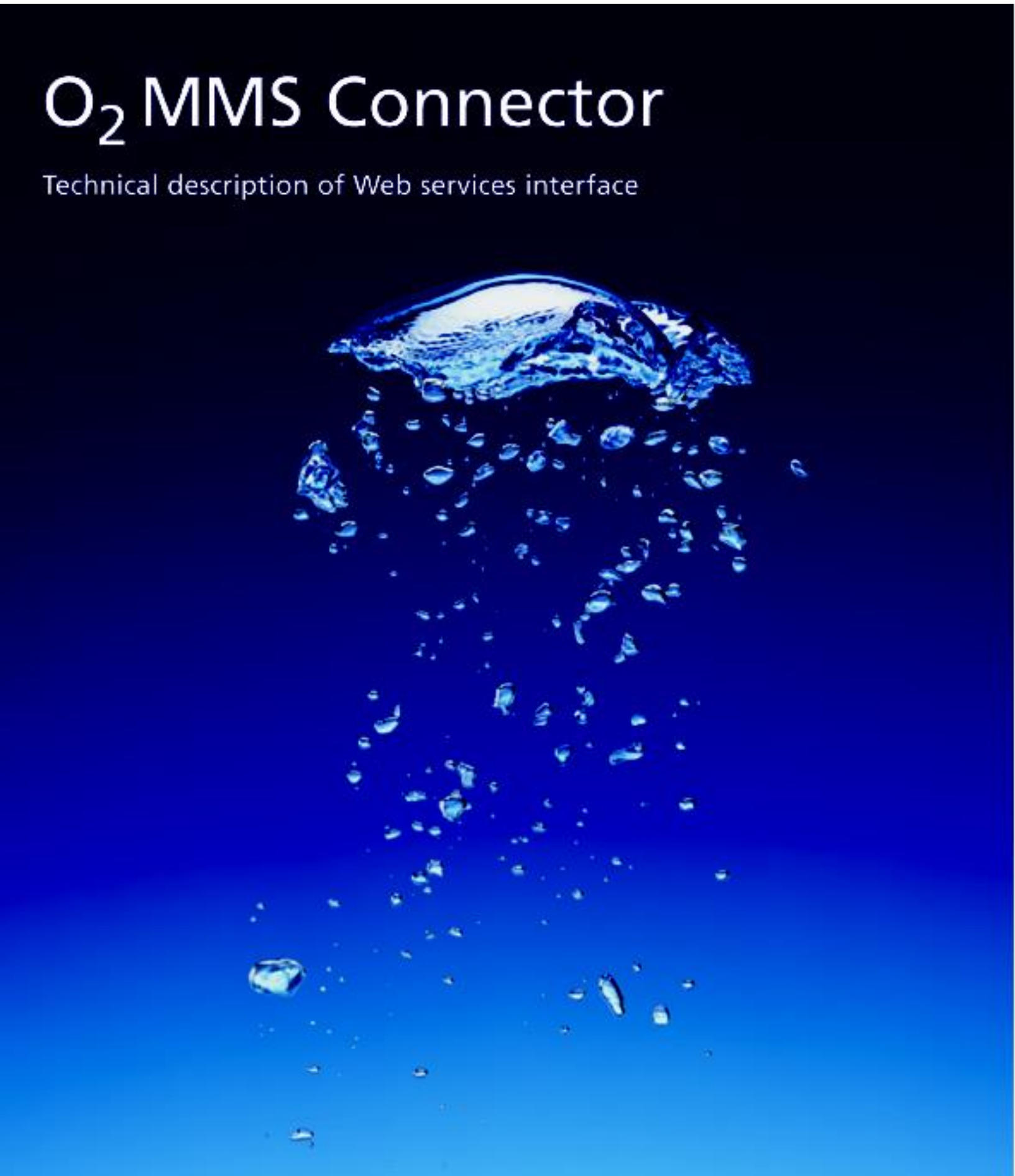
O_2 MMS Connector

Technical description of Web services interface



Content

1 Introduction

This document is for use by O2 customers wishing to send and receive MMS messages to/from mobile subscribers through Web Services. This document provides a description of the Web Services interface of the MMS Connector service.

The key words "MAY", "MUST", "MUST NOT", "OPTIONAL", "RECOMMENDED", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT" in this document are to be interpreted as described in [RFC2119].

1.1 Definitions, Acronyms and Abbreviations

Abbreviation	Definition
AO	Application-Originated - is a term used to determine the identification of the source direction or originator of a message
API	Application Program Interface
AT	Application-Terminated - is a term used to determine the identification of the target direction or destination of a message.
BA	Business Application
CA	Certification Authority
CDR	Call Detail Record
M2M Protocol	Machine-to-machine protocol. Set of communication protocols used in O2 for communication between applications.
M2M WS Client	Web Services Client comunicating with M2M Server through M2M protocol. Customers of O2 MMS Connector implement M2M WS Clients to send and receive messages with O2 MMS Conector.
M2M WS Server	Web Services Server implementing M2M Protocol. M2M WS Server is used for exposing O2 MMS Connector services.
MMS	Multimedia Message Service - is a service for sending multimedia messages to mobile phones Multimedia message content examples include images, audio, text, video and combinations of these
MMS Connector	MMS Connector Message Service – This is the product that business partners will use to send and receive MMS messages
MMSC	MMS Center – The O2 network element that the MP sends MMS messages to and receives MMS messages from mobile phones
MO	Mobile-Originated - is a term used to determine the identification of the source direction or originator of a message
MP	Messaging Platform - A logical system made up of the Web Services Gateway, HTTP Server, and other internal O2 systems
MSISDN	The mobile user's telephone number.
MT	Mobile-Terminated - is a term used to determine the identification of the target direction or destination of a message
RPC	Remote Procedure Call
SMS	Short Message Service - is a service for sending text or binary messages to mobile phones

SMSC	SMS Center – The O2 network element that the MP sends SMS messages to and receives SMS messages from mobile phones
SMSR	SMS Redirector – O2 network feature enabling to receive MO SMS from network of other operator
SOAP	Simple Object Access Protocol – a W3C recommendation / standard that is used by Web Services
Web Services	standards-based, interoperable, self-contained, modular applications that can be described, published, located, and invoked over the internet
WSDL	Web Services Definition Language.
XML	Extensible Markup Language
XSD	XML Schema Definition

2 Bibliography

[COOKIES]	Persistent Client State -- HTTP Cookies.	http://www.netscape.com/newsref/std/cookie_spec.html
[COOKIES2]	HTTP State Management Mechanism.	http://www.ietf.org/rfc/rfc2965.txt
[HTTP/TLS]	HTTP over TLS.	http://www.ietf.org/rfc/rfc2818.txt
[HTTP]	RFC 2616	http://www.w3.org/Protocols/rfc2616/rfc2616.html
[PKI]	Internet X.509 Public Key Infrastructure Certificate.	http://www.ietf.org/rfc/rfc3280.txt
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels	http://www.ietf.org/rfc/rfc2119.txt
[SOAP]	Simple Object Access Protocol (SOAP) 1.1.	http://www.w3.org/TR/soap/
[SSL]	The SSL Protocol Version 3.0 .	http://wp.netscape.com/eng/ssl3/draft302.txt
[TLS]	The TLS Protocol Version 1.0.	http://www.ietf.org/rfc/rfc2246.txt
[W3C WS]	W3C Web Services Description Working Group.	http://www.w3.org/2002/ws/desc
[WSDL]	Web Service Descrition Language (WSDL) 1.1.	http://www.w3.org/TR/wsdl
[WSI-BP]	WS-I Basic Profile 1.1.	http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html
[XML Namespaces]	XML Namespaces 1.0.	http://www.w3.org/TR/1999/REC-xml-names-19990114/
[XML Schema]	XML Schema . World Wide Web Consortium	http://w3.org/XML/Schema
[XML]	Extensible Markup Language (XML) 1.0 (Second Edition). World Wide Web Consortium	http://w3.org/XML
[GSM340]	GSM 03.40 Technical	

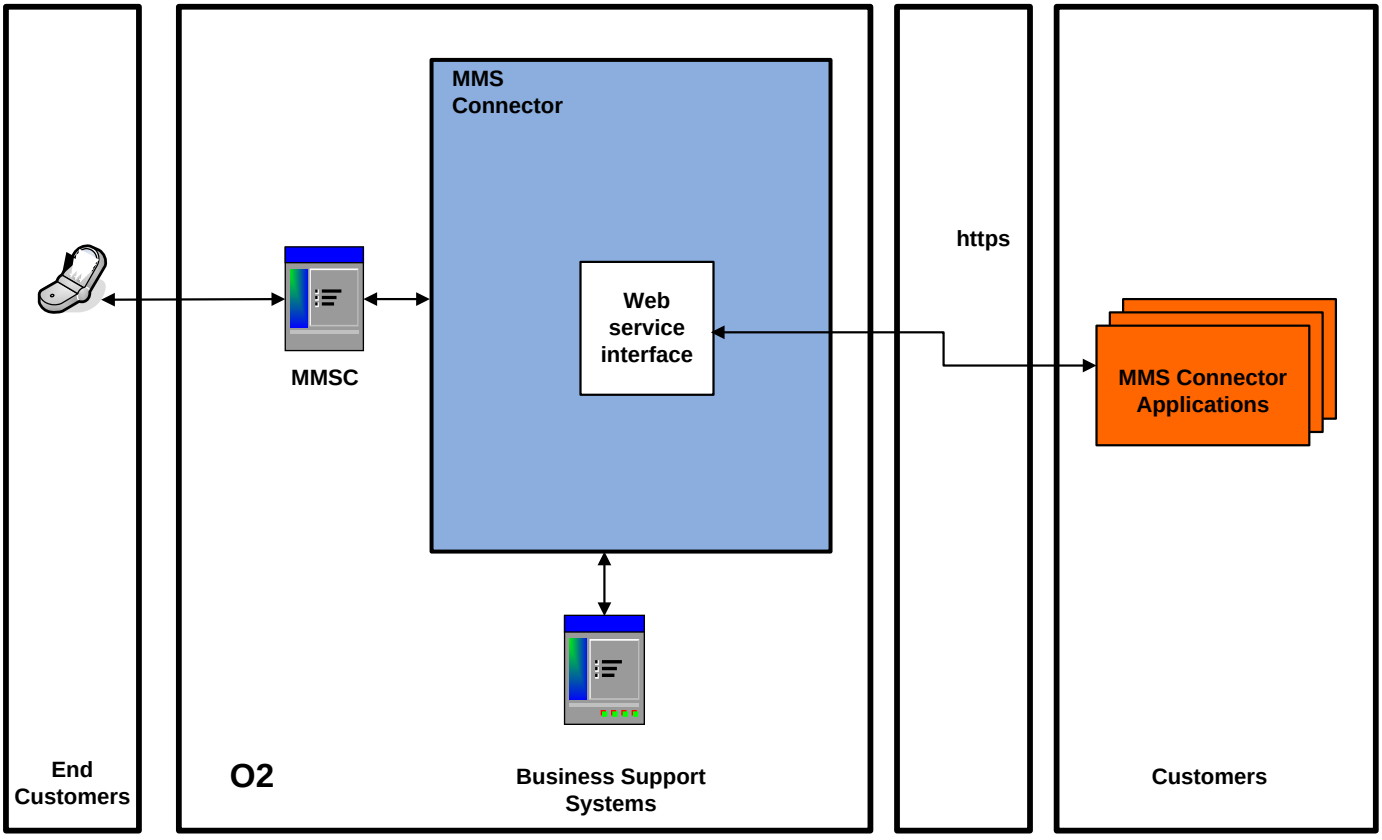
	realization of the Short Message Service
[GSM338]	GSM 03.38 Alphabets and language-specific information.
[3GPP23038]	3GPP 23.038 Alphabets and language-specific information.

3 Description of MMS Connector Web Services Interface

The MMS Connector provides business applications with a highly scalable and reliable messaging channel. It is integrated into O2's existing back-end systems such as pre- and post-paid billing, customer care, customer management, etc.

3.1 Architectural View

The following diagram shows the main components of the MMS Connector service:



O2:

Offers MMS Connector Service to the Customers. MMS Connector service communicates with customers using Web Services interface for MMS messaging. Web Services interface acts as a proxy between the inner O2 messaging platform

and the MMS Connector customer‘s application. It provides a secure method of sending and receiving messages by using HTTPS and client side authentication. The description of the services is done via WSDL file and thereby allows platform - independent access to the MMS Connector. O2 is responsible for Messaging Delivery, Billing and Customer Care.

MMS Connector Customers:

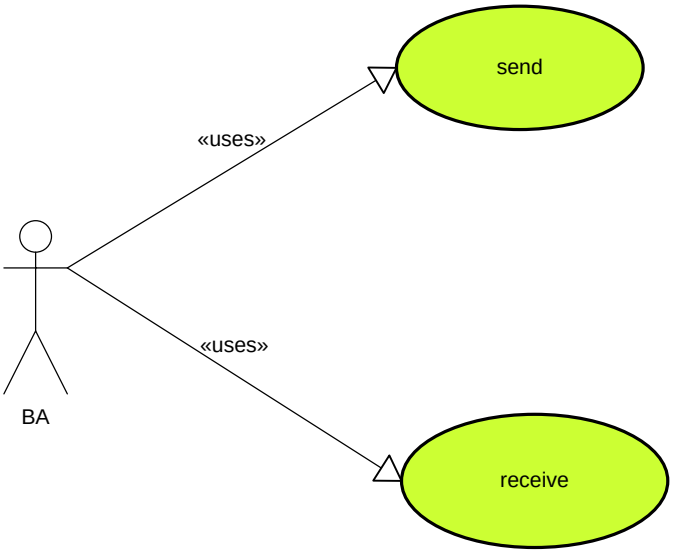
Customers using MMS Connector service. MMS Connector Customers implements applications that use the MMS Connector service for sending and receiving MMS.

End Customer:

An individual who receives/sends MMS from/to an application of a O2 MMS Connector Customer.

3.2 Usage Scenarios

The key usage scenarios are described in the form of Use Cases:



Scenario “send” enables the MMS Connector Customer’s application to send one or more AO MMS messages the to the End customer.

Scenario “receive” enables the MMS Connector Customer’s application to receive one or more MO MMS sent from End customer, and receive MMS Delivery Reports.

3.3 Operations

Scenarios are realized by invoking operations of MMS Connector Web Services interface by MMS Connector Customer’s application. The operations are:

Operation	Parameters (Type)	Returns (Type)	Description
Send	Pack (Pack)		Sends a pack of messages
Receive		Pack (Pack)	Receives a pack of messages and the id of the pack in the header
Confirm	PackID (String)		Confirms the pack of messages specified by pack id, the pack id MUST be the same as was returned by the operation Receive

Reject	PackID (String)	Rejects the pack of messages specified by pack id, the pack id MUST be the same as was returned by the operation Receive

The Operations are formally described in the sMessageService.wsdl file, their input and output parameters in file sMessageService.xsd.

3.4 Description of Scenarios

Scenarios are described in generic terms of O2 M2M (Machine to machine) protocol.

The description uses following parties of communication:

`M2M WS Server`: Web service server

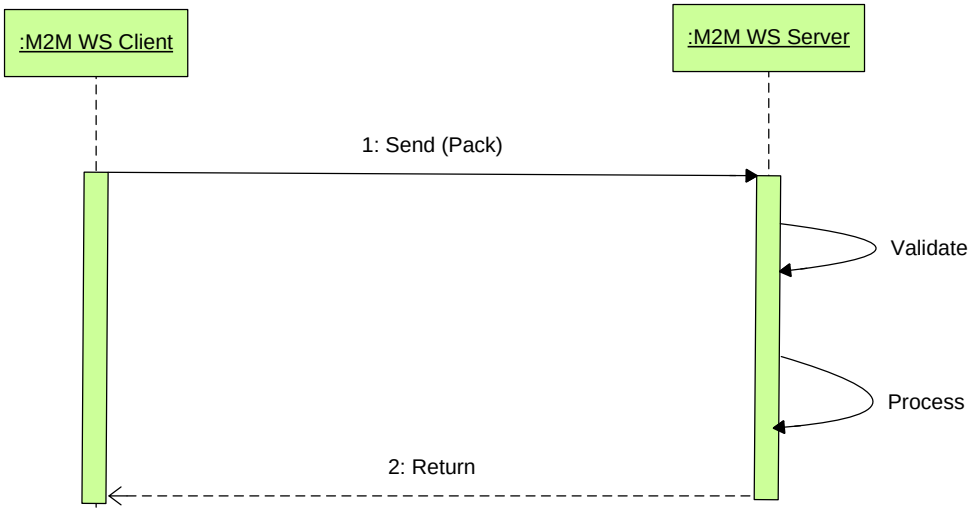
`M2M WS Client`: Web service client

`M2M WS Server` is implemented on the O2 side and customers implement the `M2M WS Client`.

The `M2M WS Server` and `M2M WS Client` mutually exchange m2m protocol messages. In case of the MMS Connector service each m2m protocol message means a single pack of messages of type MMS, Acceptance/Rejection or Delivery report.

3.4.1 Scenario Send

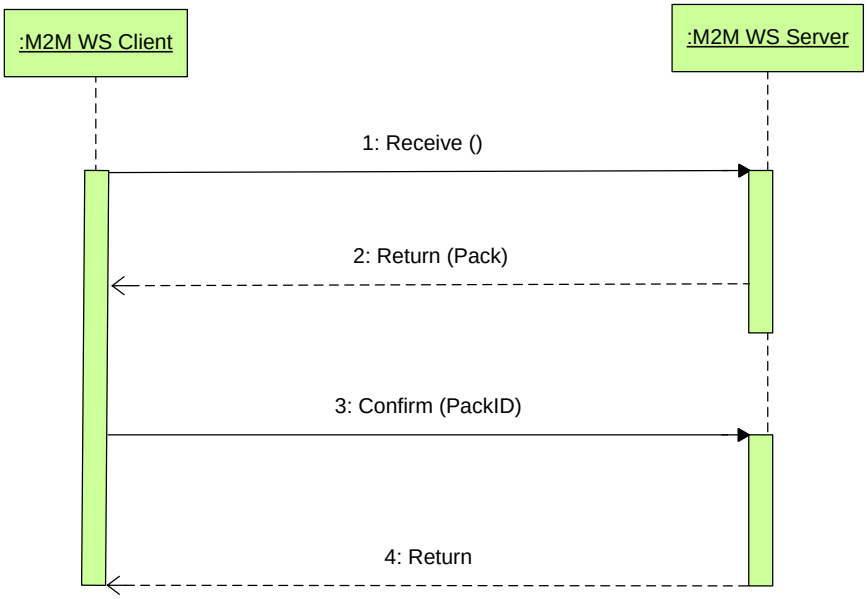
Sending a pack of messages is realized as a simple SOAP call.



`M2M WS Client` calls the *`Send`* operation to send a pack of messages. In case of MMS Connector each message can be MMS message.

3.4.2 Scenario Receive

Receiving messages is realized using a polling mechanism, where `M2M WS Client` periodically polls the `M2M WS Server` to obtain messages.



`M2M WS Client` calls the `Receive` operation to retrieve any message that is available. If there is none, the `M2M WS Client` MUST wait for an agreed amount of time (see parameter: `waiting time on empty queue`) and repeat the call. Client SHOULD NOT call `Confirm` or `Reject` because there is nothing to confirm or reject.

In case there is a message available, the *`Receive`* operation returns a `Pack` containing set of available messages. The returned pack contains a `PackId` that is an identification of the pack that MUST be used in subsequent call to `Confirm`.

In case of MMS Connector each message can one:

- MO-AT MMS message
- Acceptance / Rejection Response: Information, that an AO-MT MMS message was accepted / rejected for processing.
- MMS Delivery Report: Information, that a AO-MT MMS message was delivered to a End customer's mobile station

The `M2M WS Client` SHOULD forward the retrieved pack of messages securely for furher internal processing (i.e. client SHOULD NOT process the messages synchronously) and confirm the reception by calling the *`Confirm`* operation within an agreed timeout (see confirmation timeout) from the reception. The confirmation after timeout returns fault and the messages will be delivered again. The `M2M WS Client` MUST repeat the `PackId` returned by operation `Receive` when it calls operation `Confirm` or `Reject`.

The client MUST NOT call the operation `Reject` after operation `Confirm` with the same `PackId` or vice versa.

In case the client cannot ensure processing of the pack of messages (e.g. due to technical problems) it SHOULD call the `Reject` operation to notify the `M2M WS Server` that the message was not succesfully retrieved. In such case the messages will be redelivered in some of the subsequent invocations of `Receive` operation.

4 Faults

This chapter describes faults that can be returned when invoking MMS Connector operations.

Error	HTTP Status Code	SOAP Fault	Fault Code	Description
Request validation failed.	400 Bad Request	No	-	The XML document isn't well-formed or doesn't correspond to WSDL or XML Schemas.
Client authentication failed.	403 Forbidden	No	-	Invalid client certificate or login/password - depends on the authentication method.
General server error.	500 Internal Server Error	No	-	Some error occured on the server side. Repeating the call later MAY produce the desired result.
SOAP Request processing error.	500 Internal Server Error	Yes	Server	Some error occured during the processing of the request. Repeating the call later MAY produce the desired result.
SOAP must understand fault.	500 Internal Server Error	Yes	MustUnderstand	The SOAP request contains a mandatory header block (one that has an attribute mustUnderstand with value "1") which the server does not understand.
Too many unconfirmed packs.	500 Internal Server Error	Yes	Client.Limit.Unconfirmed	The client violated the maximum number of unconfirmed packs limit.
Pack confirmation failed.	500 Internal Server Error	Yes	Client.ConfirmFailed	The pack was confirmed later than the confirmation timeout specifies or was already confirmed by another thread.
Throughput limit exceeded.	500 Internal Server Error	Yes	Client.Limit.Throughput	The client violated the maximum throughput limit. In this case, the client SHOULD NOT repeat the call until the returned timeout expires.

Too many concurrent connections.	500 Internal Server Error	Yes	Client.Limit.Connections	The client violated the maximum number of concurrent connections limit.

4.1 Meaning of the Fault Codes

Server ({http://schemas.xmlsoap.org/soap/envelope/}Server) - the fault announces that the call to the service failed, and did not have any affect. Repeating the call later MAY produce the desired result.

Client ({http://schemas.xmlsoap.org/soap/envelope/}Client) - the fault informs us about the wrong data, which were sent to the service or the interface specification was violated or some of the limits were exceeded. The client SHOULD NOT repeat the same call. There is an exception from this, whether the fault code is Client.Limit, which means, that the client exceeded one of the limits, so MAY repeat the same call later.

4.1.1 Throughput limit exceeded Fault

This fault MAY contain a timeout value in milliseconds in element {http://cz.o2.com/sMessage/sMessageService}RetryAfter, which will be a child of element {http://schemas.xmlsoap.org/soap/envelope/}detail in SOAP fault. If the timeout is present, the `M2M WS Client` SHOULD NOT repeat the call until the returned timeout expires. See Example of throughput limit exceeded SOAP fault.

5 XML types and elements

5.1 Overview

This chapter describes XML types used in input / output parameters of operations of the MMS Connector service. They are based on XML ver 1.0 [XML] and defined using XML schemas compliant with W3C XMLSchema recommendation [XMLSCHEMA].

The XML elements and types are defined in following namespaces:

Namespace	Description	Located in
http://cz.o2.com/sMessage/sMessageService	Defines types of input and output parameters of operations defined in sMessageService.wsdl.	sMessageService.xsd
http://cz.o2.com/sMessage/Envelope	Defines XML type containing M2M Service Message (sMsg:Envelope) and Service Message pack (sMsg:Pack). Prefix used in this document: sMsg	sMessageEnvelope.xsd
http://cz.o2.com/sMessage/Payload	Defines abstract XML complex type m2mPayload:Payload, which is extended by all service specific M2M message payloads. Prefix used in this document: m2mPayload	sMessagePayload.xsd
http://cz.o2.com/sMessage/Acceptance	Defines M2M message header element and payload types for acceptance and rejection. Prefix used in this document: acceptance	sMessageAcceptance.xsd
http://cz.o2.com/Messaging	Defines Payload types for messages used in MMS Connector Service Prefix used in this document: msg	Messaging.xsd

The prefixes used are provided only for readability's sake. Actual messages MAY use any prefix.

5.2 Namespace sMessageService

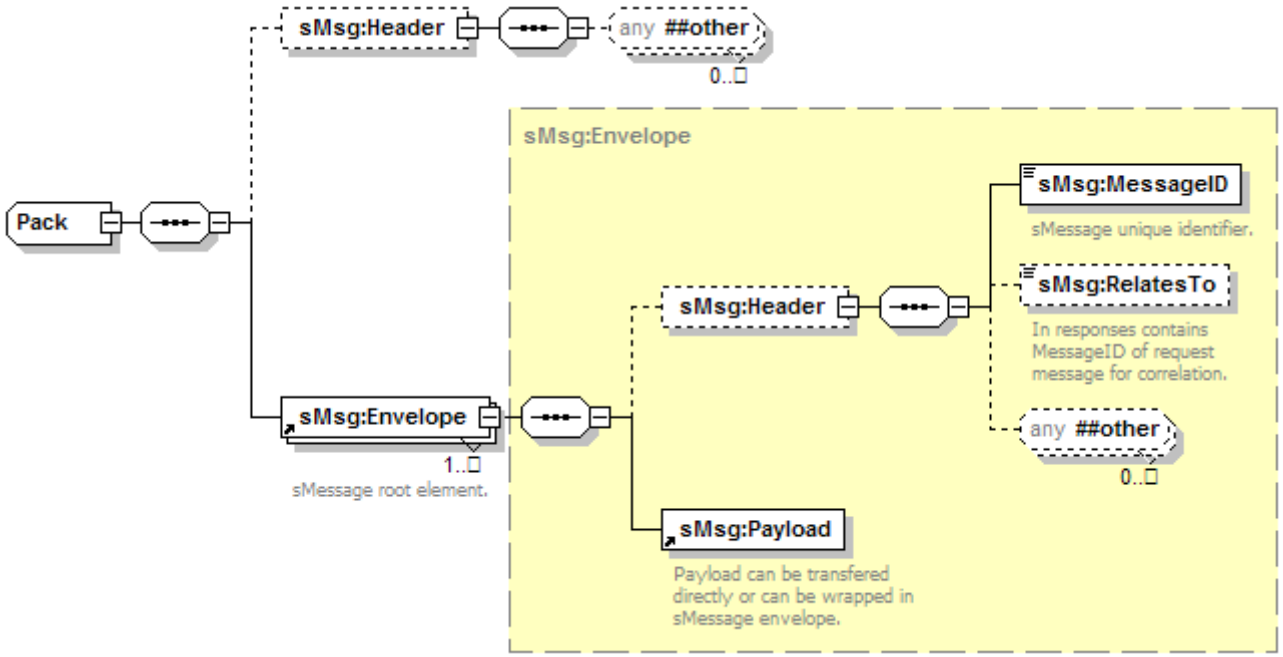
This namespace defines elements used in input and output parameters of operations defined in sMessageService.wsdl. Each operation has its corresponding type for input and for output.

The following elements are defined:

Send	Send operation request containing a Pack of messages
SendResponse	Send operation response
Receive	Receive operation request
ReceiveReponse	Receive operation response containing a Pack of messages
Confirm	Confirm operation request containing id of pack to confirm
ConfirmResponse	Confirm operation response
Reject	Reject operation request containing id of pack to reject
RejectResponse	Reject operation response
PackId	Definition of id of pack

5.3 Namespace Envelope

Definitions in namespace Envelope are depicted on the following picture:



5.3.1 sMsg:Pack Complex Type:

Description	An XML representation of multiple messages accompanied with OPTIONAL metadata. Messages contained in the Pack MAY carry various Payloads.
Content	Header, Envelope

Header Element

Description	An extension point for including additional data to the Pack
Content	xs:Any

Envelope Element

Description	Element representing a single M2M message
Type	sMsg:Envelope

5.3.2 sMsg:Pack Element

Description	Element representing a set of M2M messages
Type	sMsg:Pack

5.3.3 sMsg:Envelope Complex Type

Description	Type representing a single M2M message
Content	Header, sMsg:Payload

Header Element

Description	Message header containing common metadata (MessageId and OPTIONAL RelatesTo) and optionally other metadata.
Content	MessageID, RelatesTo, xs:Any

MessageID Element

Description	Unique identifier of message with a maximum length of 128 characters
Type	xs:string

RelatesTo Element

Description	In responses contains MessageID of request message for correlation.
Type	xs:string

Payload Element

Description	Main part of message. Contains business data. Attribute "xsi:type" (xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance") MUST be used to specify the actual type of the payload being carried in the message.
Type	sMsg:Payload

5.3.4 sMsg:Envelope Element

Description	Element representing a single M2M message.
Type	sMsg:Envelope

5.3.5 sMsg:Payload Element

Description	Element representing Main part of message.
Type	sMsgPayload:Payload

5.4 Namespace Acceptance

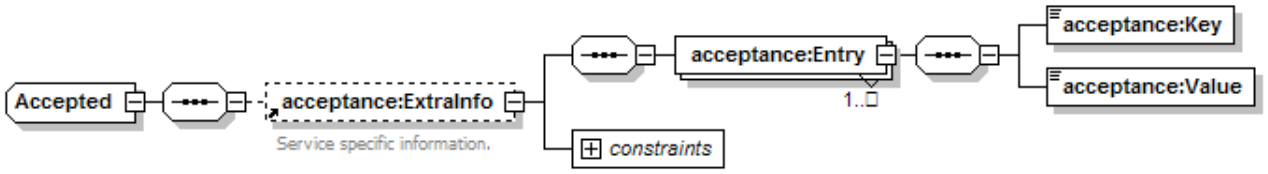
The namespace Acceptance enables the originator of the message:

- To ask for explicit information about message acceptace/rejection by including accepted:Respond element in the sMsg:Header element
- To receive informations about message acceptace/rejection by receiving messages with payload of type acceptance:Accepted or acceptance:Rejected.

5.4.1 acceptance:Respond Element

Description	Originator of the M2M message MAY ask for an explicit acceptance/rejection response by including this element in the header section of the message (sMsg:Header element inside sMsg:Envelope).
Type	restriction of xs:String
Enumeration	Never (default value): receiving party MUST NOT generate any acceptance/rejection result Rejects: receiving party MUST generate the acceptance/rejection result in case the request gets dropped from processing Always: receiving party MUST generate the acceptance/rejection result

5.4.2 acceptance: Accepted Complex Type



Description	Message containing payload of this type confirms that the message referenced using `sMsg:RelatesTo` was or certainly will be processed.
Content	Acceptance:Description, Acceptance:ExtraInfo
Type	extension of m2mPayload:Payload

Description Element

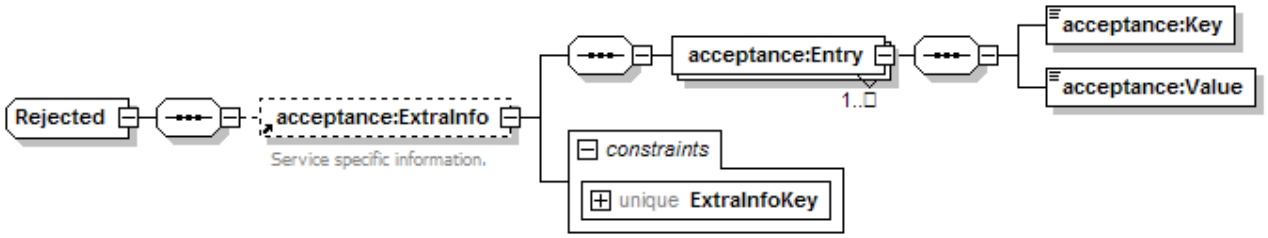
Description	Additional description of message acceptance.
Type	xs:string

ExtraInfo Element

Description	
Type	acceptance:ExtraInfo

5.4.3 acceptance:Rejected Complex type

The definition of Rejected complex type is shown in the following diagram:



Description	Message containing payload of this type confirms that the message referenced using `sMsg:RelatesTo` was rejected.
Content	Acceptance:Description, Acceptance:ExtraInfo
Type	extension of m2mPayload:Payload

Description Element

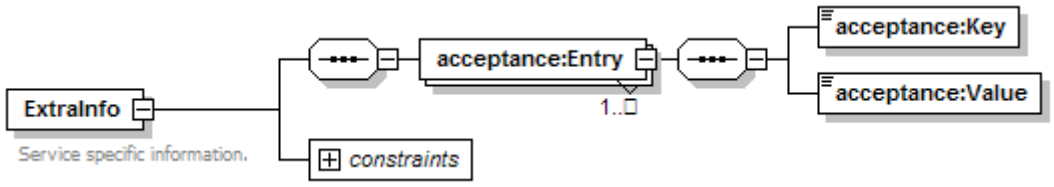
Description	Additional description of message refusal.
Type	xs:string

ExtraInfo Element

Description	
Type	acceptance:ExtraInfo

5.4.4 acceptance:ExtraInfo Complex Type

The definition of ExtraInfo Element is shown in the following diagram:



Description	Set of "key-value" pairs carrying service specific details about the acceptance or rejection.
Content	acceptance:Entry

Entry Element

Description	Additional description of message acceptance / refusal.
Content	acceptance:Key, acceptance:Value

Key Element

Description	Key
Type	xs:String

Value Element

Description	Value
Type	xs:String

5.5 Namespace Messaging

Namespace messaging contains:

- Base structure for all concrete message types, keeping common elements - UserMessage
- Type for MMS messages - MmsMessage
- Types for expressing MMS Content: MmsContent, MMSPart, PlainTextMmsPart, Base64MmsPart
- Types for expressing MMS Addresses: Address, MmsAddress, MmsPhoneNumberAddress, MmsApplicationNumberAddress, MmsEmailAddress
- Base structure for all delivery reports: DeliveryReport
- Type for MMS Delivery Report: MmsDeliveryReport
- Simple types with patterns for PhoneNumber, ApplicationNumber and Email
- Simple types with enumerations for Delivery status and RecipientType

Except these also types for SMS and WapPush messaging are defined, however these are not used in MMS Connector service.

In following subchapters all types for MMS messaging are described, for complex non-abstract types also pictures are included.

5.5.1 msg:Address Complex type

Description	An abstract address type
Abstract	true
Children	msg:MmsAddress

5.5.2 msg:MmsAddress Complex type

Description	An abstract MMS address type, a base for all MMS addresses.
Abstract	true
Children	msg:MmsPhoneNumberAddress, msg:MmsApplicationNumberAddress, msg:MmsEmailAddress
Type	extension of msg:Address

5.5.3 msg:MmsPhoneNumberAddress Complex Type

Description	The phone number address type in MMS messages.
Content	msg:PhoneNumber
Type	extension of msg:MmsAddress

PhoneNumber Element

Description	The phone number in international format with leading '+'.
Type	msg:PhoneNumber

5.5.4 msg:MmsApplicationNumberAddress Complex Type

Description	The application number address type in MMS messages.
Content	msg:ApplicationNumber
Type	extension of msg:MmsAddress

ApplicationNumber Element

Description	An application number.
Type	msg:ApplicationNumber

5.5.5 msg:MmsEmailAddress Complex Type

Description	An email address complex type in MMS messages.
Content	Email
Type	extension of msg:MmsAddress

Email Element

Description	E-mail address
Type	Msg:Email

5.5.6 msg:DeliveryReport Complex type

Description	An abstract type, a base for delivery report types. It keeps the common elements shared by all types.
Abstract	true
Children	msg:MmsDeliveryReport
Content	msg:Status, msg:Timestamp, msg:ExtraInfo
Type	extension of sMessagePayload:Payload

Status Element

Description	Indicates state of delivery of the message.
Type	msg:DeliveryStatus

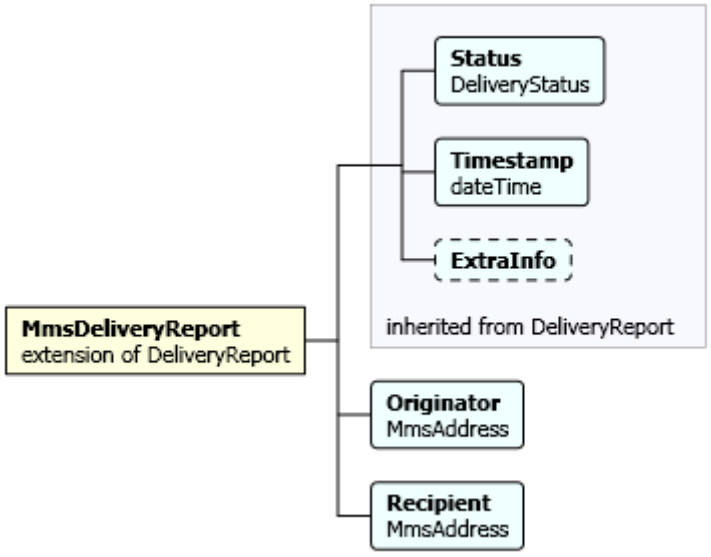
Timestamp Element

Description	A timestamp when the message was delivered by message centre in case of delivery report status delivered or was expired in message centre in case of delivery report undeliverable.
Type	xs:dateTime

ExtraInfo Element

Description	reference of msg:ExtraInfo Element

5.5.7 msg:MmsDeliveryReport Complex Type



Description	Delivery Report. Informs a message's sender about delivery of a sent MMS message.
Content	msg:Originator, msg:Recipient imported from msg:deliveryReport: msg:Status, msg:Timestamp, msg:ExtraInfo
Type	extension of msg:DeliveryReport

Originator Element

Description	MMS originator
Type	msg:MmsAddress

Recipient Element

Description	MMS recipient .
Type	msg:MmsAddress

5.5.8 msg:UserMessage Complex type

Description	An abstract, base structure for all concrete message types. It keeps the common elements shared by all message types.
Abstract	true
Children	msg:MmsMessage
Content	DeliveryReports, CreationTime, ExpirationTime, ExtraInfo
Type	extension of sMessagePayload:Payload

DeliveryReports Element

Description	A flag indicating whether a receiving of delivery reports is requested. The delivery reports MAY not be supported in some solutions.
Type	xs:Boolean

CreationTime Element

Description	The message creation time. In client application originated messages this element contains time when the message was created. In user originated messages this element contains time when the message was received from the core network systems
Type	xs:DateTime

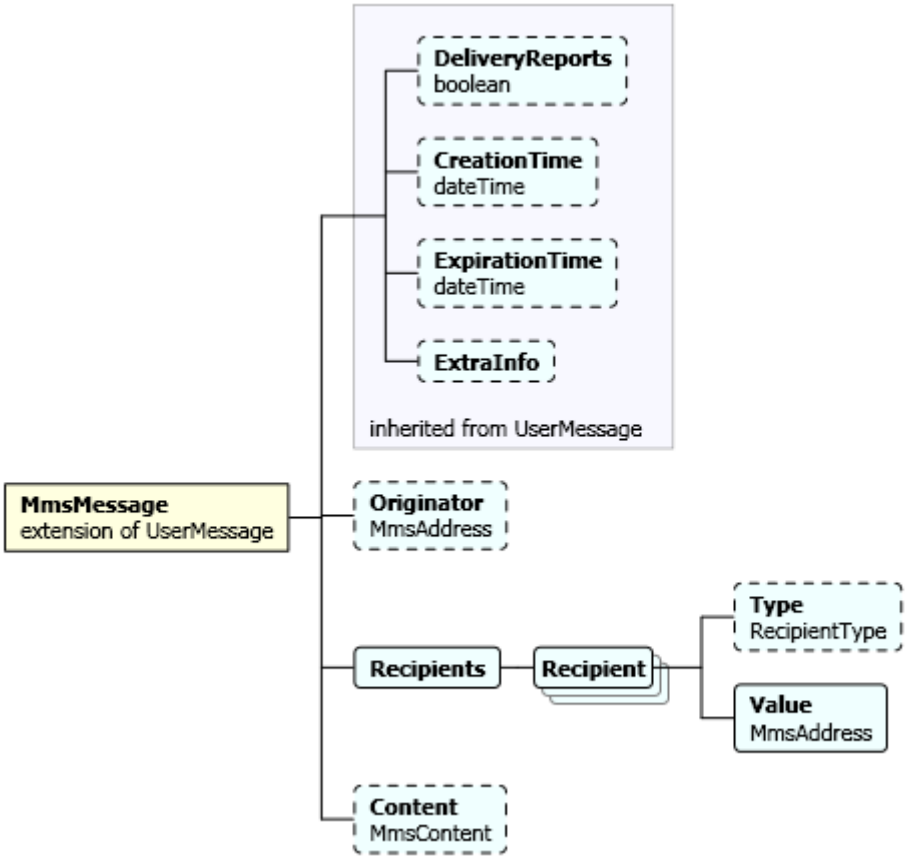
ExpirationTime Element

Description	The validity of the message, which expires at the specified time. When specified in application originated message, the value limits the time the message is waiting to be delivered to the user (in case the user is unreachable or unable to receive messages).In user originated messages the value does not have a meaning and MUST be ignored.
Type	xs:DateTime

ExtraInfo Element

Description	reference of msg:ExtraInfo Element

5.5.9 msg:MmsMessage Complex type



Description	A message containing Multi Media Service (MMS) message data.
Content	msg:Originator, msg:Recipients, msg:Content Inherited from msg:Usermessage: msg:DeliveryReports, msg:CreationTime, msg:ExpirationTime, msg:ExtraInfo
Type	extension of msg:UserMessage

Originator Element

Description	The originator of the message. The originator MAY be REQUIRED in some solutions.
Type	msg:MmsAddress

Content Element

Description	A structure holding the actual MMS content.
Type	msg:MmsContent

Recipients Element

Description	A list of recipients of the message. Duplicate recipients are not allowed.
Content	msg: Recipient
Unique	The recipients of the MMS MUST be unique xpath="msg:Recipient/msg:Value/*"

Recipient Element

Description	Represents one recipient of the message.
Content	msg:Type, msg:Value

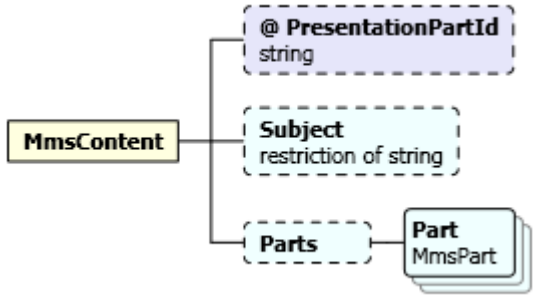
Type Element

Description	Recipient type. The enumeration of To, Cc, Bcc.
Type	msg:RecipientType

Value Element

Description	The recipient value.
Type	Msg:MmsAddress

5.5.10 msg:MMSContent Complex Type



Description	Represents the whole MMS content.
Content	msg:Subject, msg:Parts
Attributes	msg:PresentationPartId

Subject Element

Description	The subject of the MMS message.
Type	restriction of xs:string

Parts Element

Description	A list of MMS parts.
Content	msg:Part

Part Element

Description	One MMS part
Type	msg: MmsPart

PresentationPartId Attribute

Description	A reference to the presentation part of the MMS. If this attribute is specified, message is sent in “multipart related” format. Otherwise the message is sent in “multipart mixed” format.
Type	xs:string
Use	optional

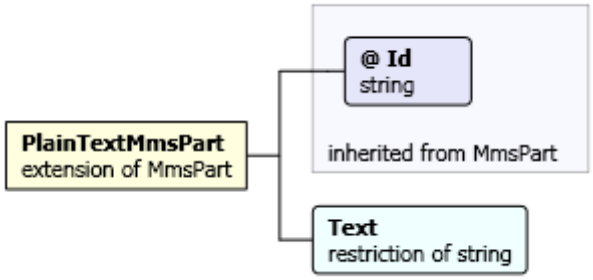
5.5.11 msg:MMSPart Complex Type

Description	An abstract, a base structure for all concrete MMS parts.
Abstract	True
Children	msg:PlainTextMmsPart, msg:Base64MmsPart
Attributes	msg:Id

Id Attribute

Description	The ID of this MMS part.
Type	xs:string
Use	required

5.5.12 msg:PlainTextMMSPart Complex Type

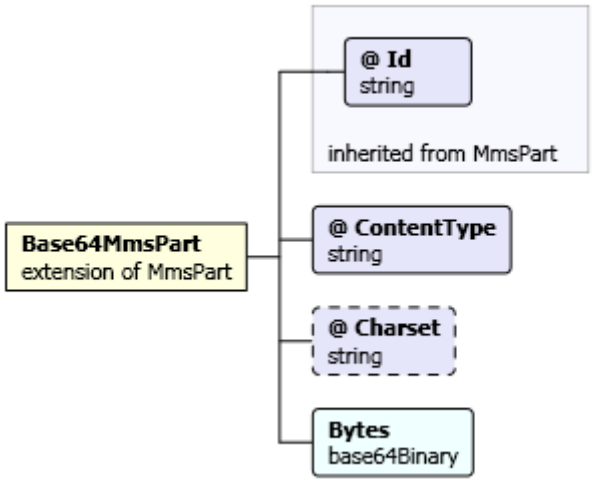


Description	A plain text MMS part complex type. Represents one text part of the MIME multipart structure. Can contain only plain text. The content-type of this part will be set to text/plain.
Content	msg:Text
Type	extension of msg:MmsPart
Attributes	Inherited from msg:MmsPart: msg:Id

Text Element

Description	The text content
Type	restriction of xs:string, minimum length: 1

5.5.13 msg:Base64MMSPart Complex Type



Description	A Base64 encoded MMS part complex type. Can contain binary content or text.
Content	msg:Bytes
Type	extension of msg:MmsPart
Attributes	ContentType, Charset Inherited from msg:MmsPart: Id

Bytes Element

Description	The Base64 encoded content.
Type	xs:base64Binary

ContentType Attribute

Description	The MIME type of the content.
Type	xs:string
Use	required

Charset Attribute

Description	The character set of the base64 encoded text. It makes sense only for text.
Type	xs:string

Use	optional

5.5.14 msg: PhoneNumber Simple type

Description	Phone number type definition
Type	restriction of xs:string
Restriction	pattern \+\d{9,20}

5.5.15 msg:ApplicationNumber Simple type

Description	Application number type definition
Type	restriction of xs:string
Restriction	pattern \+\d{3,20}

5.5.16 msg:Email Simple type

Description	Email address type definition
Type	restriction of xs:string
Restriction	pattern [A-Za-z0-9._%\-]+@[A-Za-z0-9._%\-]+\.[A-Za-z]{2,4}

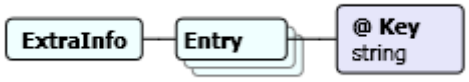
5.5.17 msg:DeliveryStatus Simple type

Description	An enumeration of statuses in delivery reports
Type	restriction of xs:string
Restriction	enumeration: Delivered enumeration: Undeliverable

5.5.18 msg:RecipientType Simple type

Description	A message Recipient type. An enumeration of To, Cc, Bcc.
Type	restriction of xs:string
Restriction	enumeration: `To` enumeration: `Cc` enumeration: `Bcc`

5.5.19 ExtraInfo Element



Description	A particular solution specific information
Content	Entry

Entry Element

Description	An actual value of the extra info item.
Attributes	msg:Key
Unique	Unique constraint. The attribute Key MUST be unique

Key Attribute

Description	An identification of the extra info item.
Type	xs:string
Use	required

6 Requirements on Client Implementation

This chapter describes requirements on implementation of the `M2M WS Client`, i.e the part of MMS Connector application that communicates with the `M2M WS Server` implemented in the MMS Connector.

6.1 Multi-Threading

To achieve higher number of messages exchanged, the `M2M WS Client` SHOULD use multiple parallel threads of control (multiple connections) when communicating with the `M2M WS Server`. Specific requirements are provided in floowing subchapters.

6.1.1 Sending messages

`M2M WS Client` MAY use multiple threads for sending messages as long as it does not exceed maximum number of concurrent connections.

6.1.2 Receiving messages

`M2M WS Client` MAY use multiple threads for receiving messages with following limitations:

Number of parallel connections MUST NOT exceed the value of parameter „maximum number of concurrent connections“

In case the `M2M WS Client` supports cookies, calls to Receive and corresponding call to Confirm MUST be made with a same set of cookies.

6.2 Location Transparency and Loadbalancing

6.2.1 Polling multiple servers

When receiving messages, `M2M WS Client` MUST allow for polling more than one `M2M WS Server`.

6.2.2 Configurable Server URL

`M2M WS Client` implementation SHOULD allow for reconfigurable Server URL i.e. for changing the URL without changing code of the application.

6.2.3 Cookie support

`M2M WS Client` SHOULD support HTTP cookies based on [COOKIES] or [COOKIES2] to allow for transparent loadbalancning on `M2M WS Server`.

To allow for better performance in multithreaded implementation, each thread SHOULD have a separate cookie "repository" (i.e. to keep its own set of cookies).

6.3 Security

6.3.1 Authentication

The client has to authenticate with SSL client certificate.

6.3.2 Transport Layer Security

HTTP with SSL/TLS transport mechanism (HTTPS) is used to achieve sufficient level of security on the transport layer (see [HTTP/TLS]).

`M2M WS Client` MUST use persistent HTTP connections.

`M2M WS Client` SHOULD use persistent SSL sessions to avoid slow and expensive SSL handshakes. Supported versions of transport layer security protocols:

- SSL version 3.0 (see [SSL])
- TLS version 1.0 (see [TLS])

`M2M WS Client` MUST check validity of the `M2M WS Server` certificate i.e.

- The certficate was issued by a trusted and previously agreed certificate authority
- The certificate has not expired or not valid yet
- Certificate is issued with a correct and previously agreed identity, which MUST be present in the Common Name in the Subject field of the certificate or in a subjectAltName extension of type dNSName (see [HTTP/TLS])

In case the above mentioned conditions are not met, client MUST NOT communicate with the server and SHOULD inform O2 representatives.

7 Implementation Notes

7.1 Application Numbers and Scenarios for Using Them

Each customer's application using MMS Connector can be assigned two application numbers, where each of them is of a different type. Each type can be used in different scenarios, therefore it is important for the customer to understand their difference and proper usage.

Application number uniquely identifies the customer's application; it is used as an `Originator` in AO-MT MMS message and as a Recipient in MO-AT MMS message. There are two types of application numbers:

BA Number (or "Short BA number")

Number adressable in MO direction only from devices of O2 mobile network customers. MMSs from End customers of other mobile operators cannot reach MMS Connector customer's application.

In MMS messages BA Numbers can be attached suffixes up to summary length of 20 digits.

MSISDN Application Number (or "Virtual MSISDN")

Number adressable in MO direction from devices of all mobile network operators End customers. MMSs can reach MMS Connector customer's application regardless of home mobile network of the customer.

MSISDN Application Number has “+420xxxxxxxxx” format; always starts with a “+” sign followed by a country prefix and six digits. The total length of MSISDN application number with a plus sign is 13. Suffixes to these numbers are not allowed in MMS messages because of general constraints on MSISDN numbers and guaranteed functionalities of MMS message centers.

The impact is, that customer of the O2 MMS Connector must decide:

- From which mobile networks are the users he wants to communicate to
- How to transfer additional information with the message provided such functionality is required.

In AO MMS messages it is possible to use each of application numbers as `Originator`. The messages can always reach the customer’s mobile phone. However, if the MMS Connector customer develops a business service using a reply scenario (i.e. End customer replies to AO message by MO message), it is necessary to keep in mind, that the reply message that the mobile phone creates, uses the `Originator`‘s value from AO message by default. So replies to AO messages from non-O2 End customers accepting default value of `Recipient` will not reach application of MMS Connector customer provided BA number is used in AO message.

MMS communication scenarios sometime utilize suffixing in AO-MT and MO-AT messages as a means to transfer additional information with the message. Additional infromation can uniquelly identify the communication between MMS Connector customer’s application and End customer, identify a service or product the communication is about etc. Because in AO-MT and MO-AT scenarios with non-O2 customers the suffixes are not supported, it is recommended to use keywords in content of messages instead.

7.2 Message Uniqueness

All nodes MUST equip all produced messages with a unique MessageId.

All nodes (both `M2M WS Client` and `M2M WS Server` MUST either ignore messages with duplicate MessageIds or behave idempotently (repeated processing of the same event MUST NOT produce undesirable results - multiplied user interaction, charging, misreporting etc). For checking of uniqueness the MessageIds are kept at least for a guaranteed period – the length of this period is set by uniqueness period deployment parameter (see uniqueness period). It means that it is guaranteed that the message with previously used MessageId MUST NOT be processed if the last use of the id occurred within uniqueness period.

8 Appendix A: WSDL and XSDs

Appendix A is a formal specification of the MMS Connector Web Services interface in form of WSDL and XSD documents. It is provided separately in MMS_Connector_AppendixA.zip file.

9 Appendix B: Examples

9.1 Examples of Operation Send

This chapter contains more examples of SOAP messages calling operation Send for sending various message types and a single example of a SOAP message response for operation Send.

9.1.1 Sending AO-MT MMS with Acceptance / Rejection Notification Requested

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Send xmlns="http://cz.o2.com/sMessage/sMessageService">
      <Pack xmlns:sMsg="http://cz.o2.com/sMessage/Envelope">
```

```
        <sMsg:Envelope>
          <sMsg:Header>
            <sMsg:MessageID>SDFW2347234829873-287348374</sMsg:MessageID>
            <acceptance:Respond
xmlns:acceptance="http://cz.o2.com/sMessage/Acceptance">Always</acceptance:Respond>
          </sMsg:Header>
          <sMsg:Payload xmlns:msg="http://cz.o2.com/Messaging" xsi:type="msg:MmsMessage"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          >
            <msg:DeliveryReports>false</msg:DeliveryReports>
            <msg:Originator xsi:type="msg:MmsApplicationNumberAddress">
              <msg:ApplicationNumber>9999975</msg:ApplicationNumber>
            </msg:Originator>
            <msg:Recipients>
              <msg:Recipient>
                <msg:Type>To</msg:Type>
                <msg:Value xsi:type="msg:MmsPhoneNumberAddress">
                  <msg:PhoneNumber>+420602608070</msg:PhoneNumber>
                </msg:Value>
              </msg:Recipient>
            </msg:Recipients>
            <msg:Content>
              <msg:Subject>Test MMS with one pic :)</msg:Subject>
              <msg:Parts>
                <msg:Part xsi:type="msg:PlainTextMmsPart" Id="txt">
                  <msg:Text>Hello, how do you like this</msg:Text>
                </msg:Part>
                <msg:Part xsi:type="msg:Base64MmsPart" ContentType="image/jpeg" Id="pic">
<msg:Bytes>/9j/4QDmRXhpZgAASUkqAAgAAAAFABIBAwABAAAAAQAAADEBAgAcAAAASgAAADIBAgAUAAAAZgAA
ABMCAwABAAAAAQAAAGmHBAABAAAAegAAAAAAAAABBQ0QgU3lzdGVtcyBEaWdpdGFsIEltYWdpbmcA
MjAwNjowODoxMCAxNjoxOTo0NwAFAACQBwAEAAAAMDIyMJCSAgAEAAAAOTMxAAKgBAABAAAAZAAA
AAOgBAABAAAAZAAAAAWgBAABAAAAvAAAAAAAAAAAACAAEAAgAEAAAAUjk4AAIABwAEAAAAMDEwMAAA
AAAAAAAA/8AAEQgAZABkAwEhAAIRAQMRAf/bAIQAAwICAgIBAwICAgMDAwMEBwQEBAQECQYGBQcK
CQsLCgkKCgwNEQ4MDBAMCgoPFA8QERITExMLDhUWFRIWERITEgEEBQUGBQYNBwcNGxIPEhsbGxsb
GxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsbGxsb/8QAlwAAAQQDAQEA
AAAAAAAAAAAABwIFBggBAwQACRAAAQMDAwIEAwUECQUAAAAAAAQIDBAUGEQASIQcxEyJBYQgUUVJx
gZGhFSMyMwkWJEJTYrHB8DRUktHhAQACAwEBAAAAAAAAAAAAAAAADBAECBQAGEQACAgEDAwMEAwEA
AAAAAAAAAQIRAwQSIRMxQSJRYSMyccFCgaHw/9oADAMBAAIRAxEAPwCqwVpW7jWaOI9vwO+loewe
+oCI3B7IxnSw7xqlFkPtGtWq1W2364UCLSoufFmO8Iz9lP2lew00URh+sdTo9NQ+zT4a1f8AWTYv
jt9wMlvcklOTknOAMnnULmVEt0hV93FCokiRRKxSGYs+K4YxlRoyY5jyEnCmXm0japJwChwAZB9d
Q+l3dGnzRClJSy+o4Qc+Rz29jq8q8FYN+R2V651zOEDVAhpJ51jP3agrQ4hQxpW4aOwCMFQ0nIB1
ARC21qU4EIBJJwAO5OjP0d6EVS7rtZqV0Nop9Biyfmuvcnw/pgHJUfQep7+uoSs6UtqCl1loT0mG
xTo9J/ZVFgNhunU1Cf5aewW4PVZ/1+7VYbvVMtmoN1pgbBBcHh5TnxVj+JIB4IAyPckn01TlTspF
```

```
pxojvV+XUpl8Gsz6bKbhViM1LZccSSh5haAUqSv+8nGMHuNpB5B0KXkFqoLhpcO5PKFdjkcj/bUt
8sLFcIIVtVkVi1EPOH9+1+7d91D1/HXesAnnVQpqIGeMa9j7vy1B1HSlXvpQOdMCiM5GNJUoAZ1U
IgqfDpQLUrPXX5+8am9EplLZDqwwgqedcWrY2hvAJ3lR4OOPy1fXplblGqF7sxf2SIFLh/2pmOtZ
cW84c7VuZyVFIBOOfNtGui+UhfL3sYutVMgu1qY7HVHLkZJUEF3yMAcZWod8Y7J8ylcJwMr189Or
MmoVy73UOpdSyySGW1jaR/mIHAPsOBwPTVcj9ROBcDGmsyq/0kbtOqxG5qrciKVCSpxTazHz5ti0
8gpUTwcjBHHGhfPaZVcxdjsuoTsSQHFhZHA9QBnQrtsbivB22JM8CvPwieHm96fvBP8AtqaqUfXU
vuFXY1k+b/7rGf8AmdccdIIxrO/TDEUeK9JUvnGdUYVBl+FCImtfFZTraSshVZcEZwAHJbAK+D6Z
Ukfhx6nVy7jkVZfXWtUSiyDCAyY6o+UlDUdtCnQP83mwD9T7apNPatvuDaTm79iJ3reRVai1tMgO
M4bYDgOGVcZcwrJWr03K7emqnXJAemViU/UJapCS6ohbjhJVyeedc3Z0Y7QeSyzQ7qhVJlQSWnHU
KV3CkuJIwR9M4499RuZBiIvdhoob8CpxG1p292Vdkn3GeD7ao0Xtp2RGGkUu/NxBSWH/AA1j6AnB
H66niiNcxuPY1Fac+msb0+2uJo2Bw6L/AEL6Rwr7se7r4uNh56lW1GbjQ4yHiymfUnzhlpSxyEJT
lagOTlIyMnRn2bEYK2kdfVno7RLTbn1KkviFSqK6mjuSH31LcqU9CcyFtIOQlpBIJ54BQPMpXAWf
S4y/tX94IPBH10OMne1jEorbuiWa/o/7TerHxmG7FgpjW3EU8VfaecyltA9+FE+ydW1r1CjXJ8TV
VlrZdXEmQJTa2mXS0tfipbCwlQ5Tyk8jR9tqK+RS/qP8AosTovKot93XdtSuN5qxoFEeS9RX5Ilq
acQ3+7c3HJQsEEqVnz55HqKOzOoNwVipvxqYmJFU4pQbdf5A+zjIwR9c49tXyRimr8hFCTT2sY6h
KrDlK8C7IjLE8Hh6KoFlwfd6H9Dpl/aJfq1OSTlxlp9lKvu8w/VX+ulZU7ou4ulYy1t4udRpzoKf
DkPrAIPqeQfzCtShurxHIyFeOnJH11VrgZx8oyJjKuQ4n89e+aa/xE/nrqLm8u7Gyo8hIJ+/Vuej
DVDuD4UkdFIlXj0espuJyfOU66SZzS220JkI+qUlCkbR22g87skmRXBiON7Zpk/6k9P6TAtiDb9E
uaPVJtOp0rbIrD6Go8FDjyfFlBXKlPZcwkYUrBKu6U4o9cUYxrrepjU+BLENRZbXDcJadGc7kFXJ
yfQ86HGNTpcjUmnjvyHz4NuqbVo9Rn7UdbSy5PdMxpxStpW620tKUHPp5ycatDfF0zKJa0mt01ak
1BTK2ihCsklSAMAj0JTn3zprdSEEksy+QK2PNrD3wt30mpV1gVO6YngKYDp8VhtKgVp2/aJUkE44
AwO51W2x6PFZrUihvpeeadUrwfEQnLSucgEdwT20vLhLk0bTk38kNvKFKp1edYkPFY3EDce2NRVt
tNMpEipzlhLoRvaRn+WlYBGfcgBWPQbfrrl2B5mr4GQrDEiOiTne4guPfVJUSofiO+uCoy3GZIRv
2qxnIPB9x7atVhMMqkcoqslIwHlfnr37Wlf4x/8ALU7RrcgyGgVN5hTLLJLjgKEADJKjwP1I0Z/i
n6NXZ03+JZpNlOSZTS2W3IvgOgPQyUJ3tLTwcBe4hXIIPtjRVjeSDa8V+zClkjimlLzf6IdDlX09
QJc6/L4ekTiwsRqWlSdysAAlawN5PPABA4ySeAIKaROSP5XbUOChwWxzeS2u3gdWKhPhw6c+zTQ1
UqSr9xNZeLanEbioJdTjzFOSArIO3CTnAOirb3VXrNdVwinUeFHuF5Y8jMdHgKSjjKju4CQceYnA
JHPODOxZO3c6VxfqXHuEl6jiu9CXIdzUKmU2rRnlrlw23TJ2uryMh1vHmKcZGTjkHQVh0SDYd8pq
Lb0puG0o70vOqUlXsAc+vOg5FKD2MdjlU8V+AT9Ubwj12svotuK9LKVkPSQnDaD9M9s+2h4/OqCf
7PUI7hClh1W9X8xfcZ+//wBfTRFFJAZS3MTSKfVLhrayFjxVKKlEn1PprRWTMpFZTHdKEOpGSnhY
SofQY41TfB5Omu4fHCUY7mNC3FrdK1HJUck57nWNx/4dGpBrLWMvusPpfjLKHWlBxtQ/uqByk/mB
q0d62rbPWJyDcNCpjkKPHo7NxXFWKpUn/CZU6kDahQJGP4j4aQSSSMd8G0vKlEwNcnUZIrRca6C5
LcnxqlGZIirQEiIWluhDnlSSkeidp9QM/jpntOs0Wq1WRDr8Srs/KwRKK4iGyVqIHlwvGMZGdVyQ
jfJbTZJRg6Cf0im9IFw5dXmtTqhV/kXERoVUiMqjJcIOVJTzladp2k8Zzxoi0HqlR6XZbqJrb1Nj
RG/BqD6XPAlvPOrWtHnUeEIjhJ4AAKyQNwBDeLpQSpgsryZW7NlRspN22nQibtm2/RH0uOswKbTT
8vIQdp3ObCVoWnZgEkk7lE4OhhffTJi8epTVFrVMr0ZmXn5GKh1MZ59GcF7YokIZ4PnUr37c67Jg
jklukdjzuEdsUSu6+hdDsu1WoEGCuHHZShuXFQ8x4DOTgBMlxIyk9yrjcM4B7aqR1Ttq2o/UNlm2
/DcjraKZDrcgPpLoWoEBY78AYVgZ54512qhDHHgvpZynLkgNHVVrfnLdiKCskjB1xVqLUqvWjOeU
As+mshPGsnU89jf6GVwUfHc5hSpQTg69+y5OidWJfoTLNIeT3B0dumr06+fh3hUQ1yLEj2fJf3Jf
mGOU/MKCmnBztUEgOpAVnB7d9G0172vdGHq19K/Zgqu6JWaxVVQ3nGG2XpMkr8NOHHGxtSFLJA2p
2jPIySRwcDUMtsVKH1Tp6X4D0fctIZK3QW392QNyiOxTgYA7jv6aLJCUGqpHW9Bq0K8plSt5qOmO
Hlq+UD5Dkd0E7khCgCE9uOcH8Ro0fDl0prd+1Nd4y2USaG7UB458MuJO1RSVHnJyU7OOTnHbXQx7
5JB3k2xb8l167b9ldP8ApPX6rdcWe61RECZKbIQNyUqGEtAfwjcvaFHsCrHJGhf8Psmrf1OqHVur
```

```
0tTtZ6g1t5zelpKA3BQ78vGhtJVnK3FJW5jttCST5QNa8mt6Rlpelsp71QvO/epM821S7cp9HgUy
W63safeebaCThS1K5SpQxg4Vj6AaCFfmQX+oMxMFsIDOxvyhKPNsG5W33Vn7hrJ1FzW5mvp3GLpE
akqLCG1vthCnE7inPY/85/HXP8wyfprHkvUz1+CSeKInxWc9hrHis/Qa6mF4DWl3AznT1at63FZl
xrqlu1ByK680Y7wT/C82e6VfkCCOQeQQdORk4SUkeXlFTi4vySKqXzQarvZYYmU+KtpC1JUr5l5T
wHnysjBST24yPXOo7VK5CrkVinwaKWZAWMtRyXFEEjdjA3EnHoBg9saa6kG+TPjpZx7M5KzY1eqL
wQKHUocMAFDtTlCItR+qlulPtzk9s6vR8Kdy2dQKPZFmR75tx6dKgvMzKM3Vozrj8xRU6QgIUQpQ
AV6nv7aewRp21X5F8qpVd/gcPjLvSfF+C9EeRRURiivQkPLbdJddQ1hwkbQCU5VgbvL5PfTZ0ol1
mvdMIlwUWJ4VOpcBMG2oyXyt11S0JDkhtJA2oBX4aSrKgUE8Y0f+bQq1aAtfVpyullIdjS5zNBEw
sohMvIaVJkpUvLg5ysqwCSpRBwRqktYuCPOvWqVSS6hRdmuvAhkpzucJTjj7vppHVJKoo0dLfLY2
SIKZbi0tvnxXFbkIWrHmPb9Pw0wFxe7HPvrMkuT0Wmm9lCfGOfXWPGPvqKGdwW0dRLYKR/atbEX9
bCu0z9dF6cjC6sDZ/Xe3P+8HPvo19KpVR6jXW0avck2m2dDiu1WRAt1KaXGUwyneUOLaAcVvwGzl
ZVlWdwxpnDJ4dzrwVcI6lpXwivVzW/CrXUx6VEZbLKSneHHS4WylSgpIUrKjjyDJOpT0trL9N66U
apw5bMZulVOMW20gkKJcA4PcAgkE6hTk6b+CHGClJR7c1/pfb4n4N3Xl8O0eM3V5tTk0t9ubIiRG
W0pXuVnaVjK1JSny4PB05dBbBTJ+AW0qHHqc6nz4CpUmZFjOhpcoCU8pLaFZACAVpUQOCeD21r7f
qX8GGp+n+yv3Xus9KEXPcj1Nrb06qQKU7FZbem722nnCG0HAyDhSsFRUcDvzqndSjVNNcZp7EdT6
UqSHQF707ePMR9fu+mszVbVJJGppVJJ2cj8j5W2jNLLjbqTt2qOTj3GOB6jUWU616Z1ntWbunpRY
krbJ7axuR9NRTGbQ5fsBpaiGhxrc3bQCcqONPnlicdCrDtC6/jZsi1b5bfk0Gr1tiDOZZdLanEuE
pSncOQCsozjBxnka+gN59Em+g3wkXq7CqMSS1PaXEo0WNDTHTTYbjiVrZ4OFkrSPNwSBg5JJ1Wak
06G8E4wuz5v1CU5CqVVpc19xiSzPcdCVApUUOBKgcHB/u/rrRNqcSnyGpUOQpttCEb8EAlSVBWPf
kd/fUz3cNF8Kjbs+g8++o9ftTx357bba2mikJkqBW4WgUAp3JBynk5BOAe550cbFn03p/wDDVQ7X
lWbLmViIxIfZdejEDwn5LziEjKgc7F7iSNoHvrb+52jz8oqNplRb5s2l3b1HWzbvS2VMkzphZY2s
rfQgNk4xjCfDJcOd/A2jJPoHetJoloUiPbU6u0cVJrztsURlsRorQyNqnGwErWVAgjkjByeNZ2eD
Vuq/ZoYJ7qTAs9KkzIXy638o8MIWoAguAds503KpwA4VrJlSdG1Ccox4EGCc8LGsfIq+2NRaLdaR
IIilCZtzxpyYSDPSk8j6aeMOJLLCabifEJaktgbXWa7BcQoHBBEhvGvrF1tShfSSoSHG0rXGpM2r
MlQz4b0ZpxaNv2QSMKxgkdiO+mNP9zfwUyv0pfJRDp90msTr/D+f6g0crqrlgPXIanBeXGkfMeGH
EpISdhbBUQAUE47knJ0DLj6NWtS5nTluPOqihc89LM3xHkHaNyf4PJx+OdE2KSt+5dTcHSLR2NbV
JuTpbSYtVY3pp0hxmKoAZb2N7ge2FHPfcCPbRU+IS5a9W/h9vkv1aQwq1JkCRAVHWUEpcjpdU0v7
Te4/w/QDnTbdQdf9wJNKU1ZQ2D1g6g1uvv0yqXDKfZ8VTh3PL8ylEqJPODk/UaiHUyCzLlww6VDc
Vbik4Jzzz9dY823yzSxRSltRHtiUU1JSOd+3P1wka1qPkGkZdzSXYRx9Br2B9BqCD//Z</msg:Bytes>
                </msg:Part>
              </msg:Parts>
            </msg:Content>
          </sMsg:Payload>
        </sMsg:Envelope>
      </Pack>
    </Send>
  </soap:Body>
</soap:Envelope>
```

9.1.2 Send Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

```
    <SendResponse xmlns="http://cz.o2.com/sMessage/sMessageService"/>
  </soap:Body>
</soap:Envelope>
```

9.2 Examples of Operation Receive

9.2.1 Receive Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Receive xmlns="http://cz.o2.com/sMessage/sMessageService"/>
  </soap:Body>
</soap:Envelope>
```

9.2.2 Receive Response – Acceptance of Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ReceiveResponse xmlns="http://cz.o2.com/sMessage/sMessageService">
      <Pack xmlns:sMsg="http://cz.o2.com/sMessage/Envelope">
        <sMsg:Header>
          <PackID>921xbc3hsgg383</PackID>
        </sMsg:Header>
        <sMsg:Envelope>
          <sMsg:Header>
            <sMsg:MessageID>123aa99eee551144</sMsg:MessageID>
            <sMsg:RelatesTo>ff1234567</sMsg:RelatesTo>
          </sMsg:Header>
          <sMsg:Payload xsi:type="acceptance:Accepted"
            xmlns:acceptance="http://cz.o2.com/sMessage/Acceptance"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <acceptance:Description>Message was accepted.</acceptance:Description>
            <acceptance:ExtraInfo>
              <acceptance:Entry>
                <acceptance:Key>Status</acceptance:Key>
                <acceptance:Value>OK</acceptance:Value>
              </acceptance:Entry>
            </acceptance:ExtraInfo>
          </sMsg:Payload>
        </sMsg:Envelope>
      </Pack>
    </ReceiveResponse>
  </soap:Body>
```

```
</soap:Envelope>
```

9.2.3 Receive Response – Rejection of Message

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ReceiveResponse xmlns="http://cz.o2.com/sMessage/sMessageService">
      <Pack xmlns:sMsg="http://cz.o2.com/sMessage/Envelope">
        <sMsg:Header>
          <PackID>921xbc3hsgg383</PackID>
        </sMsg:Header>
        <sMsg:Envelope>
          <sMsg:Header>
            <sMsg:MessageID>456</sMsg:MessageID>
            <sMsg:RelatesTo>dd455732g33</sMsg:RelatesTo>
          </sMsg:Header>
          <sMsg:Payload xsi:type="acceptance:Rejected"
            xmlns:acceptance="http://cz.o2.com/sMessage/Acceptance"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          >
            <acceptance:Description>Message was not accepted due to insufficient
credit.</acceptance:Description>
            <acceptance:ExtraInfo>
              <acceptance:Entry>
                <acceptance:Key>ERRORCODE</acceptance:Key>
                <acceptance:Value>daily_limit_exceeded</acceptance:Value>
              </acceptance:Entry>
            </acceptance:ExtraInfo>
          </sMsg:Payload>
        </sMsg:Envelope>
      </Pack>
    </ReceiveResponse>
  </soap:Body>
</soap:Envelope>
```

9.2.4 Receive Response – Delivery Report

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ReceiveResponse xmlns="http://cz.o2.com/sMessage/sMessageService">
      <Pack xmlns:sMsg="http://cz.o2.com/sMessage/Envelope">
        <sMsg:Header>
          <PackID>921xbc3hsgg383</PackID>
```

```
        </sMsg:Header>
        <sMsg:Envelope>
          <sMsg:Header>
            <sMsg:MessageID>S382472384232394235</sMsg:MessageID>
            <sMsg:RelatesTo>AOSMS000000000101</sMsg:RelatesTo>
          </sMsg:Header>
          <sMsg:Payload xmlns:msg="http://cz.o2.com/Messaging"
            xsi:type="msg:SmsDeliveryReport"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          >
            <msg:Status>Delivered</msg:Status>
            <msg:Timestamp>2006-08-10T14:36:36</msg:Timestamp>
            <msg:Originator xsi:type="msg:SmsApplicationNumberAddress">
              <msg:ApplicationNumber>999456777</msg:ApplicationNumber>
            </msg:Originator>
            <msg:Recipient xsi:type="msg:SmsPhoneNumberAddress">
              <msg:PhoneNumber>+420602608070</msg:PhoneNumber>
            </msg:Recipient>
          </sMsg:Payload>
        </sMsg:Envelope>
      </Pack>
    </ReceiveResponse>
  </soap:Body>
</soap:Envelope>
```

9.3 Examples of Operation Confirm

9.3.1 Confirm Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Confirm xmlns="http://cz.o2.com/sMessage/sMessageService">
      <PackID>921xbc3hsgg383</PackID>
    </Confirm>
  </soap:Body>
</soap:Envelope>
```

9.3.2 Confirm Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ConfirmResponse xmlns="http://cz.o2.com/sMessage/sMessageService"/>
  </soap:Body>
```

```
</soap:Envelope>
```

9.4 Examples of Operation Reject

9.4.1 Reject Request

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Reject xmlns="http://cz.o2.com/sMessage/sMessageService">
      <PackID>921xbc3hsgg383</PackID>
    </Reject>
  </soap:Body>
</soap:Envelope>
```

9.4.2 Reject Response

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RejectResponse xmlns="http://cz.o2.com/sMessage/sMessageService"/>
  </soap:Body>
</soap:Envelope>
```

9.5 Examples of Faults

9.5.1 Example of SOAP mustUnderstand Fault

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:MustUnderstand</faultcode>
      <faultstring>SOAP must understand header was not understood:
{http://test}testHeader</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

9.5.2 Example of SOAP Fault

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client.ConfirmFailed</faultcode>
      <faultstring>Pack confirmation failed.</faultstring>
    </soap:Fault>
  </soap:Body>
```

```
</soap:Envelope>
```

9.5.3 Example of Throughput limit exceeded SOAP Fault

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client.Limit.Throughput</faultcode>
      <faultstring>Throughput limit exceeded.</faultstring>
      <detail>
        <sMsg:RetryAfter
xmlns:sMsg="http://cz.o2.com/sMessage/sMessageService">8156</sMsg:RetryAfter>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

10 Appendix C: Deployment Specific Parameters

This chapter describes parameters of the `M2M WS Server` in the MMS Connector Web Services Interface and their configured values. Please note that in case of need O2 can change them without notice.

Name	Description	Configured Value
URL to Web Service endpoint and/or WSDL		specified separately
confirmation timeout	Maximum time between retrieval of the request and its confirmation. In case the client fails to confirm the retrieved request within this time, the request is considered rejected and will be redelivered to the client	3 seconds
waiting time on empty queue	how long SHOULD client wait before calling Receive() operation again when no messages were available	specified separately
maximum throughput	maximum number of messages sent per second (note it is number of messages, not calls to the operation) using Send()	Configured for particular customer application based on contract
maximum number of unconfirmed packs	maximum number of packs waiting for confirmation (or rejection or timeout see confirmation timeout)	5
maximum number of concurrent connections	maximum number of concurrent connections to the server	10
Acceptance / Rejection usage	specifies how the Accept/Reject requests are	In case of Rejected result, error codes are

	used in the deployment and what are contents of extra-info fields in the requests	filled only in following cases: -when partners daily limit is exceeded, error code is "daily_limit_exceeded" -when partners monthly limit is exceeded, error code is "monthly_limit_exceeded" Error code is placed as Rejected ExtraInfo Value where key is “errorCode”. In other cases, only Description element is filled.
uniqueness period	period for which the platform keeps data for uniqueness checks	7 days