



# **Evaluation of frameworks for creating end-to-end mobile services with OMA MMS as a use case**

by

*Andreas Häber and Lene Beate Longvastøl*

**Master's Thesis in  
Information and Communication Technology**

**Agder University College  
Faculty of Engineering and Science**

**Grimstad**

**July 2005**

## **i. Abstract**

Several frameworks are available in 3GPP networks to create mobile services, such as the Open Service Access (OSA) Application Programming Interface (API) and the native Session Initiation Protocol (SIP). Each of these frameworks has their own advantages and disadvantages. Therefore it is important for a service to use a framework which suits its own requirements as best as possible.

In this thesis we have defined a use case, TMMS Service. This use case has been designed for four frameworks selected, which are: GPRS, IMS, OSA API and OSA Parlay X Web Services. We have then evaluated the design of these services against a set of evaluation criteria. The evaluation criteria cover security, usability, modifiability, reliability, interoperability and billability.

Our evaluation has proved that none of the frameworks are superior in all areas. The best framework overall is IMS which provides a lot of end-to-end features and is also very extensible. One of the biggest disadvantages with IMS is the current lack of a specific set of documentation for application developers.

## **ii. Preface**

This thesis is written for Teleca Wireless Solutions AS and is the final part of the Master of Science degree in Information and Communication Technology at Agder University College, Faculty of Engineering and Science. The work has been carried out in the period between January and July 2005.

First of all we would like to thank our supervisor at Teleca Wireless Solutions AS, Arild Løvendahl, for guidance throughout the project period. We would also like to thank Magne Arild Haglund for grateful feedback on our report.

Finally we would like to thank Head of Studies, Stein Bergsmark, for his contributions, and our co-students for helpful feedback on our thesis.

Grimstad, July 2005.

Andreas Häber and Lene Beate Longvastøl

### iii. Table of Contents

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	BACKGROUND.....	1
1.2	THESIS DEFINITION.....	1
1.3	REPORT OUTLINE.....	1
<b>2</b>	<b>BACKGROUND INFORMATION.....</b>	<b>3</b>
2.1	OMA MULTIMEDIA MESSAGING SERVICE.....	3
2.2	JAVA 2 PLATFORM, MICRO EDITION.....	4
2.3	UNIFIED MODELING LANGUAGE.....	6
2.4	CASE DESCRIPTION: TMMS SERVICE.....	9
2.5	EVALUATION CRITERIA.....	19
2.6	SESSION INITIATION PROTOCOL.....	21
2.7	STREAMING PROTOCOLS.....	25
<b>3</b>	<b>TMMS SERVICE BASED ON GPRS.....</b>	<b>28</b>
3.1	OVERVIEW.....	28
3.2	DESIGN.....	31
3.3	EVALUATION.....	40
<b>4</b>	<b>TMMS SERVICE BASED ON IMS.....</b>	<b>43</b>
4.1	OVERVIEW.....	43
4.2	DESIGN.....	49
4.3	EVALUATION.....	61
<b>5</b>	<b>TMMS SERVICE BASED ON OSA API.....</b>	<b>64</b>
5.1	OVERVIEW.....	64
5.2	DESIGN.....	66
5.3	EVALUATION.....	71
<b>6</b>	<b>TMMS SERVICE BASED ON OSA PARLAY X WEB SERVICES.....</b>	<b>74</b>
6.1	OVERVIEW.....	74
6.2	DESIGN.....	74
6.3	EVALUATION.....	81
<b>7</b>	<b>DISCUSSION.....</b>	<b>84</b>
7.1	INTRODUCTION.....	84
7.2	SECURITY.....	84
7.3	USABILITY.....	84
7.4	MODIFIABILITY.....	85
7.5	INTEROPERABILITY.....	86
7.6	RELIABILITY.....	87
7.7	BILLABILITY.....	87
<b>8</b>	<b>CONCLUSION.....</b>	<b>88</b>
8.1	GPRS AS A FRAMEWORK FOR CREATING END-TO-END MOBILE SERVICES.....	88
8.2	IMS AS A FRAMEWORK FOR CREATING END-TO-END MOBILE SERVICES.....	88
8.3	OSA API AS A FRAMEWORK FOR CREATING END-TO-END MOBILE SERVICES.....	88
8.4	OSA PARLAY X WEB SERVICES AS A FRAMEWORK FOR CREATING END-TO-END MOBILE SERVICES.....	89
8.5	OVERALL.....	89
8.6	FURTHER WORK.....	89
<b>9</b>	<b>REFERENCES.....</b>	<b>90</b>
<b>10</b>	<b>APPENDIX.....</b>	<b>93</b>



**iv. Table of Figures**

Figure 2-1: MMS Architecture (from MMS-ARCH spec) .....	3
Figure 2-2: J2ME architecture .....	5
Figure 2-3: Use case diagram example .....	6
Figure 2-4: Static structure diagram example .....	6
Figure 2-5: Collaboration diagram example .....	7
Figure 2-6: Sequence diagram example .....	7
Figure 2-7: State chart diagram example .....	7
Figure 2-8: Activity diagram example .....	8
Figure 2-9: Component diagram example .....	9
Figure 2-10: Deployment diagram example .....	9
Figure 2-11: Use case view for service operator .....	10
Figure 2-12: Use case view for user with TMMS Client .....	11
Figure 2-13: Use case view for user with GSM SMS Client .....	12
Figure 2-14: Use case view for TMMS Proxy-Relay .....	13
Figure 2-15 Logical architecture of TMMS Service .....	15
Figure 2-16 Component view of TMMS Service .....	16
Figure 2-17 Component view of TMMS Client .....	16
Figure 2-18 Component view of Communication package .....	17
Figure 2-19 Component view of Storage package .....	17
Figure 2-20 Component view of UserInterface package .....	18
Figure 2-21 Security evaluation scenarios .....	19
Figure 2-22 Usability evaluation scenarios .....	20
Figure 2-23 Modifiability evaluation scenarios .....	20
Figure 2-24 Interoperability evaluation scenario .....	20
Figure 2-25 Reliability evaluation scenarios .....	21
Figure 2-26 Billability evaluation scenarios .....	21
Figure 2-27: SIP transaction example .....	22
Figure 2-28: SIP call establishment [20] .....	24
Figure 2-29 Interaction between Client and Server .....	26
Figure 2-30: RTP can be viewed as a sublayer of the transport layer .....	27
Figure 3-1: Overview of the GPRS architecture .....	29
Figure 3-2 Deployment of GPRS based service .....	31
Figure 3-3 TMMS Client Registration .....	33
Figure 3-4 Send MM to TMMS Client .....	34
Figure 3-5 Send MM to GSM SMS Client .....	34
Figure 3-6 TMMS Client Receives MM (Immediate Retrieval) .....	35
Figure 3-7 TMMS Client Receives MM (Deferred Retrieval) .....	36
Figure 3-8 Deferred MM Retrieval .....	36
Figure 3-9 TMMS Client Unregistration .....	36
Figure 3-10 Charging state machine for client .....	38
Figure 3-11 Charging state machine for proxy-relay .....	38
Figure 4-1: Overview of the 3GPP IMS architecture .....	44
Figure 4-2: Traffic between two security domains .....	47
Figure 4-3 WAP Push Architecture .....	48
Figure 4-4: Deployment diagram for IMS solution .....	50
Figure 4-5: Create service profile of PUI .....	51
Figure 4-6 Component view for IMS based service .....	52
Figure 4-7: Component view for TMMS Client .....	52

Figure 4-8: TMMS Client registration .....	53
Figure 4-9: TMMS Client Sends MM.....	54
Figure 4-10: TMMS Client retrieves MM immediately.....	55
Figure 4-11: TMMS Client defers retrieval of MM.....	55
Figure 4-12: Deferred MM retrieval.....	56
Figure 4-13: Streaming media retrieval of MM.....	57
Figure 4-14: TMMS Client unregistration .....	58
Figure 4-15: IMS online charging, client accepts to be charged.....	59
Figure 4-16: Offline charging in IMS.....	59
Figure 4-17: IMS online charging, client declines to be charged .....	60
Figure 5-1: Deployment of OSA API based service in IMS network environment.....	66
Figure 5-2: Establish service agreement - Initial Access .....	67
Figure 5-3: Establish service agreement - Discover Service Agreement Management interface .....	67
Figure 5-4: Establish service agreement – Sign service agreement.....	68
Figure 5-5: Component view.....	68
Figure 5-6: TMMS Client components .....	69
Figure 5-7: TMMS Proxy-Relay components .....	69
Figure 5-8: Charging of service usage .....	70
Figure 6-1: Deployment view .....	74
Figure 6-2: Component View.....	75
Figure 6-3: Component view for TMMS Client.....	76
Figure 6-4: TMMS Proxy-Relay Web Service components .....	76
Figure 6-5: Client registration.....	77
Figure 6-6: Send MM.....	77
Figure 6-7: New message notification.....	78
Figure 6-8: Deferred message retrieval .....	78
Figure 6-9: Streaming retrieval of MM.....	79
Figure 6-10: Client deregistration .....	80

**v. Table of Tables**

Table 2-1 Description of service operator use cases .....	10
Table 2-2 Description of use cases for user with TMMS Client.....	11
Table 2-3 Description of use cases for user with GSM SMS client.....	12
Table 2-4 Description of use cases for TMMS Proxy-Relay .....	13
Table 2-5 Description of Communication package components .....	17
Table 2-6 Description of Storage package components .....	18
Table 2-7 Description of UserInterface package components.....	18
Table 3-1: Evaluation of Security quality attribute for TMMS Service based on GPRS...	40
Table 3-2: Evaluation of Usability quality attribute for TMMS Service based on GPRS	.41
Table 3-3: Evaluation of Modifiability quality attribute for TMMS Service based on GPRS .....	41
Table 3-4: Evaluation of Interoperability quality attribute for TMMS Service based on GPRS .....	42
Table 3-5: Evaluation of Reliability quality attribute for TMMS Service based on GPRS	42
Table 3-6: Evaluation of Billability quality attribute for TMMS Service based on GPRS	42
Table 4-1: Evaluation of Security quality attribute for TMMS Service based on GPRS...	61
Table 4-2: Evaluation of Usability quality attribute for TMMS Service based on GPRS	.62
Table 4-3: Evaluation of Modifiability quality attribute for TMMS Service based on GPRS .....	62
Table 4-4: Evaluation of Interoperability quality attribute for TMMS Service based on GPRS .....	63
Table 4-5: Evaluation of Reliability quality attribute for TMMS Service based on GPRS	63
Table 4-6: Evaluation of Billability quality attribute for TMMS Service based on GPRS	63
Table 5-1: Evaluation of Security quality attribute for TMMS Service based on GPRS...	71
Table 5-2: Evaluation of Usability quality attribute for TMMS Service based on GPRS	.72
Table 5-3: Evaluation of Modifiability quality attribute for TMMS Service based on GPRS .....	72
Table 5-4: Evaluation of Interoperability quality attribute for TMMS Service based on GPRS .....	73
Table 5-5: Evaluation of Reliability quality attribute for TMMS Service based on GPRS	73
Table 5-6: Evaluation of Billability quality attribute for TMMS Service based on GPRS	73
Table 6-1: Evaluation of Security quality attribute for TMMS Service based on GPRS...	81
Table 6-2: Evaluation of Usability quality attribute for TMMS Service based on GPRS	.82
Table 6-3: Evaluation of Modifiability quality attribute for TMMS Service based on GPRS .....	82
Table 6-4: Evaluation of Interoperability quality attribute for TMMS Service based on GPRS .....	83
Table 6-5: Evaluation of Reliability quality attribute for TMMS Service based on GPRS	83
Table 6-6: Evaluation of Billability quality attribute for TMMS Service based on GPRS	83



## vi. Abbreviations

2.5G	Second and a half Generation
2G	Second Generation
3G	Third Generation
3GPP	Third Generation Partnership Project
3GPP2	The Third Generation Partnership Project 2
ACA	Accounting-Answer
ACR	Accounting-Request
API	Application Programming Interface, Application Programming Interface
AS	Application Server
ASCII	American Standard for Information Interchange
AVP	Attribute Value Pair
B2BUA	Back-to-back User Agent
BGCF	The Breakout Gateway Control Function
CAMEL	Customized Applications for Mobile networks Enhanced Logic
CAN	Carrier Access Network
CC	Charging Controller
CDF	Charging Data Function
CDR	Charging Data Record
CGF	Charging Gateway Functionality
CKSN	CKSNumber
CLDC	Connected Limited Device Configuration
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off The Shelf
CS	Circuit Switched
CSCF	Call/Session Control Function
DNS	Domain Name System, Domain Name System
EDGE	Enhanced Data rates for Global Evolution
ETSI	European Telecommunication Standards Institution
GERAN	GSM/EDGE RAN
GERAN BS	GERAN Base Station
GGSN	Gateway GPRS Support Node
GR	GPRS register
GSM	Global System for Mobile Communications
GSN	GPRS Support Node
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
I-CSCF	Interrogating-CSCF
IEC	Immediate Event Charging
IETF	Internet Engineering Task Force
iFC	initial Filter Criteria
IMS	IP Multimedia Subsystems
IMSI	International Mobile Subscriber Identifier
IP	Internet Protocol
IP-M	Internet Protocol Multicast
IP-MESSAGE-GW	IP Short Message Gateway
IPSec	IP Security
ISIM	IP-Multimedia Services Identity Module

J2ME	Java 2 Platform Micro Edition
J2SE	Java 2 Platform Standard Edition
JAD	Java Application Descriptor
JAR	Java Archive
JVM	Java Virtual Machine
MIDP	MID Profile, Mobile Information Device
MIME	Multipurpose Internet Mail Extensions
MM	Multimedia Message, Multimedia Message
MMS	Multimedia Messaging Service
MRFC	MRF Controller
MRFP	MRF Processor
MS	Message Store
MSISDN	Mobile Station Integrated Services Digital Network
NAI	Network Access Identifier
OEM	Original Equipment Manufacturer
OMA	Open Mobile Alliance
OSA	Open Service Access, Open Service Access
PAM	Presence and Availability Management
PAP	Push Access Protocol
PCM	Pulse Code Modulation
PDA	Personal Digital Assistant
PIP	Push Initiator
PLMN	Public Land Mobile Network
POSIX	Portable Operating system Interface
PO-TCP	PPG Originated TCP
PPG	Push Proxy Gateway
PR	Proxy-Relay
PS	Packet Switched
PSI	Public Service Identities
PSTN/CS	Public Switched Telephone Network/CS
PTM	Point-to-Multipoint
PTP	Point-to-Point
PTT	Push-To-Talk
QoS	Quality of Service
RAN	Radio Access Network, Radio Access Network
RR	RRresource ecord
RTP	Real-Time Protocol, Real-time Transport Protocol
RTSP	Real-Time Streaming protocol, Real-time Streaming Protocol
SCFs	Service Capability Features
SCS	Service Capability Server
S-CSCF	Serving-CSCF
SCUR	Session Charging with Unit Reservation
SDP	Session Description Protocol
SEG	Security Gateways
SGSN	Serving GPRS Support Node
SIA	Session Initiation Application
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol, Session Initiation Protocol
SIR	Session Initiation Request
SMS	Short Message Service
SNMP	Simple Network Management Protocol

## Abbreviations

---

SOAP	Simple Object Access Protocol
SPT	Service Point Trigger
SSL	Secure Sockets Layer
TDMA	Time Division Multiple Access
TLLI	TLLIdentity
TLS	Transport Layer Security
TMMS	Test MMS
TMMS PR	TMMS Proxy-Relay
TO-TCP	Terminal Originated TCP
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDDI	Universal Description, Discovery and Integration
UDP	User Datagram Protocol
UE	User Equipment
UICC	Universal Integrated Circuit Card
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USIM	UMTS Subscriber Identity Module
UTRAN	Universal Terrestrial Radio Access Network
VPN	Virtual Private Network
WAP	Wireless Application Protocol
WBF	WAP Billing Framework
WSDL	Web Services Description Language
WSP	Wireless Session Protocol
XML	eXtensible Markup Language

## 1 Introduction

In this chapter we will give an overview of our thesis and how this report is structured. More precisely, subsection 1.1 will give an overview of the background of the thesis. In subsection 1.2 we present the final thesis description and thesis title, before we finish this chapter by giving a summary on how this report is structured.

### 1.1 Background

The early mobile telecommunication networks were primarily focused on voice services. Now the telecommunication networks have converged with the computer networks which brings data services. This convergence opens up possibilities for new richer services not possible before, such as Push-To-Talk (PTT). Alongside a way to create these services is needed, and the Third Generation Partnership Project (3GPP) addresses this problem. If not standardized each network provider will need to develop an ad-hoc method if they want to let third parties provide services in their network.

In fact there are several frameworks available in 3GPP networks to create mobile services, such as the Open Service Access (OSA) Application Programming Interface (API) and the native Session Initiation Protocol (SIP). Each of these frameworks have their own advantages and disadvantages. Therefore it is important for a service to use a framework which suits its own requirements as best as possible. Currently, as of the authors' knowledge, there are no comparisons done between these frameworks.

### 1.2 Thesis definition

The thesis title is:

*"Evaluation of frameworks for creating end-to-end mobile services with OMA MMS as a use case"*

The final definition of the thesis is:

*Evaluate some of the different frameworks available for creating end-to-end mobile services in the Third Generation Partnership Project (3GPP) specifications, and in particular frameworks based on the fairly new IP Multimedia Subsystem (IMS). For comparison the students we will also use a framework which is not based on IMS. As a use-case the students will use a service based on Open Mobile Alliance (OMA)'s Multimedia Messaging Service (MMS) specification. The use-case service must provide a way for the service provider to bill its customers and be able to integrate with Global System for Mobile Communications (GSM)'s Short Message Service (SMS) The students will evaluate the services on the following quality attributes: security, reliability, usability, interoperability, modifiability and billability.*

### 1.3 Report outline

This report is structured as followed:

Chapter one is the introduction chapter which is the current chapter.

Chapter 2 gives an outline of the technologies used when designing our OMA based MMS service and our end-to-end mobile services. Technologies presented in this chapter are common for all the different end-to-end services. This chapter also gives a case description of our OMA based MMS service, and we present the different criteria and scenarios which the evaluation of each of the end-to-end services are based on.

Chapters 3 to 6 present the different end-to-end mobile service that we have designed and want to evaluate. These chapters first give a brief introduction to background material specific for the individual services. After this introduction, our design solution for the service follows, before each chapter ends with an evaluation of this service.

Chapter 7 presents a discussion based on the evaluation of the services given in the previous four chapters. Here we focus on advantages and drawbacks, allowing us to draw some conclusions based on chapter 7 in the final chapter.

## 2 Background information

In this chapter we give an introduction to all background material common for chapters 3 to 6. It is necessary for the reader to understand these important topics before reading the next chapters. We also introduce our case study, the Test MMS (TMMS) service which is based on OMA's MMS, and present all the evaluation criteria and scenarios. The evaluation criteria and scenarios are used in the evaluation of our end-to-end mobile services.

### 2.1 OMA Multimedia Messaging Service

OMA was formed in 2002 by some of the leading companies within the mobile world. Among some of the main goals of the OMA is to deliver high qualified technical specifications and to be a catalyst for the consolidation of standards activity within the mobile data industry. As a part of this thesis our TMMS is based on OMA's MMS [1]. Here follows an outline on how OMA's MMS works.

OMA's MMS is a messaging system for various media types (text, image, audio and video). The MMS Client can send and retrieve messages through a MMS Proxy-Relay (PR). Also MMS may interoperate with other messaging systems, such as e-mail. The Figure 2-1 shows the MMS architecture.

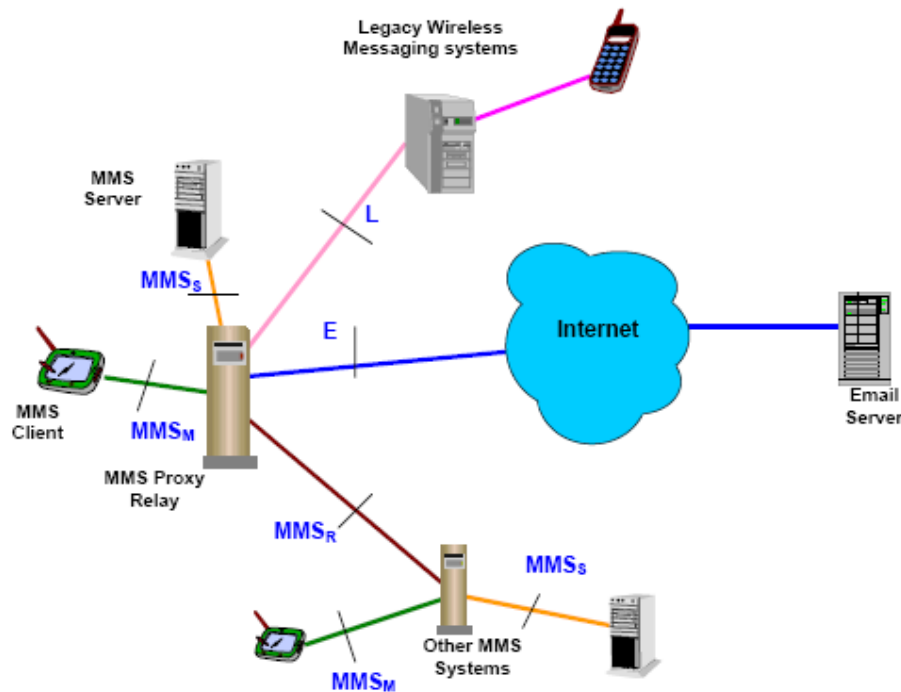


Figure 2-1: MMS Architecture (from MMS-ARCH spec)

The payload carried through the MMS system is described in the MMS Message Encapsulation standard [2]. A MMS message consists of a header part and a body part. The header part describes the message with various fields, such as sender, destination and flags. A MMS message does not need to have a body part, but if it is present it consists of one or more media objects and zero or more presentation descriptions.

The end-user interacts with the MMS Client, and it is responsible for rendering and composing of messages. Also it must be able to send and retrieve messages to/from the MMS PR.

The MMS PR is responsible for routing the messages to and from the MMS Clients. It may also provide a storage service for MMS messages, which logical is named the MMS Server. Instead of doing simple message delivery the MMS PR may decide to convert the message body into a streaming delivery, depending on various properties like message content and user profile.

### ***2.2 Java 2 Platform, Micro Edition***

#### **2.2.1 Connected Limited Device Configuration**

Connected Limited Device Configuration (CLDC) [3] is defined as a Java 2 Platform, Micro Edition (J2ME) [4] configuration and is intended to run on a wide variety of small devices. It targets application programming rather than systems programming.

The main component in a CLDC implementation is the Java Virtual Machine (JVM) which typically runs on top of a Host Operation System. On top of the JVM we find the Java libraries. A general goal for designing these libraries for the CLDC is to provide a minimum set of libraries for practical application development and a profile definition for a variety of small devices.

An essential requirement for the CLDC is the ability to support dynamic downloading of Java applications and third-party content. It must also be able to support standard Java files in addition to be able to support compressed Java Archive (JAR) files.

The security model of CLDC is defined at three different levels, which are:

- Low-level security
- Application security – a Java application can access only those libraries, system resources and components that a device and the Java application environment allows it to access
- End-to-end security – guarantees that a transaction from one device to another is protected

##### **2.2.1.1 Generic Connection Framework**

One detail which makes designing the CLDC a problem is the various needs required by different devices. This led to the Generic Connection framework, which provides a coherent way to access various types of network in resource-constrained environment. The goal is to be a functional subset of Java 2 Platform Standard Edition (J2SE) [5] classes, which easily can be mapped to common low-level hardware or to any J2SE implementation with better extensibility, flexibility and coherence in supporting new devices and protocols.

##### **2.2.2 Mobile Information Device Profile**

The Mobile Information Device (MID) Profile (MIDP) [6] is a key element of the J2ME. Combining MIDP with CLDC provides a standard Java runtime environment for mobile information devices such as cell phones and Personal Digital Assistants (PDAs). MIDP 2.0 is based on MIDP 1.0 which provides a standard API for application development. MIDP 2.0 includes many enhancements and additions. Some of these are secure networking, multimedia and gaming.

### 2.2.2.1 Architecture

The goal of MIDP is to create an open, third-party application environment for MIDs. Figure 2-2 as shown in[6] below describes a typical architecture, but it is only a suggestion on how the architecture may look like. A device that implements MIDP does not necessarily need all elements shown in the figure. Neither does the device need to layer the elements and software exactly as shown in the figure.

The lowest part in the figure is the MID hardware. On top of that is the Native System Software which includes the operating system and libraries used by the device. Above the Native System Software is the CLDC, which represents the Virtual Machine and associated libraries. CLDC provides the underlying Java functionality upon which higher-level Java APIs may be built. On top of the CLDC we find two different APIs. These are the MIDP API and an Original Equipment Manufacturer (OEM)-specific API. The three last blocks on the figure represent the application types possible on a MID, which are MIDP applications, OEM-specific applications, and native applications.

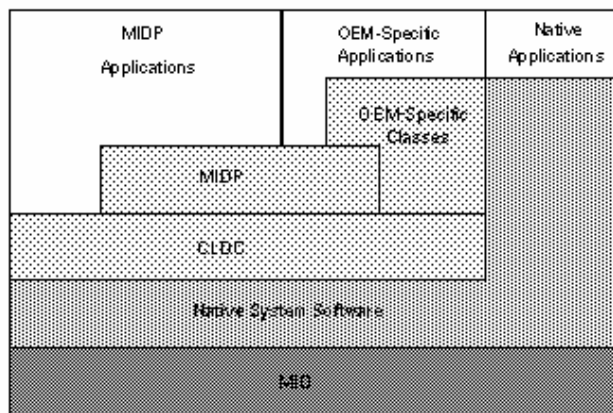


Figure 2-2: J2ME architecture

### 2.2.2.2 Security

MIDP 1.0 can only offer non-secure Hypertext Transfer Protocol (HTTP) [31] and has weak support for confidentiality, integrity and authentication. In contrast MIDP 2.0 supports the Secure Hypertext Transfer Protocol (HTTP over TLS, HTTP/TLS or HTTPS). Since HTTPS runs over the secure Transport Layer security (layer TLS), which again runs over TCP, support for encryption, integrity and authentication is taken care of. When using TLS a handshake is done between the client and the server. They first decide which algorithm to use, secondly they exchange certificate used to authenticate. This certificate is also used by the client to create and encrypt a session key. This session key is used to encrypt the rest of the session.

### 2.2.3 SIP API for J2ME

The SIP API [8] for J2ME is designed as an Optional Package that can be used with different J2ME profiles. The API is designed to be a compact and generic SIP API which provides SIP functionality in transaction level. The API is integrated into the Generic Connection Framework which we described in 2.2.1.1. The SIP API is used by applications to implement SIP User Agents (UA), such as User Agent Clients (UAC) and User Agent Server (UAS). SIP UA is explained in chapter 2.6.3.



### 2.3 Unified Modeling Language

Unified Modeling Language (UML) [9] is a graphical notation for modeling. It is based on a four-layer metamodel structure, with the following layers: user object(s), model, metamodel and meta-metamodel. Each sub-layer is an instance of its parent metalayer. This means that an instance of a user object is an instance of its class type which is an instance of the UML metamodel Class which again is an instance of the meta-metamodel MetaClass.

Several model and diagram types are defined, which are discussed in the sections below. The example figures are very basic, because an advanced description of these diagrams is out of scope for this thesis.[10] Have a small introduction to UML.

#### 2.3.1 Use Case Diagrams

A use case diagram depicts the important scenarios for a system. It shows actors and use cases. An actor is the user in a use case. To be more specific an actor is a particular role of a user, so there is no one-to-one relationship between users and actors. A system may have several actors but only one user. Actors do not have to be a human being either, but anything which interacts with the system. For example can an actor be an external system which interacts with the system.

A use case is a task which an actor wants the system to do. It should be a end-to-end task, with defined states of beginning and ending. Figure 2-3 below shows a use case of a customer who wants to withdraw money in a banking system.



Figure 2-3: Use case diagram example

#### 2.3.2 Static Structure Diagrams

Static structure diagrams can be used to describe a structural model of a system. For example a class model, relationship between processes and threads, etc. An example of the static structure showing the relationship between a Customer and an Account class is given in Figure 2-4 below.

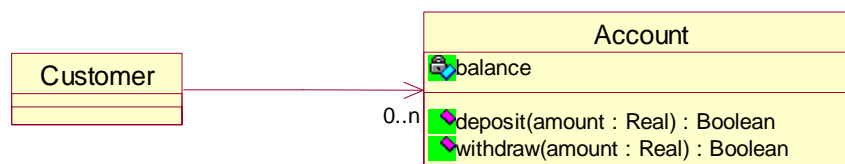


Figure 2-4: Static structure diagram example

There are several types of relationships available: Generalization, realization, dependency and assignment.

#### 2.3.3 Interaction Diagrams

There are two forms of interaction diagrams which show communication of model elements. These two types are sequence diagrams and collaboration diagrams. They are

based on the same information, except that sequence diagrams are more useful to describe complex scenarios and when timing (such as in real-time systems) is important. Figure 2-5 shows a collaboration diagram of a customer depositing money to its account.

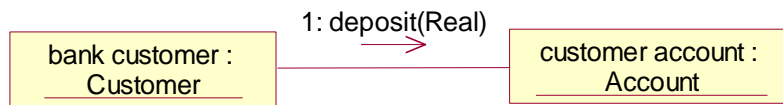


Figure 2-5: Collaboration diagram example

In the figure below the same information is displayed in a sequence diagram.

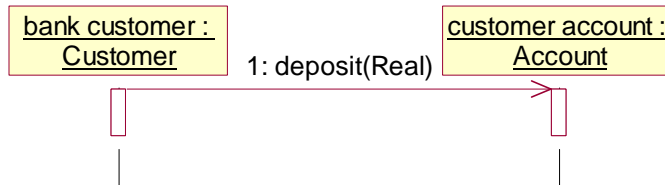


Figure 2-6: Sequence diagram example

### 2.3.4 State chart Diagrams

A state chart is a model of finite state machines, and is useful for showing the dynamic aspects of a system. The state chart diagram contains several items, which are shown in Figure 2-7. As can be seen from the figure it has a start state (the filled black circle), one or more states (the yellow rounded rectangles), an end state (the filled black circle with a surrounding circle) and transitions between them (or to itself).

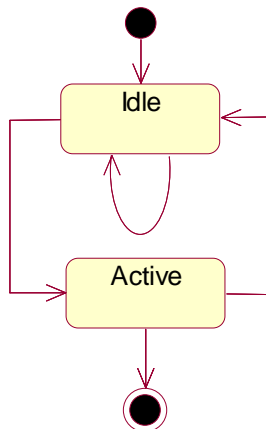


Figure 2-7: State chart diagram example

States may be nested as well and specifying actions to be performed during entry, exit and when events are signaled. The transitions may have guards on them and signal events.

### 2.3.5 Activity Diagrams

Activity diagrams are a special kind of state chart diagrams, and add some additional elements to it. In specific the additional elements are action states, swimlanes, decision points and synchronization points. The activity diagram is useful to focus on internal activities of an algorithm for example, where most of the events signal the completion of internal actions.

Action state is a special state, which has an entry action and at least one outgoing transition involving the implicit event of completing the entry action.

Swimlanes helps to organize the responsibilities of activities and sub-activities.

A decision point can be seen as an “if-then-else” statement, and indicates different possible transitions from a state.

Figure 2-8 shows an activity diagram for the deposit method of the Account class from Figure 2-4 above.

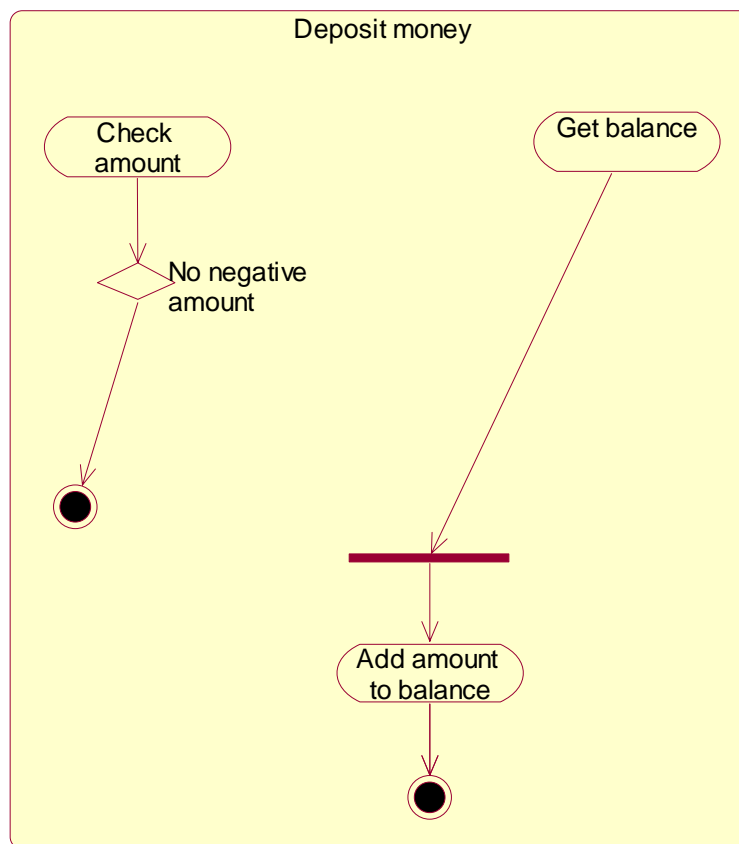


Figure 2-8: Activity diagram example

### 2.3.6 Implementation Diagrams

Implementation diagrams come in two forms, component diagrams and deployment diagrams. Component diagrams show the dependency between components. A component may be modules in a system. See Figure 2-9 below for a component diagram.

Deployment diagrams show the structure of the nodes on which the components are to be deployed. See Figure 2-10 below for an example of a deployment diagram for the same Air Traffic Control system.

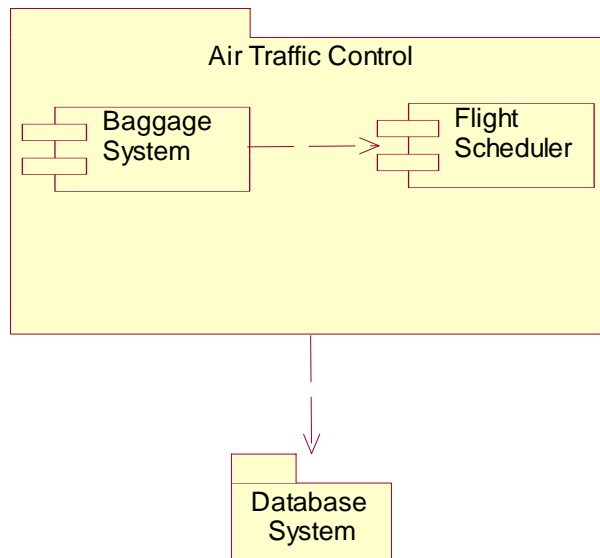


Figure 2-9: Component diagram example

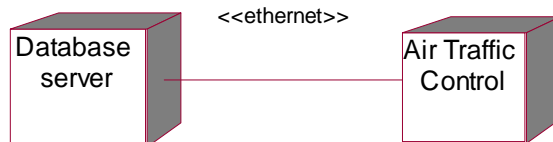


Figure 2-10: Deployment diagram example

Note that the deployment diagrams used in the report have different graphics, but the semantic of them are equal.

## 2.4 Case description: TMMS Service





### 2.4.1 Use case view

First all the actors of the system are presented, and then the use cases for each actor are described.

#### 2.4.1.1 Actors

The following table describes the actors which are part of the TMMS Service.

Table 2-1 RPR Actor descriptions

Actor	Description
 Proxy-Relay	The PR is responsible for relaying messages from sending client to destination client.
 Service operator	A client must first contact a service operator to get access to TMMS.
 User with GSM SMS Client	A user with a TMMS-enabled device.
 User with TMMS Client	A user with a GSM SMS device.

### 2.4.1.2 Service Operator

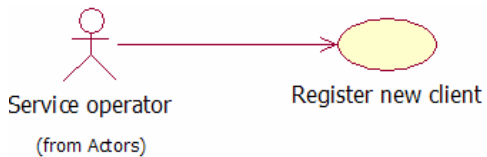


Figure 2-11: Use case view for service operator

The table below describes this use case.

Table 2-1 Description of service operator use cases

Id	Use case	Description
US01	Register new client	Before the client is authorized to use the service it must be registered by a service operator. This can be done either offline or online.  The details of this use case are out of scope for the rest of this report.

### 2.4.1.3 User with TMMS Client

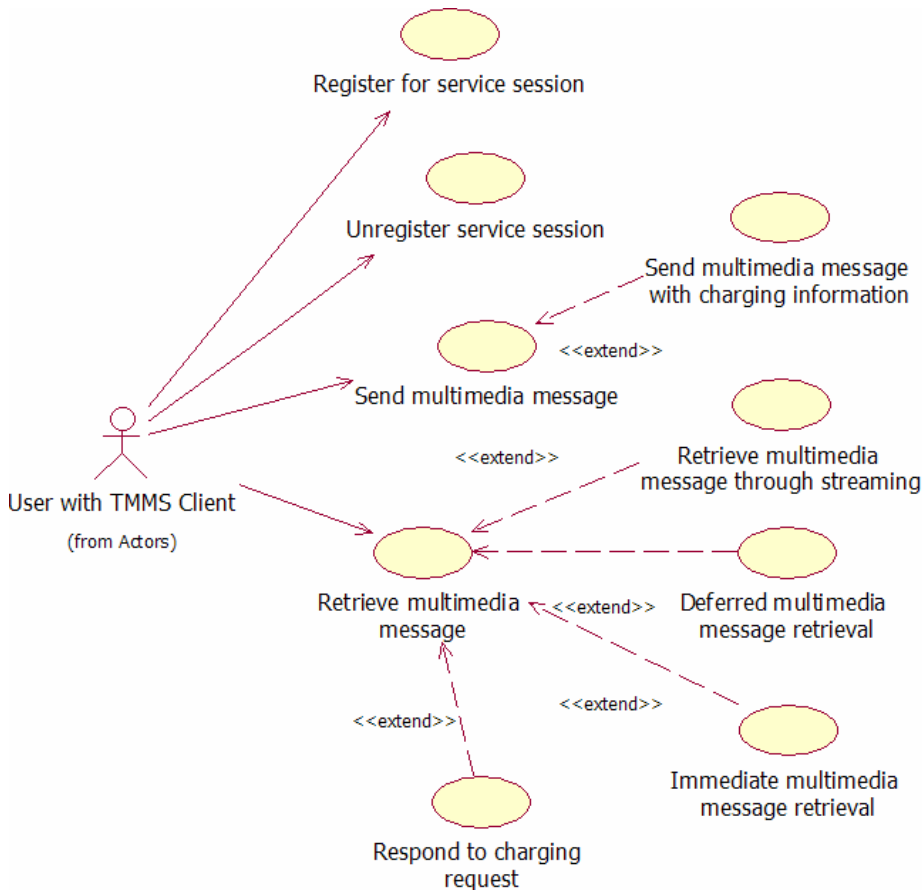


Figure 2-12: Use case view for user with TMMS Client

The table below describes the use cases.

Table 2-2 Description of use cases for user with TMMS Client

Id	Use case	Description
UT01	<b>Register for service session</b>	The client must register with the TMMS PR so it can know where to deliver notifications about new messages.
UT02	<b>Unregister service session</b>	When the client is no longer available on the network it should let the TMMS PR know so it can free resources used for the client.
UT03	<b>Send multimedia message</b>	The client sends a multimedia message to one or more recipients.
UT03.1	<b>Send multimedia message with charging information</b>	The client, as the sender, demands the recipient to be charged for the message.
UT04	<b>Retrieve multimedia message</b>	The TMMS PR sends a notification to the client's registered endpoint. Now the client can retrieve the message.

Id	Use case	Description
UT04.1	<b>Retrieve multimedia message through streaming</b>	Instead of downloading the complete message before the client can see it, the client can request the TMMS Message Store (MS) to stream it back instead.
UT04.2	<b>Deferred multimedia message retrieval</b>	The client responds to a message notification that it knows about the message, but will retrieve it at some later time.
UT04.3	<b>Immediate multimedia message retrieval</b>	The client gets a new message notification from the TMMS PR and immediately retrieves it.
UT04.4	<b>Respond to charging request</b>	The sender has demanded that the recipient is charged for the message. To avoid abuse and deceit the TMMS PR first asks the recipient if it accepts to be charged.

#### 2.4.1.4 User with GSM SMS Client

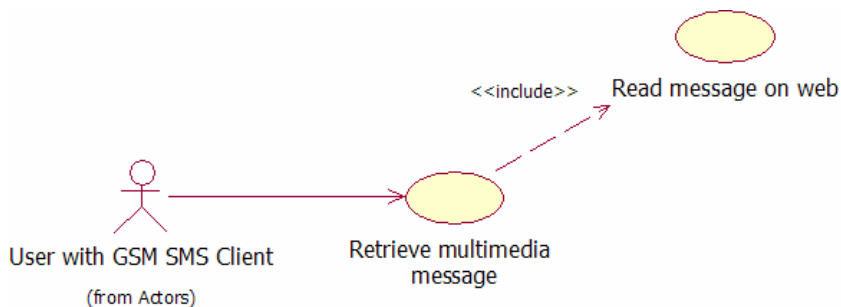


Figure 2-13: Use case view for user with GSM SMS Client

The table below describes the use cases.

Table 2-3 Description of use cases for user with GSM SMS client

Id	Use case	Description
UG01	<b>Retrieve multimedia message</b>	A TMMS Client has sent a message to a GSM SMS client. Note that GSM SMS only supports text.
UG01.1	<b>Read message on web</b>	If the message contains media elements other than text the TMMS PR will create a webpage which displays this content. A text message with a Uniform Resource Locator (URL) [11]

### 2.4.1.5 TMMS Proxy-Relay

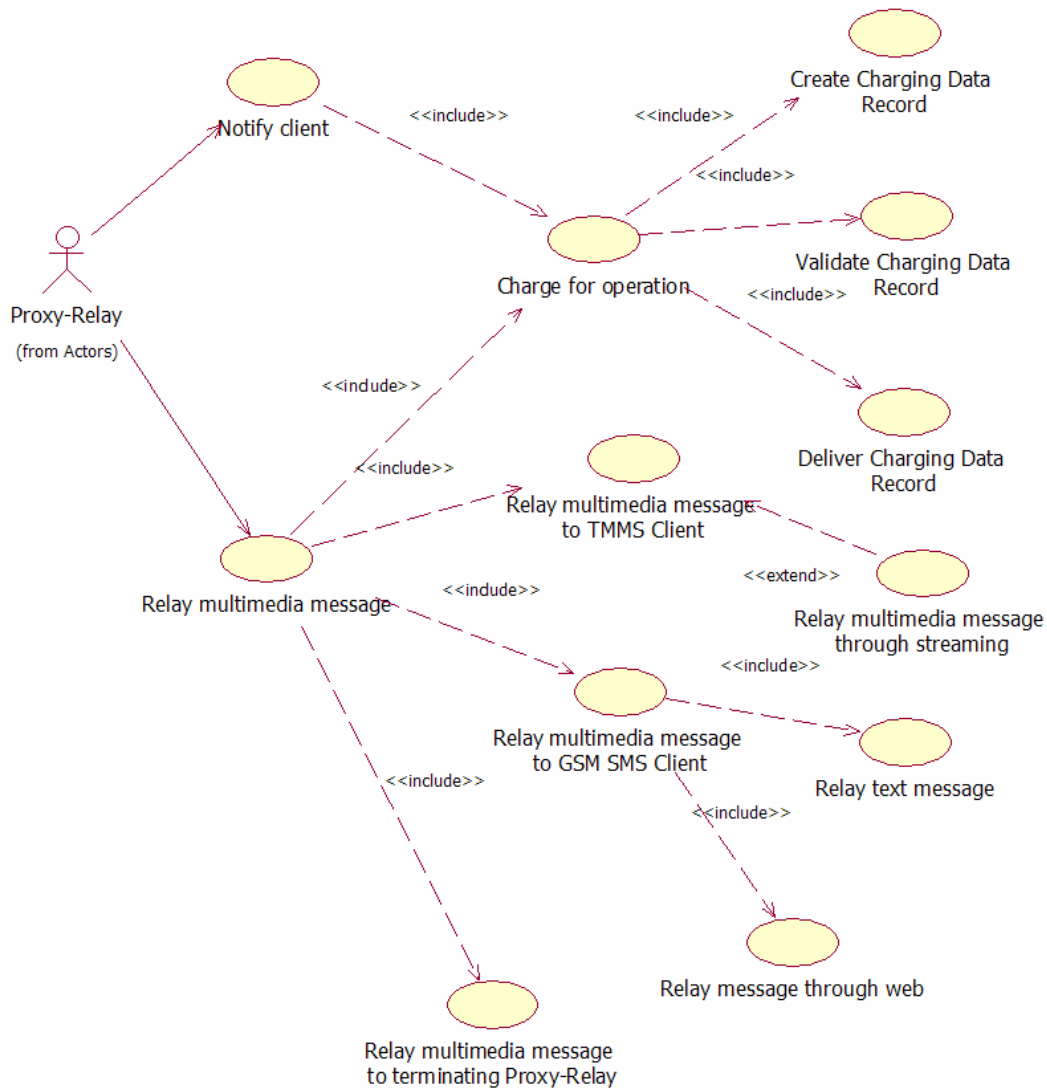


Figure 2-14: Use case view for TMMS Proxy-Relay

The table below describes the use cases. Please note that the logical nodes TMMS Proxy-Relay and TMMS Message Store are treated as one actor here. In the realization of these use cases the behavior is split between these two nodes.

Table 2-4 Description of use cases for TMMS Proxy-Relay

Id	Use case	Description
UP01	Notify client	The TMMS PR has received a new message from either a client or from another TMMS PR and notifies the client(s) registered for the recipient address.
UP02	Charge for operation	A chargeable operation has been requested and the TMMS PR processes this request.
UP03	Create Charging Data Record	For messages sent with no CDR attached the TMMS PR generates a default CDR where the



Id	Use case	Description
		sender is charged for the request.
UP04	<b>Validate Charging Data Record</b>	If the message, which the TMMS PR shall relay, contains a CDR it must be validated before the TMMS PR accepts it.
UP05	<b>Deliver Charging Data Record</b>	The TMMS PR delivers CDRs to a Charging Controller (CC)
UP06	<b>Relay multimedia message</b>	TMMS PR receives a message from either a client or a different TMMS PR and relays it to the next hop, which can be either the recipient(s) or a different TMMS PR.
UP07	<b>Relay multimedia message to TMMS Client</b>	The TMMS Client requests the TMMS PR to deliver a multimedia message which it earlier has been notified about.
UP08	<b>Relay multimedia message through streaming</b>	The TMMS Client wants to retrieve the message as a multimedia stream instead of downloading the message before it can be viewed.
UP09	<b>Relay multimedia message to GSM SMS Client</b>	The recipient of the message is a GSM SMS client, indicated by the recipient address, and the TMMS PR relays it to the recipient through interoperability with the GSM network.
UP10	<b>Relay text message</b>	The message to be relayed contains only text and is sent directly to the recipient.
UP11	<b>Relay message through web</b>	The message contains media elements which are not text. Such a message is uploaded to a webpage and an URL to the webpage is sent to the GSM SMS client as a SMS message.
UP12	<b>Relay message to terminating Proxy-Relay</b>	The recipient does not exist in this network. From the recipient address' hostname the recipient's TMMS PR is located. The message is relayed to this TMMS PR.

### 2.4.2 Logical architecture

In this section the logical architecture and its nodes will be described. In the specific TMMS Service designs described later the description of these nodes will be more specific.

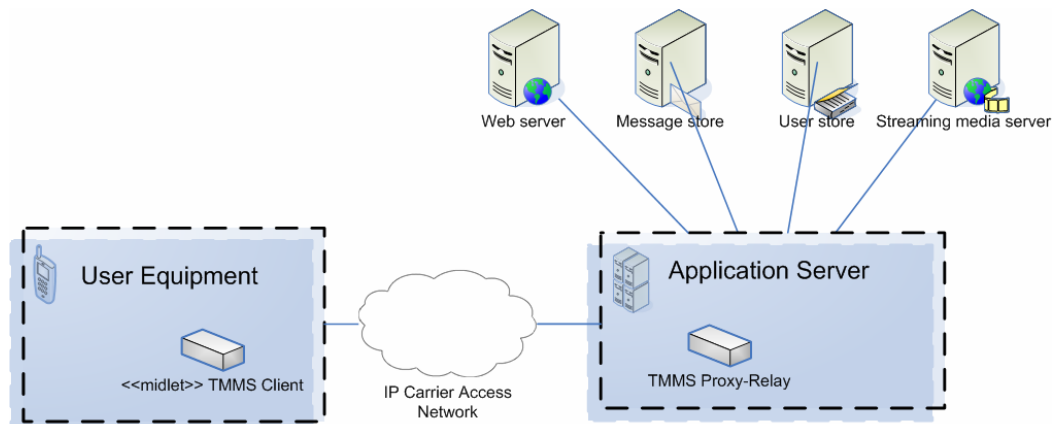


Figure 2-15 Logical architecture of TMMS Service

### 2.4.2.1 TMMS Client

TMMS Client is the application users access the TMMS Service with. For all the specific service designs it is a MIDP client application.

### 2.4.2.2 TMMS Proxy-Relay

TMMS Proxy-Relay is a server application which may run inside an operator's network or outside it. It is responsible for relaying messages from the clients it serves to the destination. The target destination is either one of its own clients, a different TMMS PR or a GSM SMS client. In addition it notifies registered clients about new messages.

### 2.4.2.3 IP Carrier Access Network

The TMMS Client and the TMMS Proxy-Relay communicates through a CAN.

### 2.4.2.4 Web server

For interoperability with GSM SMS, where media type is restricted to text, a web server is used to deliver the multimedia message. This is described in use case UP11 in Table 2-4.

### 2.4.2.5 Message store

The MS is used to store incoming messages. This can be a relational database.

### 2.4.2.6 User store

TMMS PR needs to keep track of the users it's responsible for, so only authorized users can access the service.

There are several solutions available for this. For example a relation database can be used or a directory service. Especially if integrating the service into an enterprise's network it will be necessary to integrate the service with the enterprise's directory service.

### 2.4.2.7 Streaming media server

When the TMMS MS is requested to stream a multimedia message back to the client it uploads the message to a streaming media server and directs the client to use that.

Except for the design based on IMS, which uses a standard IMS component, the streaming media server can be any Commercial Off The Shelf (COTS) streaming media server can be used. The only requirement for the streaming media server is that it must support the Real Time Streaming Protocol (RTSP) [12].

### 2.4.3 Component view

In the following figure the components which TMMS Service is dependent on depicted.

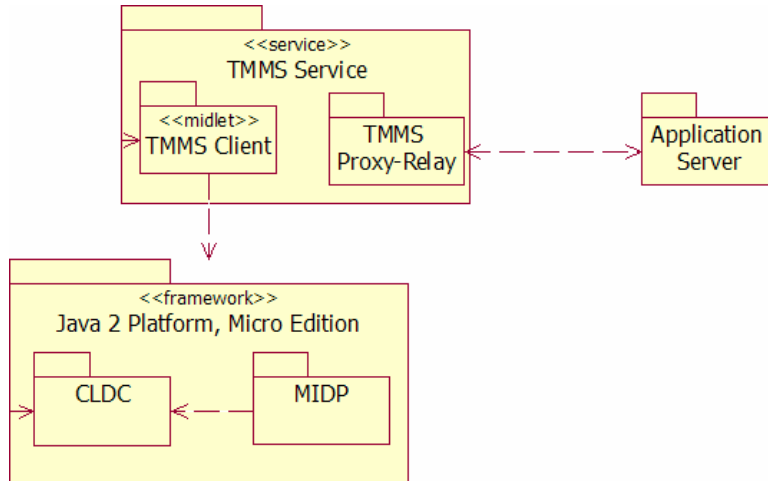


Figure 2-16 Component view of TMMS Service

The Application Server package is an abstraction of the hosting environment where the TMMS PR will be deployed. An example of such an environment is a SIP AS in IMS.

In the following subsection the internal components of TMMS Client is described.

#### 2.4.3.1 TMMS Client

The figure below depicts the components of the TMMS Client. It is partitioned into three subsystems: Storage, Communication and User Interface, which are described in the following subsections.

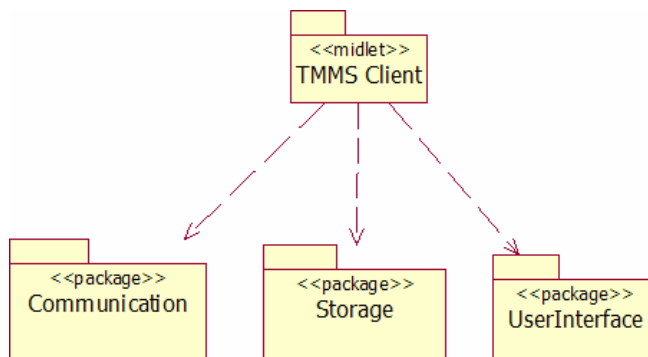


Figure 2-17 Component view of TMMS Client

TMmsClient (TMMS Client in the figure above) is the most top-level package of the application, and contains the main controller and startup class. The main controller, TMmsClient, initializes and uses the other subsystems..



The following table describes these components.

Table 2-6 Description of Storage package components

Component	Description
<b>Message</b>	This is the low level messaging format which encapsulates a TMMS Multimedia Message (MM).
<b>MessageStoreItem</b>	Encapsulates a Message and adds a message identifier to it, which is used internally by the TMMS Client for bookkeeping purposes.
<b>StorageControl</b>	An interface which gives a subset of the capabilities of the javax.microedition.rms.RecordStore class.
<b>StorageManager</b>	Implements the StorageControl interface, and is used by external classes, such as TMmsClient. It is used to control stores such as the inbox and the user settings.

2.4.3.1.3 *TMmsClient.UserInterface*

PR

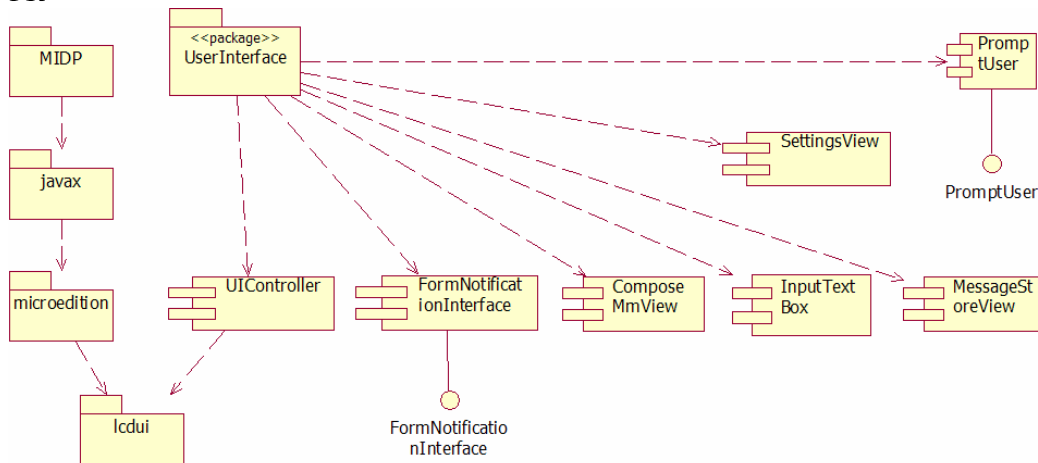


Figure 2-20 Component view of UserInterface package

The following table describes these packages.

Table 2-7 Description of UserInterface package components

Component	Description
<b>UIController</b>	Is used by the main controller, TMmsClient, to control the user interface.
<b>FormNotificationInterface (interface)</b>	Used by the InputTextBox component to return the input from the user.
<b>FormNotificationInterface (component)</b>	This is an implementation of the FormNotificationInterface interface.

Component	Description
<b>ComposeMmView</b>	A view where the user can compose multimedia messages.
<b>MessageStoreView</b>	A view to manage messages stored by a StorageManager. Is used to view the inbox and draft stores.
<b>SettingsView</b>	A view where the user can change configuration properties.
<b>InputTextBox</b>	Used by the external packages, such as Communication, to get input from the user.

The user interface provided by the UserInterface package can be seen in "Appendix A: TMMS Client user interface".

## 2.5 Evaluation criteria

For the evaluation of the different end-to-end services we have designed several quality attribute utility trees [13], which define some criteria we need to take in consideration. Each criterion has one or many scenarios which the evaluation of our services are based on.

### 2.5.1 Security

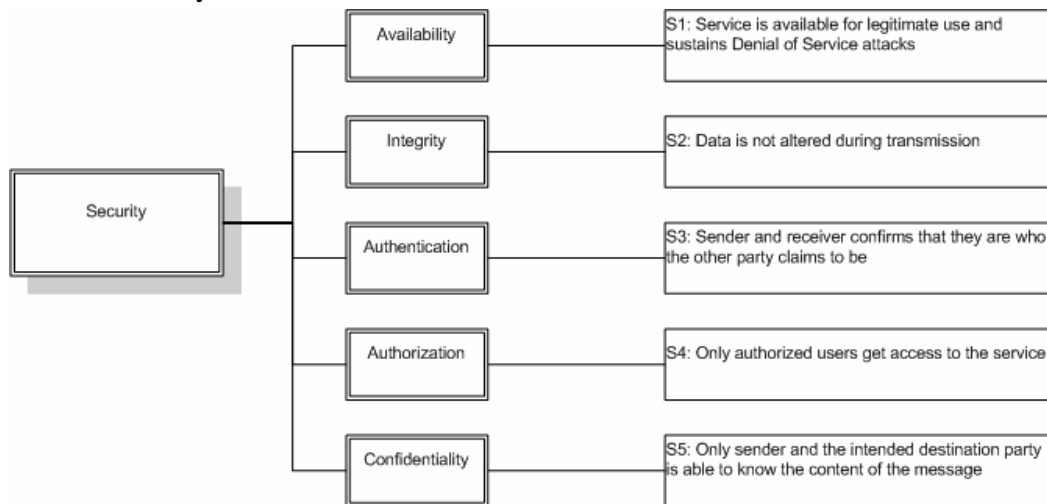


Figure 2-21 Security evaluation scenarios

### 2.5.2 Usability

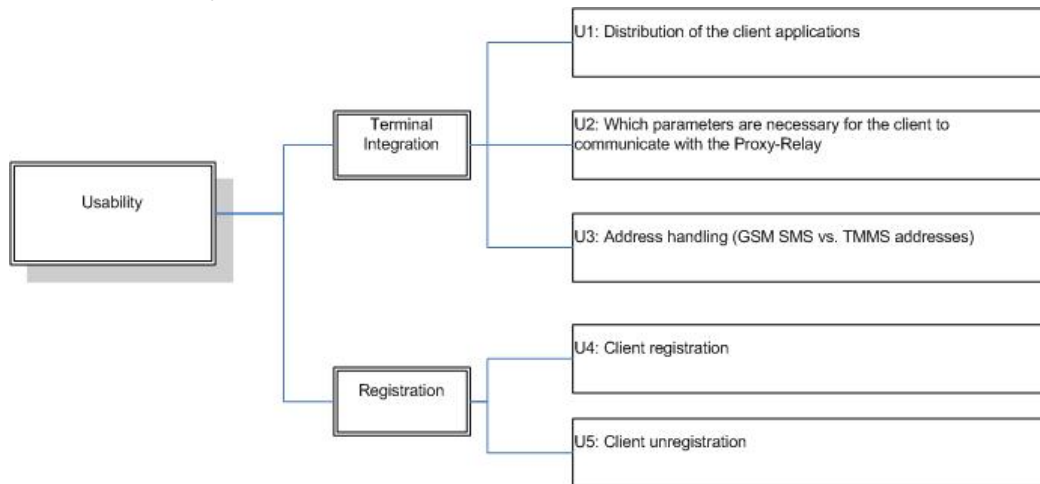


Figure 2-22 Usability evaluation scenarios

### 2.5.3 Modifiability

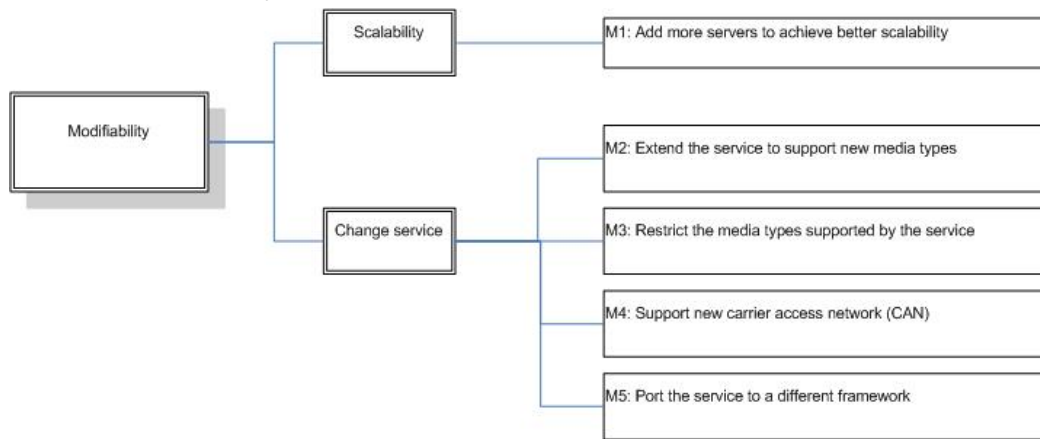


Figure 2-23 Modifiability evaluation scenarios

### 2.5.4 Interoperability

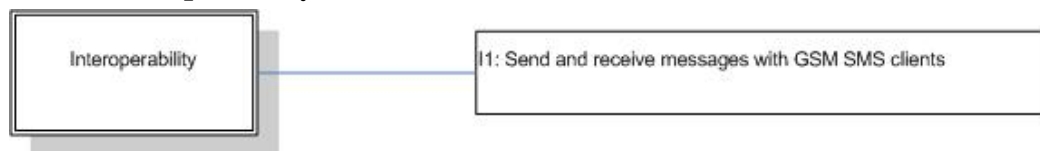


Figure 2-24 Interoperability evaluation scenario

### 2.5.5 Reliability

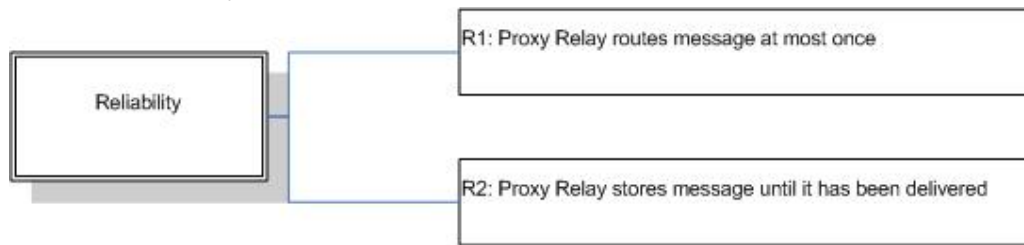


Figure 2-25 Reliability evaluation scenarios

### 2.5.6 Billability

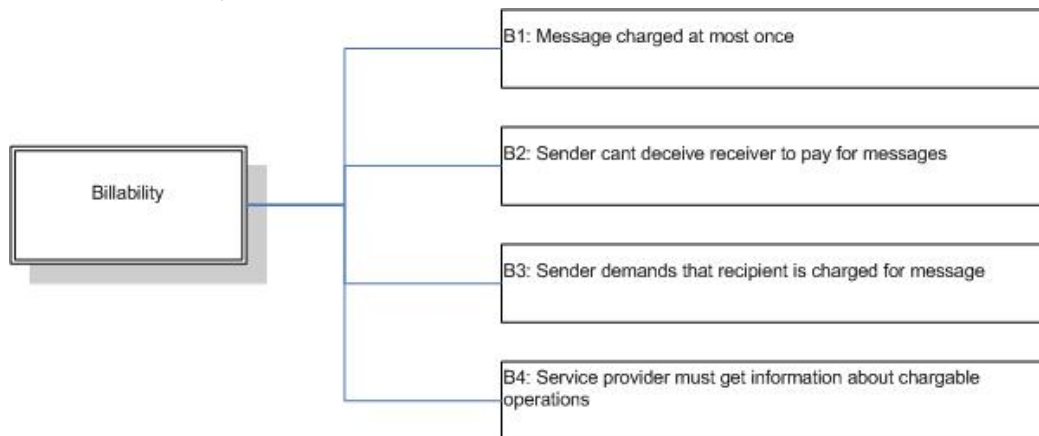


Figure 2-26 Billability evaluation scenarios

## 2.6 Session Initiation Protocol

SIP[14] is an emerging application layer protocol specified by the Internet Engineering Task Force (IETF). SIP is designed to establish and manage multimedia sessions over Packet Switched (PS) networks. It provides basic functionality for establishing, modifying, and tearing down sessions, as well as capabilities to implement a host of more advanced services.

SIP supports five facets of establishing and terminating multimedia communication, these are:

- User location
- User availability
- User capacities
- Session setup
- Session management

### 2.6.1 Architecture

SIP is based on already existing protocols, especially HTTP, but it also borrowing design principals from Simple Network Management Protocol (SNMP) [15]. This is an important strength since those two protocols are well-known and two of the most successful protocols made, and this makes SIP an attractive protocol. SIP is also to be used with other protocols such as the Real-time Transport Protocol (RTP) [16], RTSP, and Media Gateway Control Protocol (MeGaCo) [17]to build a complete multimedia



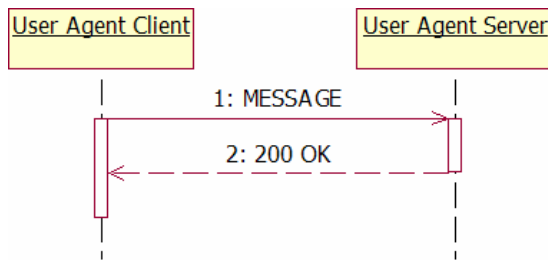
architecture. All these streaming protocols will be introduced in chapter 2.7. Since SIP is a text-based protocol, it makes it easier to extend, debug, and to build services. Other benefits are rapid programmability, and scripting capabilities [18].

SIP does the following things:

- Provides mechanisms for establishing calls between a caller and a callee over an Internet Protocol (IP) [19] network.
- Provides a mechanism for the caller to determine the Current IP address of the callee.
- Provides a mechanism for call management such as adding new media streams during the call, changing the encoding during the call, inviting new participants during the call, call transfer, and all holding.

SIP is based on a request/response transaction model, just like HTTP. Each transaction consists of a request that invokes a particular method or function on a server, and at least one response. Both types of messages consist of a start line, one or more header fields, an empty line indicating the end of the header field, and an optional message body. The most important SIP message is the INVITE message. This SIP message is used to establish a session between to participants.

An example of a SIP transaction is given in the figure below.



**Figure 2-27: SIP transaction example**

A typical SIP message may look like this [20]. In the INVITE line we find the SIP version. The SIP message attaches a Via header which indicates the IP address of the sender. Under the Via header is the From and To headers which indicates the sender and receiver addresses. In this example the SIP addresses are email addresses. Other possible forms of SIP addresses could be telephone numbers or first/middle/last name, assuming that it is unique. SIP identifiers are called SIP Uniform Resource Identifier (URI) [21]. The Call-ID line uniquely identifies the call. The Content-Type header defines the format used to describe the content contained in the SIP message. The last of the headers, the Content-Length, shows the length in bytes of the content in the message. After the empty line, which indicated the end of the header fields, comes the content. In this case the IP address to who sent the INVITE message and how the sender wants to receive the audio which is to be sent.

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-type: application/audio
Content-Length: 885
```

```
C=IN IP4 167.180.112.24
M=audio 38060 RTP/AVP 0
```

As mentioned in [22], a session which is to be established is described by a session description. This description contains enough information about the session for the remote user to join the session. The information includes IP addresses and port numbers. Session descriptions are created using standard format, and the most common one is the Session Description Protocol (SDP). The SDP protocol consists of two parts, a session-level information part and a media-level information part. SIP is session description independent, and therefore SIP doesn't need to use the SDP protocol, but can use whatever description format it wants.

### 2.6.2 Call invitation example

Here we give an example [20] in use of the SIP protocol. In this example Alice is at her PC and wants to communicate with Bob who is on his PC. To make the example on how to set up a call using SIP more understandable, let's say Alice knows Bob's IP address. The SIP session starts with an INVITE message from Alice. This message is sent over User Datagram Protocol (UDP) [23] to the well-known port 5060. This message includes an identifier for Bob, Alice's current IP address, an indication that Alice wants to receive an audio encoded in Pulse Code Modulation (PCM) format and encapsulated in RTP, and an indication that she wants to receive the audio on port 38060. Bob responds this INVITE message with a 200 OK, which includes Bob's IP address, his desired encoding and packetization for reception, and the port number which the audio should be sent.

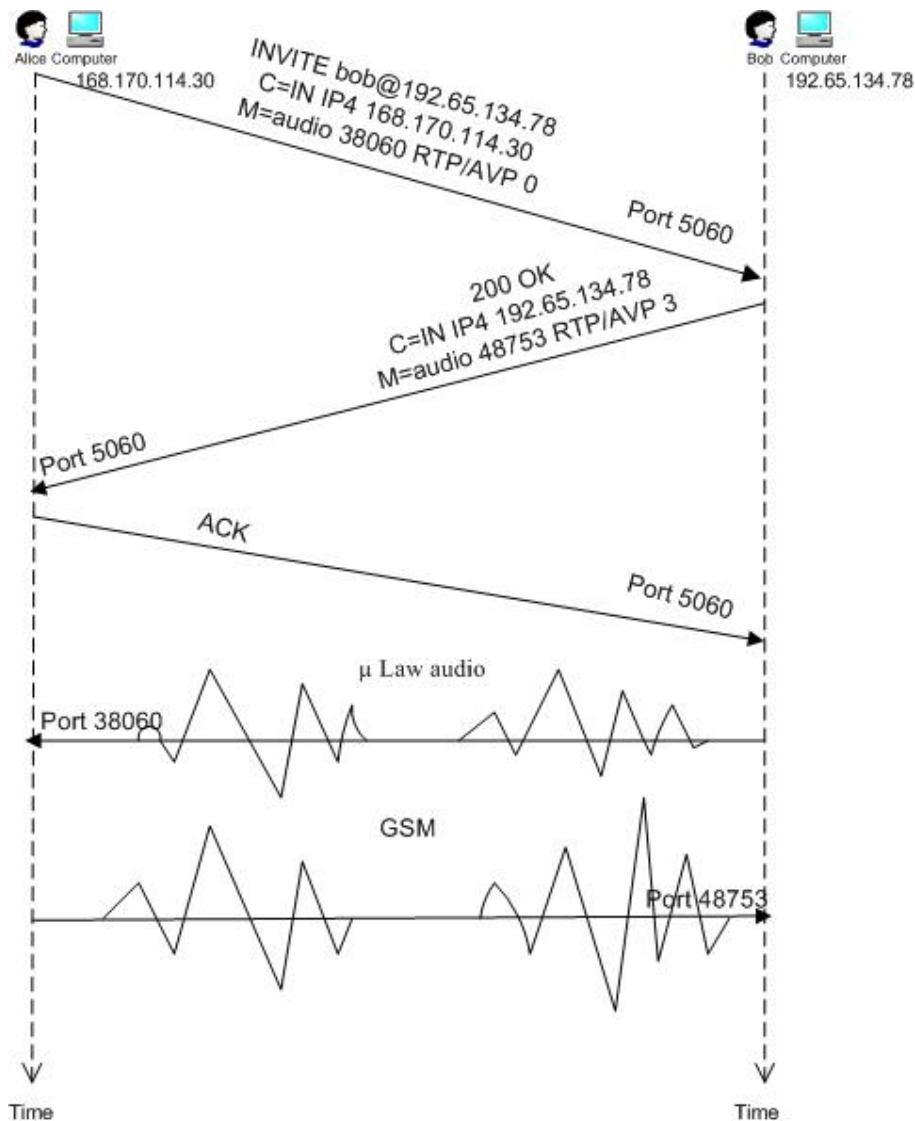


Figure 2-28: SIP call establishment [20]

In the example above, where Alice sends an INVITE message to Bob, she knows Bob's IP address. This is not particularly realistic. What Alice has to do when she doesn't know Bob's IP address, is to send the INVITE message to a SIP proxy server which is a SIP router. The SIP proxy server will inspect the request URI and respond on Alice's request with the IP address to the device Bob currently is using. The SIP proxy server knows all of Bob's IP addresses because of the SIP registrar. Whenever a user launches a SIP application on a device, the application sends an SIP register message to the registrar, and informs the registrar of its current IP address. The registrar can therefore keep track of Bob's IP address, and whenever Bob is switching to a different device, the device is sending a register message. If a user is staying on the same device for a long time, the device will send a refresh message to let the registrar know that the user is still on the same device and that the IP address is still valid.

SIP is an out-of-band protocol which means that the SIP messages are sent and received in sockets that are different from those used to send and receive the media data. As we can see from the figure, SIP also requires all messages to be acknowledged so it can be run over Transmission Control Protocol (TCP) [24] or UDP.

Additionally, in [14] SIP is structured as a layered protocol. This means that the protocols behavior is described in terms of a set of fairly independent processing stages with only a loose coupling between each stage. The lowest layer is the syntax and encoding. The next layer is the transport layer which defines how a client sends and receives messages and how servers receives requests and respond to these. The third layer is the transaction layer which handles application-layer retransmissions, matching of responses to requests, and application-layer timeouts. The next layer is the transaction user layer.

### 2.6.3 Back-to-back User Agent

In [14], a UA is defined as an entity that represents an end-system. The UA involves a UAC which sends requests to a UAS via a number of proxy servers. The UAS sends responses back to the UAC forwarded by the proxy servers. This leads us to the definition of a Back-to-back User Agent (B2BUA). A B2BUA is a logical entity that receives a request and processes it as a UAS. In order to determine how a request should be answered, it acts like a UAC and generates requests. It maintains dialog state and must participate in all requests sent on the dialogs it has established.

### 2.6.4 Security

Fundamental security services required for the SIP protocol are preserving the confidentiality and integrity of messaging, preventing replay attacks or message spoofing, providing for the authentication and privacy of the participants in a session, and preventing denial-of-service attacks. Bodies within the SIP messages separately require the security services of confidentiality, integrity, and authentication. Instead of defining new security mechanisms, SIP uses existing security models derived from HTTP and SNMP when possible.

In addition to SIP URIs, SIP also provides a secure URI called SIPS URI. The use of these SIPS URIs guarantees secure encrypted transport of SIP messages.

Example of a SIPS URI is:

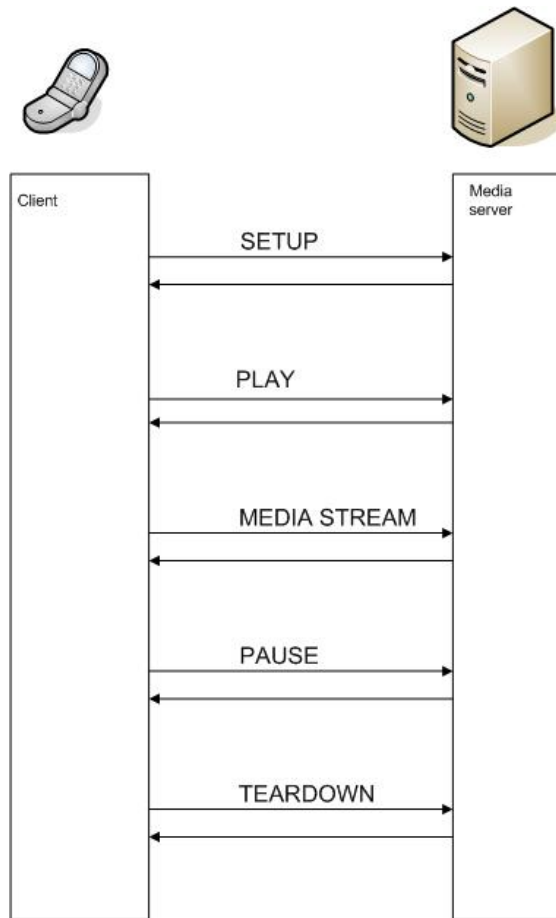
```
sips:bob@example.com
```

## 2.7 Streaming Protocols

As mentioned in 2.4.2.7 our TMMS Clients have the opportunity to use streaming to get some of the media in the TMMS. Here follow an introduction to the streaming protocols used in our services. This background material is based on [10].

### 2.7.1 Real-Time Streaming Protocol

Real-Time Streaming protocol (RTSP) is a protocol that allows users to control the transmission of a media stream. It is an out-of-band protocol which means that the RTSP messages are sent using different port numbers than the media stream.



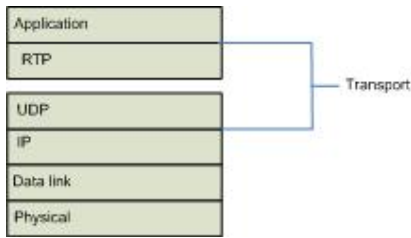
**Figure 2-29** Interaction between Client and Server

The client sends a SETUP message to the media server, which responds with an OK message. Next, the client sends a PLAY message to the media server, and the media server responds with an OK message. Now the client is ready to receive media chunks from the server. After some time the client sends a PAUSE message, and the media server responds with an OK message. When the client is finished, it sends a TEARDOWN message, and the media server confirms with an OK message.

In many ways the RTSP protocol has many similarities with HTTP and SIP. All have **American Standard Code for Information Interchange (ASCII)** text based messages. The client sends standard methods such as SETUP, PLAY and PAUSE, and the server responds with standardized reply codes such as 200 OK. However, one major difference from HTTP is that the RTSP servers keep track of the state of the client for each ongoing RTSP session. To ensure this the servers use the session and sequence numbers which are a part of each RTSP request and response. The session number is the same through the entire session, while the sequence number is incremented by one for each message the client sends.

### 2.7.2 Real-Time Protocol

Real-Time Protocol (RTP) is a protocol which typically runs on top of UDP and can therefore be viewed as a sub-layer of the transport layer.



**Figure 2-30: RTP can be viewed as a sublayer of the transport layer**

The media chunk is encapsulated in a RTP packet which its self is encapsulated in a UDP segment. The receiver extracts the RTP packet from the UDP segment and then extracts the media chunk from RTP packet. The media chunk along with the RTP header forms the RTP packet. The RTP header includes i.a. a sequence number and a timestamp.

### 2.7.3 Real-Time Control Protocol

Real-Time Control Protocol (RTCP) [16] is a protocol used in conjunction with RTP. RTCP packets are sent periodically from each participant in an RTP session to all other participants in the session using Internet Protocol Multicast (IP-M). These packets do not encapsulate media chunks as in RTP, but contains sender/receiver reports that can be useful for the application.

## 3 TMMS Service based on GPRS

In this chapter we will first give a brief introduction to GPRS, which is the technology our end-to-end mobile service in this chapter is based on. This introduction is based on how GPRS is specified in [25]. We continue this chapter with presenting how we have chosen to design our GPRS based service, before we finish the chapter with an evaluation of the service we have designed, based on the criteria and scenarios given in 2.5.

### 3.1 Overview

In the 1990's it was decided that it was necessary to develop a Third Generation (3G) cellular technology that was able to use both voice and data communications. Since these things take long time to deploy, companies developed interim protocols and standards that open for data transmission over the existing Second Generation (2G) infrastructure. This was called Second and a half Generation (2,5G) and GPRS is a part of the 2,5G infrastructure [20].

#### 3.1.1 Architecture

GPRS provides an end-to-end packet transfer service, in contrast to e.g. the data transfer service provided by GSM which is circuit switched. The intention of using packet switching is to use the network resources more efficient. This makes it cheaper to use then a circuit switched base service as well.

It depends on a Radio Access Network (RAN) from the Public Land Mobile Network (PLMN), which can be GSM/Enhanced Data rates for Global Evolution (EDGE) Radio Access Network (RAN) (GERAN)[xx] or Universal Terrestrial Radio Access Network (UTRAN)[xx]. These radio interfaces are named the Um and the Uu reference points, respectively, in the 3GPP specifications.

It is operator specific how to charge for GPRS usage, but most likely a charging scheme based on usage rather than duration will be enforced. This means that customers can stay connected and will only be charged for the packets transferred during the session. Another charging scheme is to charge for service access per month, like NTT DoCoMo's i-mode concept [26].

The main concept of GPRS is that for the new GPRS radio channel, the GSM system can allocate between one and eight time slots within a Time Division Multiple Access (TDMA) frame. These time slots are allocated on demand.

GPRS defines two different bearer service types:

- **Point-to-Point (PTP) packet transfer service**  
This service transmits packets between users, a service requester and a receiver. In this type of service one or more single packets are sent from a single sender to one single receiver.
- **Point-to-Multipoint (PTM) packet transfer service**  
This service provides transmission of packets between a service requester and a receiver group. This means transmission of a single message to multiple subscribers.

The figure below shows a simplified version of the GPRS architecture. The complete logical architecture of GPRS may be seen in [27].

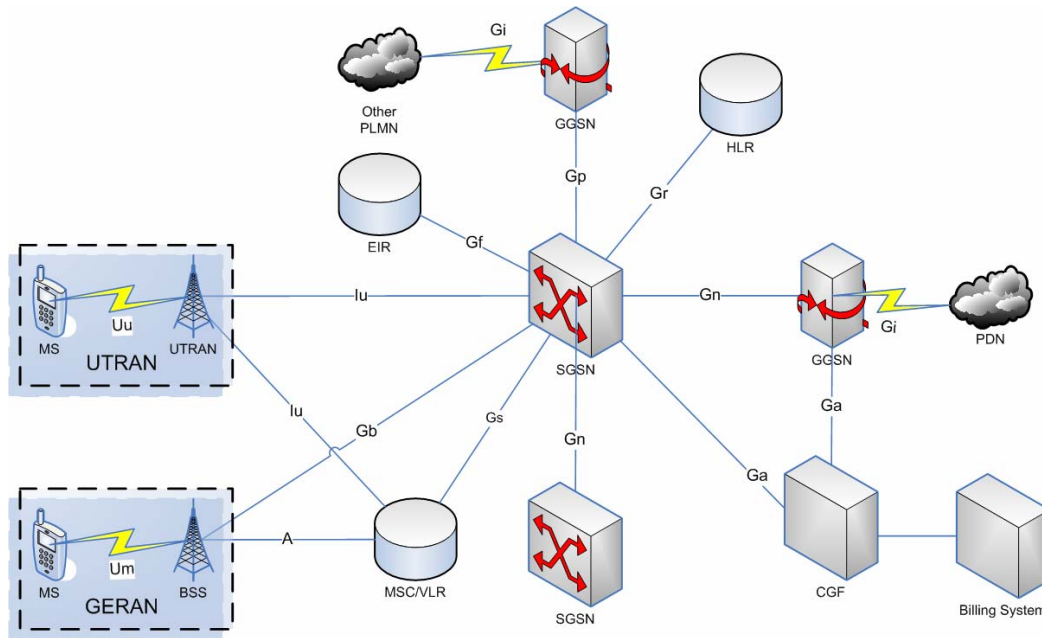


Figure 3-1: Overview of the GPRS architecture



The GPRS architecture introduces a new network node type: GPRS Support Node (GSN), which is a router and is integrated into the network architecture by some well-known reference points. It is also responsible for collecting billing information. There are two kinds of GSNs:

- **Gateway GPRS Support Node (GGSN)**  
This is a unit which provides the interface towards external data networks. GGSN contains routing information and tunnels data to a user using encapsulation. It also translates data formats, signaling protocols and address information in order to permit communication between the different networks.
- **Serving GPRS Support Node (SGSN)**  
Requests user addresses from the GPRS register (GR), keeps track of individual User Equipment (UE) locations, and handles security functions, such as access control. SGSN is connected to GERAN Base Station (BS) system through the Gb or Iu interface, and/or to the UTRAN system through the Iu interface. SGSN is the node that is serving the UEs.

The SGSN and GGSN can be combined in the same physical node or they may reside in two different nodes. When they reside in different nodes they communicate with the Gn interface. They contain IP or other routing functionality, and may be interconnected with IP routers. When the SGSN and the GGSN are in different PLMNs they are interconnected via the Gp interface.

Before the UE can send or receive data it has to be connected to the GPRS network. To connect to this network the UE assigns a temporal identifier, called TLLIIdentity (TLLI), and a CKSNNumber (CKSN), for data encryption. Then a GPRS context is created for each UE, which is stored in the UE itself as well as the corresponding SGSN. This is called a GPRS Attach. A GPRS attach is performed when the UE indicates its presence to the network. This can be done immediately after the UE has been switched on or later when the UE decides to use a GPRS service. When the UE is finished using the GPRS services it performs a GPRS detach. The network can also perform a GPRS detach. This can happen for several reasons, i.e. ill behaving UE, congested network and immediate service termination among others.

#### 3.1.2 Security

As GPRS is particularly sensitive for misuse and eavesdropping, GPRS includes several security services such as authentication to protect the network against unauthorized use, access control which means that the network can support restrictions on access by two or more subscribers, user identity confidentiality which purpose is to provide privacy of identities to the subscribers who are using GPRS radio resources, and user information confidentiality which purpose is to provide for confidentiality for user data.

#### 3.1.3 Charging

GPRS doesn't enforce a charging scheme for operators to use. Instead it collects information which the operator may use to charge its customers. The information collected is at least: source and destination of the packet sent, usage of radio interface, usage of external data networks, and usage of the packet data protocol addresses, usage of general GPRS resources and location of the UE.

It is the Charging Gateway Functionality (CGF) that collects charging records from the SGSN and GGSN.

### 3.2 Design

#### 3.2.1 Introduction

The GPRS based TMMS uses the SIP protocol for communication between clients and PRs and also for routing between PRs. The requests and responses used are described in the Scenario Implementation section below.

Note that there is not any easy mechanism available for the TMMS Proxy-Relay (PR) to get notifications about clients becoming online and offline. One mechanism available is to use the Customized Applications for Mobile networks Enhanced Logic (CAMEL) [28] framework, but that makes the service more dependent on the network operator's network, which we would like to avoid. Therefore an explicit registration and un-registration process is needed.

#### 3.2.2 Deployment view

The following figure depicts how the TMMS Service is deployed with the GPRS framework.

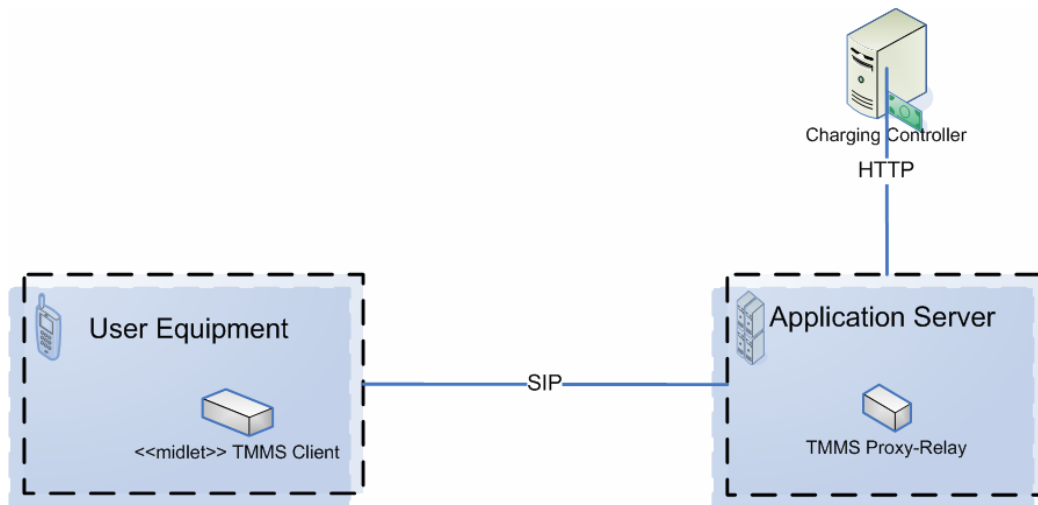


Figure 3-2 Deployment of GPRS based service

##### 3.2.2.1 TMMS Client

The service operator must provide the customer with an URL which directs the client to a Java Application Descriptor (JAD) document. A JAD contains all information necessary for the customer's UE to retrieve and download the MIDP client application. This JAD document can be seen in "Appendix E: TMMSClient-IMS.jad". The client does not enlist for Push registration in the JAD document because it must subscribe to the newmessage-event, as explained below, before it can use the service.

It needs to know the network's TMMS PR before it can access it. This can be accomplished in two ways:

- Option one is to set the TMMS PR IP address manually, where the service operator gives the IP address to the client.

- The second option is to use a DNS SVR which a client can retrieve from a DNS server to discover the local TMMS PR's address.

### 3.2.3 Component view

#### 3.2.3.1 TMMS Client

The client application uses the Generic Communication Framework [xx] to communicate with the TMMS PR. Support for the HTTP protocol comes from MIDP, and support for the SIP protocol is specified in [8].

In addition to the components described in 2.4 Case description: TMMS Service above adds this solution the package `TMmsClient.Communication.Gprs` which gives an implementation of the `TMmsClient.Communication.Messaging` interface. This implementation is used by the startup class `TMmsClient.TMmsClient` for communication.

#### 3.2.3.2 TMMS Proxy-Relay

The PR is implemented as a SIP B2BUA. It must be accessible by the clients through normal Internet routing procedures.

### 3.2.4 Security

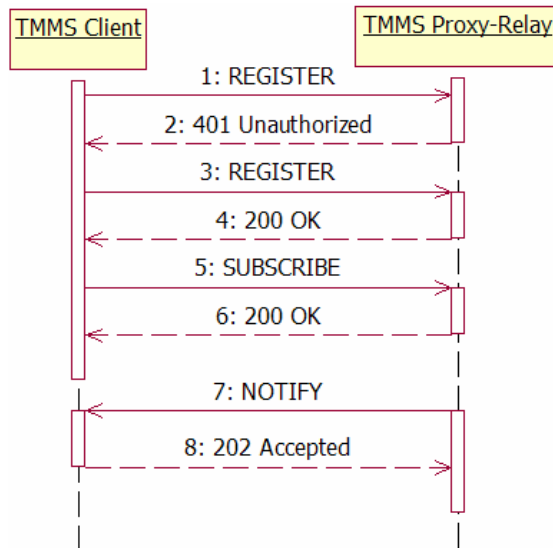
As mentioned above, GPRS does provide security for the user. But it does not offer end-to-end security, which is expected in a service. With SIP there are two possibilities offered: SIP and Secure SIP. These are distinguished by the URI used in the addressing, which are sip and sips, respectively.

In the figures below the normal SIP scheme will be used with authentication from HTTP Digest. Note that only the first figure includes the authentication part, which is the first two messages sent. For the rest of the interactions this part of the scenario is not included.

### 3.2.5 Scenario Implementation

#### 3.2.5.1 S01: TMMS Client Registration

When a client wants to use the messaging service it needs to register with the TMMS PR that it is online. The registration sequence is shown in Figure 3-3 below.



**Figure 3-3 TMMS Client Registration**

The client sends a SIP REGISTER request to its TMMS PR which acts as a registrar for its clients. The Request-URI is composed of the client's username and the TMMS PR's domain name, like [username@tmsspr1.example.com](mailto:username@tmsspr1.example.com). The same is the To and From headers set to, but with the sip(s) URI scheme in front, like "sips:username@tmsspr1.example.com". This is the client's public identity and the address for example a friend of this user will use to send her a message. To let the TMMS PR know how to contact the user the Contact header is set to an URI which contains the user's current IP-address and the port which can be used.

When the TMMS PR receives the REGISTER request from the client it first checks if the client is authenticated. If not, like in Figure 3-3 above, the TMMS PR returns a 401 (Unauthorized) response back to the client. This response will include a "WWW-Authenticate" header which the client will use to authenticate itself. Now the client will send a second REGISTER request to the TMMS PR with the authentication details included. If the authentication of the client is successful and the username in the Request-URI is a TMMS Client then the TMMS PR returns a 200 (OK) response back to the client. To check if the client is a TMMS user the TMMS PR does a search in the user store for the username.

To get notifications about new messages the client needs to subscribe to the "newmessage-alert" event, which the TMMS PR uses to notify subscribers about new messages. Subscription is accomplished by sending a SUBSCRIBE request to the TMMS PR, which will respond with 200 (OK). According to [29] the notifier needs to immediately send a NOTIFY message to the user with the current state. This message will contain a body with URIs to all new messages for the client, if any. How the client deals with the NOTIFY message is discussed in sections 3.2.5.3 and 3.2.5.3.1 below.

Now the client is registered with the TMMS PR and will receive notifications about new messages and may send messages to other clients.

### 3.2.5.2 S02: Send MM

This scenario is subdivided into sending the MM to a TMMS Client or to a GSM SMS Client.

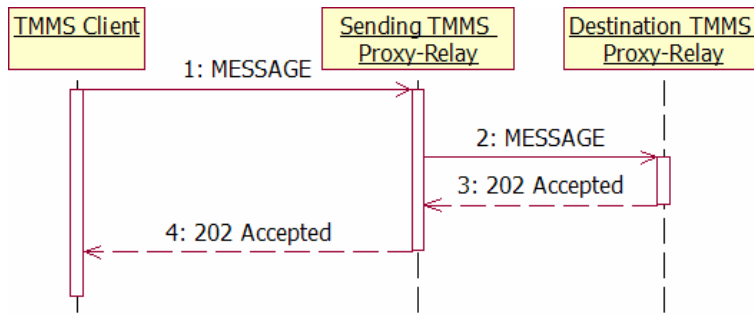
**3.2.5.2.1 S02.01: Send MM to TMMS Client**

Figure 3-4 Send MM to TMMS Client

The MM is transferred in the body of a MESSAGE request. The client sends it to its designated TMMS PR which will route it to the recipient's TMMS PR based on domain name of the "To" header(s) in the request. Since MMS is a "store and forward" service the TMMS PRs will reply with a 202 (Accepted) response back. This indicates that the TMMS PR has accepted and stored the message and will notify the recipient about it.

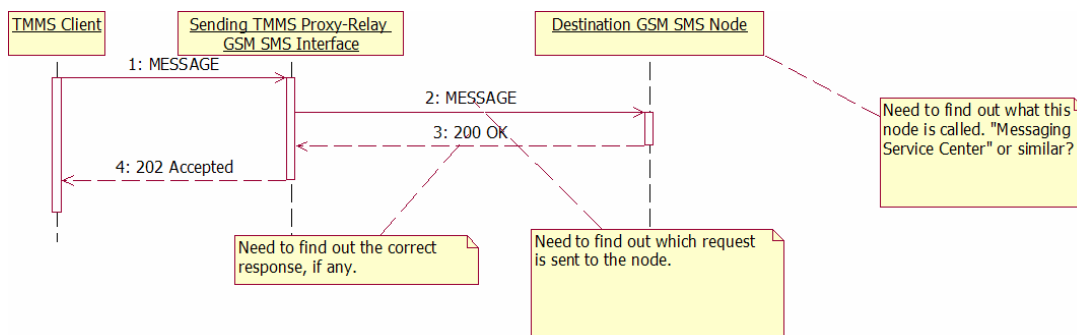
**3.2.5.2.2 S02.02: Send MM to GSM SMS Client**

Figure 3-5 Send MM to GSM SMS Client

For the client this scenario is exactly the same as in S02.01 above. When the TMMS PR examines the "To" header(s) it will detect a TEL URI [30] and assume that the recipient is a GSM client. If the message content is text only then it will forward the message as described below.

However if the message contains other media than text, which is not supported in GSM SMS, then it will need to convert it. The way this is done is to upload the original message content to a HTTP[31] server and create a HTML[32] document which presents the content. Next the TMMS PR will create a normal SMS message which will include an URI to this HTML document.

The SMS message sent to the recipient is then sent to the GSM network which will deliver it to the recipient, based on the Mobile Station Integrated Services Digital Network(MSISDN) found in the TEL URI.

### 3.2.5.3 S03: New message notification

We have now described how a TMMS Client sends a message to a recipient above. Now we will discuss how the TMMS Client will be notified about a new message. After being notified about a new message the client can choose to either get the new message immediately or defer retrieval until later. The latter is a combination of scenarios 3.2.5.3.2 and 3.2.5.4.

When a TMMS PR is forwarded a MM it will first store it in its message store. This will trigger a NOTIFY message with the "newmessage-alert" event, (3) in Figure 3-6 below, to be sent to the recipient of the MM. Since the client subscribed for this event during registration it will receive this notification. In the body of the message it will find a URI which tells the client where it can find the message. An example of such a Request-URI is: <http://inbox.tmms.example.org/username/msgid>, where username will be set to the recipient's username and msgid will be replaced with a real identifier for the message (such as msg0000). Here the scenario forks and we will discuss these two sub-scenarios in the sections below.

#### 3.2.5.3.1 S03.01: TMMS Client Receives MM (Immediate Retrieval)

Here the client retrieves the message immediately. This is done by sending a HTTP GET request with the URI given in the body of the NOTIFY message as the Request-URI. When it has downloaded the MM it will return 200 (OK) back to its TMMS PR. The TMMS PR can delete the MM when the client has retrieved it. Therefore the client needs to be careful to not respond too early back to the TMMS PR.

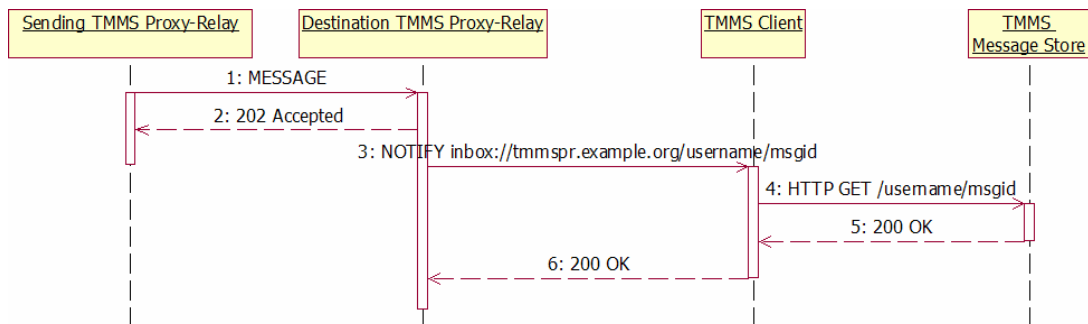


Figure 3-6 TMMS Client Receives MM (Immediate Retrieval)

#### 3.2.5.3.2 S03.02: TMMS Client Receives MM (Deferred Retrieval)

If the client does not want to retrieve the MM immediately it can reply with 202 (Accepted) to the TMMS PR. This means that the client knows about the MM but will retrieve it later.

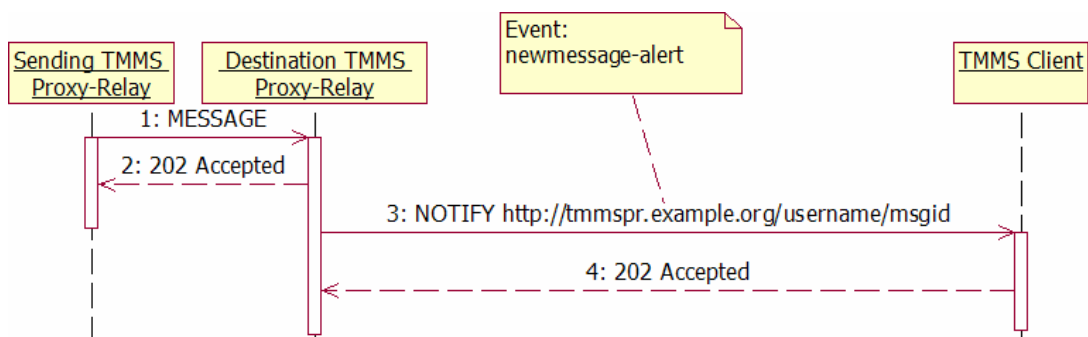


Figure 3-7 TMMS Client Receives MM (Deferred Retrieval)

### 3.2.5.4 S04: Deferred MM Retrieval

To retrieve a message deferred for retrieval the client does almost the same as when retrieving immediately. It sends a HTTP GET request with the Request-URI set to the URI it learned from the NOTIFY message it got earlier.

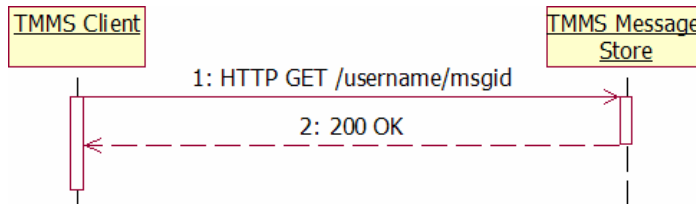


Figure 3-8 Deferred MM Retrieval

### 3.2.5.5 S05: Streaming Retrieval of MM

If the Multimedia Message (MM) contains large media parts, like audio or video parts, then the TMMS PR may decide that it will stream the content to the client instead of letting the client download the contents through HTTP GET, as discussed in the sections above. Note that the TMMS PR needs to know if the client supports streaming first. Here we will assume that the client supports streaming.

Each media part is examined and then the TMMS PR decides if that media part shall be streamed to the client. If a media part is decided to be streamed then it will be uploaded to the streaming media server. The URL to the media part, which originally will follow the http URI scheme, will be replaced with a rtsp-uri. Now the altered MM is ready to be retrieved by the client.

The client will be notified about the new

### 3.2.5.6 S06: TMMS Client Unregistration

When the client wants to disconnect from the service it first must unsubscribe from the newmessage-alert event. Next it must unregister with the TMMS PR. Both tasks are accomplished by sending the original request, SUBSCRIBE and REGISTER respectively, to the TMMS PR with the Expires-header set to 0.

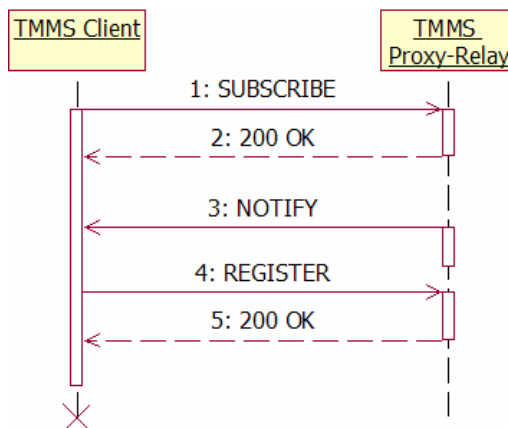


Figure 3-9 TMMS Client Unregistration

### 3.2.6 Charging of Service Usage

There are two main design choices available for charging a service. Either by developing a CAMEL application which supports the service or to let the service itself do the charging. Same as with registration/un-registration of clients a solution with CAMEL is avoided here, for the same reason.

As noted in [33] it is important for the consumer to get one bill for all services consumed, through its network operator. OMA is developing a specification for such a system [xx], but this work is not completed. Instead we have reused the logical design of OMA Wireless Application Protocol (WAP) Billing Framework v1.0 (WBF) [34] The reason for not implementing the complete specification is that it is very tightly bound to WAP, which makes it hard. However, to make it easier to interoperate in a network with WBF support the CDR used is designed to be easy to transform into a WBF CDR if needed.

The service will act as a Content Provider entity in WBF. The Charging Control and Business Support entities are also needed, which ideally are available from the network operator or another party. For example, they could be provided by a company specializing in billing customers. Otherwise the service provider must provide its own entities.

In contrast to the original WBF specification there is only one chargeable operation defined here. The chargeable operation is *message sending*. To support commercial use of the messaging service, e.g. selling multimedia, the sender may request that the receiver is charged instead.

First the charging model is described in detail. Next the eXtensible Markup Language (XML) Schema[35] for the CDR is described.

#### 3.2.6.1 Charging Model

The charging model can be seen in the state machines depicted in Figure 3-10 and Figure 3-11 below.



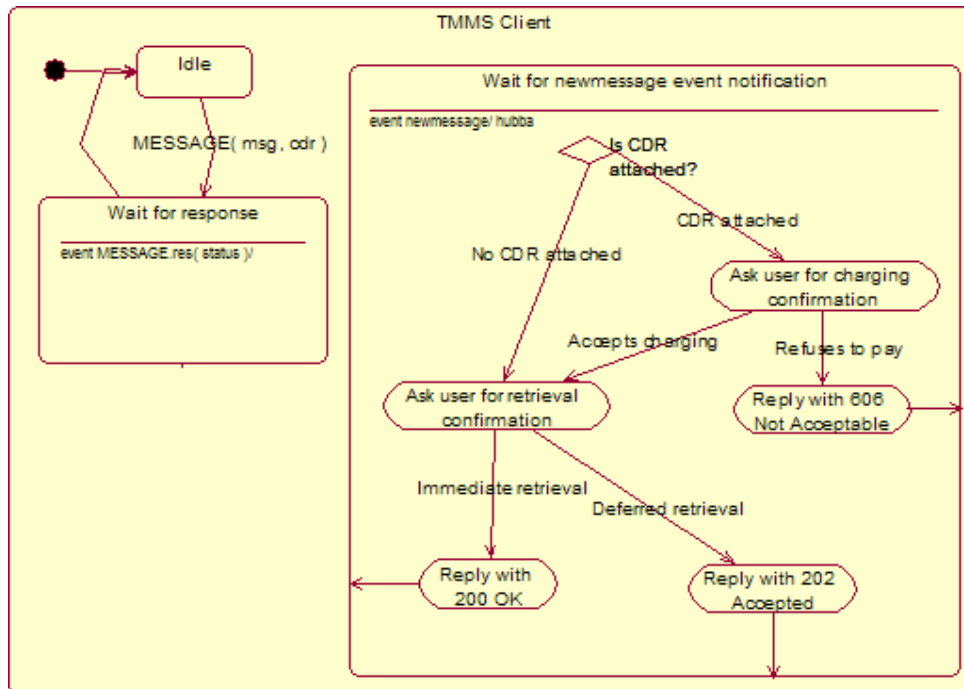


Figure 3-10 Charging state machine for client

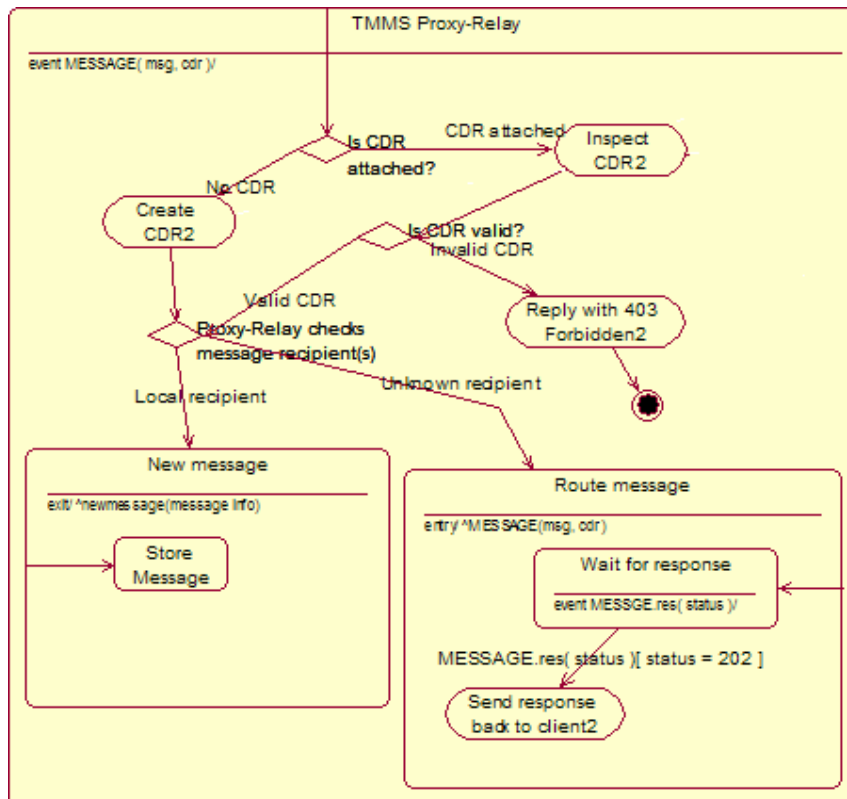


Figure 3-11 Charging state machine for proxy-relay

Both sending and receiving messages are covered in these two figures. In addition it shows inter-TMMS PR routing. Inter-TMMS PR routing is covered here since it uses the same SIP MESSAGE request as clients do.

When the TMMS PR receives a MESSAGE request it will start processing it by checking if a CDR is attached. A CDR can be piggy backed in the request by using a multipart Multipurpose Internet Mail Extensions (MIME) [36] type as the body of the request. An example of such a composite message is given in Appendix D: mm-charging.msg. If no CDR is attached then the TMMS PR will create a standard CDR, where the sender is charged. In either case the CDR will be routed to the Charging Control associated with the network. How the TMMS PR discovers the correct Charging Control is described below. The message will now be split by the TMMS PR so the CDR is sent to the Charging Control and the message received from the sender is routed to the destination, using standard routing mechanisms. Destination here means either the recipient or the recipient's TMMS PR if it is in a remote network.

Another scenario covered is when a client receives a notification about a new message. To disallow senders to trick the recipient to pay for unwanted messages, the newmessage notification contains the CDR which the client application uses to first ask the receiver if it wants to be charged for that message.

How the CDR is transmitted to the Charging Control is not specified in OMA WBF, and is implementation dependent. The same applies here, since this is up to the Charging Control to specify, which is out of scope for this report. E.g. this could be a HTTP POST request. The details of how the Charging Control and the Business Support works are also out of scope for this report. Some information on that topic is given in the OMA WBF specification.

### 3.2.6.2 Discovering the Charging Control

The Charging Control is discovered by the TMMS PR, or any other entity which needs to discover it, by issuing a query to the Domain Name System (DNS) [37] server in the network and querying for the “\_CHRGCTRL” SRV RRresource record (RR), which follows [1].

Note that [1] requires a protocol specification describing this SRV RR in its “Applicability Statement” section. However, this is out of scope for this report.

### 3.2.6.3 Charging Data Record (CDR)

A CDR is a well-formed XML document, conforming to the XML Schema found in Appendix B:charging-details-record.xsd. It contains information about the sender, proxy-relay, receiver, Charging Control and the message itself. By using all or parts of this information the operator can implement various charging schemes at the Business Support entity.

The CDR contains one or more elements describing the media included in the message. Each element is described with its MIME/type and a count of how often it is included in the message. Optionally a price per such media element may be included, which is useful for a multimedia-delivery service.

### 3.3 Evaluation

#### 3.3.1 Security

Table 3-1: Evaluation of Security quality attribute for TMMS Service based on GPRS

<b>Availability</b>	S1: Service is available for legitimate use and sustains Denial of Service attacks
	Client needs to get in contact with the nearest TMMS PR. This can be done in two ways. Option one is to set the PR's IP address manually. The other option is to use DNS.  No security precaution is implemented to take care of Denial of Service attacks in the service.
<b>Integrity</b>	S2: Data is not altered during transmission
	No end-to-end data integrity check, such as checksums, is done to ensure that data is not altered. The client needs to trust the service.
<b>Authentication</b>	S3: Sender and receiver confirms that they are who they claim to be
	The sender is authenticated with the TMMS PR, so given that the receiver trusts the TMMS PR it can also be sure that the message received comes from the claimed sender.
<b>Authorization</b>	S4: Only authorized users get access to the service
	The TMMS PR has a database which contains all authorized clients, which is queried when a client authenticates with it. In this way the TMMS PR can ensure that only legitimate users can access the service.
<b>Confidentiality</b>	S5: Only sender and the intended destination part is able to know the content of the message
	By encrypting the message content only recipients who know how to decrypt it may know the content of the message.

### 3.3.2 Usability

Table 3-2: Evaluation of Usability quality attribute for TMMS Service based on GPRS

<b>Terminal Integration</b>	U1: Distribution of the client application
	Customer must manually retrieve the client application.
	U2: Which parameters are necessary for the client to communicate with the Proxy-Relay
	Parameters needed are: the TMMS PR IP address, local port number and server port number.
	U3: Address handling (GSM SMS vs. TMMS addresses)
	SIP URIs are used to address TMMS Clients, and TEL URIs are used to address GSM SMS clients. Only the PR must treat these different.
<b>Registration</b>	U4: Client registration
	Client must register with the TMMS PR by issuing a SIP REGISTER request.
	U5: Client un-registration
	Clients must un-register with the TMMS PR by issuing a SIP REGISTER request with the Expires-header set to 0.

### 3.3.3 Modifiability

Table 3-3: Evaluation of Modifiability quality attribute for TMMS Service based on GPRS

<b>Scalability</b>	M1: Add more servers to achieve better scalability
	Since the TMMS PR is addressed through DNS it is possible to use normal DNS based network load balancing techniques to add more servers.
<b>Change Service</b>	M2: Extend the service to support new media types
	The service uses MIME types to describe the media types of the message body.
	M3: Restrict the media types supported by the service
	The TMMS PR may filter the media types in a message, based on the Content-Type, to restrict which media types are allowed to be sent between the clients.
	M4: Support new Carrier Access Network (CAN)
	The service is dependent on a packet switched data network. As long as the CAN provides such support it may be used by the service.
	M5: Port the service to a different framework
Portability depends on the constraints of the new framework.	

### 3.3.4 Interoperability

Table 3-4: Evaluation of Interoperability quality attribute for TMMS Service based on GPRS

<b>I1: Send and receive messages with GSM SMS clients</b>
A TMMS message containing media types that a GSM SMS client doesn't support is converted to a plain text message containing an URI which the recipient can use to see the message. A TMMS message which only contains plain text isn't converted since the GSM SMS client supports text.

### 3.3.5 Reliability

Table 3-5: Evaluation of Reliability quality attribute for TMMS Service based on GPRS

<b>R1: Proxy-Relay routes messages at most once</b>
The TMMS PR uses the CSeq number in the SIP messages to ensure that a message is only routed once to each recipient.
<b>R2: Service stores messages until it has been delivered</b>
When a user has retrieved a message, the service deletes messageit from the message store.

### 3.3.6 Billability

Table 3-6: Evaluation of Billability quality attribute for TMMS Service based on GPRS

<b>B1: Service provider must get information about chargeable operations</b>
The TMMS PR sends charging information to a Charging Control which handle charging on behalf of the service provider.
<b>B2: Message charged at most once</b>
The cdr-id attribute of the cdr element identifies each CDR, and the Charging Control may use it to ensure that a CDR is only used once.
<b>B3: Sender may demand that the recipient is charged for the message</b>
If the sender includes a CDR in the body of the message sent to the TMMS PR the recipient(s) will be charged for the message. Also the sender may demand what amount the recipient(s) are charged for each media type included in the message.
<b>B4: Sender can not deceive receiver to pay for any messages</b>
Receiver is asked for charging confirmation by the TMMS PR.

## 4 TMMS Service based on IMS

In this chapter we will first give a brief introduction of the IMS, which is the technology our end-to-end mobile service in this chapter is based on. This introduction is based on [xxx], and a more detailed reading about IMS can be found in [22]. We continue this chapter with presenting how we have chosen to design our IMS based service, before we end this chapter with an evaluation of the service we have designed, based on the criteria and scenarios given in 2.5.

### 4.1 Overview

IMS as defined by 3GPP is a technology intending to merge the Internet with the cellular world. It is architected to enable and enhance IP based multimedia services such as the web, email, instant messaging, presence, and videoconferencing, and make it available everywhere through the Universal Mobile Telecommunications System (UMTS) network.

#### 4.1.1 Architecture

IMS uses PS technology to perform data communication. By using this technology, the data transfer will be much faster than if Circuit Switched (CS) technology were to be used.

The IMS services architecture gives the opportunity to deal with integrated services. It allows deployment of new services by operators and third party service providers. In this way a variety of services can be developed independently and at the same time utilize the common features of the IMS infrastructure.

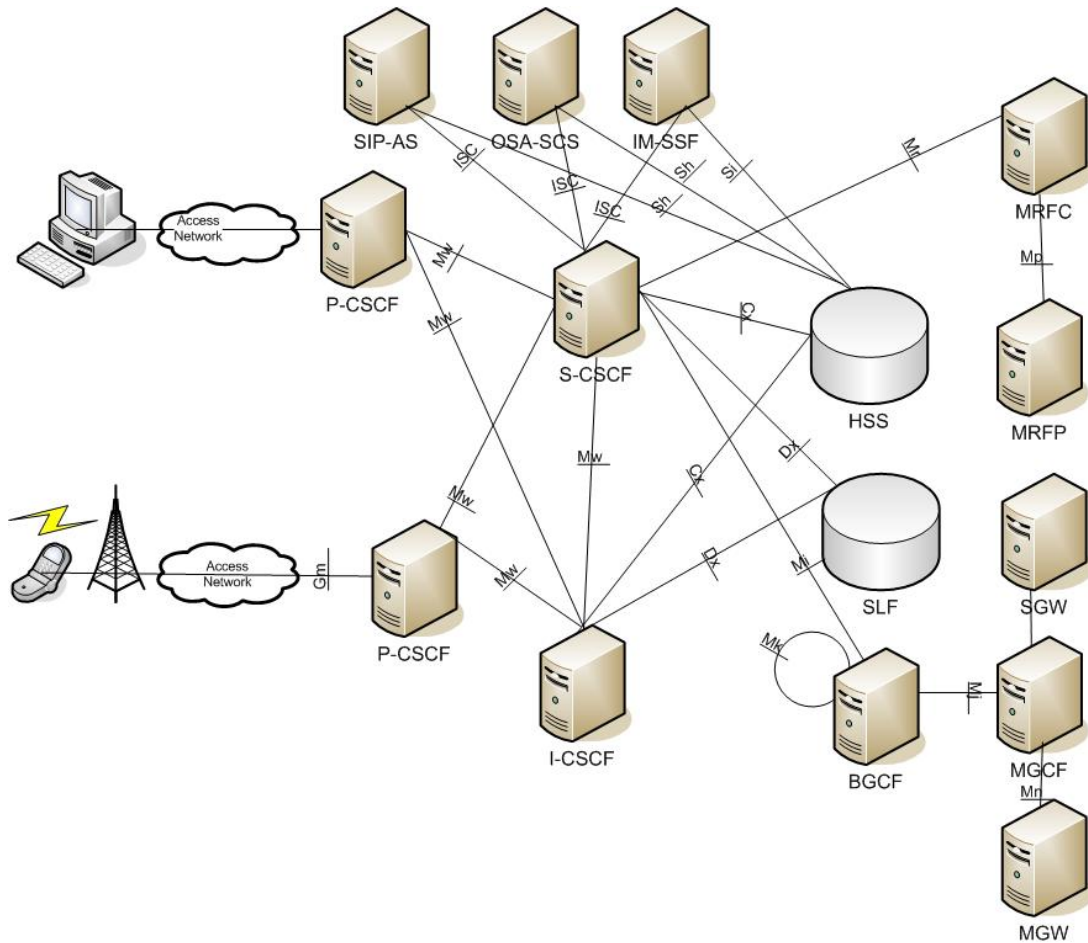


Figure 4-1: Overview of the 3GPP IMS architecture

The nodes of the IP Multimedia Core Network subsystems are described below.

#### 4.1.1.1 The Home Subscriber Server (HSS):

This node can be compared to the Home Location Register (HLR) in a GSM network. It is the master database for a given user and contains all the information about a user-related subscription data required to handle multimedia sessions. This is among others information concerning location, security, and user profile. This makes it possible for a user to find and communicate with other end users. An IMS network can contain more than one HSS. If this is the case, there is a need for another database called Subscriber Location Functions (SLF). This is a simple database that maps users' addresses to HSS.

#### 4.1.1.2 The Call/Session Control Function (CSCF):

This node is a SIP server and an essential node. It is needed for session management and support for Quality of Service (QoS) provisioning in the core network. Together with the HSS database the CSCF node is capable of routing any SIP requests/messages to devices and other networks. There are three different types of CSCF.

- **Proxy-CSCF (P-CSCF)**

The P-CSCF node is the first point of contact between the IMS terminal and the IMS network, and includes several functions. Some of these functions are the handle of emergency control, charging and resource

allocations, and security functions. The P-CSCF node has two main functions. The primary function is to be in the QoS Policy Enforcement Point within the visited network. Secondly it is responsible for providing the local control for emergency services.

- **Interrogating-CSCF (I-CSCF)**  
The I-CSCF node is a SIP proxy located at the edge of an administrative domain, and is therefore the first contact within the home network from a visited network. The I-CSCF node has an interface to both the HSS and the SLF database nodes, and retrieves user location information from the HSS and can then route the SIP messages to the appropriate destination. The I-CSCF node has the opportunity to encrypt the parts of the SIP messages that contain sensitive information of the domain. The I-CSCF node is usually located in the home network but can be located in the visited network in special circumstances.
- **Serving-CSCF (S-CSCF)**  
The S-CSCF node is the central node of the signaling plane. It is a SIP server but performs session control and can act as a SIP registrar as well. This node has the responsibility for the session management in the IMS network. One of the main functions is to provide SIP routing services, e.g. translation services between SIP URI and telephone numbers. The S-CSCF is always located at the home network.

### 4.1.1.3 The Application Servers (AS)

The AS is a SIP entity that hosts and executes services. The AS can operate in different modes depending on which service it is supposed to operate. These modes can be SIP UA or B2BUA. Different types of AS are SIP AS, OSA-Service Capability Server (SCS) and IP Multimedia Service Switching Function (IM-SSF). These types of AS all behave as SIP application servers towards the IMS network. The AS can be located in the home network or it can be in an external third-party network.

### 4.1.1.4 The Media Resource Function (MRF)

This node provides a source of media in the home network which can be used to do any sort of media analysis. The MRF node is divided into two parts, a signaling plane node and media plane node called MRF Controller (MRFC) and MRF Processor (MRFP) respectively. The MRFC acts like a SIP User Agent (UA), and has an interface towards the S-CSCF. It interprets information coming from the AS and the S-CSCF and controls the MRFP accordingly. The MRFP provides resources to be controlled by the MRFC and implements all the media-related functions. It also mixes the incoming media streams. The MRF is always located in the home network.

### 4.1.1.5 The Breakout Gateway Control Function (BGCF)

This node is basically a SIP server that handles routing functionality based on telephone numbers. It is only used in sessions that are initiated by an IMS UE and addressed to a user in a CS network. The main functions are to select an appropriate network where inter-working with the CS domains is to occur or to select an appropriate Public Switched Telephone Network/ CS (PSTN/CS) gateway, if inter-working is to occur in the same network where the BGCF is located.



#### 4.1.1.6 The PSTN/CS Gateway

This gateway makes it possible for IMS UEs to make and receive calls to and from a CS network. The PSTN/CS gateway can be decomposed into several functions:

- **Signaling Gateway (SGW)**  
The SGW interfaces the signaling plane in the CS network and performs lower layer protocol conversion.
- **Media Gateway Control Function (MGCF)**  
This node is the central node of the PSTN/CS gateway, and controls one or many Media Gateways (MGW) which is described below. It implements a state machine that does protocol conversion and maps SIP into either ISDN User Part (ISUP) over IP or Bearer-Independent Call Control (BICC) over IP. It is responsible for transferring and distributing media between participants in multimedia sessions.
- **Media Gateway**  
This gateway interfaces the media plane of the PSTN or the CS network. On one side the MGW can send and receive IMS media over RTP, and on the other side it uses one or more Pulse Code Modulation (PCM) time slots to connect to the CS network. It converts media from one format to another. It is important that the MGW performs its functions as quickly as possible so that delay is not added to transmission of the information.

#### 4.1.1.7 Home and visited Networks

A home network is the network where the infrastructure is provided by the network operator the subscriber is using. When the subscriber goes outside its home network it comes to a network with an infrastructure which is different from its home network infrastructure. This is the visited network. In order to use the visited network, the home network and the visited network need to have an agreement. This agreement may involve charging, and how to exchange accounting records.

#### 4.1.2 IMS Identifiers

IMS identifies its subscribers with allocating two different Public User Identities. This is the home networks responsibility. These identities are used by any user for requesting communication with other users. The Public User Identities can either be a SIP URI or a TEL URL. A subscriber gets two different because it is not possible to register a TEL URL in SIP. A TEL URL is also needed to make a call from an IMS UE to a PSTN phone, because PSTN numbers are represented only by digits.

```
sip:+1-123-456-7890@something.com;user=phone  
tel:+1-123-456-7890
```

An IMS subscriber is assigned a Private User Identity in addition to the Public User Identities. The Private User Identity is not on the same format as the Public User Identities. Instead it takes the format of a Network Access Identifier (NAI).

```
username@something.com
```

These identities are assigned by the home network, and are used for subscription identification, registration, authentication, authorization, administration and accounting purposes [38]. [xxx] It can be compared to the International Mobile Subscriber Identifier (IMSI) in GSM. In 3GPP release 6 a subscriber is assigned not only one Private User

Identity but with several. In 3GPP Release 6, the concept of Public Service Identities (PSI) is introduced. This is an identity located to a service hosted in an Application Server. PSI does not have an associative Private User Identity. This is because Private User Identity is used for authentication purposes and PSIs are not applicable to users.

### 4.1.3 Security

IMS provides a variety of security options. IMS is divided into two different parts, access security and network security.

#### 4.1.3.1 Access security

Users who want to use IMS services need to be authenticated and authorized before using the services. The authentication and authorization is done by using a REGISTER transaction. The S-CSCF downloads the information it needs from the HSS database while the P-CSCF and the IMS UE establish the necessary IP Security (IPSec)[39] security associations. The P-CSCF and the IMS UE establish two IPSec security associations, and to be able to establish these, the IMS UE and the P-CSCF have to agree on some parameters first. These IPSec security associations are used to protect the traffic between the IMS UE and the P-CSCF.

The security functions are not implemented directly on the IMS UE. Instead the security functions are to be found in a smart card which is inserted in the UE. One of the smart cards that exist is the Universal Integrated Circuit Card (UICC). The smart cards contain different applications. Among these applications we can mention IP-Multimedia Services Identity Module (ISIM), UMTS Subscriber Identity Module (USIM), and Subscriber Identity Module (SIM). With UICC it's preferred to use ISIM, but USIM is allowed in cases where the smart card is not upgraded to contain the ISIM application. The SIM application is not allowed when using IMS because of the poor security functions in the SIM application.

#### 4.1.3.2 Network security

Network security deals with security traffic between different security domains. As showed in the figure Figure 4-2 taken from [22] traffic going from one security domain to another has to go through two Security Gateways (SEG), one from each security domain. When sending traffic between the different SEGs the traffic is encapsulated and tunneled.

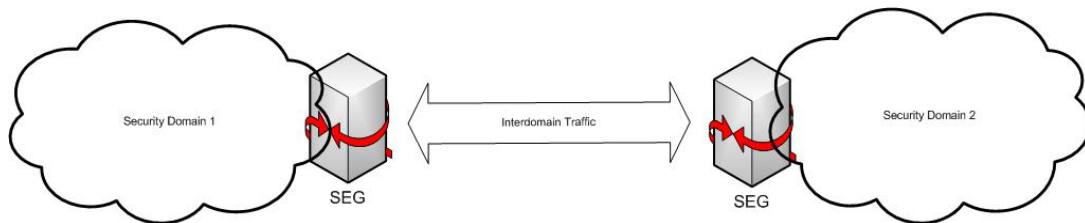


Figure 4-2: Traffic between two security domains

### 4.1.4 Charging

In PS networks without IMS the user will be charged based on the amount of bytes the user sends. In a videoconferencing e.g. this amount of bytes will be huge and the operator will not be able to know the contents of the bytes. Therefore it could be expensive for the user using such type of services. Using IMS the operator will be able to

know the content of the bytes and be able to decide an appropriate charging which also will benefit the user. This is taken care of by the AAA protocol. One of the main functionalities this protocol has is that the accounting part is dealing with billing among other things. It provides a mechanism to transfer charging information from the IMS nodes to the network operator's chosen Billing System. The charging functions are based on the IMS network nodes reporting accounting information upon reception of various SIP methods or ISUP messages, as most of the accounting relevant information is contained in these messages.

In IMS there are two different charging models. These are online charging and offline charging.

#### 4.1.4.1 Online charging

In [40] online charging is also called credit based charging and is used to charge prepaid services. Online charging uses Credit-Control-Request (CCR) and Credit-Control-Answer (CCA) messages. The requests perform rating of the IMS service and reserves units on the users accounts. The CCR messages are sent to the Online Charging System (OCA) which responds with a CCA message. These messages carry Attribute Value Pairs (AVP) There are three types of online charging in IMS. These are Immediate Event Charging (IEC) and Event Charging with Unit Reservation (ECUR) , and Session Charging with Unit Reservation (SCUR).

#### 4.1.4.2 Offline charging

Offline charging is used for users who only pay for their services periodically, e.g. once a month. Offline charging is based on IMS nodes that report accounting information. This is done by sending Accounting-Requests (ACR) and the corresponding Accounting-Answers (ACA). These messages are based on the Diameter protocol and carry AVP just like online charging. The ACR messages are sent to the Charging Data Function (CDF) which responds with an ACA.

#### 4.1.5 WAP PUSH

This background information is based on [[41], [42]]

#### 4.1.6 Architecture

With the term push we mean that a server push information to a mobile terminal without any requests from the terminal. We can therefore say that the push technology is server-initiated.

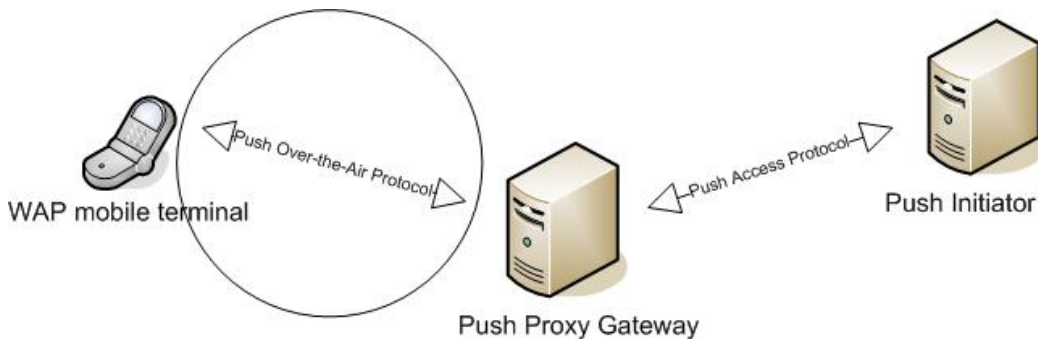


Figure 4-3 WAP Push Architecture

From figure Figure 4-3 we can see that in WAP the push technology has three important nodes. These are the Push Initiator (PI), the Push Proxy Gateway (PPG) and a WAP mobile terminal. A PI pushes content and delivery instructions to a PPG using the Push Access Protocol (PAP). Further on the PPG pushes the content to the right mobile terminal using the WAP Push Over-the-Air (OTA) protocol. This protocol runs on top of either HTTP or Wireless Session Protocol (WSP) and are of that reason called OTA-HTTP and OTA-WSP. Since our services use OTA-HTTP, we will focus on this. Further readings about the OTA-WSP can be done in [[41], [42]]. OTA-HTTP uses as the name imply HTTP for communication between the PPG and the mobile terminal. It uses HTTP POST to deliver content to the terminals. OTA-HTTP is a connection-oriented service and sets up a TCP connection before any content can be pushed from the PPG and the mobile terminal. There are two different methods to set up a TCP connection. The first one is called PPG Originated TCP (PO-TCP), and in this case a TCP connection is established by the PPG towards the terminal. In this method it is assumed that the terminal has IP connectivity with the network and that this IP address is known to the PPG. The other method is Terminal Originated TCP (TO-TCP). In this method the TCP connection is established from the terminal towards the PPG.

Summarized the core features of the OTA-HTTP are:

- IP connectivity procedure
- TCP connection procedure
- Registration – PPG becomes aware of the terminal’s current capabilities and preferences
- Content push – Delivery of content from PPG to terminals using HTTP POST method.

In addition of mentioned features, OTA-HTTP provides mechanisms for identifying and optionally authenticates both PPG and the mobile terminal during registration and delivery.

### 4.1.7 Session Initiation Application

The Session Initiation Application (SIA) is a terminal side application which allows a PPG to establish a push session or an active TCP connection using a special bearer. This application needs to be supported by both the terminal and the PPG. While the terminal has the SIA, the PPG has Session Initiation Requests (SIR) which the terminals listens to. In OTA-HTTP the SIR mechanism is used in cases like when the terminals IP address it not known by the PPG, and in cases when the PPG cannot activate the desired bearer.

### 4.1.8 Security

It is important that the PI gets authenticated and it exist several methods to do so. Among these are the uses of session-level certificates like TLS and Secure Sockets Layer (SSL), HTTP Authentication, a combination of those two mentioned and trusted networks. If a non-trusted or a non-authenticated PI tries to push content to a PPG, the PPG has the ability to perform filtering and access control to discard the pushed content.

## 4.2 Design

### 4.2.1 Introduction

The service acts as a SIP B2BUA in a SIP AS, connected to a S-CSCF. Also it... how does it communicate with GSM?

It is designed to take advantage of as many of IMS' features as possible.

### 4.2.2 Deployment view

The figure below depicts how these components are deployed in an IMS network.

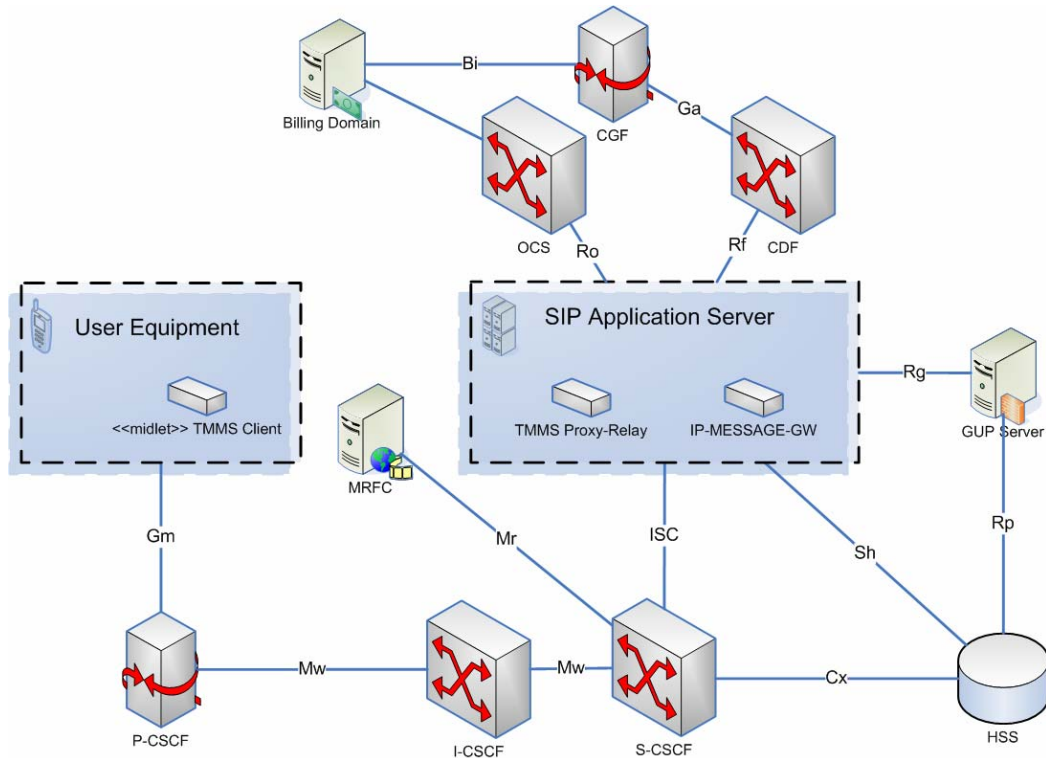


Figure 4-4: Deployment diagram for IMS solution

The next sections will describe in detail how the TMMS Client and the TMMS PR applications are deployed to their hosting environments.

#### 4.2.2.1 TMMS Client

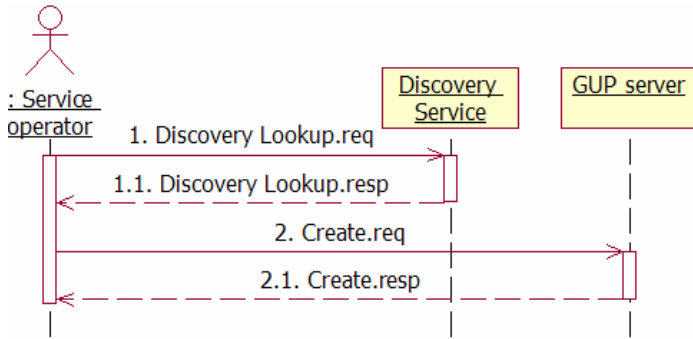
The client may use the standard JAD-based deployment, as explained in 3.2.2.1. In addition a push-based deployment method is also possible, because the TMMS PR can obtain the client's address from the Via-header of its initial SIP REGISTER request.

The TMMS PR acts as a PI and requests version information for the TMMS Client from the client. If it is not installed or a newer version is available then TMMS PR will send a push request to the client.

#### 4.2.2.2 TMMS Proxy-Relay

The service is deployed to a SIP Application Server. For the service to be available to a user its Service Profile GUP component and initial Filter Criteria (iFC) must be updated to include the TMMS PR.

The figure below shows how the TMMS PR updates these elements.



**Figure 4-5: Create service profile of PUI**

First the service operator uses the discovery service to get the address of the GUP Server and an authorization token for the Create request. Next the Create request is sent to the GUP Server with the authorization token. Service information included in the Create request includes service information:

- Authorization token, from the Discovery Service
- PUI to give service access to.
- iFC component for the service.

The specification for this is currently a bit unfinished, so a complete example of this can not be included here. An example of the iFC added to the Service Profile is given in Appendix F:InitialFilterCriteria.xml. The trigger point used in the example has two Service Point Triggers (SPT) in an AND condition, which are explained below.

- **SIP Method MESSAGE**  
The service will only trigger on MESSAGE requests.
- **Terminating side of the session**  
Since the service will only store and forward incoming messages it is only required to be part of the session on the terminating side. IMS routing will take care of the originating side.

Note that this will enable the service for all MESSAGE requests, which is not always desired. By adding a header like "X-TMMS-Message" and a SPT which requires this header to be included in the request, the service will only be invoked for these special MESSAGE requests.

### 4.2.3 Component view

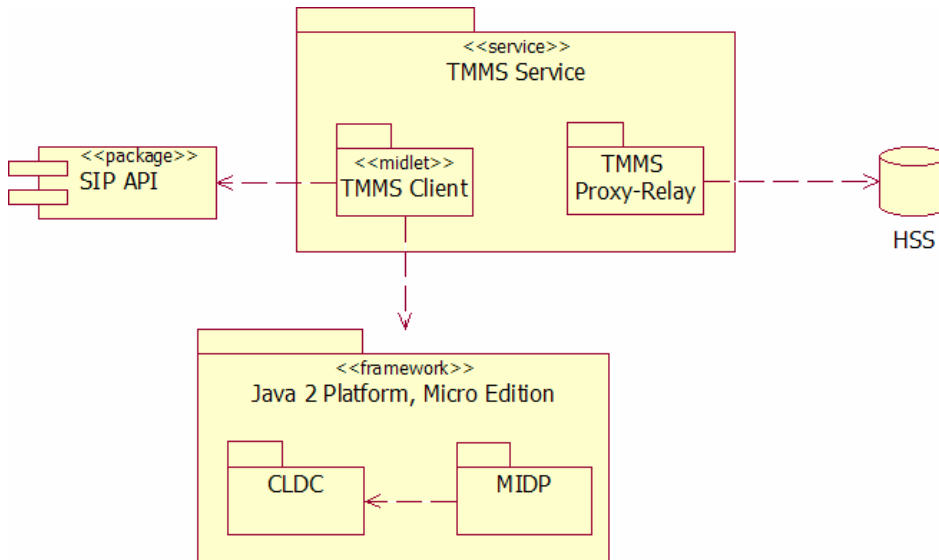


Figure 4-6 Component view for IMS based service

The figure above gives a high level overview of the components of the system, and shows the components which TMMS Service is dependent on. In the following subsection the internal components of TMMS Client is described above.

The internal components of TMMS PR are not described here since that is out of scope for this report.

#### 4.2.3.1 TMMS Client

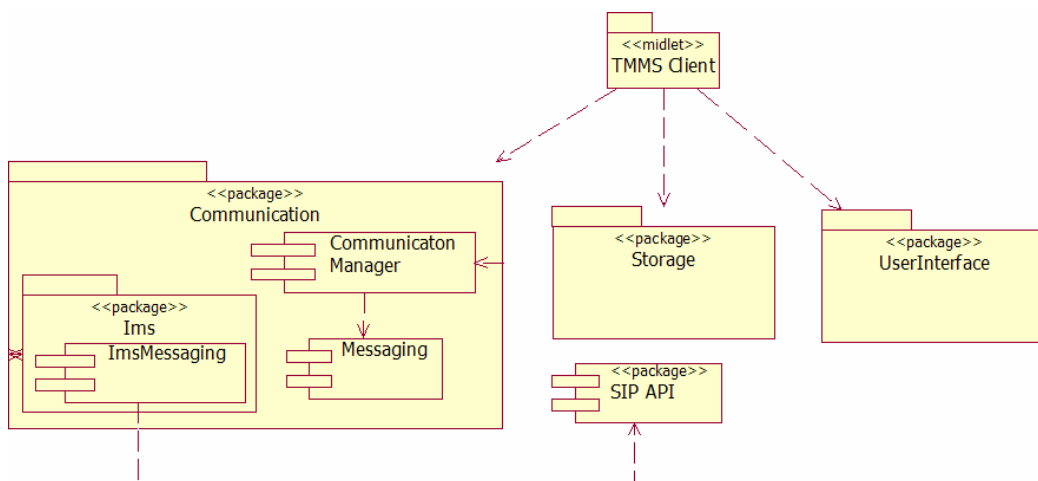


Figure 4-7: Component view for TMMS Client

The TMMS Client has a *TMmsClient.Communication.Ims* package which contains its own implementation, *ImsMessaging*, of the *TMmsClient.Communication.Messaging* interface. *ImsMessaging* uses the SIP API component to communicate with the TMMS PR.

### 4.2.4 Security

The service trusts IMS to provide a secure environment. Only authorization for service usage is dealt with by the service itself.

### 4.2.5 Scenario Implementations

In the sequence diagrams below the message paths have been simplified, to only include communication relevant to the service. Nodes which are not included are P-CSCF, I-CSCF and I-CSCF (THIG). This communication is described in [38].

#### 4.2.5.1 S01: TMMS Client Registration

To get notifications about new messages the client needs to subscribe to the “newmessage-alert” event.

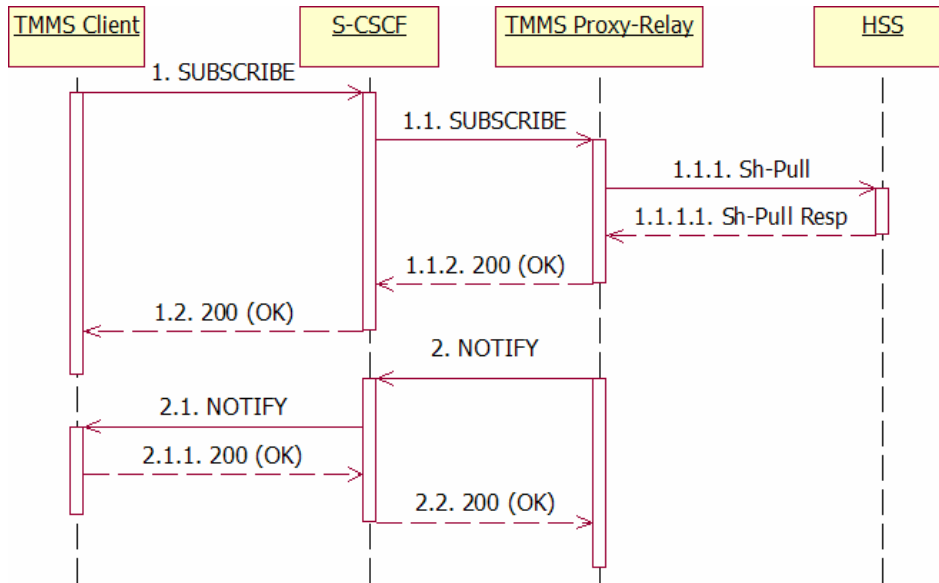


Figure 4-8: TMMS Client registration

When the TMMS PR receives the SUBSCRIPTION request it tries to download its Repository Data from the HSS. This data is used to authorize the request, and if no data is found then the request will not be authorized. TMMS PR sends a 403 (Forbidden) response and a 403 (Forbidden) response.

According to [29] the notifier must immediately send a NOTIFY message to the user with the current state after a successful subscription. This message will contain a body with URIs to all new messages for the client, if any. The URIs are retrieved from the Repository Data pulled from the HSS in step 1.1.1. How the client handles the NOTIFY message is discussed in 4.2.5.3.



#### 4.2.5.2 S02: Send MM

The client composes a MM and sends it to the recipient(s). Standard IMS routing principles will take care of routing the MM to the recipient(s). This is described in clause 5.16 in [38]. Here it is assumed that a list server is already present in the network to allow clients to send a message to multiple recipients.

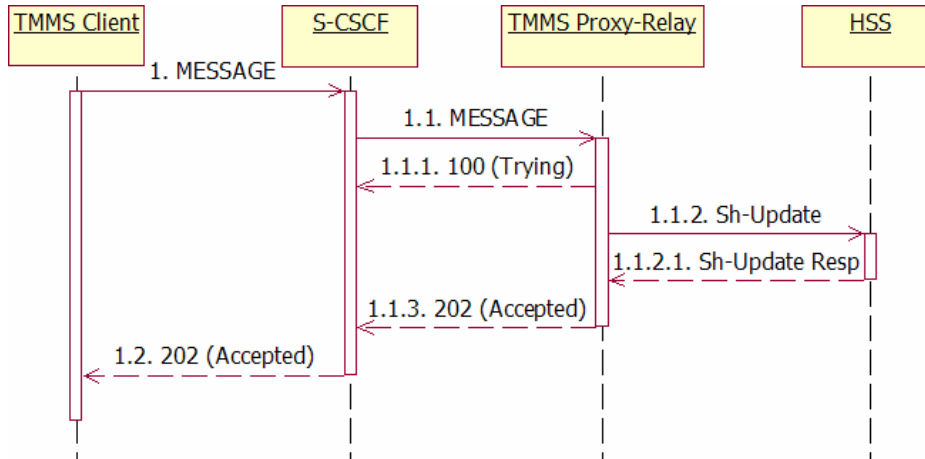


Figure 4-9: TMMS Client Sends MM

After the TMMS PR has sent the response back to the originator it stores the message in the Repository Data section of the recipient's service profile. Then a newmessage-alert notification is sent to all subscribers matching the Request-URI of the original SIP MESSAGE request.

For interoperability with GSM SMS a IP Short Message Gateway (IP-MESSAGE-GW)[53] node may be present in the network, which will relay messages between IMS and GSM SMS clients..

#### 4.2.5.3 S03: New message notification

Messages routed to the client by IMS will be picked up by the TMMS PR on the terminating side of the network. Then a notification event is produced and sent to subscribers. The subscribers may then choose to either immediately download the message or to download it later. In the figure below this is depicted until where the scenario forks.

The next subsections describe how the NOTIFY message is handled.

**4.2.5.3.1 S03.01: TMMS Client Receives MM (Immediate Retrieval)**

In this scenario the client downloads the MM immediately from the TMMS Message Store using the HTTP protocol.

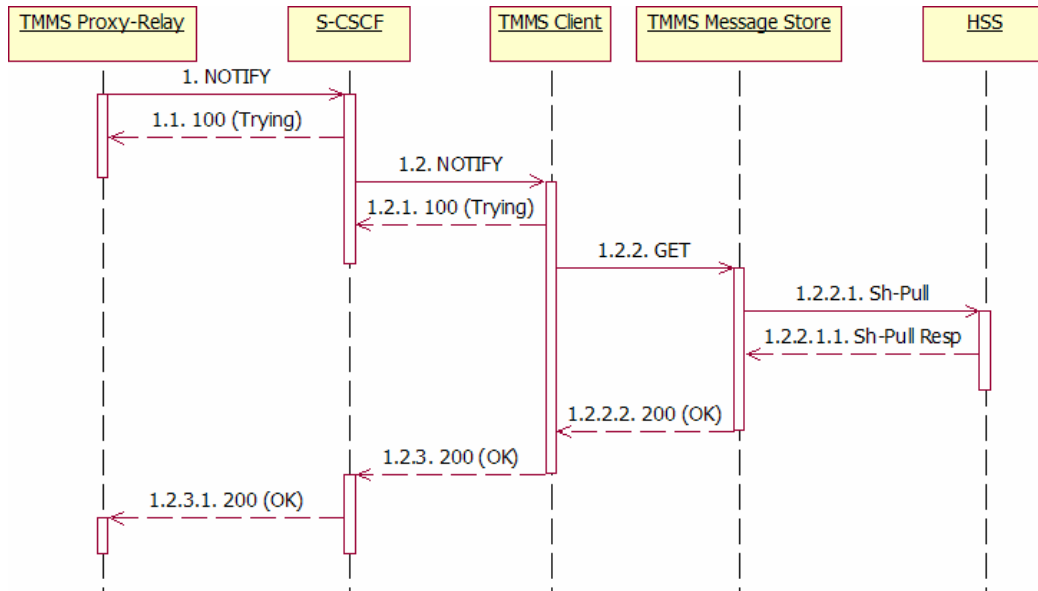


Figure 4-10: TMMS Client retrieves MM immediately

**4.2.5.3.2 S03.02: TMMS Client Receives MM (Deferred Retrieval)**

In this scenario the client defers retrieval of the MM by returning 202 (Accepted) back to the TMMS PR.

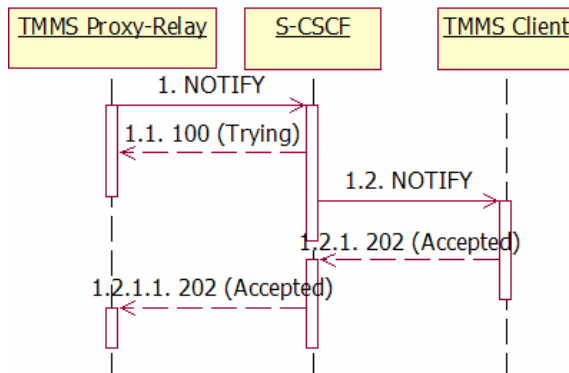


Figure 4-11: TMMS Client defers retrieval of MM

#### 4.2.5.4 S04: Deferred MM Retrieval

If the client has deferred retrieval of a MM then it can retrieve it by sending a HTTP GET request to the TMMS Message Store. The Request-URI for the GET request is the URI which was passed to the client from the TMMS PR in the newmessage-alert notification. If the message exists then the TMMS Message Store will return it in the body of the response message.

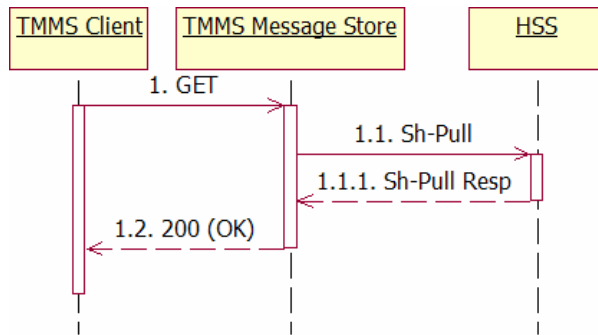


Figure 4-12: Deferred MM retrieval

### 4.2.5.5 S05: Streaming Retrieval of MM

Instead of using HTTP to download the MM from the TMMS Message Store, the client can instead request that it is streamed to it. This scenario is shown in Figure 4-13. The number in brackets, e.g. [1], indicates which SIP dialog the request or response is part of.

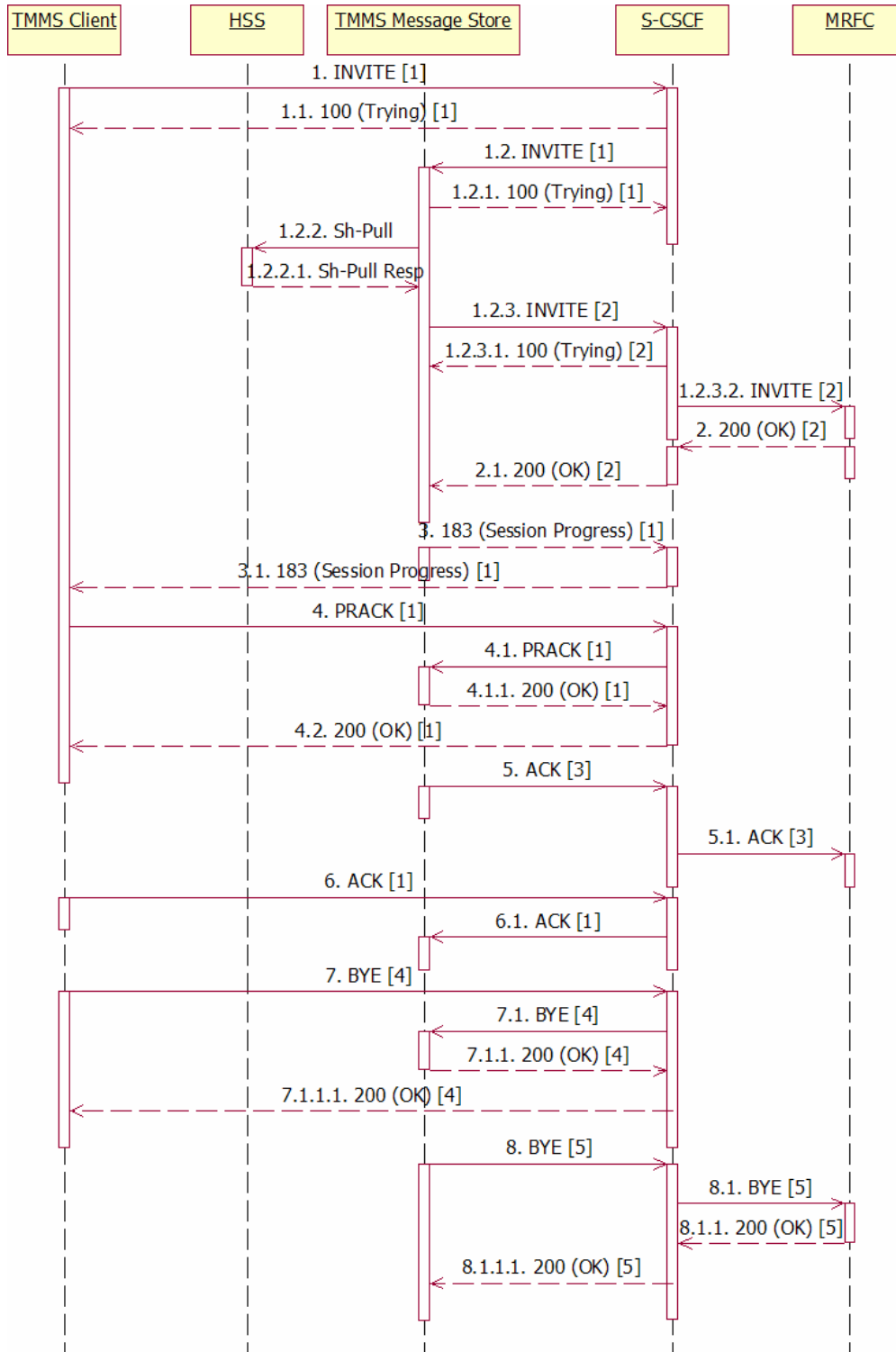


Figure 4-13: Streaming media retrieval of MM

The client sends a INVITE message to the TMMS Message Store with a SDP message in the body. This SDP contains information such as which message is to be streamed and which codecs should be used for the session. This is marked as dialog #1 in the figure, and is routed by the S-CSCF to the TMMS Message Store.

Next the TMMS Message Store pulls the message from the HSS. Then, in dialog #2, the TMMS Message Store sends an INVITE request to the MRFC with the SDP included. Then the MRFC and the TMMS Client will negotiate which codecs to use for the streaming session before they reserve resources for it.

After the resources have been reserved the TMMS Message Store sends an ACK request to the MRFC, in dialog #3. This starts the streaming session. The messages exchanged during the streaming session are not included here.

When the client is finished it sends a BYE request, which is dialog #4. In dialog #5 the TMMS Message Store sends a BYE request to the MRFC, so it can release the resources reserved earlier.

At this point the TMMS Message Store may delete the MM, since the client has viewed it.

#### 4.2.5.6 S06: TMMS Client Unregistration

The TMMS Client unregisters from the service by unsubscribing from the newmessage-alert event. This is done by sending the same SUBSCRIBE request as in S01: TMMS Client Registration, with the Expires header set to 0.

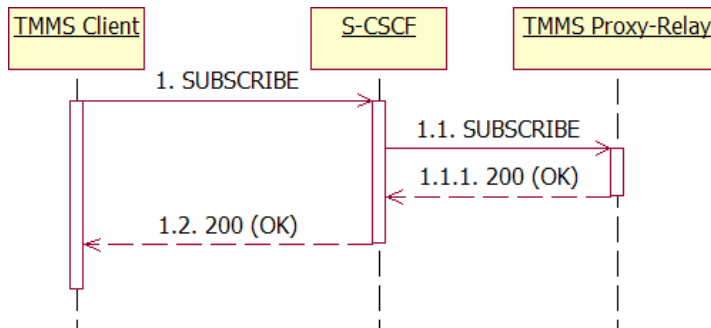


Figure 4-14: TMMS Client unregistration

#### 4.2.6 Charging of Service Usage

The only chargeable operation is if the client responds with 200 (OK) or 202 (Accepted) to a NOTIFY request. If a P-Charging-Vector header is included in the MESSAGE request then the TMMS PR will honor that when issuing charging. This can be used by the sender to indicate that the recipient shall be charged for the message.

Similar to the GPRS charging model, see 3.2.6.1, the NOTIFY request will include the CDR if available.

##### 4.2.6.1 Online Charging

After the client receives the NOTIFY request it can either respond with a success response, such as 200 (OK) or 202 (Accepted), or with an error response. If the error response is 606 (Not Acceptable) that means the client is not willing to be charged for the message.

The success scenario is depicted in below and the error scenario in Figure 4-15.

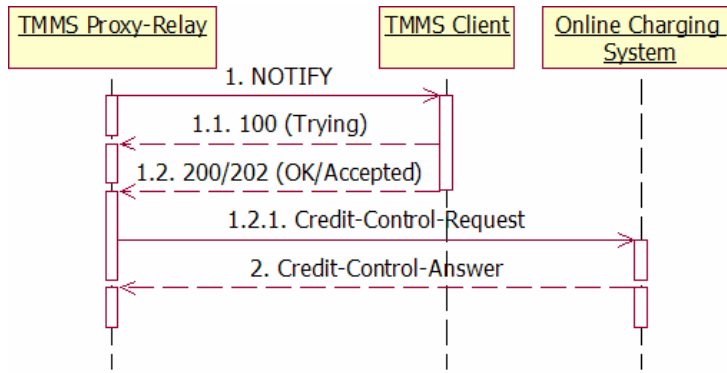


Figure 4-15: IMS online charging, client accepts to be charged

TMMS PR uses Immediate Event Charging (IEC) to do charge the client. In the CCR it will have the CC-Request-Type AVP set to EVENT\_REQUEST and the Requested-Action AVP set to DIRECT\_DEBITING. The TMMS PR will then wait until OCS answers the request. The client can now retrieve the message from the TMMS Message Store using the URI provided in the NOTIFY request message.

#### 4.2.6.2 Offline Charging

For offline charging the TMMS PR sends an Accounting-Request, with the Accounting-Record-Type AVP set to EVENT\_RECORD, to the CDF after each scenario has finished.

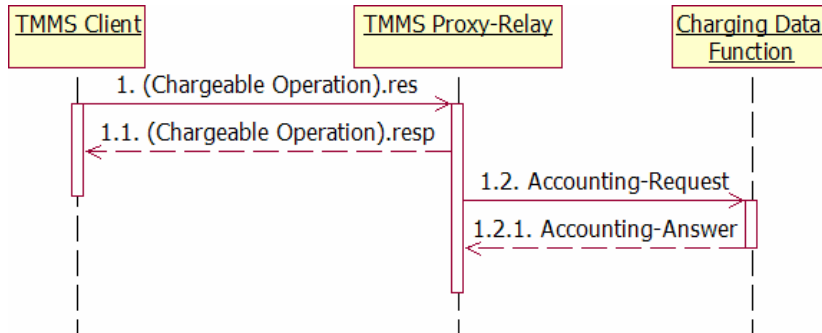


Figure 4-16: Offline charging in IMS

### 4.2.6.3 Charging Request Not Accepted

This subsection applies to both section 4.2.6.1 and section 4.2.6.2.

If the client returns 606 (Not Acceptable) that means the client does not accept to be charged. Hence the TMMS PR sends a Sh-Update request to the HSS which deletes the message. The figure below shows this.

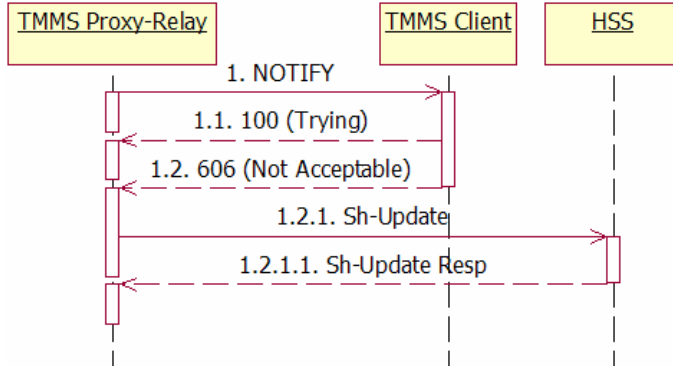


Figure 4-17: IMS online charging, client declines to be charged

### 4.3 Evaluation

#### 4.3.1 Security

Table 4-1: Evaluation of Security quality attribute for TMMS Service based on IMS

Availability	S1: Service is available for legitimate use and sustains Denial of Service attacks
	Service gets available to users through updating their Service Profile GUP and iFC to include TMMS PR
Integrity	S2: Data is not altered during transmission
	IMS uses IPSec and the ESP protocol to ensure integrity.
Authentication	S3: Sender and receiver confirms that they are who they claim to be
	IMS Specific. The service must trust that the IMS network already has authenticated the users of the service.
Authorization	S4: Only authorized users get access to the service
	Through Sh interface the service can check with HSS if a client is authorized.
Confidentiality	S5: Only sender and the intended destination part is able to know the content of the message
	IMS uses IPSec to ensure confidentiality.



### 4.3.2 Usability

Table 4-2: Evaluation of Usability quality attribute for TMMS Service based on IMS

<b>Terminal Integration</b>	U1: Distribution of the client application
	Client can choose between manually and automatically retrieving of the client application.
	U2: Which parameters are necessary for the client to communicate with the Proxy-Relay
	Parameters needed are: The TMMS PR PSI, the port number where the client will receive NOTIFY messages.
	U3: Address handling (GSM SMS vs. TMMS addresses)
	IMS specific. SIP URIs are used to address TMMS Clientsclients, and TEL URIs are used to address GSM SMS clients. Only the PRproxy-relay must treat these different.
<b>Registration</b>	U4: Client registration
	Client must subscribe to the newmessage-alert.
	U5: Client unregistration
	Client unregisters to the service by unsubscribing from the newmessage-alert.

### 4.3.3 Modifiability

Table 4-3: Evaluation of Modifiability quality attribute for TMMS Service based on IMS

<b>Scalability</b>	M1: Add more servers to achieve better scalability
	IMS helps in some way Since the TMMS PR is addressed through DNS is it possible to use normal DNS based network load balancing techniques to add more servers.
<b>Change Service</b>	M2: Extend the service to support new media types
	The service uses MIME types to describe the media types of the message body. Only the client application must be upgraded to present new media types to the user.
	M3: Restrict the media types supported by the service
	The TMMS PR may filter the media types in a message, based on the Content-Type, to restrict which media types are allowed to be sent between clients.
	M4: Support new Carrier Access Network (CAN)
	IMS specific.
	M5: Port the service to a different framework
Difficult because of the IMS framework.	

#### 4.3.4 Interoperability

Table 4-4: Evaluation of Interoperability quality attribute for TMMS Service based on IMS

I1: Send and receive messages with GSM SMS clients
If A TMMS message contains media types that a GSM SMS client doesn't support, an ENUM translation needs to be performed. The TMMS gets routed based on the translation result. A TMMS message which only contains plain text isn't converted since the GSM SMS client supports text.

#### 4.3.5 Reliability

Table 4-5: Evaluation of Reliability quality attribute for TMMS Service based on IMS

R1: Proxy-Relay routes messages at most once
IMS supports forking so messages can be routed to more than one destination.
R2: Service stores messages until it has been delivered
(The message store does not confirm if the HTTP response with the message is received by the client). When user has retrieved message, the service deletes message from message store.

#### 4.3.6 Billability

Table 4-6: Evaluation of Billability quality attribute for TMMS Service based on IMS

<b>B1: Service provider must get information about chargeable operations</b>
IMS specific. The TMMS PR sends charging information to the CDF in offline charging and OCS in online charging.. CDF and OCS handle charging on behalf of the service provider.
<b>B2: Message charged at most once</b>
IMS specific. The IMS network elements mark the ACR/CCR messages with a T-flag if duplicate messages are to be sent. A check is done based on Session-ID and CC-Request number AVP pairs.
<b>B3: Sender may demand that the recipient is charged for the message</b>
Yes. Uses private charging field.
<b>B4: Sender can not deceive receiver to pay for any messages</b>
No support.

## 5 TMMS Service based on OSA API

In this chapter we will first give a brief introduction of OSA. A more detailed reading about OSA can be found in [43]. The end-to-end mobile service designed in this chapter is based on OSA API. We continue this chapter presenting to service, before we end this chapter with an evaluation of the service based on the criteria and scenarios in 2.5.

### 5.1 Overview

Earlier the circuit switched telecommunication networks were separate from the packet switched computer networks. In the last few years this has changed and these networks have converged. This makes it possible to create new kinds of service, and has brought forward a need for a standard interface to create services. With no standard interface it becomes harder to deploy a service in networks from different providers, and in computer networks it has been common to have such a standard interface, for example the Portable Operating system Interface (POSIX) and the Common Object Request Broker Architecture (CORBA), but this is quite new for telecommunication networks. The OSA interface is jointly developed by 3GPP, The Third Generation Partnership Project 2 (3GPP2), European Telecommunication Standards Institution (ETSI) and the Parlay Group to be such an interface. Through the OSA interface applications get access to network functionality, and makes applications independent from the underlying network technology.

From [TS23.198-100], the OSA consists of three parts:

- **Applications**  
Implemented in one or more Application Server(s) and makes use of the other parts. The MMS as discussed later in this paper is one such application.
- **Framework**  
The framework provides basic functionality for applications so they can use the service capabilities available to them.
- **Service Capability Servers**  
Gives access to some network functionality for the application to use.

To get access to the network the application must perform three steps. First it must contact the framework, then authenticate with the framework and if authentication succeeds it may authorize with the network and starts discovering which Service Capability Features (SCFs) are available.

#### 5.1.1 Service Capability Features

Network functionality offered to applications is described through SCFs. A network provider may only support a subset of the SCFs described in the OSA API specification [TS29.198], and the application can find out which are available through discovery. This also makes it possible to easily add new services without breaking old applications.

Several SCFs are defined: Call Control, Data Session Control, Terminal Capabilities, Mobility, User Interaction, Charging, Account Management, Presence, and Multimedia Messaging.

### 5.1.2 Charging

OSA supports different kinds of charging for services, and is based on a subscriber's account. An application can charge the subscriber's account for both e-commerce (like shopping of books, food and ringtones) and service usage (like chatting and call forwarding).

From [44], assuming that the underlying network supports the features, the Charging SCF can be used by a service to:

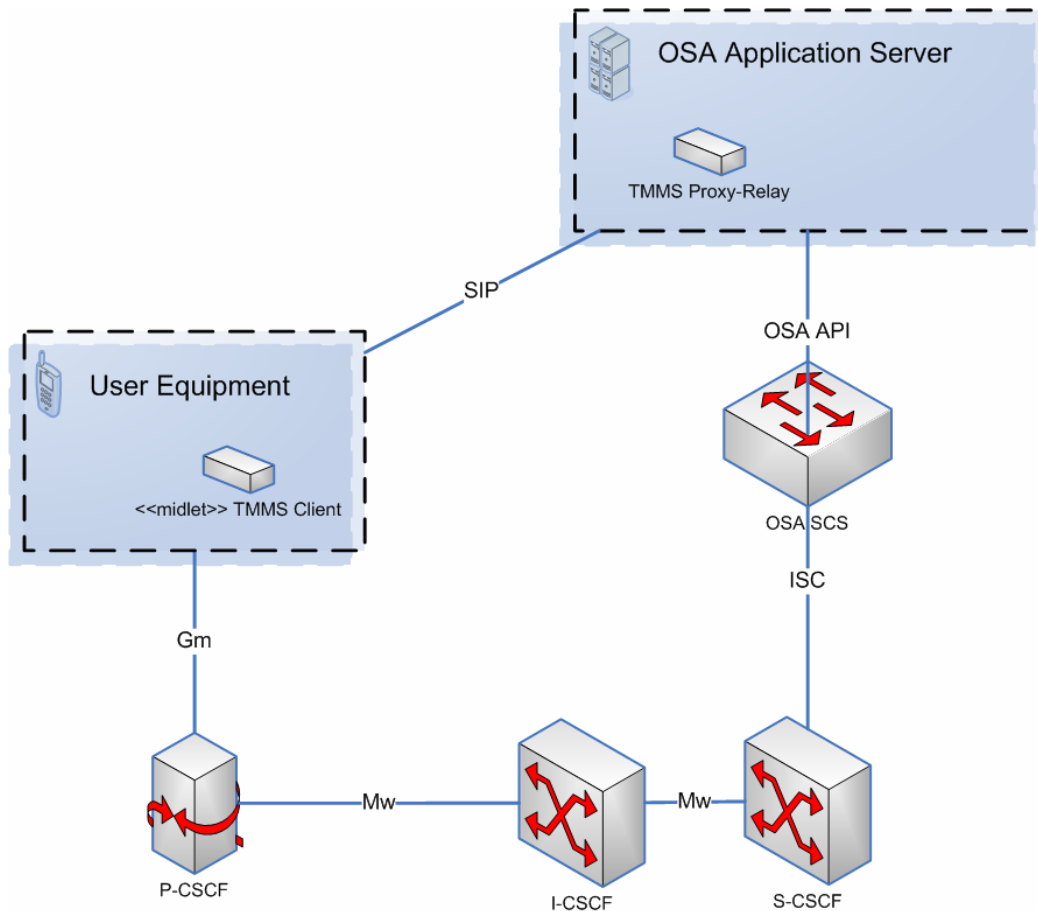
- Check, if – for the service to be provided by the application – the charge is covered by the subscribers account or credit limit.
- Reserve – for the service to be provided by the application – a charge in the subscribers account, that can be deducted from the account after service delivery.
- Deduct an amount from the subscriber's account.
- Release a reservation acquired earlier.
- Add non-monetary units to a subscriber's account.
- Deduct non-monetary units from a subscriber's account.
- Reverse a completed charge transaction, e.g. after repudiation.

## 5.2 Design

### 5.2.1 Introduction

### 5.2.2 Deployment view

The figure below depicts how the nodes are deployed in an IMS network.



**Figure 5-1: Deployment of OSA API based service in IMS network environment**

The *OSA API* interface of the OSA AS is used by TMMS PR to use OSA SCFs, such as the Charging SCF which is used for charging of service usage. The *SIP* interface is used to communicate with the client.

The next sections will describe in detail how the TMMS Client and the TMMS PR applications are deployed.

#### 5.2.2.1 TMMS Client

The TMMS Client is deployed as in the GPRS based solution, as explained in 3.2.2.1.

#### 5.2.2.2 TMMS Proxy-Relay

The TMMS PR is deployed at an AS. The kind of AS and how it is deployed there is an implementation detail which is out of scope for this report. This AS can either be located in the client's home network or at an external service platform.

Before the TMMS PR can access OSA services it needs to establish a service agreement with the OSA service provider. The procedure for establishing a service agreement may include an offline provider specific part which is not included here. There are multiple steps which needs to be executed to establish the service agreement which is described in the subsections below.

**5.2.2.2.1 Initial Access**

First TMMS PR must initialize with the OSA Framework and agree on which API version which should be used. Figure 5-2 shows this part of the scenario.

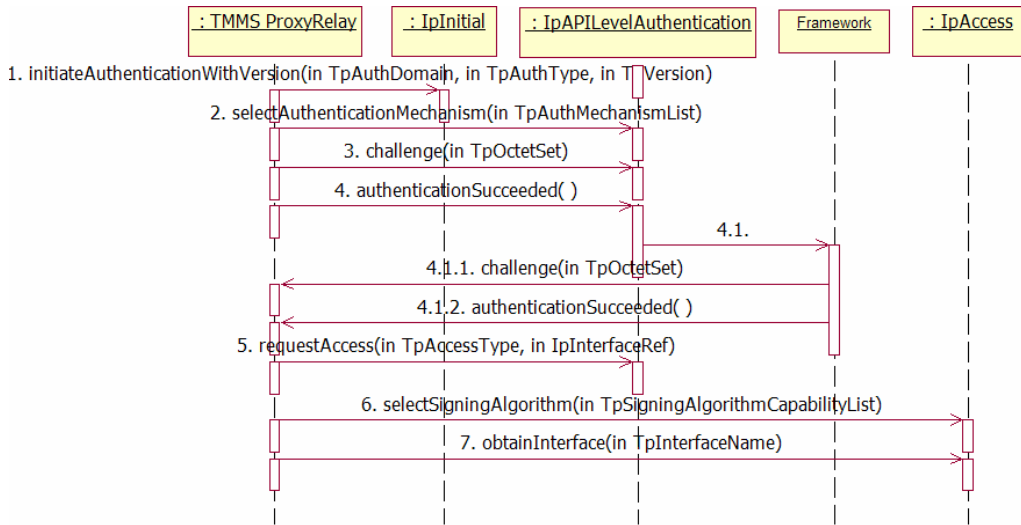


Figure 5-2: Establish service agreement - Initial Access

**5.2.2.2.2 Discover Service Agreement Management Interface**

Next part of the scenario is to use the interface obtained from IpAccess, in step 7 of Figure 5-2, to discover the IpServiceAgreementManagement interface. Figure 5-3 shows this step.

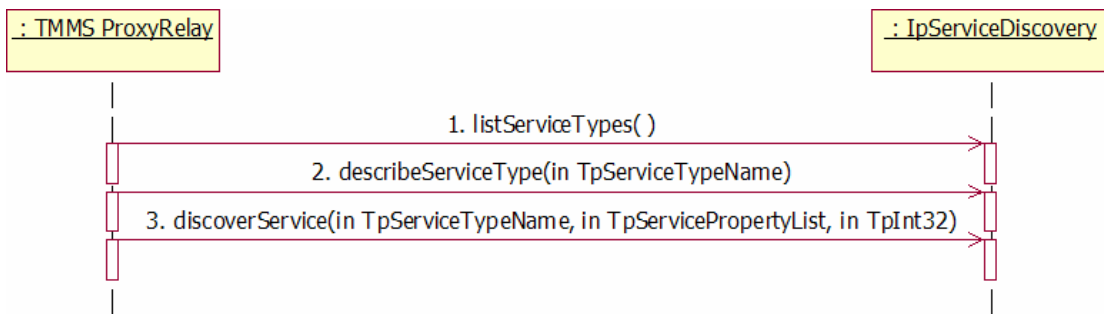


Figure 5-3: Establish service agreement - Discover Service Agreement Management interface

**5.2.2.2.3 Sign Service Agreement**

Finally, by using the interface returned by IpServiceDiscovery in step 3 of Figure 5-3, TMMS PR can sign the service agreement through the IpServiceAgreementManagement interface.

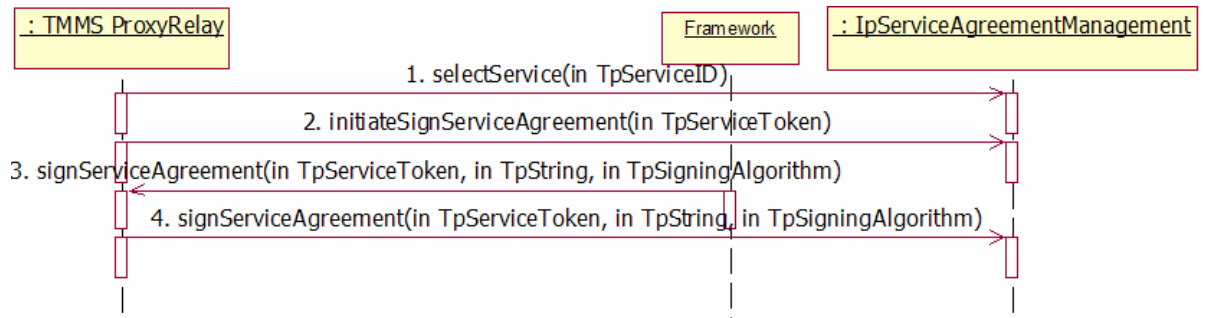


Figure 5-4: Establish service agreement – Sign service agreement

Now TMMS PR is authenticated and can use the other SCFs provided by the Framework which it is authorized to use.

### 5.2.3 Component view

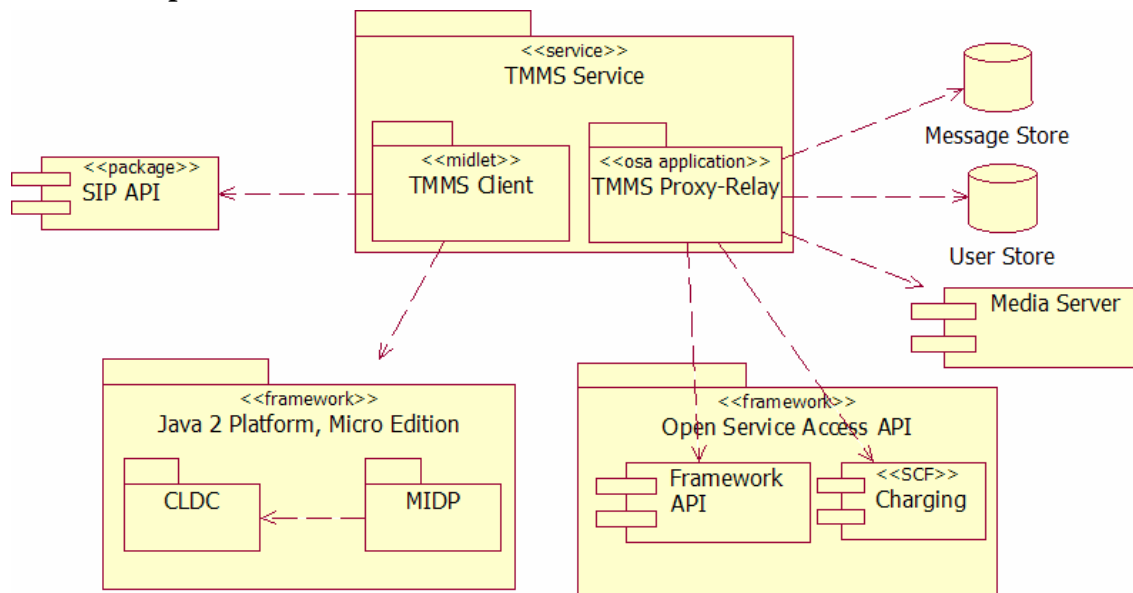


Figure 5-5: Component view

The figure above gives a high level overview of the components of the system. In the next two subsections the internal components of TMMS Service will be described.

### 5.2.3.1 TMMS Client

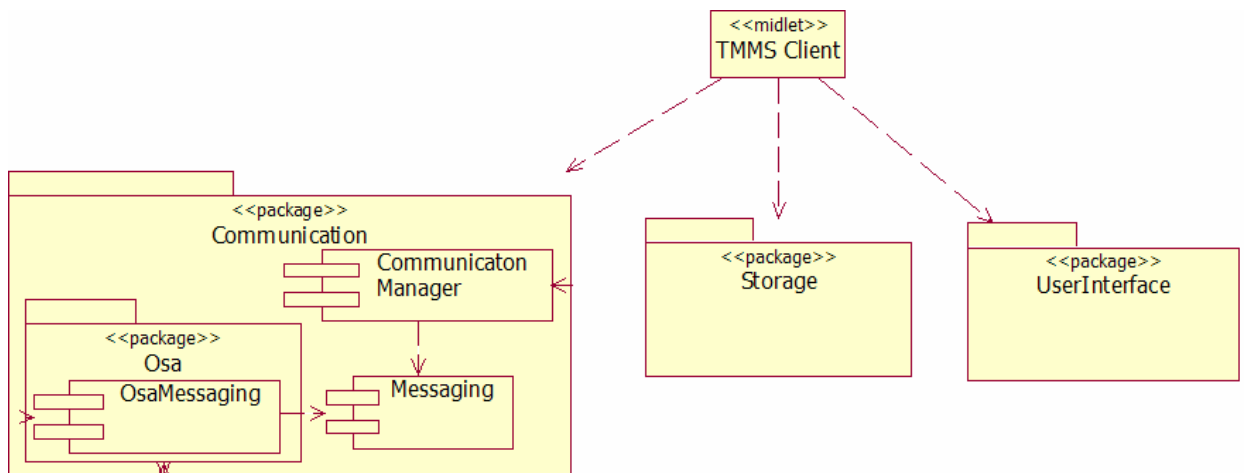


Figure 5-6: TMMS Client components

The TMMS Client has a *TMmsClient.Communication.Osa* package which contains its own implementation, *OsaMessaging*, of the *Messaging* interface from the *TMmsClient.Communication* package. This component is used to communicate with the TMMS PR.

### 5.2.3.2 TMMS Proxy-Relay

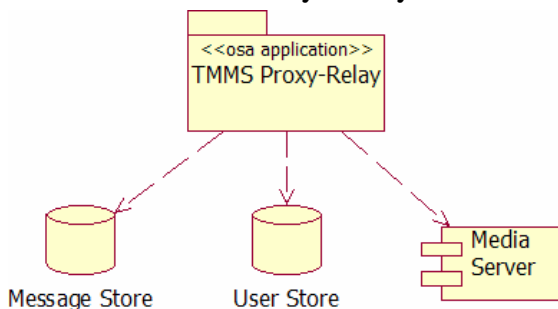


Figure 5-7: TMMS Proxy-Relay components

## 5.2.4 Security

## 5.2.5 Scenario Implementation

### 5.2.5.1 S01: TMMS Client Registration

This scenario is exactly the same as for the GPRS based service. Please see 3.2.5.1 for the description.

### 5.2.5.2 S02: Send MM

This scenario is exactly the same as for the GPRS based service. Please see 3.2.5.2 for the description.



### 5.2.5.3 S03: New message notification

This scenario is exactly the same as for the GPRS based service. Please see 3.2.5.3 for the description.

### 5.2.5.4 S04: Deferred MM Retrieval

This scenario is exactly the same as for the GPRS based service. Please see 3.2.5.4 for the description.

### 5.2.5.5 S05: Streaming Retrieval of MM

This scenario is exactly the same as for the GPRS based service. Please see 3.2.5.5 for the description.

### 5.2.5.6 S05: TMMS Client Unregistration

This scenario is exactly the same as for the GPRS based service. Please see 3.2.5.6 for the description.

## 5.2.6 Charging of Service Usage

Also the charging model presented in 3.2.6 is reused. But the service will use a Charging SCF to debit for the service usage. The data contained in the CDR will be transformed into a *TpChargingParameterSet*. The figure below shows how the TMMS PR uses the Charging SCF to charge for service usage.

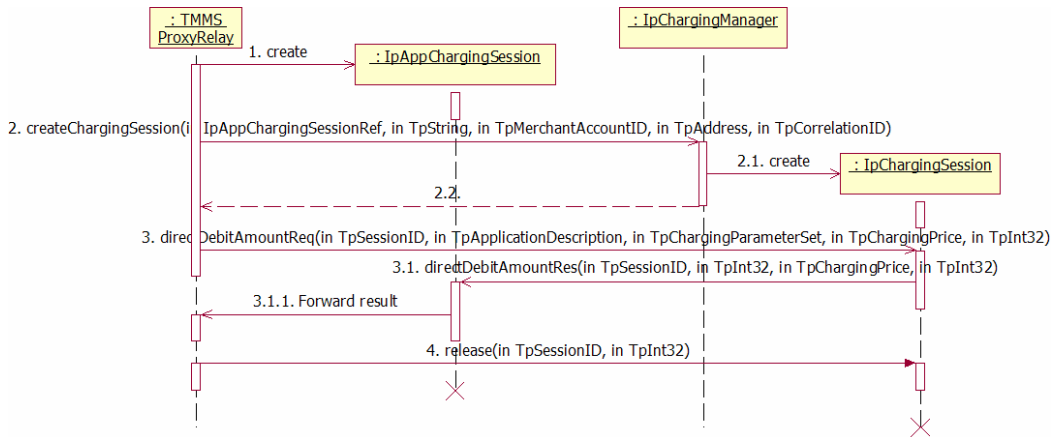


Figure 5-8: Charging of service usage

### 5.3 Evaluation

#### 5.3.1 Security

Table 5-1: Evaluation of Security quality attribute for TMMS Service based on OSA API

Availability	S1: Service is available for legitimate use and sustains Denial of Service attacks
	Service is made available through the service agreement between the TMMS PR and the OSA provider.
Integrity	S2: Data is not altered during transmission
	No end-to-end data integrity check, such as checksums, is done to ensure that data is not altered. The client needs to trust the service.
Authentication	S3: Sender and receiver confirms that they are who they claim to be
	The sender is authenticated with the TMMS PR, so given that the receiver trusts the TMMS PR it can also be sure that the message received comes from the claimed sender.
Authorization	S4: Only authorized users get access to the service
	The TMMS PR needs to check with User Storage if a user is authorized to use the service.
Confidentiality	S5: Only sender and the intended destination part is able to know the content of the message
	By encrypting the message content only recipients who know how to decrypt it may know the content of the message.

### 5.3.2 Usability

Table 5-2: Evaluation of Usability quality attribute for TMMS Service based on OSA API

<b>Terminal Integration</b>	U1: Distribution of the client application
	Client must manually retrieve the client application.
	U2: Which parameters are necessary for the client to communicate with the Proxy-Relay
	Parameters needed are: the TMMS PR IP address, local port number and server port number.
	U3: Address handling (GSM SMS vs. TMMS addresses)
	SIP URIs are used to address TMMS Clients, and TEL URI are used to address GSM SMS clients.
<b>Registration</b>	U4: Client registration
	Clients must register with the TMMS PR by using a SIP REGISTER request.
	U5: Client un-registration
	Clients must un-register with the TMMS PR by using a SIP REGISTER request with the Expires-header set to 0.

### 5.3.3 Modifiability

Table 5-3: Evaluation of Modifiability quality attribute for TMMS Service based on OSA API

<b>Scalability</b>	M1: Add more servers to achieve better scalability
	Since the TMMS PR is addressed through DNS is it possible to use normal DNS based network load balancing techniques to add more servers.
<b>Change Service</b>	M2: Extend the service to support new media types
	The service uses MIME types to describe the media types of the message body. Only the client application must be upgraded to present new media types to the user.
	M3: Restrict the media types supported by the service
	The TMMS PR may filter the media types in a message, based on the Content-Type, to restrict which media types are allowed to be sent between the clients.
	M4: Support new Carrier Access Network (CAN)
	EMPTYEMPTY EMPTYEMPTY EMPTYEMPTY EMPTYEMPTY
	M5: Port the service to a different framework
	EMPTYEMPTY EMPTYEMPTY EMPTYEMPTY EMPTYEMPTY

### 5.3.4 Interoperability

Table 5-4: Evaluation of Interoperability quality attribute for TMMS Service based on OSA API

I1: Send and receive messages with GSM SMS clients
The Service can use Multimedia Messaging SCF to send messages to a GSM SMS client.

### 5.3.5 Reliability

Table 5-5: Evaluation of Reliability quality attribute for TMMS Service based on OSA API

R1: Proxy-Relay routes messages at most once
The TMMS PR uses a Cseq number in the SIP messages to ensure that a message is only routed once to each recipient.
R2: Service stores messages until it has been delivered
When user has retrieved message, the service deletes message from message store.

### 5.3.6 Billability

Table 5-6: Evaluation of Billability quality attribute for TMMS Service based on OSA API

<b>B1: Service provider must get information about chargeable operations</b>
The TMMS PR uses Charging SCF to charge for service usage
<b>B2: Message charged at most once</b>
The cdr-id attribute in the cdr element which identifies each CDR gets transformed and the Charging SCF may use it to ensure that the transformed CDR is only used once.
<b>B3: Sender may demand that the recipient is charged for the message</b>
If the sender includes a CDR in the body of the message sent to the TMMS PR the recipient(s) will be charged for the message. Also the sender may demand what amount the recipient(s) are charged for each media type included in the message.
<b>B4: Sender can not deceive receiver to pay for any messages</b>
Receiver is asked for charging confirmation by the TMMS PR.

## 6 TMMS Service based on OSA Parlay X Web Services

In this chapter we will first give a brief introduction of the OSA Parlay X Web Services [45], which is the technology our end-to-end mobile service designed in this chapter is based on OSA Parlay X Web Services. We continue this chapter with presenting how we have chosen to design our OSA Parlay X Web Services based service, before we end this chapter with an evaluation of the service, based on the criteria and scenarios in .

### 6.1 Overview

The OSA Parlay-X Web Services are abstractions built on top of the OSA API, to enable OSA applications to be built using web services. This removes the need to map the OSA API to different programming languages, such as Java, since web services can be accessed through almost any programming language.

The OSA Parlay-X web services are described in Web Services Description Language (WSDL), and the Universal Description, Discovery and Integration (UDDI) standard is used to publish the services. When an application has discovered the service it uses the Simple Object Access Protocol (SOAP) to invoke it.

### 6.2 Design

#### 6.2.1 Introduction

This design is a web service abstraction of the TMMS Service described in chapter 5. The TMMS PR implements the necessary Parlay-X Web Service interfaces which are consumed by the clients.

#### 6.2.2 Deployment view

The figure depicts how the service is deployed. The following sections will discuss the nodes in the figure.

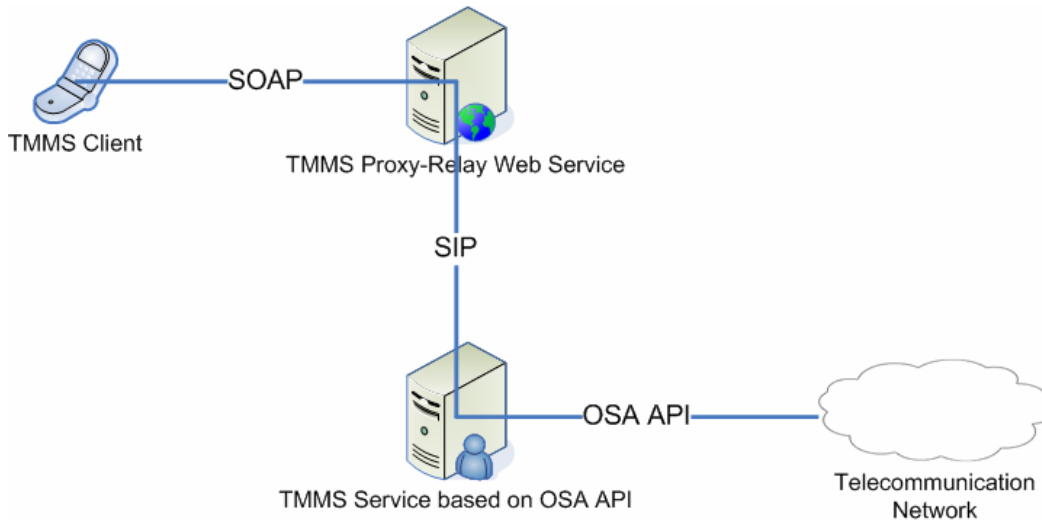


Figure 6-1: Deployment view

### 6.2.2.1 TMMS Client

The TMMS Client is deployed as in the GPRS based solution, as explained in 3.2.2.1. Through an IP-CAN it consumes the web service.

### 6.2.2.2 TMMS Proxy-Relay Web Service

The TMMS Proxy-Relay Web Service node provides a Web Service interface to the TMMS Service. It acts as the TMMS Client in Figure 5-1.

### 6.2.2.3 TMMS Service based on OSA API

This is the TMMS Proxy-Relay node as in Figure 5-1.

### 6.2.3 Component view

The figure below depicts a high level overview of the components of this service. The *TMMS Proxy-Relay Web Service* and *TMMS Client* components will be described in more details in the following subsections.

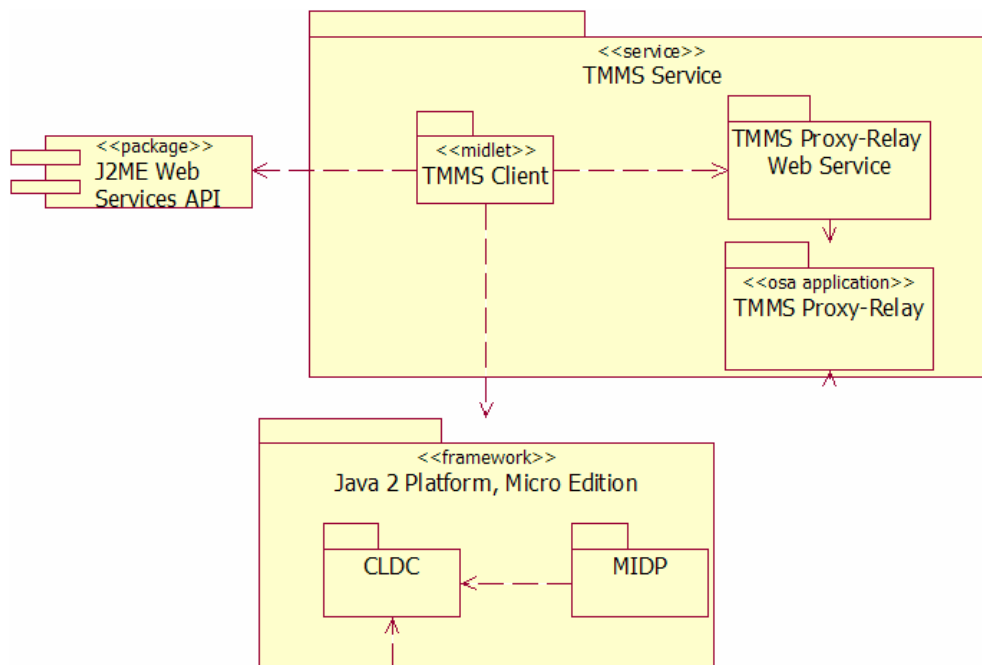


Figure 6-2: Component View

### 6.2.3.1 TMMS Client

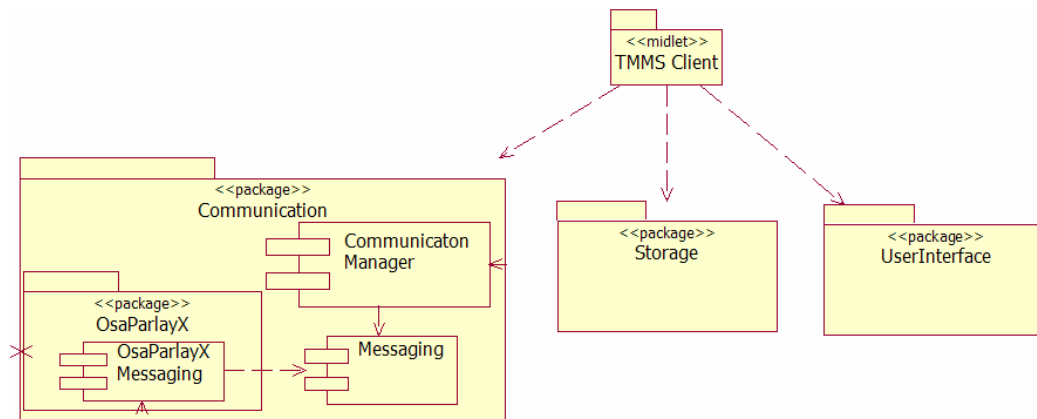


Figure 6-3: Component view for TMMS Client

The TMMS Client has a `TMmsClient.Communication.OsaParlayX` package which contains its own implementation, `OsaParlayXMessaging`, of the `Messaging` interface from the `TMmsClient.Communication` package. This component is used to communicate with the TMMS PR. It uses the J2ME Web Services API [46] component for the communication.

### 6.2.3.2 TMMS Proxy-Relay Web Service

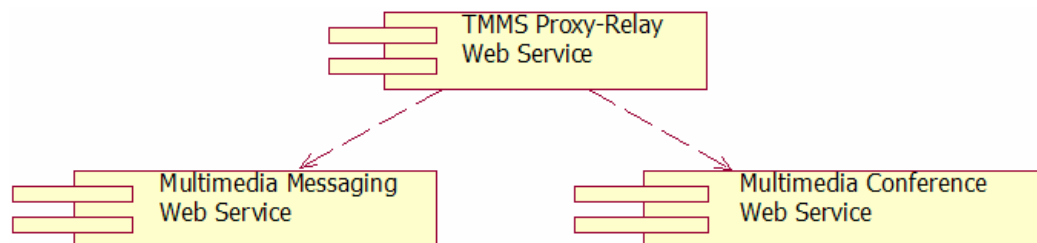


Figure 6-4: TMMS Proxy-Relay Web Service components

The Multimedia Messaging Web Service is specified in [47] and communicates with the TMMS Proxy-Relay described in 5.2.3.2.

### 6.2.4 Security

Security for the Parlay X Web Service is described in clause 4.3 in [48]

## 6.2.5 Scenario Implementation

### 6.2.5.1 S01: TMMS Client Registration

The TMMS PR WS performs the SIP registration on behalf of the client.

See 3.2.5.1 for a description of step 1.1 and 2 of the sequence.

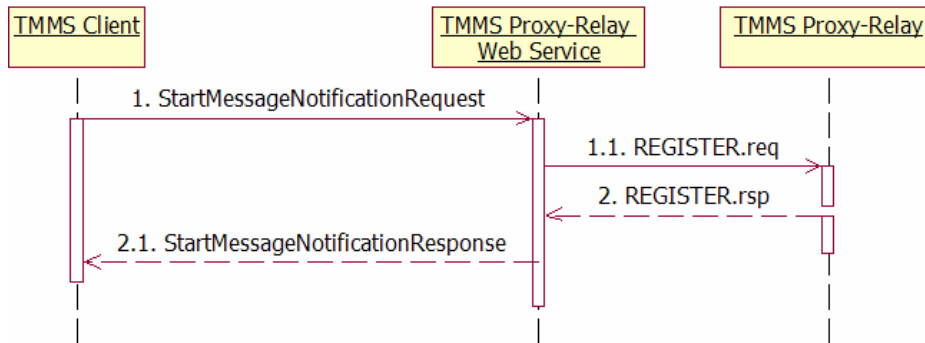


Figure 6-5: Client registration

### 6.2.5.2 S02: Send MM

Here the client sends the message to the TMMS PR WS which delivers it to the TMMS PR. The client polls the TMMS PR WS with the GetMessageDeliveryStatus operation to get the delivery status.

See 3.2.5.23.2.5.3 for a description of step 1.2 of the sequence.

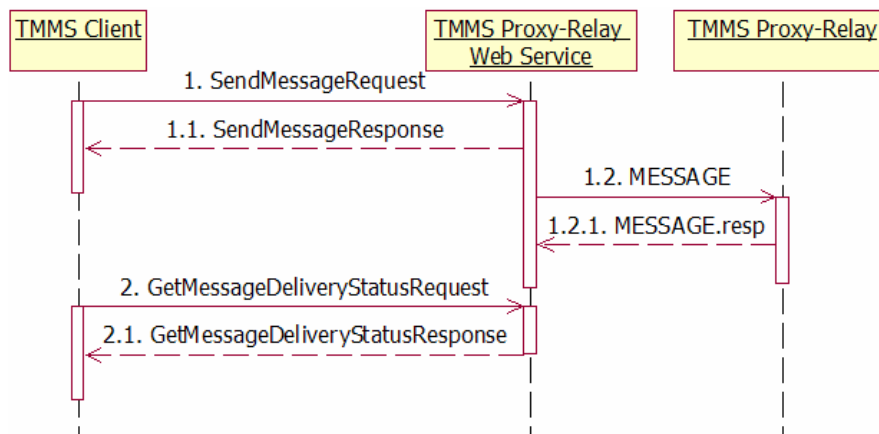


Figure 6-6: Send MM

### 6.2.5.3 S03: New message notification

When the TMMS PR WS gets a *newmessage-alert* on behalf of the client it uses the client's *MessageNotification* interface to notify about the new message alert. The message URI received in the NOTIFY request is used as the *MessageReference*. Only deferred retrieval of messages is supported.

See 3.2.5.3 for a description of the steps 1, 1.1 and 1.3 of the sequence.



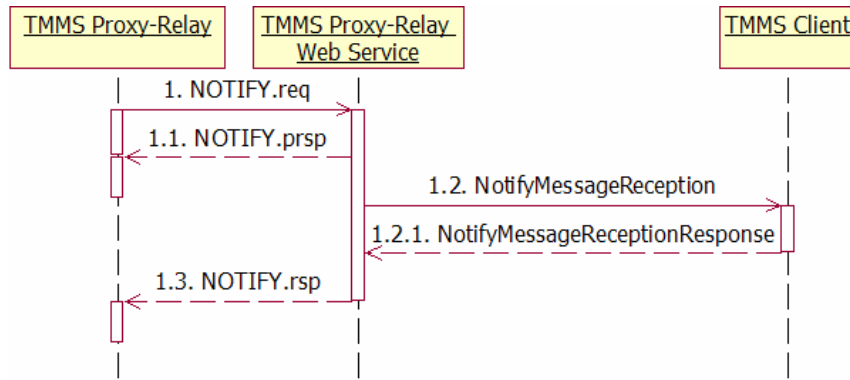


Figure 6-7: New message notification

### 6.2.5.4 S04: Deferred MM Retrieval

The TMMS PR WS fetches the message referenced in the *GetMessageRequest* from the TMMS MS and returns it to the client.

See 3.2.5.4 for a description of step 1.1 of the sequence.

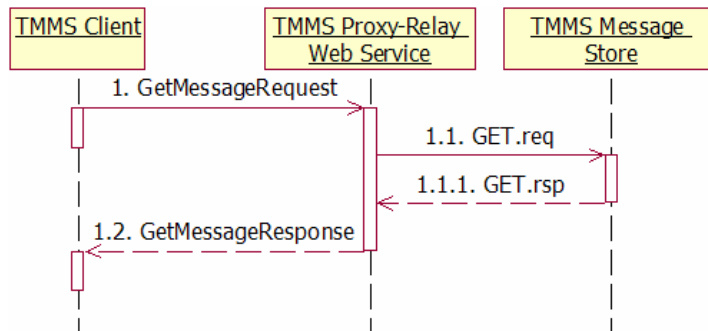


Figure 6-8: Deferred message retrieval

### 6.2.5.5 S05: Streaming Retrieval of MM

Streaming retrieval is accomplished using the Multimedia conference web service.

The client starts streaming retrieval by creating a conference(step 1). It sets the name of the conference as the message reference which it wants the TMMS PR WS to stream back to it. Next it invites itself(step 2) and the TMMS PR WS(step 3) to the conference, respectively. When the TMMS PR WS notices that it is added to a conference it sets it up for streaming the message as indicated in the conference information.

First it retrieves the conference info, where the message reference is put by the client. Then it gets information about the conference owner to know which codecs it supports. Now it is ready to setup streaming with the TMMS MS, which is described in 3.2.5.5.

Then it adds the streaming media for the client, which will then be able to retrieve the message. When the client is finished it ends the conference.

Note that all the messages the TMMS PR WS sends to itself should rather be executed internally, without going through the WS interface.

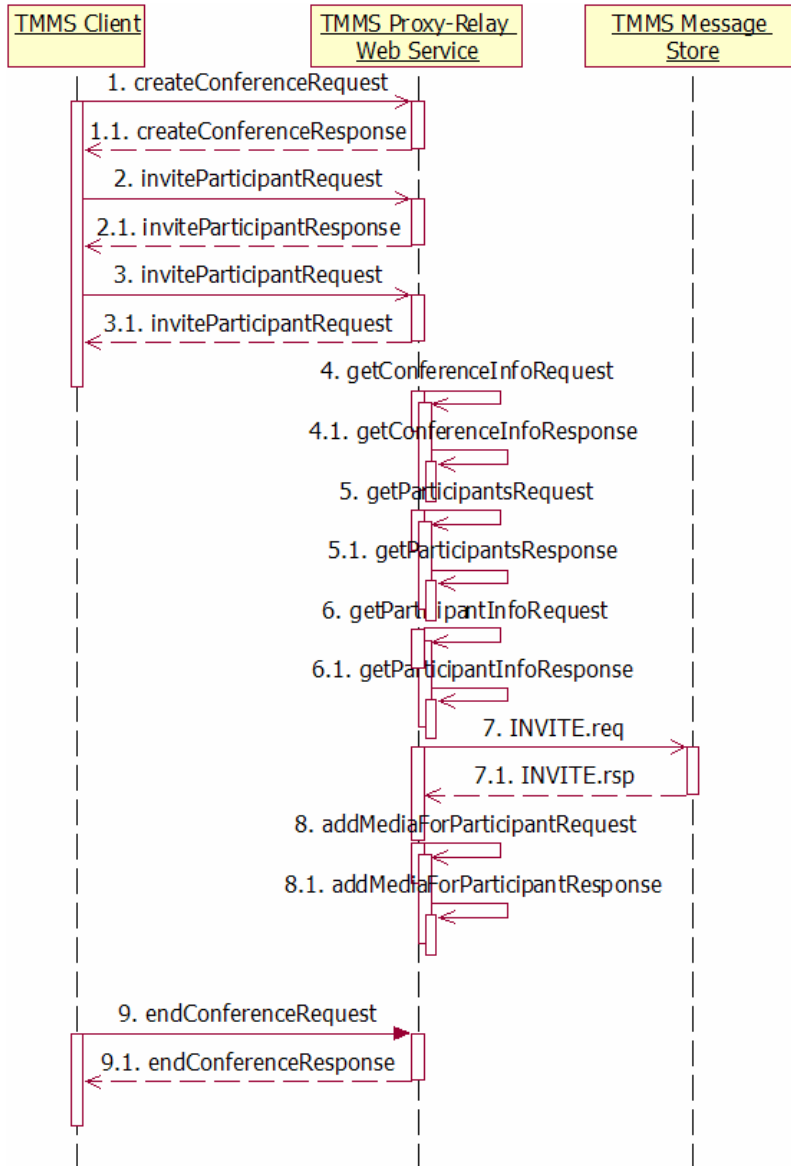


Figure 6-9: Streaming retrieval of MM

### 6.2.5.6 S05: TMMS Client Unregistration

When the client sends a *StopMessageNotificationRequest* the TMMS PR WS performs the unregistration, as described in 3.2.5.6.

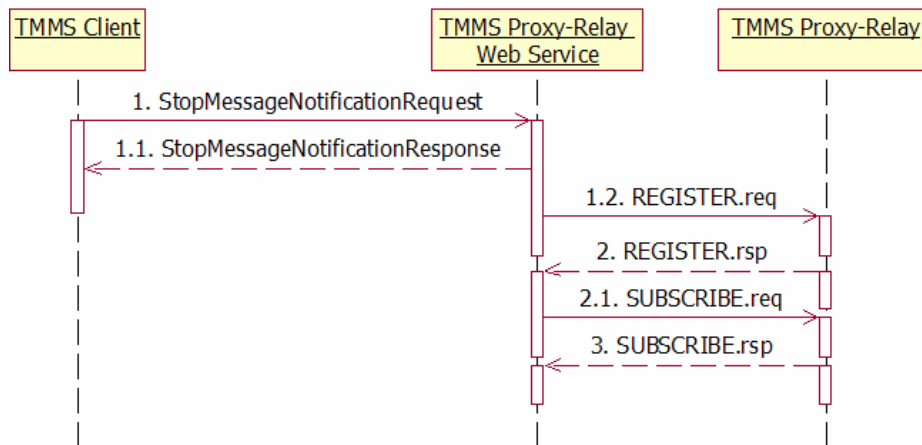


Figure 6-10: Client deregistration

### 6.2.6 Charging of Service Usage

Since the TMMS PR WS acts on the behalf of the client, the client will be charged per the charging model in section 3.2.6.1.

If the client includes *ChargingInformation* in the *SendMessageRequest* then the TMMS PR WS will transform that structure into a CDR which will be used by the TMMS PR to charge the recipient.

### 6.3 Evaluation

#### 6.3.1 Security

Table 6-1: Evaluation of Security quality attribute for TMMS Service based on OSA Parlay X Web Services

Availability	S1: Service is available for legitimate use and sustains Denial of Service attacks
	Configuration issue. Need to know which server to contact.
Integrity	S2: Data is not altered during transmission
	On top of OSA API security, TMMS PR WS ensures integrity by using XML Digital Signature.
Authentication	S3: Sender and receiver confirms that they are who they claim to be
	Client gets authenticated by the service when it registers to the service.
Authorization	S4: Only authorized users get access to the service
	TMMS PR WS needs to check with TMMS PR if client is authorized to use the service.
Confidentiality	S5: Only sender and the intended destination part is able to know the content of the message
	On top of OSA API security, TMMS PR WS ensures confidentiality by either using a Virtual Private Network (VPN) which gets administrated independent of the Web Service implementation, or to use HTTP over TLS0

### 6.3.2 Usability

Table 6-2: Evaluation of Usability quality attribute for TMMS Service based on OSA Parlay X Web Services

Terminal Integration	U1: Distribution of the client application
	Client must retrieve the client application manually.
	U2: Which parameters are necessary for the client to communicate with the Proxy-Relay
	Client need to know what SOAP binding the TMMS PR WS supports, address to web service.
	U3: Address handling (GSM SMS vs. TMMS addresses)
Registration	SIP URIs are used to address TMMS Clientsclients, and TEL URIs are used to address GSM SMS clients.
	U4: Client registration
	TMMS PR WS performs a SIP registration on behalf of the client.
	U5: Client un-registration
	The TMMS PR WS performs a un-registration on behalf of the client.

### 6.3.3 Modifiability

Table 6-3: Evaluation of Modifiability quality attribute for TMMS Service based on OSA Parlay X Web Services

Scalability	M1: Add more servers to achieve better scalability
	Since the TMMS PR is addressed through DNS is it possible to use normal DNS based network load balancing techniques to add more servers.
Change Service	M2: Extend the service to support new media types
	The service uses MIME types to describe the media types of the message body. Only the client application must be upgraded to present new media types to the user.
	M3: Restrict the media types supported by the service
	The TMMS PR may filter the media types in a message, based on the Content-Type, to restrict which media types are allowed to be sent between the clients.
	M4: Support new Carrier Access Network (CAN)
	Yes.
M5: Port the service to a different framework	Supported through WS interoperability.

### 6.3.4 Interoperability

Table 6-4: Evaluation of Interoperability quality attribute for TMMS Service based on OSA Parlay X Web Services

	I1: Send and receive messages with GSM SMS clients
	The Service can use Multimedia Messaging SCF to send messages to a GSM SMS client.

### 6.3.5 Reliability

Table 6-5: Evaluation of Reliability quality attribute for TMMS Service based on OSA Parlay X Web Services

	R1: Proxy-Relay routes messages at most once
	Use of Cseq in SIP messages, SOAP?
	R2: Service stores messages until it has been delivered
	When user has retrieved message, the service deletes message from message store.

### 6.3.6 Billability

Table 6-6: Evaluation of Billability quality attribute for TMMS Service based on OSA Parlay X Web Services

	<b>B1: Service provider must get information about chargeable operations</b>
	The TMMS PR sends charging information to a Charging Control which handles charging on behalf of the service provider.
	<b>B2: Message charged at most once</b>
	The cdr-id attribute of the cdr element identifies each CDR, and the Charging Control may use it to ensure that a CDR is only used once.
	<b>B3: Sender may demand that the recipient is charged for the message</b>
	Sender may include ChargingInformation in the message.
	<b>B4: Sender can not deceive receiver to pay for any messages</b>
	Receiver is asked for charging confirmation by the TMMS PR.

## 7 Discussion

### 7.1 Introduction

In this chapter we will compare the evaluation of the four TMMS Service designs.

### 7.2 Security

#### 7.2.1 Service is available for legitimate use and sustains Denial of Service attacks

None of the services implements any intrusion detection. The IMS based design is the most robust of them since the S-CSCF filters access for it.

#### 7.2.2 Data is not altered during transmission

Both the IMS and the OSA Parlay X based designs provides data integrity on the data transport.

It is possible to explicitly add such a feature in the other designs too.

#### 7.2.3 Sender and receiver confirms that they are who they claim to be

The GPRS, OSA API and OSA Parlay X services all builds a relationship of trust with the TMMS PR as the root. Clients must trust that the TMMS PR has successfully authenticated the other party.

On the other hand IMS authenticates all parties who want to communicate on the network. This is a better end-to-end approach then the rest of the designs.

#### 7.2.4 Only authorized users get access to the service

All of the designs use some kind of user store to authorize the clients. The IMS based service utilizes the IMS user store for this purpose, but it could use a custom user store like the other designs.

In some service deployment scenarios it may be important to be able to use a custom user store so the service can easily integrate with e.g. an enterprise's user store.

#### 7.2.5 Only sender and the intended destination part is able to know the content of the message

The services provide the same level of support as in 7.2.2 here. All of them *can* provide confidentiality, but it is only the IMS and OSA Parlay X frameworks which provide confidentiality by default.

### 7.3 Usability

#### 7.3.1 Distribution of the client application

All of the designs provide a manual distribution mechanism, where the client must start to download the client application itself.

IMS, in addition, can also support a push-based distribution. This comes from the fact that the TMMS PR is notified when clients registers to IMS. Note that the iFC for the service must also be setup correctly for this to work.

As described push-based distribution depends on that the TMMS PR is notified when clients register to the network. For the GPRS based service this information should be possible with support from a CAMEL application. Using the Presence and Availability Management (PAM) SCF of the OSA API and the Presence WS of the OSA Parlay X, respectively, it should also be possible to do this on those frameworks too.

But it seems much easier to accomplish this feature on IMS since the service provider only need to add another part to the iFC of the client PUI's service profile. This compared to e.g. developing a CAMEL application.

### **7.3.2 Which parameters are necessary for the client to communicate with the Proxy-Relay**

All of the designs has a similar parameter set.

### **7.3.3 Address handling (GSM SMS vs. TMMS addresses)**

All of the service designs do address handling similar, SIP URIs are used to identify TMMS clients and TEL URIs are used to identify GSM SMS clients.

Except for the IMS based service, the TMMS PR of the services must look after TEL URIs and relay them to GSM SMS clients itself. In IMS, with the IP-MESSAGE-GW node present, messages sent to GSM SMS clients will be routed to the recipients with no influence from the TMMS PR.

### **7.3.4 Client registration**

GPRS and OSA API has the same registration procedure, and it is the most complex procedure. First the client must send a SIP REGISTER request to the TMMS PR to register its contact information. Next the client must subscribe to the newmessage-alert event.

In IMS only the last step, event subscription, is necessary from the TMMS Service's point of view.

Similar to the IMS design, in the Parlay X WS design the client only subscribes to notifications about new messages.

### **7.3.5 Client unregistration**

All the unregistration procedures are similar to the registration procedures.

## **7.4 Modifiability**

### **7.4.1 Add more servers to achieve better scalability**

All of the designs use DNS to access the TMMS PR, which means they can all achieve better scalability through normal DNS based network load balancing techniques.

The IMS design, which use IMS' HSS node as user- and message store benefits from the scalability 3GPP has designed for this node.



#### **7.4.2 Extend the service to support new media types**

All of the service designs use MIME types to describe the media types used. This makes the media type transparent to the TMMS PR. Only the streaming media server and the client application must be upgraded to support new media types.

#### **7.4.3 Restrict the media types supported by the service**

In all the service designs the TMMS PR may filter the media type of a message based on the Content-Type header. Additionally a content negotiation scheme, like the HTTP Accept header, can be added.

#### **7.4.4 Support new Carrier Access Network**

If the CAN supports packet switching then it is supported by all the frameworks. However the binding between a new CAN and the framework may be unspecified by the framework and must be defined first.

#### **7.4.5 Port the service to a different framework**

All of the service designs use standard open protocols for communication with the nodes of the framework, which is a good basis for portability.

But in the end it depends on the features and constraints of the target framework. If the target framework does not support these protocols then an adapter must be introduced to interoperate between the protocols of the service and the target framework.

GPRS sets the least constraints to the protocols used by the service, since it only provides data transport. IMS on the other hand dictates most of the protocols used by the service. This makes GPRS the least common multiplier of the services and all of them can be ported to GPRS.

### ***7.5 Interoperability***

#### **7.5.1 Send and receive messages with GSM SMS clients**

Since GSM SMS is a text based messaging service, at least for the end-user, multimedia content can not be sent directly. UP11 in Table 2-4 describes how TMMS messages with multimedia content are adapted to GSM SMS. This is equal for all four of our service designs.

The difference between the frameworks is how routing from TMMS PR to GSM SMS is supported. In our service design based on GPRS the TMMS PR gets no support to interoperate with GSM SMS, and must provide such support itself.

On the other hand IMS supports routing of TEL URI, and an IP-MESSAGE-GW node can be introduced which picks up messages with a Request-URI of the TEL uri-scheme and routes them to a GSM SMSC.

Both the OSA API and the OSA Parlay X Web Service frameworks have methods available which the TMMS PR can use to send messages to GSM SMS clients.

## **7.6 Reliability**

### **7.6.1 Proxy-Relay routes messages at most once**

The service designs based on the GPRS, IMS and OSA API frameworks all use SIP as the protocol between sender and TMMS PR. SIP uses a CSeq header field to identify and order SIP transactions.

SOAP, which is the transport protocol used by the OSA Parlay X Web Service framework, defines no reliability. By adding support for WS-Reliability at both the TMMS Client and TMMS PR WS the service can guarantee that messages are routed at most once.

### **7.6.2 Service stores messages until it has been delivered**

None of the services deletes the message until it has been received by the client. However messages which the recipient does not accept to be charged for are deleted immediately after the recipient has refused to be charged for it.

## **7.7 Billability**

### **7.7.1 Service provider must get information about chargeable operations**

All of our service designs depends on a charging controller where the TMMS PR sends charging information to for chargeable operations.

### **7.7.2 Message charged at most once**

The charging information in all the frameworks includes an identifier for the CDR, which is used to check for duplicates.

### **7.7.3 Sender may demand that the recipient is charged for the message**

Only the IMS and the OSA Parlay X Web Services frameworks have support for this, but support is explicitly added in the rest of the service designs.

### **7.7.4 Sender can not deceive receiver to pay for any messages**

None of the frameworks provides any means for this, so all of our service designs explicitly ask the recipient before charging it.

## 8 Conclusion

From the discussion we can see that many of the evaluation criteria are fulfilled equally by all the service designs. In this chapter we will make conclusions based on our discussion above and our experiences with designing the services.

### ***8.1 GPRS as a framework for creating end-to-end mobile services***

GPRS provides only a data transport protocol, which means that the service designer must select all application level protocols herself. However this is a blade with two edges.

This freedom is unnecessary for creating end-to-end services which are quite simple in functionality, like our use case, but still require important properties such as security and reliability. But in scenarios where support for proprietary and/or legacy application protocols is needed then it is easy to use these with GPRS.

It seems like a CAMEL application can provide the missing information and events to the GPRS TMMS PR, so it can provide a similar service experience as the service based on IMS. A disadvantage with this is that the service becomes dependent on the operator's network.

A disadvantage of GPRS is that it does not provide an end-to-end environment, since the TMMS PR lives outside of the GPRS network. This makes it harder to achieve e.g. end-to-end security.

### ***8.2 IMS as a framework for creating end-to-end mobile services***

IMS is almost the opposite of the GPRS environment. There are lots of features available and most, if not all, application level protocols have been specified.

Fortunately, IMS is also easy to extend and adapt to the service designer's needs. E.g. it provides support for hosting both OSA and CAMEL application servers by adapting SIP to the expected input/output of those frameworks.

A problem with IMS is the size of the documentation, which is huge in contrast to the other frameworks evaluated. It is difficult to say exactly which specifications are necessary to know about for a service designer since most of the IMS specifications touch this subject.

Finally, IMS provides a true end-to-end environment in contrast to the other frameworks evaluated.

### ***8.3 OSA API as a framework for creating end-to-end mobile services***

The various SCFs in the OSA API framework provide a feature rich environment for the service provider. But it was hard to find a good way to exchange data between the client and the PR, except setting up a "backdoor" like in our design.

This "backdoor" destroys the end-to-end properties of the OSA API framework, and it ends up very similar to the GPRS framework instead.

In contrast to IMS, OSA API has a small, finite set of documentation which is easy to find. Also since it is divided into several and it is easy to see which are relevant to a service.

#### 8.4 Conclusion: : OSA Parlay X Web Services as a framework for creating end-to-end mobile services

---

*Unfortunately the authors did not spend enough research time to learn all the features of the OSA API framework. Therefore we acknowledge, with regret, that the service design is a little flawed since it doesn't use much of the OSA API features.*

#### **8.4 OSA Parlay X Web Services as a framework for creating end-to-end mobile services**

OSA Parlay X WS provides a simple subset of OSA API's features. But it is rich enough so the TMMS Service based on OSA Parlay X WS provides all the necessary features, except for immediate retrieval of messages.

Based on WS-I Basic Profile it provides an end-to-end environment which supports all the security quality attributes evaluated.. Reliable messaging, however, is not provided in the environment. This quality attribute can be added by the service provider, but both the client and the PR must enable support for it.

#### **8.5 Overall**

We introduced by stating that all frameworks has their own advantages and disadvantages, and this has been proven correct in this thesis. Therefore we can not state that any framework is preferred over the other frameworks without knowing the constraints and requirements of the service.

#### **8.6 Further Work**

##### **8.6.1 OSA API**

It should be examined if adding a Multimedia Messaging SCF with support for TMMS helps to make the end-to-end service more natural. This should make it possible to remove the "backdoor", so the traffic between the TMMS Client and TMMS PR follows the normal network instead.

##### **8.6.2 Different Use Case**

Our use case, a multimedia messaging service, is very common and all of the frameworks have a lot of support for messaging already. This has made it hard to use all features of the frameworks to avoid building a messaging service upon a messaging service. The design of the IMS based service is closest to cross the line here. But since it adds value on top of the messaging features of IMS we feel that the design is legal for this evaluation.

A different use case which is not directly supported by the frameworks could shed light on areas which have been missed in this thesis.

##### **8.6.3 Quantitative Testing**

This has been a theoretical study of the frameworks, which means it is hard to get quantitative data about how the services work when deployed. By implementing the different designs it should be easier to investigate their advantages and disadvantages.

## 9 References

- [1] OMA-MMS-ARCH-V1\_2-20031217-C, Multimedia Messaging Service, Architecture Overview, Open Mobile Alliance, Dec. 2003.
- [2] OMA-MMS-ENC-V1\_2-20040323-C, Multimedia Messaging Service Encapsulation Protocol, Open Mobile Alliance, Mar. 2003.
- [3] *JSR030, Connected Limited Device Configuration Version 1.1*, Sun Microsystems, Mar. 2003; <http://jcp.org/en/jsr/detail?id=30>.
- [4] *JSR068, J2ME Platform Specification*, Sun Microsystems; <http://jcp.org/en/jsr/detail?id=68>.
- [5] J2SE, <http://java.sun.com/j2se/1.5.0/index.jsp>.
- [6] *JSR118, Mobile Information Device Profile 2.0 Version 2.0*, Sun Microsystems, Nov. 2002; <http://jcp.org/en/jsr/detail?id=118>.
- [7] R. Fielding et al., *Hypertext Transfer Protocol – HTTP/1.1*, IETF RFC 2616, Jun. 1999; <http://www.ietf.org/rfc/rfc2616.txt>.
- [8] *JSR180, SIP API version 1.0.1*, JSR 180 Expert Group, 2004; <http://jcp.org/en/jsr/detail?id=180>
- [9] *03-03-01, Unified Modeling Language (UML)*, Object Management Group, 2003; <http://www.omg.org/technology/documents/formal/uml.htm>.
- [10] *TS 32.152 V6.3.0, Telecommunication management; Integration Reference Point (IRP) Information Service (IS) Unified Modeling Language (UML) repertoire*, 3GPP, Jun. 2005; <http://www.3gpp.org/ftp/Specs/html-info/32152.htm>.
- [11] T. Berners-Lee et al., *Uniform Resource Locators (URL)*, IETF RFC 1738, Dec. 1994; <http://www.ietf.org/rfc/rfc1738.txt>.
- [12] H. Schulzrinne, A. Rao, R. Lanphier, *Real Time Streaming Protocol (RTSP)*, IETF RFC 2326, Apr. 1998; <http://www.ietf.org/rfc/rfc2326.txt>.
- [13] L. Bass, P. Clements and Rick Kazman, *Software Architecture in Practice*, 2nd ed. Addison-Wesley, Reading, MA., 2003; pp. 294-295.
- [14] Rosenberg et al., *Session Initiation Protocol*, IETF RFC 3261, Jun. 2002; <http://www.ietf.org/rfc/rfc3261.txt>.
- [15] M. O'Dell et al., *A Simple Network Protocol (SNMP)*, IETF RFC 1157, May 1990; <http://www.ietf.org/rfc/rfc1157.txt>.
- [16] H. Schulzrinne et al., *A Transport Protocol for Real-Time Applications*, IETF RFC 3550, Jul. 2003; <http://www.ietf.org/rfc/rfc1889.txt>.
- [17] F. Cuervo et al., *Megaco Protocol Version 1.0*, IETF RFC 3015, Nov. 2000; <http://www.ietf.org/rfc/rfc3015.txt>.
- [18] G. Cote, “Advanced SIP Series: Extending SIP”, Awards Solutions, 2001; [http://www.sipcenter.com/sip/nsf/html/WEBB6ARJZ7/\\$FILE/Award\\_Extending\\_SIP.pdf](http://www.sipcenter.com/sip/nsf/html/WEBB6ARJZ7/$FILE/Award_Extending_SIP.pdf).
- [19] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6)*, IETF RFC 2460, Dec. 1998; <http://www.ietf.org/rfc/rfc2460.txt>.
- [20] J.F. Kurose and K.W. Ross, *Computer Networking, A Top-Down Approach Featuring the Internet, third edition*, 2005.
- [21] T. Berners-Lee et al., *Uniform Resource Identifiers (URI): Generic Syntax*, IETF RFC 2396, Aug. 1998; <http://www.ietf.org/rfc/rfc2396.txt>.
- [22] G. Camarillo and M.A. Garcia-Martin, *The 3G IP Multimedia Subsystem, Merging the Internet and the Cellular Worlds*, John Wiley & Sons, Ltd, 2004.

- [23] J. Postel, *User Datagram Protocol*, IETF RFC 768, Aug. 1980;  
<http://www.ietf.org/rfc/rfc768.txt>.
- [24] M. del Rey, *Transmission Control Protocol*, IETF RFC 793, Sep. 1981;  
<http://www.ietf.org/rfc/rfc0793.txt>.
- [25] *TS 22.060 Version 6.0.0, General Packet Radio Service (GPRS) Service description; Stage 1*, 3GPP, Mar.;  
<http://www.3gpp.org/ftp/Specs/html-info/22060.htm>.
- [26] NTT DoCoMo, "i-mode", NTT DoCoMo, 2005;  
[http://www.nttdocomo.co.jp/english/p\\_s/imode/what/what/index.html](http://www.nttdocomo.co.jp/english/p_s/imode/what/what/index.html)
- [27] *TS 23.060 Version 6.7.0, General Packet Radio Service (GPRS) Service description; Stage 2*, 3GPP, Dec. 2004;  
<http://www.3gpp.org/ftp/Specs/html-info/23060.htm>.
- [28] *TS 22.078 V6.6.0, Customised Applications for Mobile network Enhanced Logic (CAMEL) Service description; Stage 1*, 3GPP, Sep. 2003;  
<http://www.3gpp.org/ftp/Specs/html-info/22078.htm>.
- [29] A.B. Roach, *Session Initiation Protocol (SIP)-Specific Event Notification*, IETF RFC 3265, Jun. 2002;  
<http://www.ietf.org/rfc/rfc2782.txt>.
- [30] H. Schulzrinne, *The tel URI for Telephone Numbers*, IETF RFC 3966, Dec. 2004;  
<http://www.ietf.org/rfc/rfc3966.txt>.
- [31] R. Fielding et al., *Hypertext Transfer Protocol – HTTP/1.1*, IETF RFC 2616, Jun. 1999;  
<http://www.ietf.org/rfc/rfc2616.txt>
- [32] D. Ragget, A. Le Hors and Ian Jacobs, "HTML 4.0 Specification", World Wide Web Consortium (W3C) recommendation, Apr. 1998;  
<http://www.w3.org/TR/1998/REC-html40-19980424/>.
- [33] M. Koutsopoulou and A. Alonistioti, "Charging, accounting and billing as a sophisticated and reconfigurable discrete service for the next generation mobile networks" Proc. In Vehicular Technology Conference, 2002, pp. 2342-2345 vol.4.
- [34] Open Mobile Alliance, *WAP Billing Framework Version 1.0*, Nov. 2002;  
<http://www.openmobilealliance.org>.
- [35] D.C. Fallside and P. Walmsley, "XML Schema Part 0: Primer second edition", World Wide Web Consortium (W3C) recommendation, Oct.2004  
<http://www.w3.org/TR/xmlschema-0/>.
- [36] N. Borenstein and N. Freed, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, IETF RFC 2045, Nov. 1996;  
<http://www.ietf.org/rfc/rfc2045.txt>.
- [37] J. Postel, *Domain Name System Structure and Delegation*, IETF RFC 1591, Mar.1994;  
<http://www.ietf.org/rfc/rfc1591.txt>
- [38] *TS 23.228 V6.9.0, IP Multimedia Subsystem (IMS); Stage 2*, 3GPP, Mar. 2005;  
<http://www.3gpp.org/ftp/Specs/html-info/23228.htm>.
- [39] S. Kent and R. Atkinson, *Security Architecture for the Internet Protocol*, IETF RFC 2401, Nov. 1998;  
<http://www.ietf.org/rfc/rfc2401.txt>.
- [40] *TS 32.260 V6.1.0, Telecommunication management; Charging management; IP Multimedia Subsystem (IMS) charging*, 3GPP, Mar. 2005;  
<http://www.3gpp.org/ftp/Specs/html-info/32260.htm>.
- [41] WAP Push Architectural Overview, Open Mobile Alliance, 2001;  
<http://www.openmobile.alliance.org>.
- [42] Push OTA Protocol, Open Mobile Alliance, 2001;  
<http://www.openmobilealliance.org>.

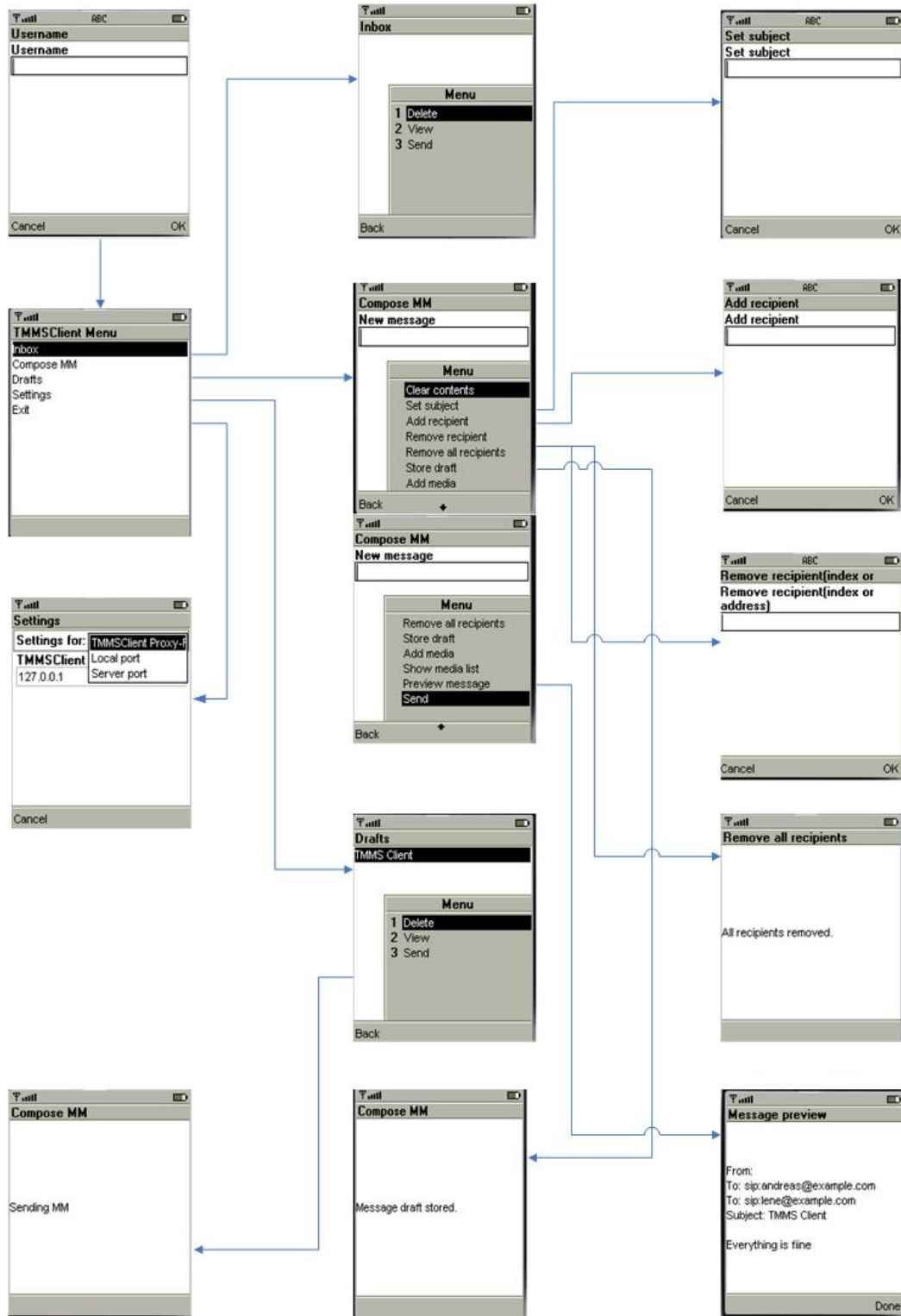
- [43] *TS 22.127 V6.7.0, Service Requirements for the Open Services Access (OSA); Stage 1*, 3GPP, Jan. 2005;  
<http://www.3gpp.org/ftp/Specs/html-info/22127.htm>.
- [44] *TS 23.198 V1.0.0, Open service access (OSA); Stage 2*, 3PP, December 2004;  
<http://www.3gpp.org/ftp/Specs/html-info/23198.htm>.
- [45] *TS 29.199-1 V6.0.0, Open Service Access (OSA); Parlay X Web Services; Part 1:Common*, 3GPP, Sep. 2004;  
<http://www.3gpp.org/ftp/Specs/html-info/29199-01.htm>.
- [46] *JSR172, J2ME Web Services 1.0*, Sun Micro Systems, Oct. 2003;  
<http://jcp.org/en/jsr/detail?id=172>
- [47] *TS 29.199-5 V6.1.0, Open Service Access (OSA); Parlay X web services; Part 5: Multimedia messaging*, 3GPP, Dec. 2004
- [48] *TS 29.199-8 V6.0.0, Open Service Access (OSA); Parlay X web services; Part 8: Terminal status*, 3GPP, Sep. 2004
- [49] *TR 23.804 Version 7.0.0, Support of SMS and MMS over generic 3GPP IP access*, 3GPP, Jun. 2005;  
<http://www.3gpp.org/ftp/Specs/html-info/23804.htm>.

## 10 Appendix

APPENDIX A: <b>TMMS CLIENT USER INTERFACE</b> .....	<b>1</b>
APPENDIX B: <b>CHARGING-DETAILS-RECORD.XSD</b> .....	<b>2</b>
APPENDIX C: <b>GPRS-CHARGING-SEND.CDR</b> .....	<b>7</b>
APPENDIX D: <b>MM-CHARGING.MSG</b> .....	<b>8</b>
APPENDIX E: <b>TMMSCLIENT-IMS.JAD</b> .....	<b>9</b>
APPENDIX F: <b>INITIALFILTERCRITERIA.XML</b> .....	<b>10</b>



## Appendix A: TMMS Client user interface



## Appendix B: charging-details-record.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:cdr="http://tmms.example.org/Charging"
  targetNamespace="http://tmms.example.org/Charging"
  elementFormDefault="qualified" attributeFormDefault="qualified">
  <xsd:element name="cdr">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="1">
          <xsd:element name="record-receive" type="cdr:receive"
maxOccurs="1" />
          <xsd:element name="record-send" type="cdr:send"
maxOccurs="1" />
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="recording-entity" type="xsd:anyURI"
use="required" />
      <xsd:attribute name="cdr-id" type="xsd:positiveInteger"
use="required" />
      <xsd:attribute name="chargeable-operation-id-number"
type="xsd:nonNegativeInteger" use="required" />
      <xsd:attribute name="timestamp" type="xsd:dateTime"
use="required" />
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="send">
    <xsd:sequence>
      <xsd:element name="media-description" type="cdr:mediaDetail"
minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="status" type="cdr:recordStatus" />
    <xsd:attribute name="sender-id" type="xsd:anyURI" />
    <xsd:attribute name="connection" type="cdr:connectionType" />
    <xsd:attribute name="charging-data-provider" type="xsd:anyURI" />
    <xsd:attribute name="partial-record-sequence-number"
type="xsd:positiveInteger" />
    <xsd:attribute name="merchant-id" type="xsd:string" />
    <xsd:attribute name="service-user-id" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>
          Format according to WAP ClientID spec or
          implementation dependent string.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="charged-party" type="xsd:string"
use="optional" />
    <xsd:attribute name="transaction-id" type="xsd:string"
use="required" />
    <xsd:attribute name="descriptive-text" type="xsd:string"
use="optional" />
    <xsd:attribute name="destination" type="xsd:anyURI"
use="optional">
      <xsd:annotation>
        <xsd:documentation>
          The URL from where the content was retrieved.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:schema>
```

This attribute is provided only for compatibility with OMA WBF.

```
</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="content-type" type="cdr:mimeType"/>
<xsd:attribute name="bearer" type="cdr:bearerType" use="required"
/>
<xsd:attribute name="header-volume" type="xsd:positiveInteger"
use="required" />
<xsd:attribute name="data-volume" type="xsd:nonNegativeInteger"
use="required" />
<xsd:attribute name="iresult" type="xsd:string" use="optional" />
<xsd:attribute name="wresult" type="cdr:wresultType"
use="required" />
</xsd:complexType>
<xsd:complexType name="receive">
<xsd:sequence>
<xsd:choice>
<xsd:element name="push-submission"
type="cdr:pushSubmission"/>
<xsd:element name="push-message" type="cdr:pushMessage"/>
<xsd:element name="push-cancellation"
type="cdr:pushCancellation"/>
<xsd:element name="push-query" type="cdr:pushQuery"/>
</xsd:choice>
</xsd:sequence>
<xsd:attribute name="pi-id" type="xsd:anyURI">
<xsd:annotation>
<xsd:documentation>The identity of the Push
Initiator(PI)</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="ppg-id" type="xsd:anyURI">
<xsd:annotation>
<xsd:documentation>The identity of the Push Proxy
Gateway(PPG)</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="push-id" type="xsd:positiveInteger">
<xsd:annotation>
<xsd:documentation>A push message identity provided by the
PI</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<xsd:complexType name="pushSubmission">
<xsd:sequence>
</xsd:sequence>
<xsd:attribute name="replace-push-id" type="xsd:positiveInteger"
use="optional">
<xsd:annotation>
<xsd:documentation>
In case replacement of a previously submitted push is
requested
, then the same Push ID is repeated here to indicate that
request
</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
```

```
<xsd:attribute name="push-content-length"
type="xsd:positiveInteger"/>
<xsd:attribute name="push-content-type" type="cdr:mimeType"/>
<xsd:attribute name="priority" type="cdr:priorityType"/>
<xsd:attribute name="number-of-recipients"
type="xsd:positiveInteger" use="optional">
  <xsd:annotation>
    <xsd:documentation>
      The number of recipients concerned with the message
      delivery initiation of the push message. Only to be output
      if more then 1 recipient is addressed
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="status-code" type="cdr:wappapStatusCode"
use="optional">
  <xsd:annotation>
    <xsd:documentation>
      The status of the message immediately after submission as
      defined in [WAPPap]
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<xsd:complexType name="pushMessage">
  <xsd:attribute name="push-client-id" type="xsd:anyURI"
use="optional">
    <xsd:annotation>
      <xsd:documentation>
        Served terminal client identity as identified in the mobile
        access network, e.g. MSISDN, proxy user identity, global client
        id.... (bearer dependent, billing model dependent).

        Only to be output if the chargeable data are generated by
        the PPG.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="recipient-address" type="xsd:string"
use="required"/>
  <xsd:attribute name="delivery-result" type="cdr:deliveryType"
use="required"/>
  <xsd:attribute name="bearer" type="cdr:bearerType"
use="optional"/>
  <xsd:attribute name="message-state" type="cdr:messageStateType"
use="optional"/>
</xsd:complexType>
<xsd:complexType name="pushCancellation">
  <xsd:annotation>
    <xsd:documentation>
      Empty by purpose
    </xsd:documentation>
  </xsd:annotation>
</xsd:complexType>
<xsd:complexType name="pushQuery">
  <xsd:attribute name="response-code"
type="cdr:pushQueryResponseCode" use="optional"/>
</xsd:complexType>
<xsd:simpleType name="recordStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="start" />
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:enumeration value="stop" />
<xsd:enumeration value="intermediate" />
<xsd:enumeration value="single" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="connectionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="connection-oriented" />
    <xsd:enumeration value="secure-connection-oriented" />
    <xsd:enumeration value="connectionless" />
    <xsd:enumeration value="secure-connectionless" />
    <xsd:enumeration value="unknown" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="mimeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[a-zA-Z]*(/.)[a-zA-Z]*" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="bearerType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Any" />
    <xsd:enumeration value="CDPD" />
    <xsd:enumeration value="CSD" />
    <xsd:enumeration value="FLEX" />
    <xsd:enumeration value="GHOST/R_DATA" />
    <xsd:enumeration value="GPRS" />
    <xsd:enumeration value="GUTS/R-Data" />
    <xsd:enumeration value="MPAK" />
    <xsd:enumeration value="Packet Data" />
    <xsd:enumeration value="ReFLEX" />
    <xsd:enumeration value="Reserved" />
    <xsd:enumeration value="SDS" />
    <xsd:enumeration value="SMS" />
    <xsd:enumeration value="USSD" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="wresultType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="successful" />
    <xsd:enumeration value="failed" />
    <xsd:enumeration value="unknown" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="priceType">
  <xsd:sequence>
    <xsd:element name="value" type="xsd:int" />
    <xsd:element name="currency">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:length value="3" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="priorityType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="high"/>
    <xsd:enumeration value="medium"/>
    <xsd:enumeration value="low"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="wapPapStatusCode">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:totalDigits value="4"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="deliveryType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="unconfirmed-pi"/>
    <xsd:enumeration value="confirmed-push-success"/>
    <xsd:enumeration value="confirmed-push-failure"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="messageStateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="rejected"/>
    <xsd:enumeration value="pending"/>
    <xsd:enumeration value="delivered"/>
    <xsd:enumeration value="undeliverable"/>
    <xsd:enumeration value="expired"/>
    <xsd:enumeration value="aborted"/>
    <xsd:enumeration value="timeout"/>
    <xsd:enumeration value="cancelled"/>
    <xsd:enumeration value="unknown"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="pushQueryResponseCode">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Successful query"/>
    <xsd:enumeration value="Query denied"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="mediaDetail">
  <xsd:attribute name="mime-type" type="cdr:mimeType"/>
  <xsd:attribute name="count" type="xsd:positiveInteger"
default="1"/>
</xsd:complexType>
</xsd:schema>
```

## Appendix C: Gprs-charging-send.cdr

```
<?xml version="1.0" encoding="utf-8" ?>
<cdr:cdr xmlns:cdr="http://tmms.example.org/Charging"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tmms.example.org/Charging TMMS-
Charging-GPRS.xsd"
  cdr:recording-entity="sip:pr1.tmms.example.org"
  cdr:cdr-id="42"
  cdr:chargeable-operation-id-number="0"
  cdr:timestamp="2005-05-26T12:00:00+02:00">
  <cdr:record-send
    cdr:sender-id="sip:user1@tmms.example.org"
    cdr:connection="secure-connectionless"
    cdr:partial-record-sequence-number="1"
    cdr:content-type="application/x-tmms.cdr"
    cdr:transaction-id="1"
    cdr:bearer="GPRS"
    cdr:header-volume="90"
    cdr:data-volume="2048"
    cdr:iresult="200"
    cdr:wresult="successful">
    <cdr:media-description cdr:mime-type="text/plain" cdr:count="2"/>
    <cdr:media-description cdr:mime-type="video/mpeg"/>
    <cdr:media-description cdr:mime-type="audio/punk"/>
  </cdr:record-send>
</cdr:cdr>
```

## Appendix D: mm-charging.msg

```
MESSAGE sip:TMMS PR1.example.org SIP/2.0
From: sip:user1@example.org;tag=2091110539
To: sip:user2@example.org
Call-ID: 1571773536@example.org
CSeq: 1 MESSAGE
Max-Forwards: 70
Via: SIP/2.0/UDP TMMS PR1.example.org:5060;branch=z9hG4bK1571773536
MIME-Version: 1.0
Content-Length: <total body length>
Content-Type: multipart/mixed; boundary="abcd1234"
Content-Description: Multimedia message with Charging Data Record

--abcd1234
Content-Type: application/x-tmms
Content-Length: <length of message>

<message>
--abcd1234
Content-Type: application/x-tmms+xml
Content-Length: <length of cdr>

<?xml version="1.0" encoding="utf-8" ?>
<cdr:cdr xmlns:cdr="http://tmms.example.org/Charging"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://tmms.example.org/Charging TMMS-
Charging-GPRS.xsd"
  cdr:recording-entity="sip:TMMS PR1.example.org"
  cdr:cdr-id="42"
  cdr:chargeable-operation-id-number="0"
  cdr:timestamp="2005-05-26T12:00:00+02:00">
  <cdr:record-send
    cdr:sender-id="sip:user1@tmms.example.org"
    cdr:connection="secure-connectionless"
    cdr:partial-record-sequence-number="1"
    cdr:content-type="application/x-tmms.cdr"
    cdr:transaction-id="1"
    cdr:bearer="GPRS"
    cdr:header-volume="90"
    cdr:data-volume="2048"
    cdr:iresult="200"
    cdr:wresult="successful">
    <cdr:media-description cdr:mime-type="text/plain" cdr:count="2"/>
  </cdr:record-send>
</cdr:cdr>
--abcd1234--
```



## Appendix E: **TMMSCient-IMS.jad**

```
MIDlet-Jar-URL: http://tmms.example.org/client/TMMSCient-IMS.jar
MIDlet-Jar-Size: 107304
MIDlet-Name: TMMSCient
MIDlet-Description: Client for using the TMMS service which stores
incoming messages and delivers them when requested.
MIDlet-Vendor: Andreas Haeber and Lene Beate Longvastoel
MIDlet-Version: 1.0
MicroEdition-Configuration: CLDC-1.1
MicroEdition-Profile: MIDP-2.0
MIDlet-Info-URL: http://student.grm.hia.no/master/ikt05/ikt590/g11
MIDlet-1: TMMSCient-IMS, , TMmsClient.IMS
MIDlet-Permissions: javax.microedition.io.Connector.http,
                    javax.microedition.io.Connector.sip
```

## Appendix F: **InitialFilterCriteria.xml**

```
<InitialFilterCriteria>
  <Priority>9999</Priority>
  <TriggerPoint>
    <ConditionTypeCNF>0</ConditionTypeCNF>
    <SPT>
      <ConditionNegated>0</ConditionNegated>
      <Group>0</Group>
      <Method>MESSAGE</Method>
    </SPT>
    <SPT>
      <ConditionNegated>0</ConditionNegated>
      <Group>0</Group>
      <SessionCase>1</SessionCase>
    </SPT>
  </TriggerPoint>
  <ApplicationServer>
    <ServerName>sip:TMMS PR@example.org</ServerName>
    <DefaultHandling>0</DefaultHandling>
  </ApplicationServer>
</InitialFilterCriteria>
```