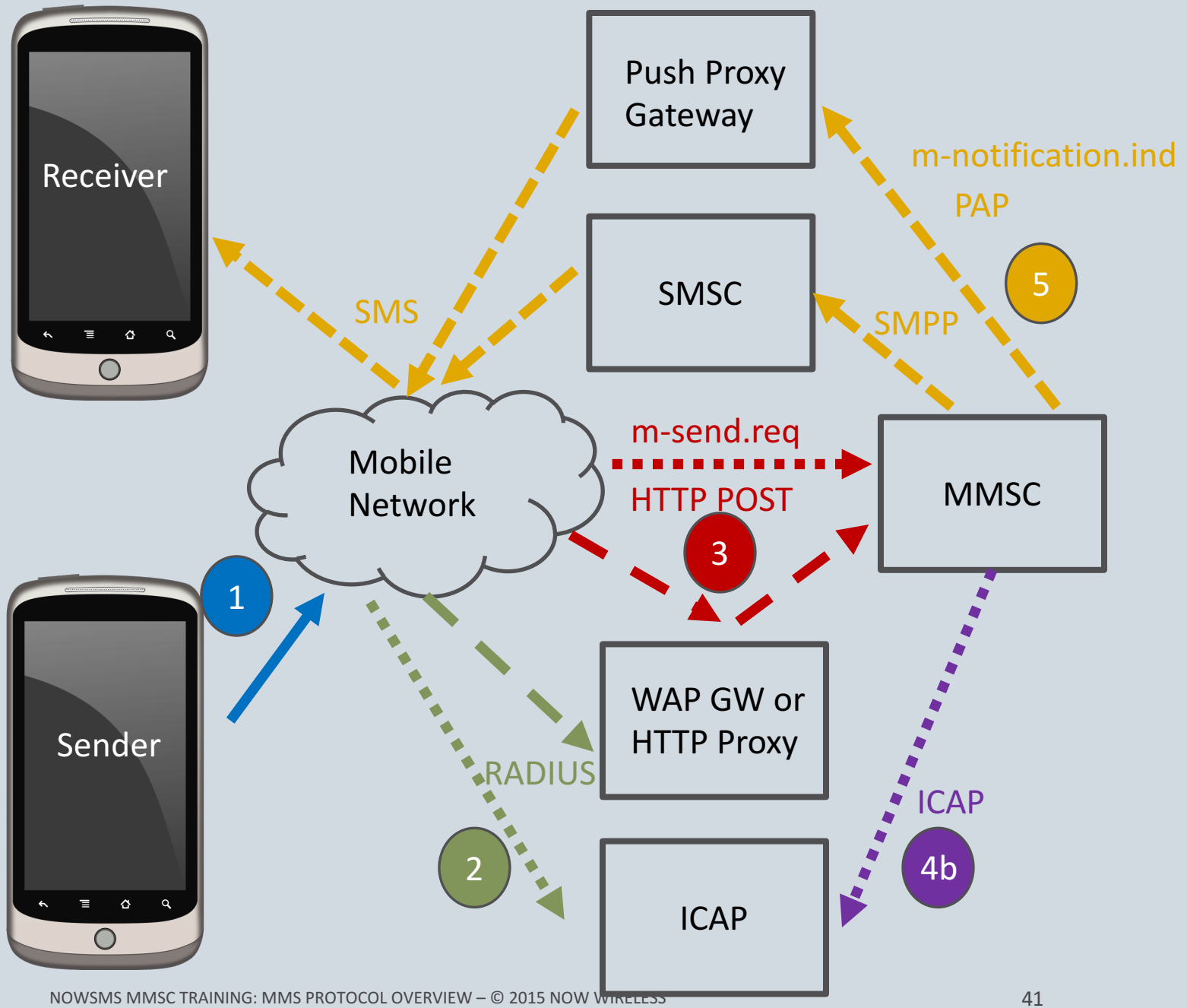


MM1 Transaction: Step 5 (continued)

m-notification.ind WAP push may be submitted to a Push Proxy Gateway using the Push Access Protocol (PAP)

Or it may be encoded directly to SMS format and submitted using SMPP

SMS message generated is usually a 2 part long message

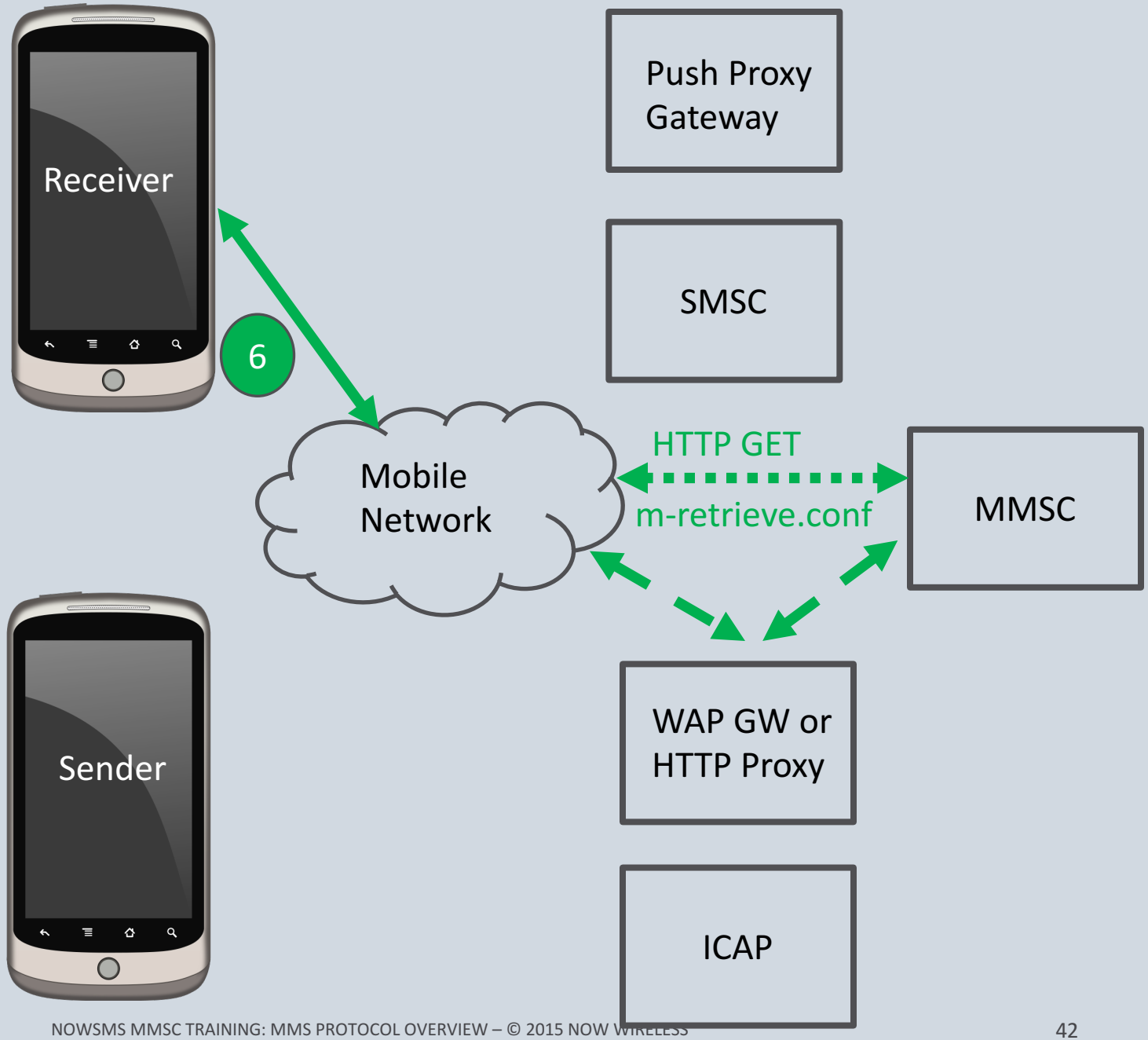


MM1 Transaction: Step 6

The recipient phone activates data connection to MMS APN. (APN setting configured in the phone)

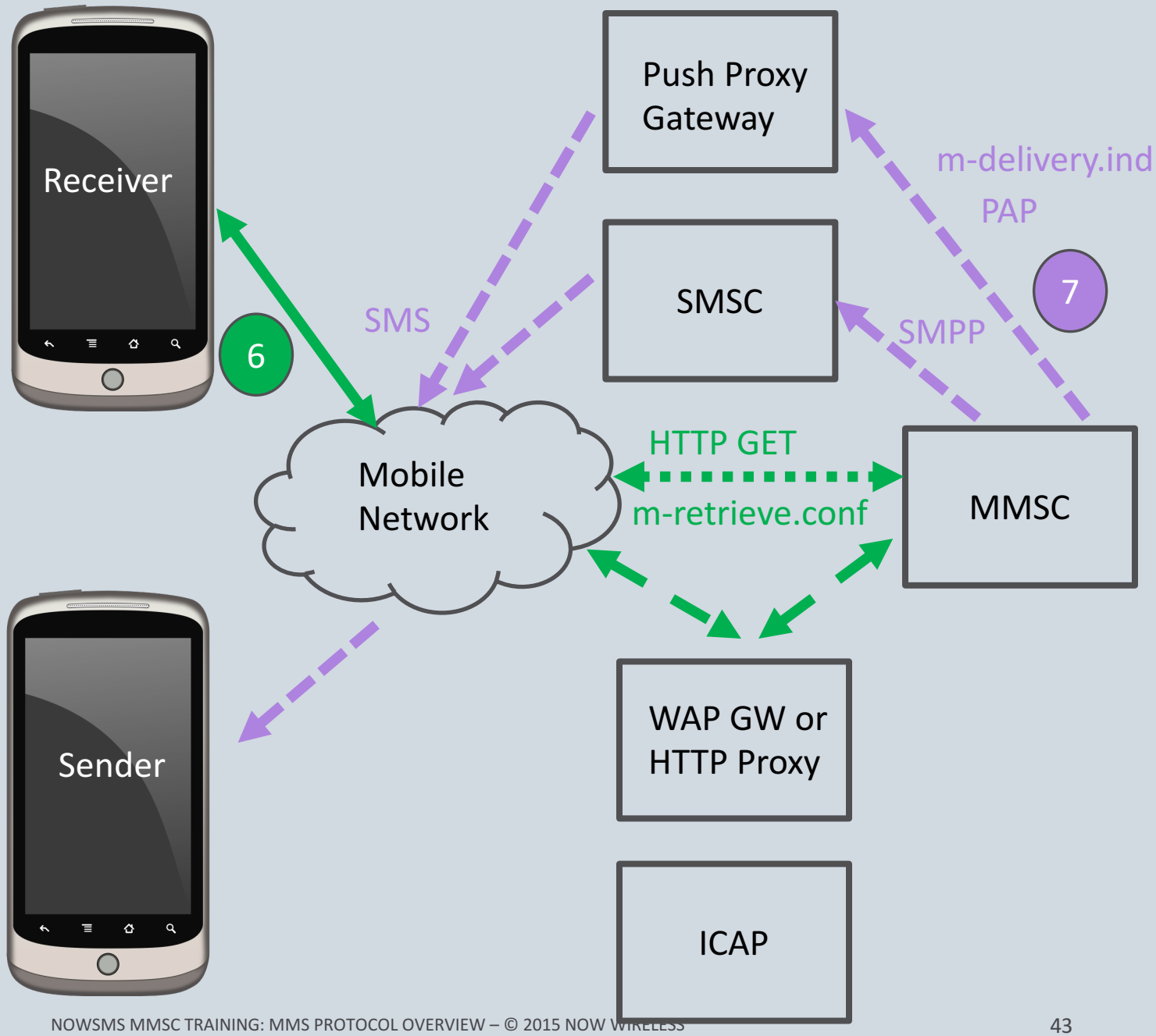
The recipient phone performs an HTTP (or WSP) GET to retrieve the MMS message content URL from the MMSC.

The HTTP response is the MMS message in an m-retrieve.conf PDU.



MM1 Transaction: Step 7

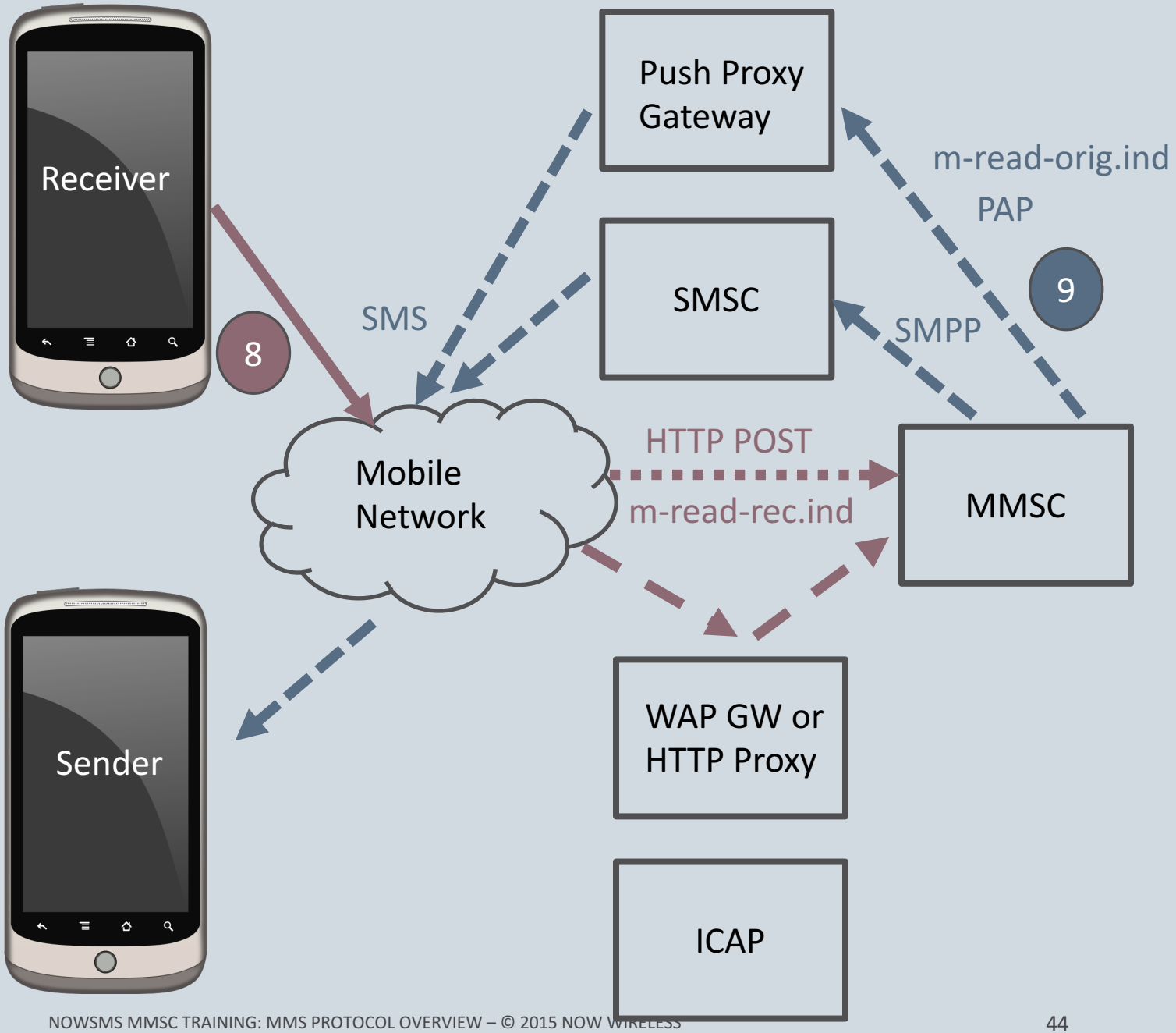
If the sender requested a delivery report, the MMSC generates an MMS delivery report (m-delivery.ind), which is sent via WAP Push over SMS to the original sender.



MM1 Transaction: Step 8

If the sender requested a read report, the receiver may generate a report (m-read-rec.ind) when the subscriber reads the message. (This is dependent on client configuration.)

The MMSC translates the format of the read report (m-read-orig.ind), which is sent via WAP Push over SMS to the original sender.



MMS MO (Mobile-Originated) Flow

1. Sender activates data connection to MMS APN. (APN setting configured in the phone)
2. GGSN sends RADIUS accounting message to notify the network about the device IP assignment.
3. Sending phone submits an MM1 m-send.req to the MMSC. (MMSC Server URL configured in phone.)
4. MMSC must identify and authenticate sender (X-MSISDN header inserted by WAPGW or ICAP)
5. The MMSC accepts and processes the content of the MMS message. If to a local recipient, the MMSC stores and makes it available as a dynamically generated URL link.

MMS MT (Mobile-Terminated) Flow

1. The MMSC generates an MMS notification message (m-notification.ind), which is sent via WAP Push to the recipient(s). This MMS notification message contains a URL pointer to the dynamically generated MMS content.
2. The recipient phone activates data connection to MMS APN. (APN setting configured in the phone)
3. The recipient phone performs an HTTP (or WSP) GET to retrieve the MMS message content URL from the MMSC.
4. Delivery report and/or read report activity may occur

What is WAP Push?

SMS supports a concept of port numbers to allow applications to register to receive messages that are sent to a specific port number.

(WAP Push uses SMS port number 2948)

WAP Push messages are binary messages that can contain different binary content types, one of which is an MMS notification type.

When this MMS notification message is received, the phone routes it to the MMS client for processing.

About WAP Push

WAP Push plays a key role in the MMS delivery process.

By default, the NowSMS MMSC expects to be able to send MMS Notification via its built-in WAP Push Proxy Gateway (PPG), which delivers MMS Notifications using WAP Push over SMS.

For GSM/UMTS/WCDMA environments, NowSMS automatically generates UDH (user data header) in the SMS messages.

For CDMA/CDMA2000 environments, the NowSMS SMPP implementation supports WDP Adaptation so that MMS notification messages can be delivered via SMS using the WAP teleservice.

For other environments, NowSMS can use the Push Access Protocol (PAP) to send MMS notification messages via a separate WAP Push Proxy Gateway (PPG).

MM1 User Authentication Issues

The MM1 Protocol does not define how the MMSC identifies and authenticates the subscriber when the MMS client on a mobile device sends or receives a message.

The MMSC is expected to interface with other operator network components to identify and authenticate the subscriber.

HTTP Header Enrichment

MMS clients send HTTP POST requests to the MMSC over TCP/IP.

Accepted industry practice is to use HTTP Header Enrichment services to insert additional HTTP headers into these requests to provide user identification and authentication.

For example, the subscriber MSISDN, is frequently inserted into one of the following headers:

- `X-MSISDN:`
- `msisdn:`
- `X-MDN:`
- `X-Device-MIN:`

Who performs HTTP Header Enrichment?

The first IP-based mobile operator services were based on WAP technologies, where a WAP Gateway acted as a proxy for all IP-based services.

These WAP gateways often performed HTTP Header Enrichment to allow IP-based services to identify subscribers.

The usual process is to configure the GGSN to send RADIUS Accounting messages to the WAPGW every time a subscriber connects to or disconnects from the GGSN. The WAPGW maintains a table of active connections that allow it to map IP addresses to device phone number for HTTP Header Enrichment.

HTTP Header Enrichment without a GW

As mobile usage has grown, gateways and proxies have become network bottlenecks.

The ICAP protocol allows an HTTP based service to request HTTP Header Enrichment on demand, without forcing clients to interface through a gateway or proxy.

The NowSMS MMSC can be configured to request HTTP HHE via ICAP, and NowWAP can be configured to provide this ICAP based service.

MM4 – MMSC Interconnect

The MM4 protocol is used to interconnect MMSCs.

- SMTP based protocol defined by 3GPP TS 23.040
- Additional X-MMS SMTP headers are defined
- Different PDUs are mapped to SMTP headers

When a subscriber sends an MMS message to a subscriber on a different network, the MM4 protocol is used to transfer the message between MMSCs

MM4 – MMSC Interconnect

MM4 connections can exist directly between mobile operators.

Frequently a single MM4 connection to an interconnect provider is used.

MM4 is not used for roaming, only for interconnect.

MM4 Message Encoding

The **MMS Envelope** contains Routing Attributes (sender and recipient) for an instance of an in-transit message. In MM4, this envelope is represented by the standard SMTP MAIL FROM: and RCPT TO: commands.

The **MMS Headers** are represented as SMTP Headers, with MMS specific headers prefixed by X-MMS- (e.g., X-MMS-Message-Type:). These headers are defined in 3GPP TS 23.140.

The **MMS Content** is encoded as a standard SMTP MIME multipart object. Note that even if there is only a single content object (e.g., image only), many implementations still expect a MIME multipart with only 1 part.

Sample MM4 Transaction

SMTP AUTH is optional and rarely used

```
IN: 220 SMTP Ready
OUT: HELO client.name (or EHLO client.name)
IN: 250 OK (or a multiline response if EHLO was used)
OUT: AUTH LOGIN
IN: 334 VXNlcm5hbWU6
(NOTE: "Username:" BASE64 encoded)
OUT: dGVzdA==
(NOTE: "test" BASE64 encoded)
IN: 334 UGFzc3dvcmQ6
(NOTE: "Password:" BASE64 encoded)
OUT: dGVzdA==
(NOTE: "test" BASE64 encoded)
IN: 235 Ok
OUT: MAIL FROM: <+447779998888/TYPE=PLMN@mms.oper.com>
IN: 250 Ok
OUT: RCPT TO: <+447778889999/TYPE=PLMN@mms.domain.com>
IN: 250 Ok
OUT: DATA
IN: 354 Ok, end with "." on a new line...
OUT: (Transmit MIME encoded message, then end with a line with only the . character)
IN: 250 Message Accepted
OUT: QUIT
```

Sample MM4 Message

```
X-Mms-3GPP-MMS-Version: 5.9.0
X-Mms-Message-Type: MM4_forward.REQ
X-Mms-Transaction-ID: "60755231.94578772@mmsc"
X-Mms-Message-ID: "148478240@mmsc.hub"
X-Mms-Ack-Request: Yes
X-Mms-Originator-System: system-user@mmsc.hub
Message-ID: <60755231.94578772@mmsc>
Date: Tue, 8 May 2007 12:18:32 +0300
To: +447778889999/TYPE=PLMN@mms.domain.com
From: +447779998888/TYPE=PLMN@mms.oper.com
Subject: News for today
Content-Type: multipart/mixed; boundary="StoryParts-74526-8432-2002-77645"
Content-ID:<SaturnPics-01020930@news.tnn.com>

--StoryParts-74526-8432-2002-77645
Content-Type: text/plain; charset="us-ascii"

Science news, new Saturn pictures...
--StoryParts-74526-8432-2002-77645
Content-Type: image/gif
Content-ID:<saturn.gif>
Content-Transfer-Encoding: base64

R0lGODdhZAAwAOMAAAAAAIGJjGltcDE0OOfWo6Ochbi1n1pmcbGojpKbnP/lpW54fBMTE1RYXEFO
...
--StoryParts-74526-8432-2002-77645--
```

MM4 PDU Kinds

There are two kinds of MM4 PDUs:

1. **Request**: Denoted as MM4_type-name.REQ
2. **Response**: Denoted as MM4_type-name.RES
   (Often referred to as ACKs or acknowledgments)

MM4 PDU Transmission

MM4 PDUs are sent using SMTP.

Because the SMTP protocol is uni-directional and contains limited status report capability, requests and their associated response can never be transmitted over the same connection,

When an MMSC has a request to transmit to another MMSC over MM4, it initiates an SMTP connection to that MMSC to transfer the request PDU.

The receiving MMSC must then initiate an SMTP connection back to the requesting MMSC to transfer the response PDU.

MM4 PDUs

Transaction	PDU Type Name
Send Message	MM4_forward.REQ
	MM4_forward.RES
Delivery Report	MM4_delivery_report.REQ
	MM4_delivery_report.RES
Read Report	MM4_read_reply_report.REQ
	MM4_read_reply_report.RES

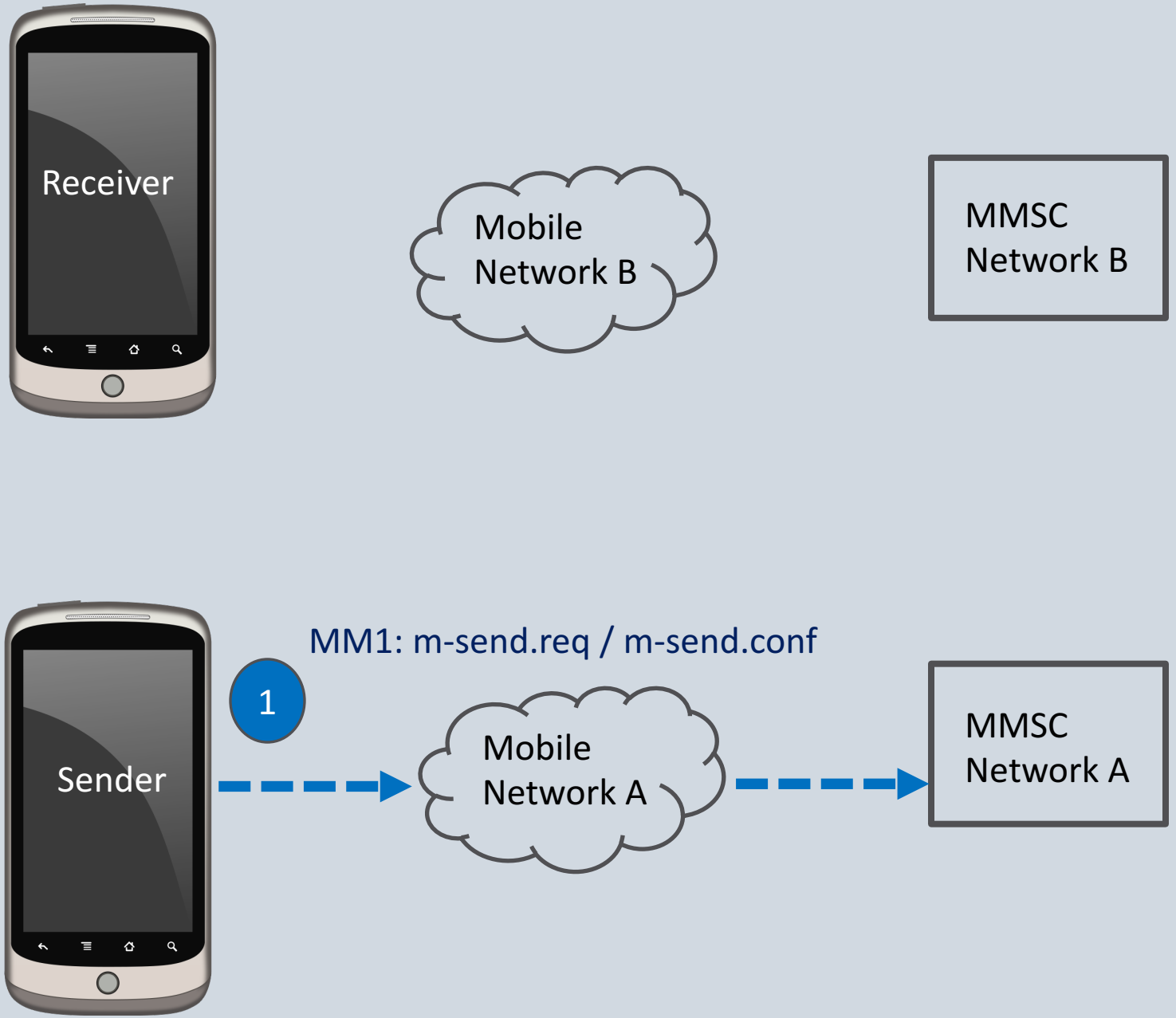
MM4 Transaction: Simplified

Step 1

Standard MM1 MMS MO Logic Flow

MM1 m-send.req from sending client to their operator MMSC

MM1 m-send.conf confirms MMSC acceptance of message

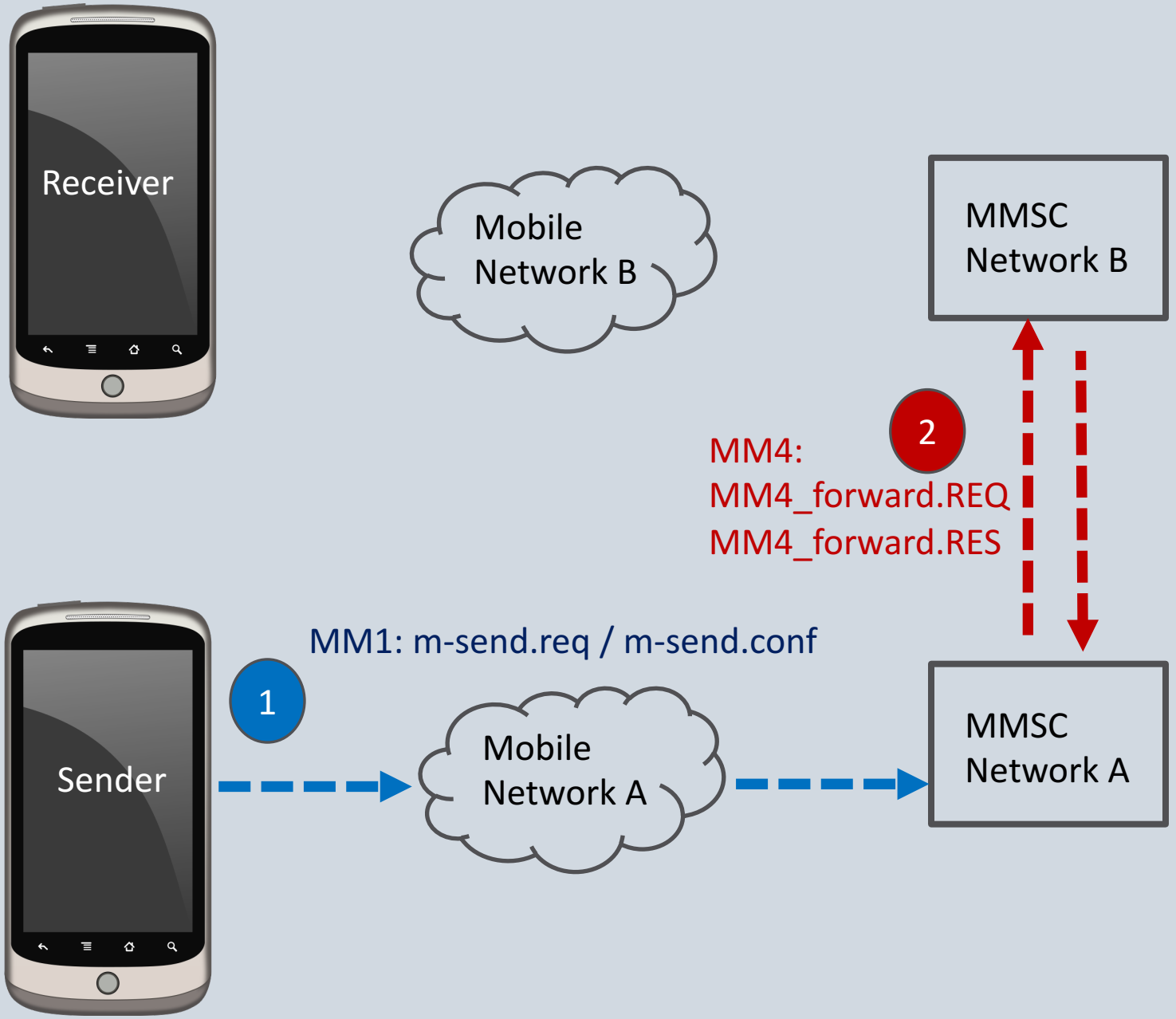


MM4 Transaction: Simplified

Step 2

MMSC initiates MM4 connection to other operator MMSC and uses MM4_forward.REQ transaction to transfer message.

Other MMSC initiates MM4 connection back to originator MMSC and uses MM4_forward.RES transaction to confirm MMSC acceptance of message

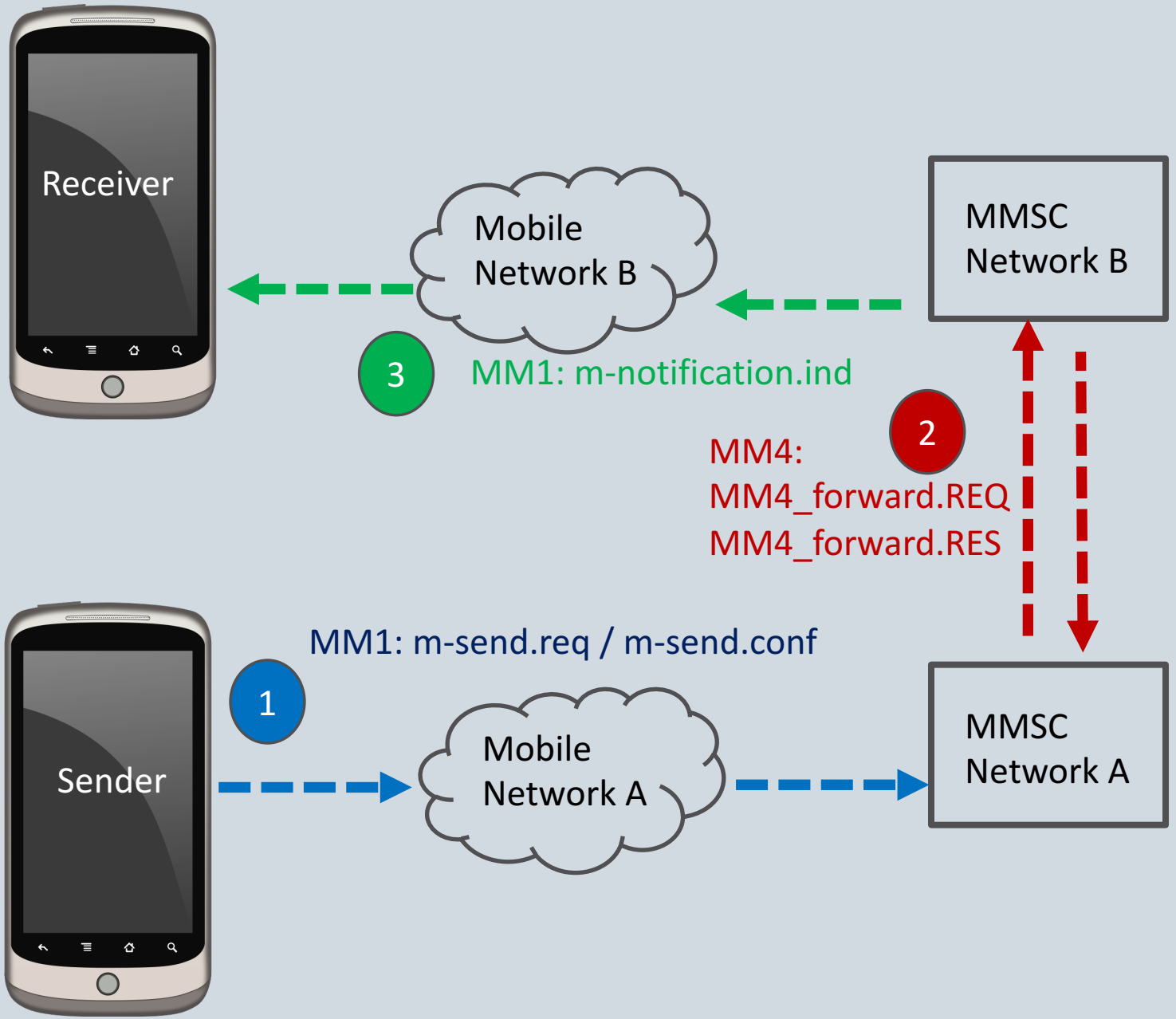


MM4 Transaction: Simplified

Step 3

Standard MM1 MMS MT Logic Flow

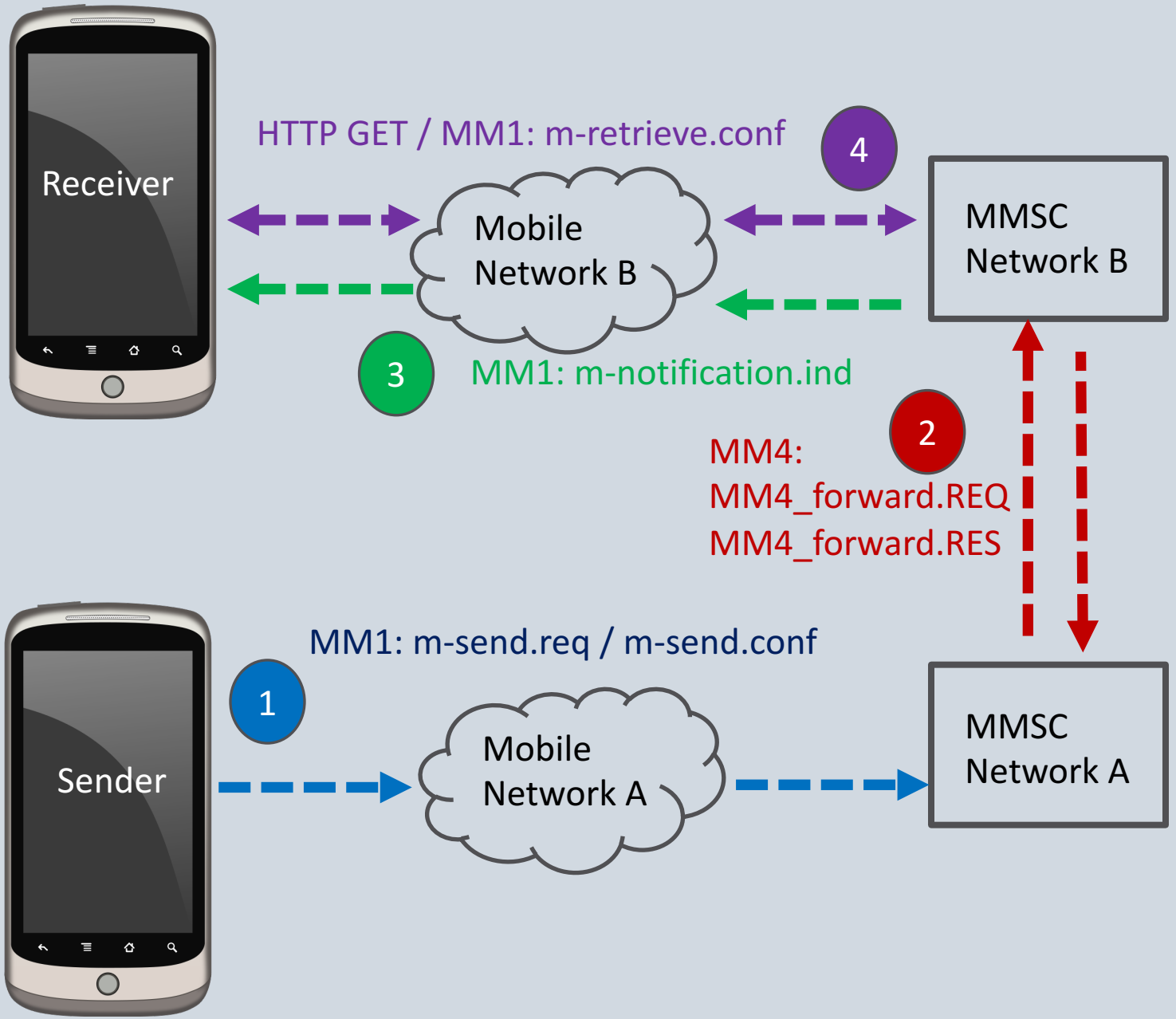
Receiver MMSC posts MM1 m-notification.ind to receiving device over WAP Push



MM4 Transaction: Simplified

The recipient phone performs an HTTP (or WSP) GET to retrieve the MMS message content URL from the MMSC.

The HTTP response is the MMS message in an m-retrieve.conf PDU.



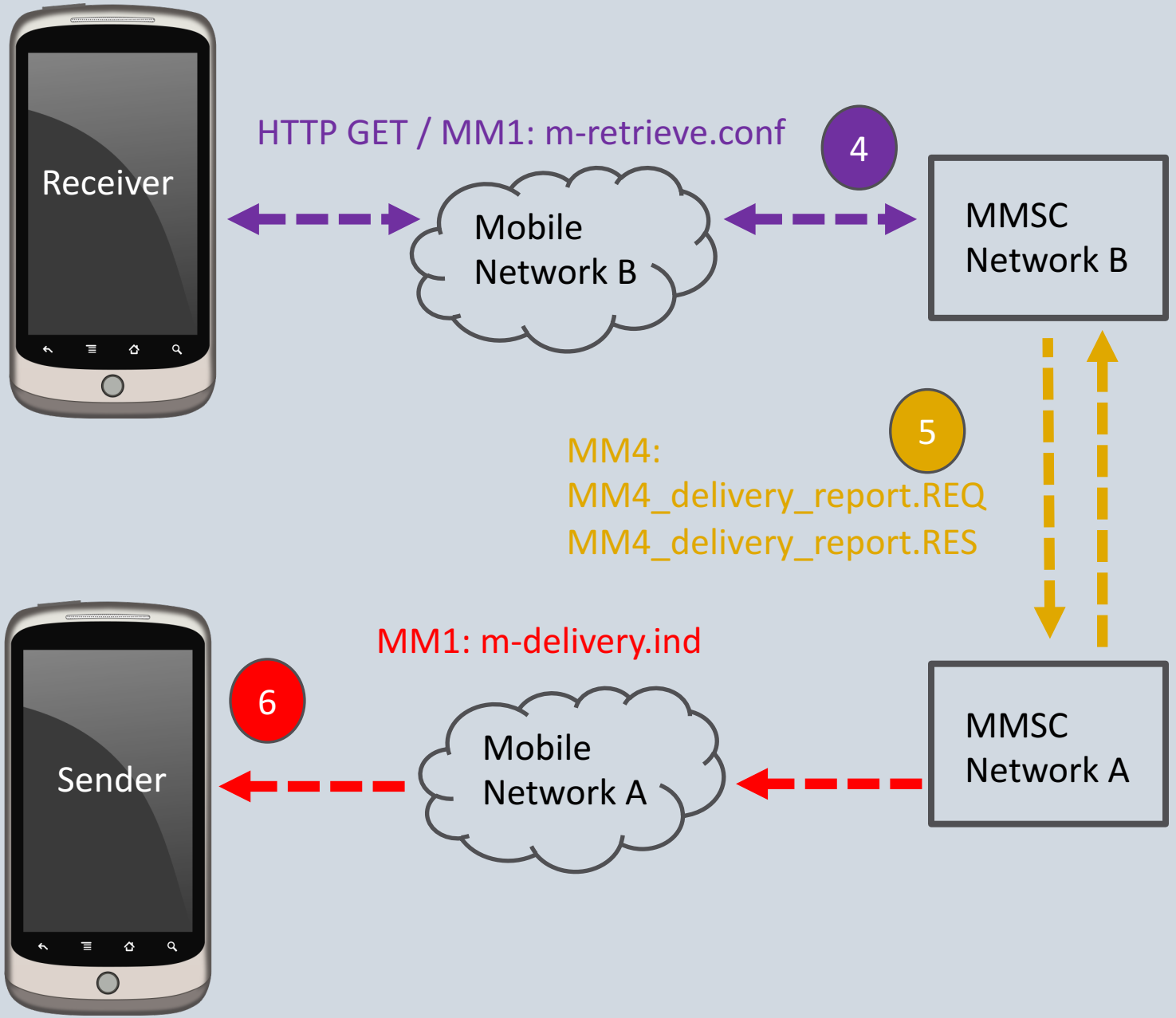
MM4 Transaction: Simplified

After MMS delivery, if a delivery report was requested:

Receiver MMSC initiates MM4 connection to originator MMSC and issues MM4_delivery_report.REQ.

Originator MMSC initiates MM4 connection back to receiver MMSC and uses MM4_delivery_report.RES transaction to confirm MMSC acceptance of delivery report.

MM1 delivery report is pushed to sender.



MM4 Transaction: Simplified

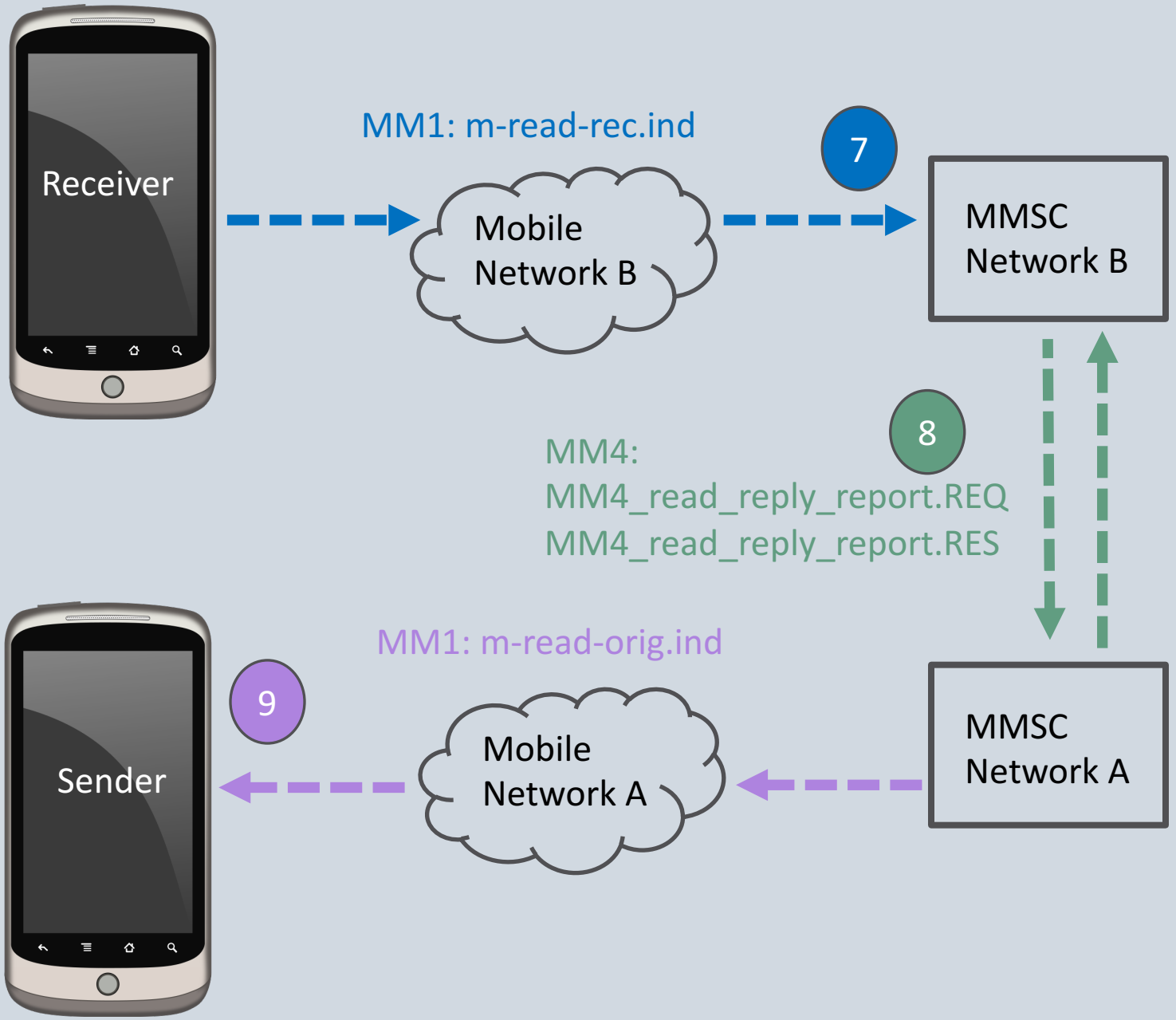
After MMS is read, if a read report was requested and receiver allows read reports:

Receiver posts an MM1 m-read-rec.ind to their MMSC

Receiver MMSC initiates MM4 connection to originator MMSC and issues MM4_read_reply_report.REQ.

Originator MMSC initiates MM4 connection back to receiver MMSC and uses MM4_read_reply_report.RES transaction to confirm MMSC acceptance of delivery report.

MM1 read report (m-read-orig.ind)fs is pushed to sender.



MM7 – Value Added Service Providers

- Interface between Value Added Service Provider (VASP) applications and MMSC.
- SOAP/XML/HTTP POST based protocol defined by 3GPP TS 23.040
- MMS headers encoded as XML document
- MMS content encoded as MIME multipart object
- Bi-directional HTTP POST – MMSC and VASP can both initiate connections

MM7 Variations and Interoperability

Early versions of the 3GPP TS 23.140 specification defined MM7 only as a conceptual protocol.

This lead several vendors to define their own MM7 protocols (Ericsson, LogicaCMG and Nokia).

Different versions of the MM7 specification have defined more than 10 different XML schemas with some significant differences and incompatibilities. The most significant problems are related to international number encoding and sender address formats.

SOAP = Simple Object Access Protocol

SOAP is a lightweight protocol for the exchange of information in distributed environments.

SOAP transactions are represented using XML.

A SOAP transaction contains a SOAP header, SOAP body and an optional SOAP attachment.

MMS headers are encoded in the SOAP header and SOAP body.

MMS content is placed in the SOAP attachment as a MIME multipart.

SOAP MIME Types

When a SOAP transaction contains only a SOAP header and body (no MMS content objects), such as a delivery or read report, the MIME type text/xml is used.

When a SOAP transaction contains a SOAP attachment (one or more MMS content objects), the MIME type multipart/related is used. This multipart will always consist of exactly two parts, even if the MMS message contains more than one content object. One part is the text/xml SOAP header and body. The other part is a multipart object containing the MMS content objects. This can be confusing as one multipart object is nested inside another.

Sample MM7 Message

The first object in the outer multipart object is the SOAP header and body, which is an XML document containing MMS headers

```
POST /mm7 HTTP/1.1
Host: mms.omms.com
Content-Type: multipart/related; boundary="NextPart_000_0028_01C19839.84698430"; type=te
xt/xml; start="</tnn-200102/mm7-submit>"
Content-Length: nnnn
SOAPAction: ""

--NextPart_000_0028_01C19839.84698430
Content-Type:text/xml; charset="utf-8"
Content-ID: </tnn-200102/mm7-submit>

<?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/sch
ema/REL-5-MM7-1-3" env:mustUnderstand="1">
vas00001-sub
</mm7:TransactionID>
</env:Header>
<env:Body>
<SubmitReq xmlns="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-MM
7-1-3">
<MM7Version>5.6.0</MM7Version>
<SenderIdentification>
<VASPID>TNN</VASPID>
<VASID>News</VASID>
</SenderIdentification>
<Recipients>
<To>
<Number>7255441234</Number>
</To>
<Cc>
<Number>7255443333</Number>
</Cc>
<Bcc>
<RFC2822Address>7255444444@OMMS.com</RFC2822Address>
</Bcc>
</Recipients>
<ServiceCode>gold-sp33-im42</ServiceCode>
<LinkedID>mms00016666</LinkedID>
<MessageClass>Informational</MessageClass>
<TimeStamp>2002-01-02T09:30:47-05:00</TimeStamp>
<DeliveryReport>true</DeliveryReport>
<Priority>Normal</Priority>
<Subject>News for today</Subject>
<ChargedParty>Sender</ChargedParty>
<DistributionIndicator>true</DistributionIndicator>
<Content href="cid:SaturnPics-01020930@news.tnn.com" allowAdaptations="true"/>
</SubmitReq>
</env:Body>
</env:Envelope>
```

Sample MM7 Message (continued)

The second object in the outer multipart object is the SOAP attachment, which contains the MMS content objects.

This is a multipart object encoded inside another multipart object.

```
--NextPart_000_0028_01C19839.84698430
Content-Type: multipart/mixed; boundary="StoryParts-74526-8432-2002-77645"
Content-ID:<SaturnPics-01020930@news.tnn.com>

--StoryParts-74526-8432-2002-77645
Content-Type: text/plain; charset="us-ascii"

Science news, new Saturn pictures...

--StoryParts-74526-8432-2002-77645
Content-Type: image/gif
Content-ID:<saturn.gif>
Content-Transfer-Encoding: base64

R0lGODdhZAAwAOMAAAAAAIGJjGltcDE0OOfWo6Ochbi1n1pmcbGojpKbnP/lpW54fBMTE1RYXEFO

...

--StoryParts 74526-8432-2002-77645--
--NextPart_000_0028_01C19839.84698430--
```

MM7 PDU Kinds

There are two kinds of MM7 PDUs:

1. **Request**: Denoted as MM7_type-name.REQ
2. **Response**: Denoted as MM7_type-name.RES

MM7 PDU Transmission

MM7 PDUs are sent using HTTP POST.

The payload of the HTTP request contains the MM7 Request.

The payload of the HTTP response contains the MM7 Response.

MM7 PDUs

Transaction	Flow of Transaction	PDU Type Name
Submit Message	POST: VASP to MMSC	MM7_submit.REQ
	Response from MMSC	MM7_submit.RES
Deliver Message	POST: MMSC to VASP	MM7_deliver.REQ
	Response from VASP	MM7_deliver.RES
Delivery Report	POST: MMSC to VASP	MM7_delivery_report.REQ
	Response from VASP	MM7_delivery_report.RES
Read Report	POST: MMSC to VASP	MM7_read_reply_report.REQ
	Response from VASP	MM7_read_reply_report.RES
MMSC Error	Response from MMSC	MM7_RS_error.RES
VASP Error	Response from VASP	MM7_VASP_error.RES

MM9 – Billing & Charging Interface

- Billing generally refers to post-paid CDR generation
- Charging generally refers to real time charging required by pre-paid
- NowSMS MMSC supports HTTP based accounting callbacks which some customers use to generate CDRs (or can be used for real time charging)
- NowSMS MMSC supports DIAMETER Credit Control or Base Accounting for real time charging.

DIAMETER/MM9

NowSMS MMSC implements Diameter Credit Control for charging based upon the following specifications:

- [DIAMBASE] – **RFC 3588 – Diameter Base Protocol**
- [DIAMCCA] – **RFC 4006 – Diameter Credit Control Application**
- [3GPPDIAM] – **3GPP TS 32.299 – 3GPP Diameter Charging Applications**
- [SCAPv2] – **Ericsson CCN SCAPv2**

DIAMETER/MM9 (continued)

[3GPPDIAM] defines a standard format for implementing MMS charging over Diameter, known as MM9 in the MMS protocol specifications. While the MMSC preference is to use MM9, it is also possible to configure the MMSC to use generic charging primitives defined in [DIAMCCA] in order to facilitate interoperability with a wider base of charging systems.

Optionally, Diameter extensions defined in Ericsson's [SCAPv2] protocol can be enabled.

DIAMETER/MM9 (continued)

The MMSC Diameter MM9 implementation is highly configurable, with configuration settings that allow the Diameter requests to be tuned to meet the requirements of different charging systems. The core Diameter charging requests use Credit-Control-Request as defined in [DIAMCCA]. Several extension sets are defined that provide additional Diameter parameters to be include in the charging request.

For maximum flexibility, XML template files are used to define the underlying Diameter requests, allowing for custom parameters to be added or deleted as required.

As an example of this flexibility, templates have been created to support Ericsson SCAPv1, which uses Diameter Base Accounting instead of Diameter Credit Control.

DIAMETER/MM9 (continued)

More on the web site at

http://www.nowsms.com/mmsc-diameter-mm9-implementation

MMSC Accounting Callbacks

MMSC accounting callbacks provide an interface between the NowSMS MMSC and external billing and charging systems.

These MMSC accounting callbacks are HTTP-based. When accounting callbacks are enabled, the MMSC will issue HTTP requests to a customer supplied URL in order to interface with the customer billing and charging systems.

To enable MMSC accounting callbacks, it is necessary to manually edit the MMSC.INI configuration file, and define the callback URL under the [MMSC] section header, using the following configuration parameter:

MMSAccountingURL=http://server/path

MMSC Accounting Callbacks (continued)

Whenever the MMSC processes an MMS message, it issues an accounting callback by issuing an HTTP transaction to the callback URL. Variables describing the MMS transaction are appended to the MMSAccoutingURL as HTTP GET CGI-style variables, with standard URL escaping applied for encoding reserved characters.

For example:

http://server/path?PreAuth=Yes&Type=MMSSend&From=%2B449999999999&To=%2B447777777777&MsgCount=1

MMSC Accounting Callbacks (continued)

Most of the accounting callbacks are informational only, and exist to record charging information after the MMSC has processed a transaction.

However, there are also pre-authorisation callbacks which occur before the MMSC processes a transaction. These pre-authorisation callbacks exist to allow the customer billing system to decide whether or not the transaction should be allowed. In this scenario, the callback could check available credit and reject an MMS message transaction before it is accepted by the MMSC.

MMSSend PreAuth Callback

This callback is executed when an MMS subscriber, Value Added Service Provider (VASP) or MMSC interconnect partner, is requesting to send a message.

This is a “pre-authorisation” request, and does not mean that the message will actually be accepted by NowSMS for delivery. If NowSMS cannot successfully connect to the accounting URL, or the URL returns a response other than a standard “HTTP 200 OK” response, the user request to send a message will be blocked. A “PreAuth” request to send a message will also be blocked if the HTTP response content includes the text “PreAuth=Deny”.

MMSSend PreAuth Callback (continued)

The following parameter variables may be set for the MMSSend pre-authorisation request:

PreAuth=Yes

The presence of this parameter indicates that this callback is a pre-authorisation request.

Type=MMSSend

The transaction type is MMSSend, indicating that a request is being made to send an MMS message.

From=SenderPhoneNumber

This parameter contains the phone number of the subscriber that is sending the message. Note that URL escaping rules require the “+” symbol to be encoded as “%2B”.

To=RecipientPhoneNumber (may be a comma delimited list with multiple recipients)

This parameter contains one or more recipient phone numbers. If more than one phone number is present, this will be a comma delimited list of recipient phone numbers. (Note that URL escaping rules require the “,” symbol to be encoded as “%2C”.)

MMSSend PreAuth Callback (continued)

VASPIN=MmscVaspName

This parameter is present if the message is arriving from a Value Added Service Provider or MMSC interconnect partner. The value of this parameter refers to the account name as defined in the “MMSC VASP” list.

Note that some versions of NowSMS may preface the MmscVaspName with the text “VASP:”.

VASP=MmscOutboundRoute (may be a comma delimited list if multiple recipients)

This parameter is present if the MMSC has determined that the message must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the “MMSC Routing” list.

If the message is being sent to multiple recipients, this field may contain a comma delimited list of routes with a route listed for each recipient. If there is a mix of local and remote recipients, local recipients will have a blank entry within the comma delimited list of routes.

MMSSend PreAuth Callback (continued)

VASPIN=MmscVaspName

This parameter is present if the message is arriving from a Value Added Service Provider or MMSC interconnect partner. The value of this parameter refers to the account name as defined in the “MMSC VASP” list.

Note that some versions of NowSMS may preface the MmscVaspName with the text “VASP:”.

VASP=MmscOutboundRoute (may be a comma delimited list if multiple recipients)

This parameter is present if the MMSC has determined that the message must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the “MMSC Routing” list.

If the message is being sent to multiple recipients, this field may contain a comma delimited list of routes with a route listed for each recipient. If there is a mix of local and remote recipients, local recipients will have a blank entry within the comma delimited list of routes.

MMSSend PreAuth Callback (continued)

MsgCount=####

This parameter specifies the number of recipients for this MMS message transaction.

(Note that the MMSSend PreAuth callback is issued only once if an MMS message is being sent to multiple recipients. The MMSSend Charging callback, which records billing and charging information after the MMSC has accepted the message, issues a separate callback for each recipient.)

Size=####

This parameter specifies the size of the MMS message in bytes.

MMSSend PreAuth Callback (continued)

Depending on system configuration, additional parameters such as SGSN Address, SGSN MCC MNC (used to detect roaming subscribers) and IMSI may also be available to the callback. For information on these parameters, see **Detecting Roaming Subscribers** at http://www.nowsms.com/operator-mmsc-accounting-detecting-roaming-subscribers

Additional callback documentation can be found at http://www.nowsms.com/doc/advanced-configuration-settings/mms-accounting-callbacks

MM5/MNP/MMS Routing

The NowSMS MMSC implements a dynamic MMS routing callback facility for environments where more advanced MMS routing capabilities are required.

The standard NowSMS MMSC configuration allows for MMS routing based upon phone number prefixes. However, in MNP environments, it may be necessary to query HLR or a database to determine how to properly route an MMS message.

When the MMS Routing callback is enabled in NowSMS, each time the MMSC receives a message, it will connect to a configurable customer-provided routing callback URL, passing the message recipient to the URL. The customer provided URL can return a response to indicate that the message should be routed via a specific route defined in the “MMSC Routing” page of the NowSMS configuration dialog, or the response can indicate that the message should be rejected.

MM5/MNP/MMS Routing (continued)

The MMS routing callback URL is defined in the MMSC.INI file, under the [MMSC] section header:

MMSRoutingURL=http://server.name/path

The variables listed below will be added to the MMSRoutingURL when the URL is executed by the gateway as HTTP GET (CGI-style) parameters.

Type=MMSRouteCheck (Note: Future “Type” values may be added in the future.)

From=SenderPhoneNumber or e-mail addressVASPIN=VASPname (present if the message was received via a specific account defined in the “MMSC VASP” list)

To=RecipientPhoneNumber

MM5/MNP/MMS Routing (continued)

Example:

http://server.name/path?Type=MMSRouteCheck&From=%2B1234567&VASPIN=test&To=%2B9999999999

(Note: The “%2B” in the above examples is standard URL escaping for the “+” character.)

To specify which of the routes defined in the “MMSC Routing” list should be used to route this message, the URL must return a standard HTTP 200 OK response, and include the following text somewhere in the response:

Route=xxxxxxx

“xxxxxxx” should match an “Account Name” defined in the “MMSC Routing” list, or it can use the predefined values of “Direct” (signifying MMSC Direct Delivery), “WAPPush” (signifying “Convert to Multimedia WAP Push”), or “BlockMessage” (signifying the MMSC should reject the message).