DIPLOMA THESIS

Integration of a Proof of Delivery in a WAP Content Billing Architecture for Third Party Services

Winter Term 2002/2003

Mathias Kienholz, February 2003

Tutor: David Watrin, Swisscom Innovations
Supervisor: Prof. Dr. Burkhard Stiller, TIK ETHZ

Abstract

To compete on the market, an operator of a mobile communication network needs to find new service opportunities and to offer a better QoS and a better customer care compared to his business rivals.
A good opportunity to offer added value to the customers is the so-called third party business:
This allows content providers offering their content under a brand of their own. The network operator does the billing on behalf of the content provider and keeps a certain percentage or a fixed amount of the revenue. The advantage for the customer is that he gets only one bill. The network operator can offer interesting services to his customers, without the need to implement them and the third party benefits from the experience of the network operator in billing and from its customer base.

The services are delivered to the customer over a GSM/GPRS or in the future over an UMTS network. On the way to the customer, the air interface is the weakest part: A lot of bit errors occur, the signal fades depending on the distance to the base station, bad weather conditions and fast moving terminals can influence the transmission. A critical situation is also the handover, especially if already a lot of users are connected to a cell. Last but not least also the user behaviour has an impact on the service delivery, for instance if the battery is empty.

These problems are drivers to implement a “Proof of Delivery” (PoD). The network operator and also the third party (at least for expensive services) like to know, if the service has arrived at the customer’s handset. The network operator can store the PoD to be ready if there are customer requests or claims. But the PoD can also be used as a proactive instrument to offer discounts to the customer’s account in case of problems during a service delivery and so to increase the customer satisfaction.
But the highest customer satisfaction can still be reached by a reliable service delivery. Discounts should only be a measure in exceptional cases.

The considered services in the scope of this thesis are downloads and streams. The PoD for downloads can easily be generated just after the arrival of the last bit belonging to the content at the mobile device. The content is afterwards used locally on the handset. Streaming services instead are transmitted and reproduced in parallel and therefore they have to be investigated during the whole delivery. It isn’t enough to look just at the last bit, if there are a lot of interruptions during the stream.

Four different implementation variants for the PoD are discussed within this thesis:

1. A manual feedback of the customer by clicking on a hyperlink (i.e. HTTP GET request) or by entering a code to unlock the content before it can be used locally on the handset.
2. Looking at end-to-end protocols on the server side (e.g. TCP or RTSP). The states of the protocols can help to decide, if the content has arrived at the client.
3. Using probes to sniff the traffic to the customer or the control flow back to the server. Access to the server of the third party is not needed with this solution.
4. Using a software agent on the mobile device, which automatically generates the PoD message and sends it back to the network operator.

The last variant seems to be the best and also the most interesting from the technical point of view. Java provides already a solution, which goes in this direction for the download of MIDlets.
The software agent is installed on the device (needs to support J2ME) before the client uses the third party services of the network operator the first time. Together with each service delivery a Java class is sent to the device. The agent on the device generates then with the service specific information out of this class the PoD and sends it via an HTTP request to a server of the network operator. A lot of technical details need to be considered by implementing such an agent. It is also very important to communicate properly the advantages (possible discounts) to the customers, because people usually don’t like if data is sent from their devices back to a server.

The integration of the event “PoD” in the billing process can be considered more or less independently from the implementation. It is assumed that the PoD arrives at the system and contains the needed information, i.e. mainly the delivery status (ok, not ok, partially ok) and the reason for a possible failure (handset, network, server). The user gets for instance a discount if the delivery isn’t ok and it isn’t due to a failure on the handset. The discounts influence of course also the settlement between the network operator and the third party. The revenue shares are reduced and it is even thinkable that the two parties have to pay a fine respectively if a certain percentage of the deliveries fail due to their respective errors. The discounts have to be clearly declared to the subscribers and statistics have to be collected concerning the revenue loss of the network operator and of the TP.

Zusammenfassung

Obwohl der Internetboom fürs Erste vorbei zu sein scheint und nicht wenige Telekommunikations-Unternehmen den Kater spüren nach den durch grenzenlos scheinende Euphorie geprägten UMTS-Auktionen, sind die Anbieter mobiler Kommunikationsdienste weiterhin daran interessiert, durch neuere und bessere Dienste Mehrwerte für ihre Kunden zu schaffen.
In diese Richtung geht das sogenannte Third Party (TP) Business: Ein Content Provider kann unter der eigenen Marke Dienste verkaufen (z.B. News-Services, Spiele oder Streams etc.). Diese Dienste sind kostenpflichtig. Allerdings stellt der Content Provider dem Kunden nicht direkt eine Rechnung aus, sondern der Netzwerkanbieter übernimmt das integriert in seine regulären Rechnungen. Für die Übernahme des Billings durch den Netzwerkanbieter zahlt die TP einen bestimmten Preis.

Die Inhalte des Content Providers werden über das Mobilfunknetz (GSM/GPRS oder in absehbarer Zeit UMTS) übertragen. Es ist allerdings nicht garantiert, dass der Content auch wirklich zum Kunden ausgeliefert werden kann. Der unzuverlässigste Abschnitt auf dem Weg zum Kunden ist die Funkverbindung ab der Basisstation. Hohe Bitfehler-Raten, ungleiche Signalstärken innerhalb der Zellen, Probleme beim Handover, aber natürlich auch sich schnell bewegende Kunden (Zug, Auto) sowie Geländehindernisse (Tunnels, Hügel) oder Wettereinflüsse, genauso wie Benutzerverhalten (Batterie leer) können die Auslieferung der Dienste verhindern oder zumindest negativ beeinflussen.

Auf diesen Tatsachen beruht die Idee für einen „Proof of Delivery“ (PoD). Dieser informiert den Netzwerkbetreiber (und evtl. auch die TP), dass der Kunde den Inhalt ausgeliefert erhalten hat. Der PoD kann dann genutzt werden, um bei Kundennachfragen (Reklamationen) die nötige Information zur Hand zu haben. Er kann aber auch proaktiv verwendet werden, um dem Kunden nur effektiv ausgelieferte oder minimalen Qualitätsansprüchen genügende Dienste in Rechnung zu stellen oder Rabatte zu gewähren, was natürlich die Kundenzufriedenheit (und evtl. auch die Nutzungsintensität solcher Dienste) erhöht.

Ein PoD kann nicht für alle Arten von Diensten auf die gleiche Art implementiert werden. Im Rahmen dieser Arbeit wird zwischen Download und Straming-Services unterschieden. Bei ersteren wird der Dienst nachher lokal auf dem Gerät genutzt. Das letzte Bit des Downloads ist ausschlaggebend für den PoD. Streams müssen während der ganzen Dauer untersucht werden, d.h. es reicht nicht, nur das letzte Bit des Streams zu berücksichtigen.

Es werden vier verschiedene Implementationsvarianten untersucht:
1. Manuelles Feedback durch den Kunden (Klick auf Link oder Eingabe eines Freischaltcodes).
2. PoD auf Protokollebene. Die Protokolle (TCP oder RTSP) serverseitig betrachten.
3. Einsatz von Packet Sniffern, um den Daten- oder Kontrollfluss zu untersuchen.
4. Automatisches Feedback durch einen Software-Agenten auf dem Gerät des Kunden.

Variante 4 scheint die sinnvollste zu sein, da sie unabhängig vom Kunden und vom Server der TP ist.
Der Software Agent wird auf das Mobiltelephon (MIDP 1.0 und CLDC) geladen, bevor das erste Mal ein Dienst genutzt wird. Mit jedem Content wird eine Klasse mit service-spezifischen Parametern mitgeliefert, mit deren Hilfe der Agent die PoD Meldung generieren kann, welche anschliessend via HTTP Request an den Netzwerkbetreiber geschickt wird.
Es gilt dabei, etliche Details zu berücksichtigen und auch zu bedenken, dass die Kunden nicht gerne haben, wenn Daten ohne ihr Zutun von ihren Geräten aus verschickt werden („Big Brother“). Der ganze Ablauf muss positiv kommuniziert werden und klar als Vorteil für den Kunden deklariert sein!

Die Integration des PoDs in die Billing Architektur kann unabhängig von der Implementation behandelt werden. Es wird angenommen, dass der PoD im System ankommt und die nötigen Informationen enthält, v.a. den Delivery Status (ok, not ok, teilweise ok) und die allfällige Fehlerursache (Endgerät, Netzwerk, Server). Ist Auslieferung beispielsweise nicht ok und liegt der Fehler nicht beim Endgerät, erhält der Kunde einen Rabatt. Dieser wirkt sich dann natürlich auch auf die Einnahmenverteilung zwischen dem Netzwerkbetreiber und der TP aus. Beide erhalten reduzierte Geldbeträge. Es ist denkbar, dass die beiden Parteien untereinander auch Bussen zahlen müssen, wenn eine bestimmte Anzahl der Dienste aufgrund jeweiliger Fehler nicht bei den Kunden ankommt. Die allfälligen Rabatte müssen auf der Rechnung deutlich ersichtlich sein. Ebenso müssen Statistiken geführt werden, die es dem Netzwerkbetreiber erlauben, die Einnahmeverluste aufgrund der Rabatte zu bestimmen.

Preface

This diploma thesis has been developed at Swisscom Innovations in Bern to conclude my studies in Information Technology and Electrical Engineering at the Swiss Federal Institute of Technology (ETH) in Zurich.
The task for this thesis has been verbalised by Swisscom Innovations and has been accepted by the Computer Engineering and Networks Laboratory at the ETH.

My goal was to do a practical thesis in a telecommunication enterprise with the intention to have the opportunity to work in a possible future professional environment and mainly to gather further practical experiences.

The thesis is kept as much general as possible. Most of the parts of this thesis have been elaborated independently of Swisscom specific requirements. There are only a few chapters, which have been really focussed on Swisscom and their needs.

At the beginning there was a longer period of literature studies to get into the topic and to look for related work. With this background the first attempts to design a concept for the Proof of Delivery have been started. Paper and pencil helped to brainstorm and to settle the requirements to the PoD.
These ideas have been concretised afterwards and the technical feasibility (of the different implementation variants) has been studied.
During the second phase of this thesis the focus has been laid on one implementation variant (software agents) and on the treatment of the event “PoD” in the billing stream.
This report has been developed continuously during the elaboration of this thesis, so that there hasn’t been too much stress during the last weeks of the four months at Swisscom Innovations.

It has been an interesting and a good time working in the team of Swisscom Innovations. I think that I have learnt a lot and have made some good and helpful experiences. It is clear that a thesis can probably not be developed as freely as it could be done in an ETH lab. Nevertheless I had always my liberties and I could also always bring in my ideas and work independently.

I’d like to thank very much David Watrin from Swisscom Innovations. He has been my tutor, organised the thesis, supported me with useful hints and enabled me to establish contacts to other persons with specific knowledge. I enjoyed the agreeable collaboration with him!
I’d also like to thank Prof. Dr. Burkhard Stiller from the ETH, who has been the supervisor of the thesis, who has steered the schedule in the background and has also always provided helpful clues.

Last, but not least, I’d also like to thank my parents, who facilitated me my studies in Zurich, which now have been terminated and also all my friends and my girlfriend for the funny and nice time beside the work-intensive moments during the studies or this thesis.

Table of Contents

Abstract ... III
Zusammenfassung ... V
Preface ... VII
Table of Contents ... IX
Index of Figures ... XI
Index of Tables ... XI
Abbreviations ... XII
1 Introduction ... 1
1.1 Scope of the Thesis ... 1
1.1.1 Global System for Mobile Communication (GSM) ... 1
1.1.1.1 High Speed Circuit Switched Data (HSCSD) ... 2
1.1.2 General Packet Radio Service (GPRS) ... 2
1.1.3 Wireless Application Protocol (WAP) ... 3
1.1.4 Universal Mobile Telecommunication System (UMTS) ... 4
2 Services and Quality of Service (QoS) ... 5
2.1 Mobile Services ... 5
2.2 Quality of Service (QoS) ... 6
2.3 Wireless QoS ... 7
3 Billing ... 9
3.1 Terms ... 9
3.1.1 Aggregation ... 9
3.1.2 Billing ... 9
3.1.3 Charging ... 9
3.1.4 Data Collection ... 10
3.1.5 Mediation ... 10
3.1.6 Payment ... 10
3.1.7 Pricing ... 10
3.1.8 Rating ... 10
3.2 Third Party Business ... 10
3.3 Swisscom Mobile WAP Content Billing ... 12
3.3.1 Application Flow for WAP Content Billing ... 13
3.3.2 Proof of Order ... 13
3.3.3 Proof of Delivery ... 14
3.3.4 Service Data Records ... 14
4 Proof of Delivery - Fundamentals ... 15
4.1 Introduction ... 15
4.2 Assumptions ... 15
4.3 Reasons for a Proof of Delivery ... 16
4.3.1 Bit Errors ... 16
4.3.2 Bandwidth Limitations ... 16
4.3.3 Resources in the Device ... 17
4.3.4 Handover ... 17
4.4 Related Work ... 18
4.4.1 Confirmed Download ... 18
4.4.1.1 Nokia OTA Download for Generic Content ... 19
4.4.1.2 OMA Generic Content Download Over The Air ... 20
4.4.2 Other Work ... 21
4.4.2.1 Project SOQUET ... 21
5 Proof of Delivery – Design ... 23
5.1 Service Categories ... 23
5.1.1 Atomic and Non-atomic Services ... 23
5.1.2 1-phase and 2-phases Services ... 24
5.1.2.1 2-phases Services ... 24
5.1.2.2 1-phase Services ... 25
5.1.2.3 Special Services ... 25
5.1.2.4 Services with States ... 26
5.1.2.5 From the Theory to the Practice ... 26
5.2 Implementation Variants of a Proof of Delivery ... 27
5.2.1 Explicit Feedback from the Subscriber ... 27

5.2.1.1 Clicking on a Hyperlink ... 27
5.2.1.2 Activation Code ... 28
5.2.2 Automatic Feedback generated by an Agent on the Device ... 28
5.2.2.1 Agent Basics ... 28
5.2.2.2 Agent Platform ... 29
5.2.2.3 Implementation of the PoD ... 29
5.2.3 Proof of Delivery using Protocols ... 30
5.2.3.1 Transmission Control Protocol (TCP) ... 30
5.2.3.2 Real Time Streaming Protocol ... 30
5.2.4 Proof of Delivery using Network Probes ... 32
5.2.4.1 Network Probes ... 32
5.2.5 Pros and Cons of the different Solutions ... 33
5.2.5.1 Explicit Feedback from the Subscriber ... 33
5.2.5.2 Automatic Feedback generated by Agents on the Device ... 33
5.2.5.3 Proof of Delivery using Protocols ... 34
5.2.5.4 Proof of Delivery using Network Probes ... 34
5.3 Conclusion ... 34
6 Proof of Delivery with Software Agents ... 35
6.1 Java 2 Micro Edition ... 35
6.1.1 Configuration ... 35
6.1.2 Mobile Information Device Profile (MIDP) ... 36
6.2 Agent based Solution for the PoD ... 36
6.2.1 Agent Projects ... 37
6.2.1.1 Project Sniff: Java Mobile Agent for Wireless Devices ... 37
6.2.1.2 Project Smart Agent ... 37
6.3 PoD with the Smart Agent ... 38
6.3.1 Requirements ... 38
6.3.2 Implementation ... 38
6.3.3 Problems and Challenges ... 39
7 Integration of the PoD in the Billing Stream ... 41
7.1 Objectives ... 41
7.2 Structure of the PoD ... 42
7.2.1 Identifier ... 42
7.2.2 MSISDN ... 42
7.2.3 Third Party ID ... 42
7.2.4 Service ID ... 42
7.2.5 Transmission Medium ... 42
7.2.6 Delivery Status ... 42
7.2.7 Failure Reason ... 43
7.2.8 Responsibility ... 43
7.2.9 Timestamp ... 43
7.2.10 Associated Proof of Order ... 43
7.2.11 Flags ... 43
7.3 Discounts for the End Customer ... 44
7.4 Settlement Rules ... 45
7.5 Integration of the PoD in the Architecture of SCM ... 46
7.5.1 Data Flow ... 46
7.5.1.1 Common Billing Stream ... 47
7.5.1.2 Customer Billing Stream ... 48
7.5.1.3 Partner Billing Stream ... 49
7.5.2 Recommendations ... 49
8 Conclusions and Outlook ... 51
8.1 Implementation ... 51
8.1.1 Interaction between Agent and Mobile Device ... 51
8.1.2 Reliability ... 51
8.1.3 Usability Studies ... 51
8.2 PoD in the Billing stream ... 52
References ... 53

Index of Figures

Figure 1: GPRS Architecture [SiemensGPRS] 2
Figure 2: The GPRS protocol stack [Acterna] 3
Figure 3: The WAP model [WAPForum] 3
Figure 4: The evolution of mobile services [NokiaBusiness] 6
Figure 5: Mediation and billing process [Uni-x] 9
Figure 6: Third Party Business Case 1 11
Figure 7: Third Party Business Case 2 11
Figure 8: Part of the Swisscom Mobile WCB Architecture 12
Figure 9: The WCB disclaimer 13
Figure 10: Mobile IP packet flow [Ericsson] 18
Figure 11: COD Schema [NokiaOTA] 19
Figure 12: Possible approach for a categorisation of services 23
Figure 13: Schema of 2-phases services 25
Figure 14: Schema of 1-phase services 25
Figure 15: The activation code on the display 28
Figure 16: PoD generated by a software agent on the device 28
Figure 17: The RTSP in use 32
Figure 18: The various Java editions [Muchow] 36
Figure 19: Sniff mobile agent 37
Figure 20: PoD with a Smart Agent 38
Figure 21: The integration of the PoD in the billing process of Swisscom Mobile 41
Figure 22: Settlement between the network operator (Swisscom) and the third party 45
Figure 23: The data flow for the PoD integration in the content billing 46
Figure 24: The Common Billing Stream 47
Figure 25: The Customer Billing Stream 48
Figure 26: The Partner Billing Stream 49

Index of Tables

Table 1: Examples of possible mobile services 5
Table 2: Explicit feedback from the subscriber 33
Table 3: Automatic feedback generated by agents 33
Table 4: PoD using protocols 34
Table 5: PoD using network probes 34
Table 6: The structure of the PoD 42

Abbreviations

3G	Third Generation (of mobile communication)
AAA	Authentication, Authorisation and Accounting
BER	Bit Error Rate
BSS	Base Station Subsystem
BTS	Base Transceiving Station
CDC	Connected Device Configuration
CLDC	Connected Limited Device Configuration
FDMA	Frequency Division Multiple Access
GGSN	Gateway GPRS Support Node
GUI	Graphical User Interface
GPRS	Global Packet Radio Service
GSM	Global System for Mobile communication
HLR	Home Location Register
HO	Handover
HTML	Hyper Text Markup Language
HSCSD	High Speed Circuit Switched Data
HTTP(S)	Hyper Text Transfer Protocol (over Secure Socket Layer SLL)
IETF	Internet Engineering Task Force
IP	Internet Protocol
J2ME	Java 2 Micro Environment
KVM	Kilobyte Java Virtual Machine
LBS	Location Based Services
MBS	Micro Billing System
MIND	Mobile Information Device Profile
MMS	MultiMedia SMS
MSISDN	Mobile Station ISDN number
MS	Mobile Station
MT	Mobile Terminal
OSS	Operations Support System
OTA	Over The Air
PoD	Proof of Delivery
PoO	Proof of Order
QoS	Quality of Service
RFC	Request For Comments
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SLA	Service Level Agreement
SCM	Swisscom Mobile
SDR	Service Data Record

SMS	Short Message System
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TP	Third Party
TPB	Third Party Business
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication Service
WAP	Wireless Application Protocol
WCB	WAP Content Billing
WCDMA	Wideband Code Division Multiple Access
WML	Wireless Markup Language
XML	Extensible Markup Language

1 Introduction

Although the Internet boom seems to be (temporary) over, the endeavour of the service providers to implement new services and revenue sources in the Internet goes on. The continuous process to improve the existing technologies and to discover new ones, allows more and more to use services, which have been bound to the wired Internet up to now, also in a wireless environment and even to introduce mobility specific services (so-called location based services, LBS).
The catchwords in this context are 3G or UMTS. The euphoria in this area at the beginning of the 21st century induced a lot of network operators (especially in Europe) to pay ruinously prices only for the licences to have the right to build up and maintain an UMTS network. Now, there is a huge hangover and it will take certainly still some time until the UMTS networks can be used all over the countries. Swisscom for instance, the market leader in Switzerland, plans to start the UMTS communication only in 2004 (the first plans have scheduled 2002…)
It is not yet clear, if UMTS will be a success, even though its opportunities seem to be really fascinating. Certainly a lot depends on the willingness of the customers to pay for the new enabled services.

However, the network operators are interested in the success of UMTS and regardless the problems with the launch of UMTS, they try to find new business opportunities. The key challenge, which operators face is the development of value added services for their customers [Aran].
The willingness of the customers to pay for content in the Internet isn't very high, because they are used to free services. In a mobile environment it is easier to charge customers not only for the connection but also for content. The subscribers are used to pay more for mobile communication compared to the communication over wired lines and so the tolerance for premium services is higher [TagAnz].
One of these opportunities, which are already in use, is the so-called third party business, which means that a provider can offer content under an own brand, but the network operator is responsible for the billing on his regular bills. This interaction of the three parties offers advantages to all of them (see chapter 3).

Because the markets are saturated and deregulated, it is important for the network providers to offer new innovative services to diversify their product portfolio against their competitors and to be able to create added value for their customers by assuring a better quality of service than the competitors. For most customers the value of goods or services is based on price and quality!
The error-free delivery of data services (and also of voice) to mobile devices can't be guaranteed. There are a lot of influences (especially in the air interface), which can constrict a satisfying delivery of services to the customer. So it would be valuable for the network operator (and also for the third party) to get information about the status of the service delivery to the customer. This information can help the network operator to steadily improve his infrastructure and also to offer an added value, if he is able to charge the customers only for services, which really have been delivered.

This is the coarse outline of this thesis. It is structured as follows:
The rest of chapter 1 gives a short introduction into the mobile communication, while chapter 2 treats some important aspects of services and their quality (QoS). Chapter 3 introduces the subject billing and explains the third party business. Chapter 4 introduces the main subject of this thesis: the Proof of Delivery (PoD). Chapter 5 contains the conception of this PoD, while chapter 6 treats one possible implementation in particular. It is not only important to investigate, how such a PoD can be implemented in the network, but also how the event PoD is treated in the billing chain. This will be discussed in chapter 7. Chapter 8 contains then a short outlook and names some open questions.

1.1 Scope of the Thesis

The main scope of this thesis is certainly the mobile communication. In the following subchapters present some facts about mobile communication technologies to give the reader a background and also to anchor the main chapters of this thesis.

1.1.1 Global System for Mobile Communication (GSM)

The biggest success in the (short) history of mobile communication is the GSM (Global System for Mobile Communication). Its commercial launch was in 1992. It is also referenced as 2G (standing for "second generation"; the "first generation" is no longer used, e.g. the Natel C network in Switzerland).

GSM is a circuit switched digital communications system, used in more than 180 countries worldwide (April 2002) and serving almost one billion of subscribers [GSMWorld].
Though the GSM system is commonly operated in 900 MHz and 1800 MHz bands, 450 MHz, 850 MHz and 1900 MHz bands are also used. It requires a paired spectrum and supports a carrier bandwidth granularity of 200 kHz. The GSM radio interface uses a combination of Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). The basic GSM system supports voice bearers at 13 Kbps (full rate codec) or 6.5 Kbps (half rate codec) as well as circuit switched data services at maximal 14.4 Kbps [Siemens3G].
The main service of GSM is voice but it is also widely used to send and receive SMS, short messages of 160 characters, routed via the control channel. The success of SMS couldn't have been foreseen, but it shows that people also like to use their mobile devices for data transmission.

1.1.1.1 High Speed Circuit Switched Data (HSCSD)

To achieve faster transmission rates for data communication, the existing GSM has been enhanced with the technology of HSCSD. HSCSD bundles up to 8 channels to be able to transmit at data rates of 57.6 Kbps. But HSCSD was considered only as a temporal solution on the way to 2.5G and 3G mobile networks [Loetscher].

1.1.2 General Packet Radio Service (GPRS)

Also GPRS is an enhancement to GSM. It introduces packet-based transmission over TCP/IP. This allows a mobile subscriber to communicate with foreign data networks such as the Internet or corporate intranets.
One of the main benefits of the packet-switched technology is that the mobile terminals (MT) may be conceived to be always on, allowing incoming and outgoing voice communication to be made via GSM circuit switched channels in parallel with data communication over the packet switched network. Another important feature is that users may be charged based on the amount of data being transmitted, i.e. it is not important for the user how long he looks at a Web page.
The GPRS technology submits an optimised use of radio network resources for several mobile stations by sharing the radio spectrum efficiently. The introduction of new coding schemes offers higher data transmission rates, up to 21.4 Kbps per time slot. By bundling up to eight timeslots separate for uplink and downlink, data rate of up to 170 Kbps can be achieved [SiemensGPRS].
But it has to be considered that 30-40 Kbps seem to be more realistic, because normally only two or three timeslots can be allocated to one user and not eight [Loetscher]. The bandwidth of 30-40 Kbps is sufficient for services like email or limited Internet browsing. With packet switching the radio spectrum efficiency is higher because network resources and bandwidth are only used when data is actually transmitted.

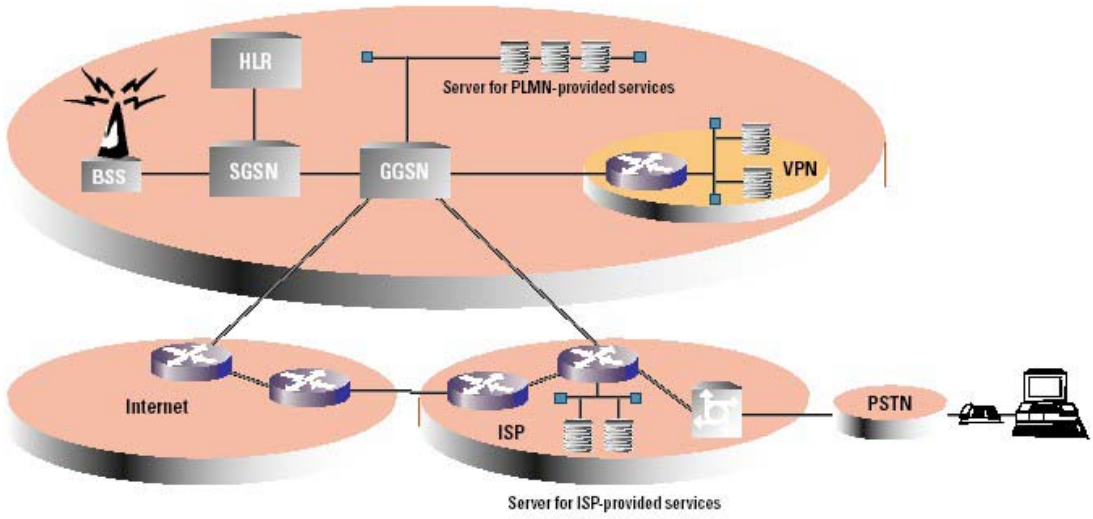

Figure 1: GPRS Architecture [SiemensGPRS]

GPRS introduces two new network elements in the existing GSM architecture, namely the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN):

The SGSN is responsible for the communication to and from the mobile station (within its area). It maintains information about currently attached nodes and performs authentication procedures. It is connected to the base station subsystem over a Frame Relay network on one side and to the GGSN over an IP backbone network on the other side.
The GGSN serves as a gateway to other networks such as public data networks (e.g. the Internet) or other GPRS networks. It is responsible for converting packet addresses from the external packet address format to the GSM address of the mobile terminal and vice versa.

Figure 1 presents a graphical overview: The Mobile Station (MS) is attached to the Base Station Subsystem (BSS), which is connected to the SGSN. The Home Location Register (HLR) stores user profile information and the current location for each user, whose home network is controlled by this HLR. The GGSN is as already mentioned the gateway to other networks.

Figure 2 shows the GPRS protocol stack. It can be seen that the MS gets an IP address and can so send and receive IP packets. This allows also connections over TCP and UDP. But discussing the whole protocol stack of GPRS would go beyond the scope of this thesis. It is mainly important that there are IP end-to-end connections.

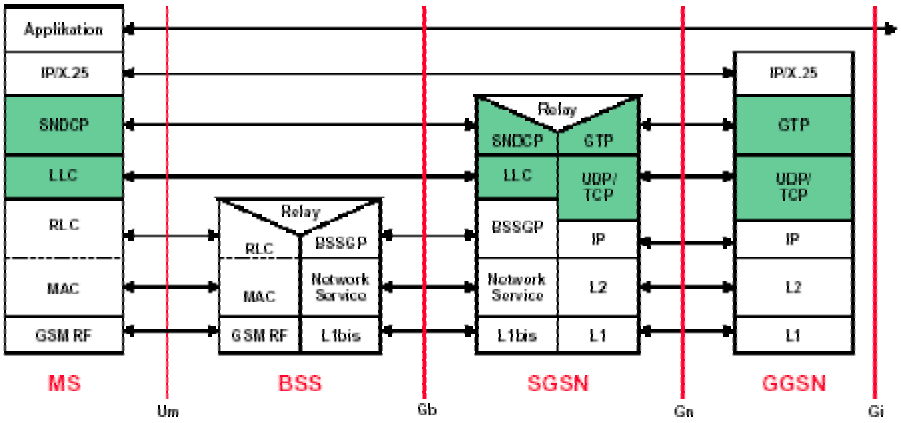

Figure 2: The GPRS protocol stack [Acterna]

1.1.3 Wireless Application Protocol (WAP)

The WAP technology unifies two network technologies: the mobile GSM network and the Internet. Due to the fact that a GSM mobile device has only limited possibilities to implement browsing functions, a new language has been developed do describe web pages. This language is called Wireless Markup Language (WML) and is quite similar to the conventional Hypertext Markup Language (HTML), but takes into account the modest presentation possibilities of the displays of mobile terminals and the limited bandwidth of the GSM network.

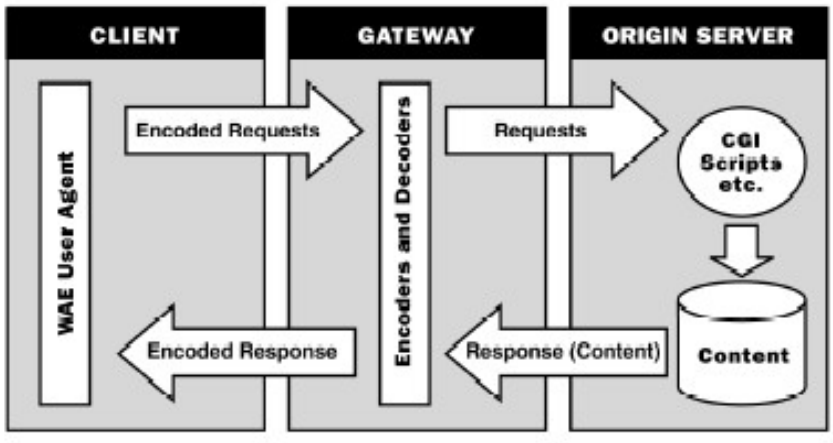

Figure 3: The WAP model [WAPForum]

The WML pages can of course also be transmitted over HSCSD or GRPS with the correspondent higher bandwidths. Even SMS can be used as a bearer for WAP.
The interface between the Internet and the mobile network is called WAP gateway. To get a WML page from the Internet over a circuit-switched GSM-connection, the subscriber has to call the number of the WAP gateway. The gateway establishes a connection to the appropriate web server. HTTP is used between the WAP gateway and the web server. The web server sends then the requested page within the HTTP response to the gateway. The headers and data in an HTTP transfer are too large for wireless users and also a lot of information is redundant. Therefore the data gets binary encoded and is afterwards sent to the mobile device. The WAP browser interprets the binary data and presents the result on the display [UMTSLink, NokiaWAPGPRS]. This process is illustrated in Figure 3.
A GPRS capable WAP client requests for WAP content via an access point (SGSN, GSGN). The GPRS network assigns dynamic IP addresses to WAP clients for temporary use [NokiaWAPGPRS].

1.1.4 Universal Mobile Telecommunication System (UMTS)

UMTS is the European standard of the third generation of mobile communication (3G). The international name for this standard is “IMT-2000” (International Mobile Telecommunication).
UMTS bases on the UMTS Terrestrial Radio Access Network (UTRAN). UTRAN is completely new and differs explicitly from the GSM technology. It allows a more efficient use of the available frequencies and faster data services and permits also a better support for QoS aspects.
The European 3G market will be ruled by the Wideband Code Division Multiple Access (W-CDMA) technology. W-CDMA allows several subscribers to communicate using only one frequency channel. To make this possible, the signal of each subscriber is marked with a binary code, so that the receiver can filter the signal, when he knows the binary code [UMTSlink, Siemens3G].
The maximum transmission rate is 2.05 Mbps, but only 384 Kbps seem to be realistic, when the MS is stationary or moving slowly and even less (128 Kbps) in a faster moving vehicle. Within an UMTS system, the GPRS network provides the packet data capability [LucentGPRS].

After the auctions to buy UMTS licences (in several countries at almost ruinously prices in the background of the Internet hype at the end of the nineties) the telecom operators are currently building up the UMTS networks. These networks are rolled out in steps. Deployment starts in urban areas, where a specific demand for data services is anticipated. In order to provide full coverage for service continuity from the beginning, networks and terminals are designed to enable roaming and handover between GSM/GPRS and UTMS [Siemens3G].
The driving force behind UMTS will be the availability of attractive applications and services. With the higher data rates it should be possible to provide web browsing, streaming, location based services or content download in a satisfying manner and also a better voice quality for normal phone calls.

UMTS in the mass market can only be expected at the end of 2003 or even in 2004. One problem is the lack of reliable end devices. But also the large sums of money, which have been spent by most of the operators to get an UMTS licence, are a problem. Some providers are now having financial problems or are very thoughtful to carefully implement their networks, so that UMTS will be a success from the beginning and doesn’t have starting problems such as WAP [X-media].

2 Services and Quality of Service (QoS)

The main topic of this thesis is the design of a Proof of Delivery (PoD) for third party services. This PoD can confirm just the delivery of the service, but it is also thinkable that it contains some information about the quality of the service. This chapter gives a short overview about services in a mobile context and also an introduction into some QoS aspects.

2.1 Mobile Services

With the increasing bandwidth it is more and more possible to use enhanced services also in a mobile environment and not only on a device attached to the Internet over a wired line.
Of course a mobile phone can also be used to connect a portable computer to the Internet to browse on this computer. But the scope of this thesis concentrates on the usage of services on mobile phones. These devices have smaller screens and will probably always have limited memory and CPU resources compared to desktop or portable computers. However, most of the services can also be used on such devices, only the experience of the service use (in particular for pictures and films) is probably not as pleasant as it is on larger displays.
One kind of service is special in a mobile context: the location based services (LBS). LBS are services that exploit knowledge about where a device user is located. So the user can be provided for instance with event notifications or traffic information depending on his current position. But beside the location specific content these services aren't different from "normal" services.

The following table contains an outline over the main services and some examples, which seem to make sense in a mobile environment. It isn't sure if the "killer application", which should become a "gold donkey" (if there would be one at all) for the providers, is covered by this list.

Service category	Examples
Communication	• Email • Multimedia SMS • Videoconferencing • Pinboard
Information	• News (Weather, politics, finance, sports etc.) • Events announce (e.g. location based) • Navigation (but also traffic info, schedules etc.) • Translator • Remote monitoring
Entertainment	• Games • Video clips • Music • Dating services
M-commerce	• M-shopping • M-banking • M-ticketing
Personal Information Manager	• My calendar • My reminder • My to-do-list

Table 1: Examples of possible mobile services

The mobile terminal is fast becoming a personal device that meets user needs for personalised, timely and relevant information and communication. Figure 4 contains an illustration of the evolution of mobile services. This representation appears to be realistic and not exaggerated. On the Web still a lot of documents can be found, which seem to have their origin in the times of the absolute Internet hype, when also the euphoria about the upcoming UMTS seemed to be boundless. However, times have changed, the expectations are more modest compared to the ones two or three years ago. Nevertheless, new services and new opportunities for the users of mobile communications will be established in the near future.

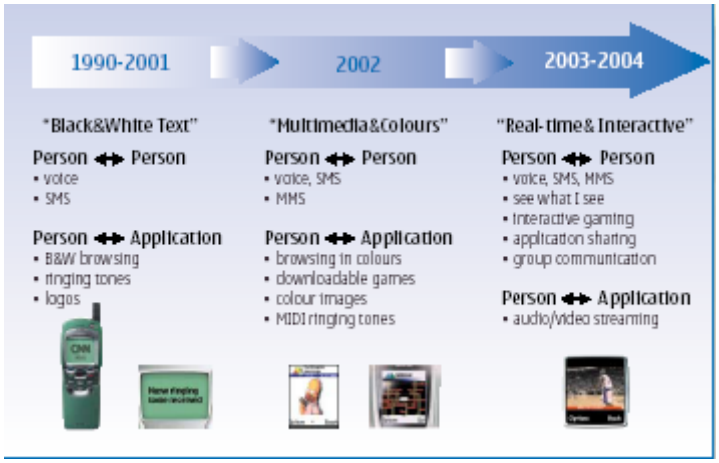

Figure 4: The evolution of mobile services [NokiaBusiness]

2.2 Quality of Service (QoS)

Quality of Service is a broad term used to describe the overall experience of a user or an application receiving a service over a network [NortelQoS].
From the end-user's point of view, QoS defines the characteristics of service delivery that impact most critically his perception of the service. QoS is in fact something very subjective; the perception of the quality may differ from one user to another.
The three main characteristics of the QoS of a service experienced on a subjective level by the user are [NortelQoS]:

- **Availability of the service**
  The service is immediately available or only reasonably delayed.
- **Quality of information**
  The information is received uncorrupted and usable without any errors.
- **Consistent delivery**
  The information is delivered throughout the session at a perceivably consistent speed and quality.

On a more technical level, QoS can be characterised by a set of measurable parameters [NortelQoS, Nortel3GQoS]:

- **Network availability**
  The availability of the network can have a significant affect on QoS. Only short periods of network outage can cause unpredictable influence on the performance perceived by the user. Redundancy in the network architecture helps to ensure a higher availability.
- **Delay**
  The network delay is the transit-time a service experiences from the ingress point of a network to the egress point of the network. Delay can cause significant QoS issues with services such as voice or video/audio streams.
  Delay can be fixed or variable. Fixed delay is for instance the application-based delay (codec processing time or the packetisation delay) and the propagation delay depending on the transmission distance. Variable delay may be the ingress or egress queuing delay for traffic entering or exiting a network node.
- **Delay variation (Jitter)**
  Jitter is the measure of delay variation between consecutive packets for a given traffic flow. Jitter is of particular importance for real-time applications. They expect to receive packets at a fairly constant rate with fixed delay between consecutive packets. All networks introduce some jitter (variable queuing times in the network nodes). But up to a certain level jitter is tolerable and by using a clever buffering there are also means to live with these variations.
- **Packet loss rate**
  Packet loss can occur due to errors introduced by the physical transmission medium. Wired lines have usually a very low bit error rate (BER) compared to wireless connection such as

satellite or mobile networks. The BER may there even vary due to environment or geographical conditions (physical obstacles, weather conditions etc.).

- **Throughput / Bandwidth**
  Throughput is the effective amount of data passing the network within a timeframe. Bandwidth is the theoretical available amount of data, which could pass.

From the fixed network point of view, the easiest way to provide a certain quality of service is to ensure that sufficient bandwidth is available all times to all users and all services. But this leads to an inefficient use of network resources and to excessive costs. The real challenge is to manage the existing network bandwidth in such a way that appropriate service levels can be delivered to all users all the time [NokiaE2E].

The fact behind all these assumptions about QoS is that the Internet relies on the Internet Protocol (IP). IP only provides a best-effort service to the higher protocol layers. Best-effort means that there is no guarantee that packets arrive in order, in time or even at all at their destination.
There are already a lot of measures and mechanisms to assure QoS in the Internet. Examples are Integrated Services, Differentiated Services, Resource Reservation Protocol (RSVP) or Multi-Protocol Label Switching (MPLS).
It would go beyond the scope of this thesis to discuss these mechanisms. A lot of material about these themes can be found on the WWW.

2.3 Wireless QoS

Generally, the QoS is not something more special for a service transmitted over one or more wireless networks on its way to the client. What is really important and what is of interest for the user is the so-called end-to-end QoS. The QoS perceived by the client is only as good as it is on the weakest link. The weakest link in the wireless environment is the air interface. It is the most bandwidth-limiting factor in the network. Radio is also a hostile medium for access, with problems arising from error-prone and time-variant link characteristics [NokiaE2E]. Special situations in a mobile environment are the handovers (HO), if a moving mobile terminal gets attached to another base station. HO associated parameters, of spatial (e.g. HO rate per cell) or temporal (HO rate per call) significance, play a dominant role in the performance of mobile networks. In addition the user experiences inconsistent delays, depending on the point, where he is located relative to the base station antenna. Data rates cannot be evenly distributed across the entire base-station coverage area [QoSMobileIP].
These facts will be discussed in more details in chapter 4.3 of this thesis, because they are all reasons, which justify implementing the Proof of Delivery.

Table 1 (chapter 2.1) lists some examples of mobile services. They are categorised based on their purpose. From a more technical point of view, the categorisation can also be done as follows [NokiaE2E, NortelQoS]:

- Rich call (voice, video conferencing)
- Streaming (e.g. infotainment)
- Browsing (Internet/intranet access)
- Messaging (multimedia, email)
- M-commerce transactions

What are missing here are services, which need a download (e.g. a video clip, which is not delivered over a stream) and then are reproduced locally on the handset.
Applications such as rich call and streaming are real-time by nature, involving periodic transmissions of information in the network.
Transaction-type applications such as M-commerce and Internet browsing produce less regular traffic patterns, but nevertheless pose delay limits for transmission, thanks to their interactive nature.
Interactive real-time applications, such as voice or video conferencing, have the most stringent end-to-end delay requirements. Such applications typically are UDP-based and cannot retransmit lost or dropped packets as TCP-based applications, because retransmission wouldn't be beneficial (causes additional delay).
Interactive transaction traffic types such as browsing are more tolerant to longer delays.
For streaming, delays up to some seconds are acceptable.
Instant messaging benefits from short delays, but in email delivery delay is not an important issue.

Anyway, too much delay is never desirable. The best content becomes worthless if it is delivered too slowly or too late: According to Zona Research the delay (for web browsing) becomes critical if it exceeds 7-8 seconds [Zona]. Then the users usually interrupt the page download and look for another page.

3 Billing

The chapter before acts as an introduction into services and their quality. These services have a price and the user is charged if he uses them. A Proof of Delivery informs the network operator if the service has been delivered to the client. This information can have an impact on the invoice sent to the subscriber (mainly if there was a problem during the delivery). This thesis not only treats the possible ways to implement a PoD on the network level but also how an event “PoD” can be treated in the billing chain. Therefore this chapter contains a short introduction in the billing world to give the reader a background for the chapter 7.

3.1 Terms

There are a lot of different terms in the context of billing. It is important to distinguish between them [Uni-x, Loetscher, Valtakari, Whatis].

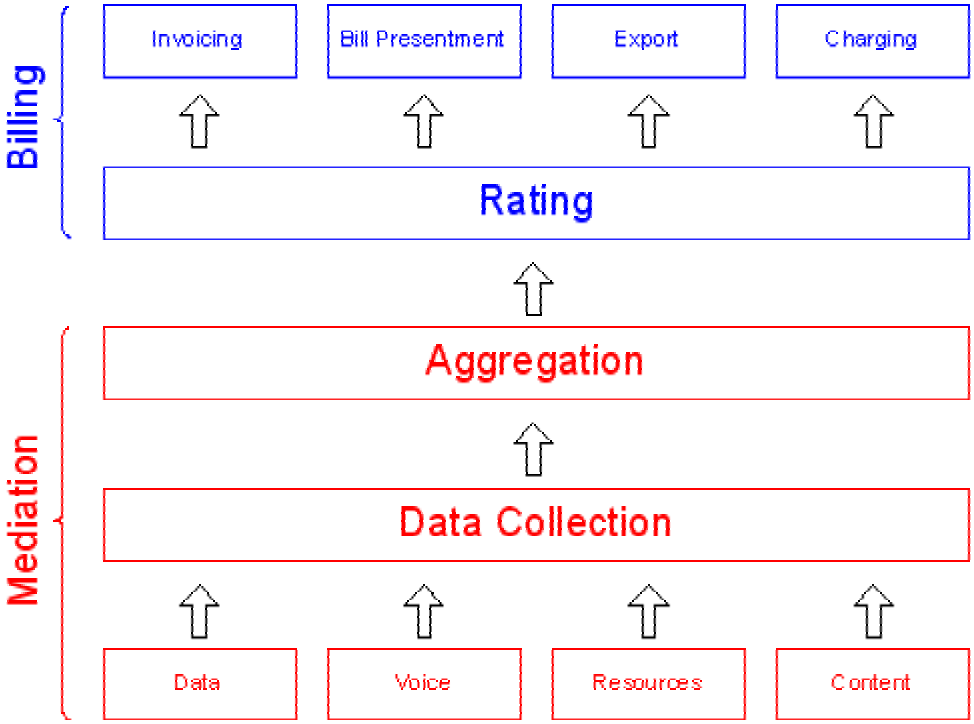

Figure 5: Mediation and billing process [Uni-x]

3.1.1 Aggregation

Aggregation names any process, in which information is gathered and expressed in a summary form, for purposes such as statistical analysis.

3.1.2 Billing

Billing is a generic term for all processes for the calculation of the usage of services. Billing covers beside the equitable allocation of the costs the creation of the invoice (rating), the delivery of the bill and also the dunning process.

3.1.3 Charging

Charging stands for determining the relevant activities for the invoice and for transferring the collected data to a system, which generates the invoices.

3.1.4 Data Collection

The data collection measures the resources a user consumes during access. This can include the amount of system time or the amount of data sent and/or received during a session. Data collection is carried out by logging of session statistics and usage information and is used for authorisation control, billing, trend analysis, resource utilisation, and capacity planning activities.

3.1.5 Mediation

Mediation terms all processes, which collect and format the data of utilisation and store them afterwards in a standardised format in a database.

3.1.6 Payment

Payment determines the process of exchanging monetary values for the utilisation of a resource or a service.

3.1.7 Pricing

Pricing determines prices for services. A pricing scheme contains prices for each service that is going to be billed. The most usual pricing schemes are time-based (client pays depending on the time of the service use) or volume-based (client pays depending on the sum of the transferred data). Well-known is also a flat rate pricing, where the client pays only once and can then use the service as much as he wants (probably within some restrictions, e.g. a limitation of the data volume or a limited access time).

3.1.8 Rating

The idea of rating is to process the collected data simply and fast according to usage prices. Rating gets the records in date order and processes them in one pass. It outputs rating records, which are a list of customer activities with a regular cost.

3.2 Third Party Business

As an example, some services are mentioned in chapter 2. They are all content services. The customer gets delivered data (a content), which is of use for him. This content is transported over a network. Also this transport is a service; a service of the network operator, which offers its customers network access (for a certain price bound to certain conditions).
Relevant for this thesis are only content services. It is thinkable that the network operator acts also as a provider for such services. This means that he offers to its customers (perhaps also to subscribers of other network operators) beside the network access also the possibility to download ring tones or weather forecasts as an example.

But often enterprises or institutions offer services under their own brand. They use the network of the operator as a transport medium to be able to deliver the content to their customers. Of course they want the customers to pay for this content. This could happen by sending an invoice to the customer after he has used a service and the user would then pay his fee directly to the content provider. Beside this fee he pays the normal invoice from his network provider for the network access.

In another potential business model the network operator buys the content from the content provider and sells it then to its customers. The customer pays the network operator for the network access, but also for the content. In this case, both, the billing system and the content server are located at the network operator. He has the complete control over the needed information for the billing process. This model has advantages for all, the network operator, the content provider and also the customer.
The network operator can offer to his clients more services to gain attraction compared to competing network providers. He has already a billing system and just has to extend it.
The content provider has access to more potential customers and doesn't have to build up an own billing system.
The advantage for the customer is that he only gets one bill [Ringenbach] and has also only just one contact in case of problems.

Figure 6 illustrates the situation: The network operator delivers the content (which has been produced by the third party and which is stored on his servers) to the customer (1). The user gets the normal periodical invoice (2) and he has to pay for the network usage and the content (3). The network operator pays then the third party accordingly to their agreement (4).

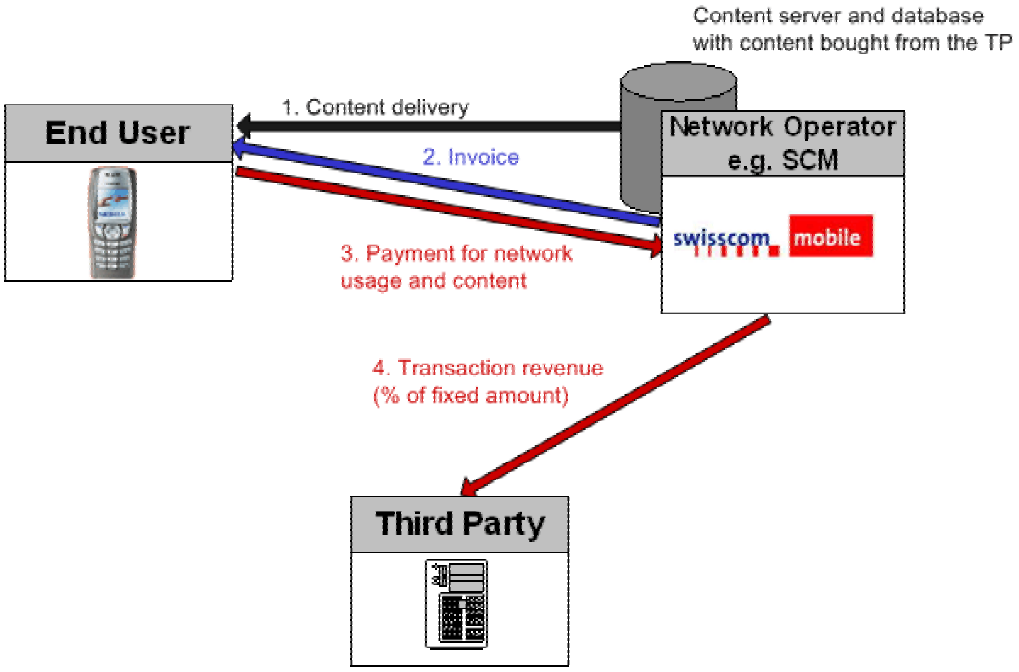

Figure 6: Third Party Business Case 1

In a second business model the user is also charged by the network operator for both, network access and ordered content services. But the content server is located at the content provider. The network operator doesn't buy the content to resell it to its customer, but he still does the billing on behalf of the content provider. This one can benefit from the large customer base of the network operator and from his experiences in billing. The bills of Swisscom, for instance, are paid with one of the highest payment morality in Switzerland [Watrin]. Also here the customer has the advantage that he only gets one bill and has only one contact in case of questions or problems. A proper arrangement between the network operator and the TP to define the responsibilities is obviously needed.

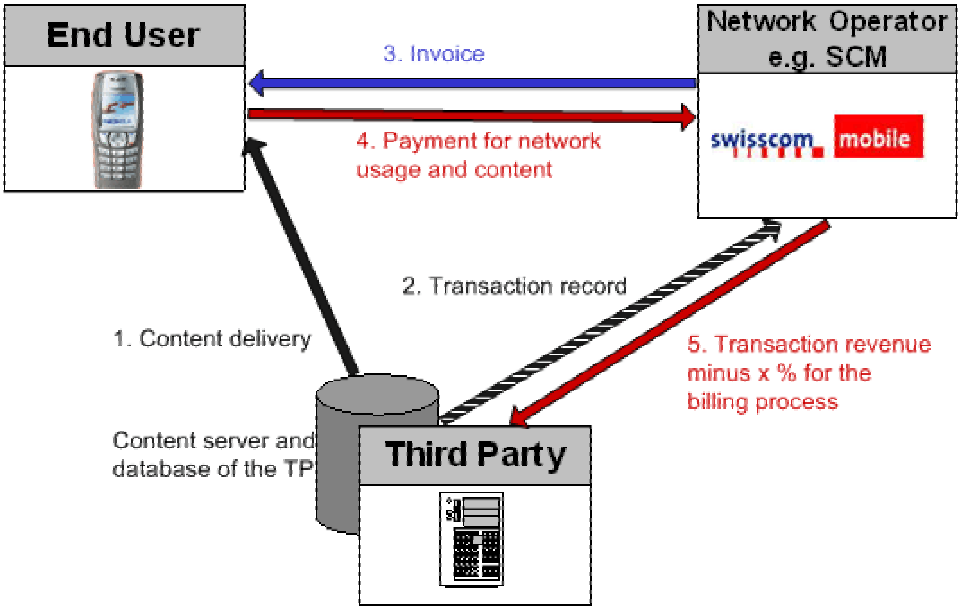

Figure 7: Third Party Business Case 2

Figure 7 shows this second business case: The content is directly delivered to the customer from the third party (1). The network operator gets a transaction record (2) and sends the periodical invoice to the customer and charges him for the network usage and the content delivered by the third party (3). The customer pays the bill (4) and the third party gets the transaction revenue minus some percent to pay the network operator for the billing process.

These business models are called “third party (TP) billing”. A content provider benefits from the experience and the billing infrastructure of a network operator.
This thesis concentrates on the second business model above. The TP party provides its content directly (from its own servers) to the clients and the network operator is responsible for the billing process. Of course the TP has to pay the network operator for executing the billing process. Normally the network operator doesn’t deliver the whole charged amount to the TP and makes a deduction of a certain percentage.

Third party business could even go a step further and there are already implementations in this direction: Swisscom Mobile offers for instance the possibility to buy beverages from a vending machine and to pay them via SMS. The client gets then the consumed beverage charged on his mobile phone invoice. However, this thesis treats only the third party business in the context of services, which are used on the mobile device and not the selling of physical goods.

3.3 Swisscom Mobile WAP Content Billing

Swisscom Mobile WAP Content Billing (SCM WCB) allows third parties to offer premium content (billable services) to mobile customers and to bill them via the Swisscom Mobile invoice. With WCB, the third party has an opportunity to act as an independent provider of high-quality WAP content and so to upgrade its portfolio of free content with premium content.
The basis of the billing of this premium content is revenue sharing. This regulates the division of the proceeds of the sale between the third party and the network operator.

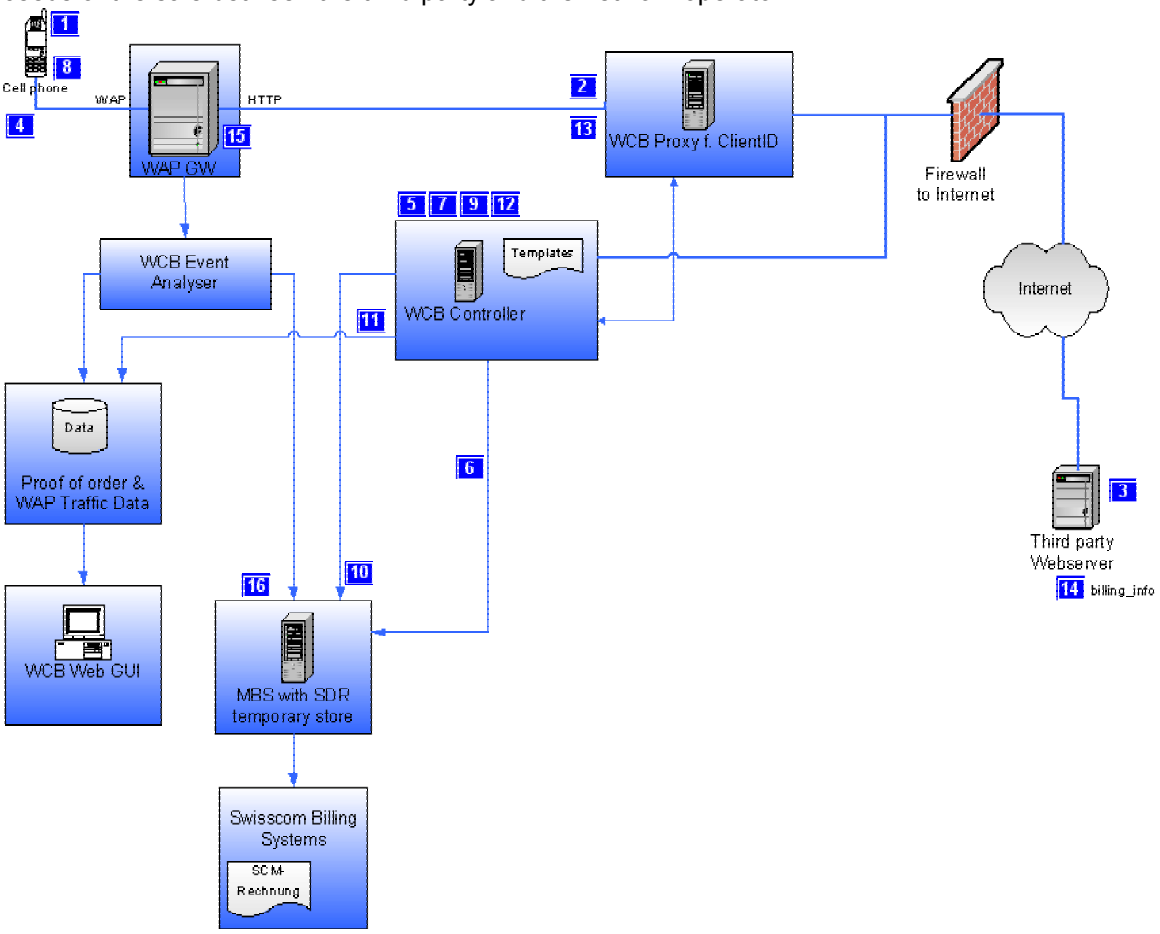

Figure 8: Part of the Swisscom Mobile WCB Architecture

If the mobile customer chooses a billable service, then it will be delivered to the mobile terminal. The delivery assumes an explicit approval by the customer, before it will be executed.

3.3.1 Application Flow for WAP Content Billing

This chapter describes the architecture and the application flow for WCB. The description isn't too detailed. Only the most important events and actions are presented to have a background to introduce the extension of the architecture and the additional flow of events for the Proof of Delivery later. Figure 8 shows a schema of a part (relevant for the PoD afterwards) of the architecture, while below the different steps are explained.

1. The user selects a hyperlink on his device, which points to a TP.
2. The request passes the WCP Proxy. The whole WAP traffic passes this proxy, which can add or change a header field in the HTTP request. For the WCB, the proxy can introduce a new header field, a so-called Client ID, which provides an identification of the client without uncovering his MSISDN. According to the Swiss law, an operator must request the permission of a mobile customer, whenever his MSISDN is transmitted to another party.
3. The request arrives at the TP. The application recognizes that the possible following request contains premium content and generates a page with a link, which points to the WCB Controller of SCM. This links contains some arguments, such as the TP ID or the service identification.
4. The customer selects this dynamically generated link.
5. The WCB Controller is the core of WCB architecture. It controls the dialogue with the customer and does the billing for the TP. The controller first checks, if
6. the MSISDN belongs to a subscriber of SCM.
7. Then a WML page is generated with the templates based on the arguments of the TP and the user_agent parameter in the request. This page contains the bill text and the bill amount.
8. The customer receives this page and is asked if he agrees to be billed (see Figure 9). If the customer accepts, an URL is requested, which points again to the WCB Controller.
9. The WCB Controllerrecognises that the customer accepted.
10. Then, the Micro Billing Server (MBS) is called with the needed parameters. If the parameters (TP, service etc.) have been checked successfully, the MBS puts a Service Data Record (SDR) in a temporary memory and gives back a unique sequence number (bill ID) to the WCB Controller.
11. Then a Proof of Order (PoO) is generated and stored in the database.
12. The WCB Controller attaches the unique sequence number to the URL of the TP.
13. The proxy recognises the URL of the TP and attaches if necessary the Client ID in the request.
14. The TP composes dynamically a page and sets a header field to the bill ID.
15. The WAP Gateway is able to generate events. An event is of instance fetching an URL and contains several key/value pairs, such as the URL, MSISDN, IP address, packet size, timestamp etc. The header field with the bill ID causes the gateway to generate an event, which is recognised by the WCB Event Analyser.
16. The MBS is then notified that the SDR is valid.

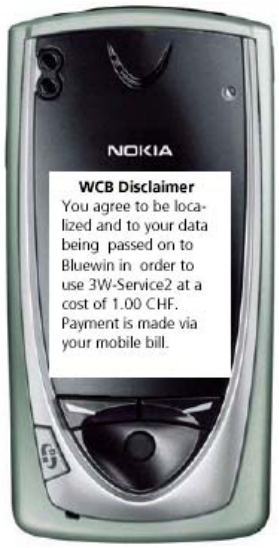

Figure 9: The WCB disclaimer

3.3.2 Proof of Order

In the chapter above, point 11 mentions the Proof of Order (PoO). If the customer accepts to pay for the premium content by clicking on the appropriate link, the WCB Controller waits for a sequence number of the MBS. Then a PoO can be stored. A PoO documents that a client has ordered a certain service at a certain time. This information is needed in case of a later request from the customer. The Customer Care (CuC) has access to this stored data over a web interface.
It is also thinkable that the customer can directly query his stored PoO data, i.e. over SMS or WAP.

The PoO has a special relevance in this thesis because it can be considered as the counterpart of the Proof of Delivery. They are respectively an answer to the request and to the response of a service delivery.

The difference is that the PoO is only stored to enable the CuC to answer user requests. The PoD can also have an influence on the billing process. This will be discussed later in chapter 7.

3.3.3 Proof of Delivery

There is already some kind of Proof of Delivery (PoD) integrated in the WCB architecture. The WAP gateway generates an event, if the transmission of the page of the third party to the handset is connection-orientated. The gateway generates the event, when the confirmation from the handset comes back.
But this PoD works only for Web pages and also only if the connections go via WAP gateway. It is the goal of this thesis to show also other ways, how a PoD could be implemented, independent of the WAP gateway (e.g. in case of a direct HTTP connection from the device) and for every kind of service, thus also for streaming etc.
The ideas for possible designs can be found in the chapters 5 and 6.

3.3.4 Service Data Records

A Service Data Record (SDR, in the context of voice often also called CDR for Call Data Record) is a network report that includes details about service use, such as type, time, duration or destination.

Within the WCB architecture of SCM a SDR is generated for the delivery of content service.
The SDR contains among other things the following parameters:

- Date and time of the Proof of Delivery (feedback from the WAP gateway)
- MSISDN of the subscriber
- Billing text (Name of the TP and name of the WAP Service)
- Billing amount
- Third party ID

The generated SDRs are afterwards transmitted to the billing systems. The date, the billing text and the billing amount appear finally on the invoice.

4 Proof of Delivery - Fundamentals

This chapter is the basis of the main topic of this thesis. It explains the needs for a Proof of Delivery (PoD) for services over GSM or GPRS and gives an overview over related work.

4.1 Introduction

Let's imagine the situation that someone has ordered a good, a PC for instance, and gets it delivered at home some days later. The delivery is accomplished by a parcel service on behalf of the PC manufacturer or the PC shop. When the customer gets the parcels handed out by the employee of the parcel service, he has to sign an acknowledgement of receipt. So, the employee has a proof that he has done his job and the client received the parcels.
If the customer later claims not to have received one parcel (e.g. the one with the keyboard in it) the manufacturer and the parcel service company can show him the signed acknowledge. They have a proof that the goods have been delivered. This proof is based upon the fact that the client had checked the delivered parcels (counting them, looking at the content description etc.) before he signed.

Now, let's imagine the situation that the same customer uses a service on his mobile device. A company specialised to this kind of service produces the content (e.g. a sports information agency). This agency is a third party (chapter 3.2) and uses the network infrastructure of a network operator to deliver the service. The agency can be compared with the computer manufacturer from above and the network operator with the parcel service. The network operator is responsible for the delivery and likes also to have a proof that he has done its job.

Suppose now, that the customer is charged to have used the service from the sports agency 140 times, but he believes that this is really too much. So he calls the customer care (CuC) of the network provider and criticises the invoice. The employee of the CuC can then search for the client's data in the systems and then tell him, that he ordered the service really 140 times (Proof of Order) and got it delivered also 140 times (Proof of Delivery). Of course the customer can get this information in written form too and sees that he has no choice and has to pay the bill.
The fact that the client pays the bill then without any further complaints, implies that he can trust the PoD of the network provider (if he isn't aware anymore of his "excessive" service use during the last month…).

It exists also a slightly different sight on the things: If a carrier can determine when an individual wireless customer is having a hard time downloading files or is experiencing numerous dropped calls in a single day, then perhaps that carrier can reach out to the customer and offer a credit to his account before he calls to complain or even worse decides to look for another provider [Bill0502]. This means that the carrier doesn't use the PoD only to be ready for user complaints, but instead uses it as a proactive instrument to already satisfy the users as much as possible before it is too late.

The parcel delivery is confirmed with a signature. But the delivery of a service to the mobile device can't be confirmed in such a way. Therefore solutions have to be found, which can be considered as being equivalent to the signature for the parcel carrier. These solutions have to be secure, user-friendly, fast and convenient.

4.2 Assumptions

Before the reasons justifying a Proof of Delivery and possible implementations can be discussed, some assumptions have to be fixed.

The PoD is designed for services from a third party delivered over a GSM circuit-switched or a GPRS packet-switched network to the end customer. The PoD will also be applicable in the upcoming 3G networks.
The network operator gathers the information about the status of the service delivery. The third party is also informed about the status of delivery. Probably not in real-time or only about failed deliveries, which can have an influence on the revenue sharing between the network operator and the third party.
This means that the PoD is used as a proactive instrument to be able to charge the client according to the success of the service delivery.

After Figure 4, the currently relevant services are the download of content (e.g. web pages, ring tones, pictures or video/audio clips) and in the near future (time horizon 1-2 years) also streaming services (video/audio) and interactive gaming. These services can be of business and of private (entertainment) use. The investigations about the PoD in the scope of this thesis concentrate on download and streaming.
Of course it is also thinkable to have a PoD for emails or SMS/MMS. But such services are in the scope of this work neither.

4.3 Reasons for a Proof of Delivery

But why a proof of delivery is needed at all? If the delivery over the network were 100% reliable, then a PoD wouldn't be needed.

As it is showed in chapter 2, the network can't be considered as 100% reliable. GPRS and also future mobile network technologies base on all-IP-networks. IP has a lot of advantages, but has, as already mentioned, a lack of reliability. This missing reliability and also some other technical matters are reasons, which justify the implementation of a PoD.

The main problems occur in the wireless environment. The air interface is likely to be one of the major bandwidth limiting factors in a wireless data network. In the core network there is normally enough bandwidth and there are also used QoS assuring mechanisms. The air interface instead is a hostile medium for access, with problems arising from error-prone and time-variant link characteristics [NokiaE2E]. Wireless networks tend to have more jitter, more delay, less bandwidth and higher error rates compared to wired networks. These features may change randomly, for example as a result of vehicular traffic or atmospheric disturbance. They may also change when the terminal moves and handover occurs [MIND].

4.3.1 Bit Errors

A lot of bit errors occur on the wireless link. The link is unreliable (optimal packet lengths have to be small) and relatively insecure. Packet loss in a mobile environment is an important issue to be considered because of the limited bandwidth of the network and the possible fading and blackout situations that can occur when a mobile moves from one cell to another.
Due to mobility, wireless applications cannot use a consistent data rate. The data rates can't be evenly distributed across the entire area covered by the Base Transceiver Station (BTS). A high data rate may be granted when the mobile terminal is close to the BTS with small shadow fading. But if the user is in the fade zone or in the fringe of the cell, the data rate will drop considerably. This uneven data rate leads to inconsistent delays depending on where the mobile terminal is located relative to the BTS [QoSMobileIP].
This is mainly a problem for streaming services. For downloads delay variations are less relevant, the disturbing fact is the delay by nature.

The high and unpredictable bit error rate causes also problems with TCP: TCP is a protocol initially designed for working in fixed networks, where the main problem is congestion. Therefore TCP has mechanisms to avoid sending useless segments into the network when a congestion situation is detected. However, the problems in wireless networks vary: bursty packet losses, high packet delays depending on the wireless network, variable throughput etc. [RenCasFan]. TCP works under the implicit assumption that all packet losses have their origin in congestion. Hence, TCP reduces its congestion window before retransmitting packets and backs off its retransmission timer.
This will unnecessarily result in severe throughput degradation and very high interactive delays when packets are lost for reasons other than congestion [QoSMobileIP].
Every service delivered over TCP is affected by this degradation. Streams normally use UDP, so the problem is more relevant for downloads. High delays by themselves don't force the need for a PoD. But with high delays the danger increases that the user aborts the download or that other problems occur during the (longer) download.

4.3.2 Bandwidth Limitations

The bandwidth of the wireless link connecting a mobile device to the static segment of the network is significantly lower than the one of the wired links between static hosts. This causes a serious degradation to the traffic performance especially for real time applications.

Many applications have very unpredictable traffic patterns. They may lay dormant for a while and then send a burst of data as in the case of email and Web browsing. Static allocation of bandwidth does not use the precious air resource efficiently.
The limited bandwidth can be a problem in particular at metropolitan areas. To use a higher bandwidth, the GPRS channels have to be bundled. But since GPRS channels have a lower priority than voice channels, there is no guarantee for available GPRS channels to bundle [W&BGPRS].
A limited bandwidth doesn't prevent the service to be delivered; a download takes only more time. But the situation is different for streaming services, where the quality of service is influenced very much by the available bandwidth.

4.3.3 Resources in the Device

Problems are also the constricted resources in the mobile terminal. The devices have a limited memory and storage capacity. Mobile systems use small power batteries and hence have power restrictions. Transmitting and receiving of packets expend power and will have to be controlled when the mobile battery power is low. An ideal solution would be that the mobile terminal transmits nothing expect when actively connected to the network. But in order to allow the network to efficiently forward the incoming packets destined for it, a mobile terminal must periodically transmit a beacon packet to inform the network of its current location. This must be done even when the device isn't in an active state and can be very power consuming [QoSMobileIP].
Due to low battery power it is imaginable that a service can't be delivered entirely, because the connection gets cut off. Managing of the battery power is in the exclusive responsibility of the user, so a provider can't influence the delivery problem arising out of this fact. But the provider should be aware of it and be able to prove that the task has been fulfilled as much as it lies in his area of responsibility.

4.3.4 Handover

Handover describes a mechanism in cellular networks that transfers the association of a mobile end system from one base station – which is presently active – to a new base station. The handover between wireless cells of the same type (in terms of coverage, data rate and mobility) is often referred to as horizontal handover, whereas the handover between wireless cells of different type (for example between WLAN and GPRS or UMTS) is characterised as vertical handover [TUBerlin].

One of the difficulties in moving from one cell to another is the fact that GPRS data is competing against both voice calls and circuit-switched data transmission on the GSM network. A cell may have only one slot available for GPRS; if this is already taken, any attempt to transfer another GPRS connection to this cell will fail [Transcomm].

With GPRS, the mobile device is assigned a temporary IP address by the network (received from the SGSN) for each session, since phone numbers cannot be used as addresses for packet data. When the user is on the move and requires a handover from one base station to the next, the network has to ensure that the same temporary IP address is used. In practice there are often difficulties maintaining the session while making the change. If a session is prematurely terminated, the user has to make up a new connection, and if data has been partially transferred, the transfer process may have to start all over again (i.e. the process cannot just pick up where it left off) [Transcomm].

A solution for the problem with changing IP addresses is Mobile IP. The basic concept of Mobile IP is that the mobile terminal is able to communicate using the same IP address at all times, regardless of its access points. Mobile IP uses a couple of addresses to manage the movements of the user. Each time the mobile terminal connects to a foreign network, it obtains a temporary address, called Care-of-Address (COA), from the foreign agent (node in the current network). This address remains valid only while the mobile terminal stays connected to this network. The mobile terminal must inform its home agent in its home network. If the home agent knows the current foreign agent of the mobile terminal, packets destined for it can be routed towards it using IP tunnelling [ReinBona].
Figure 10 shows the packet flow in the Mobile IP context. The correspondent host (CH) sends a packet to the mobile node (MN) via the home agent and the foreign agent.

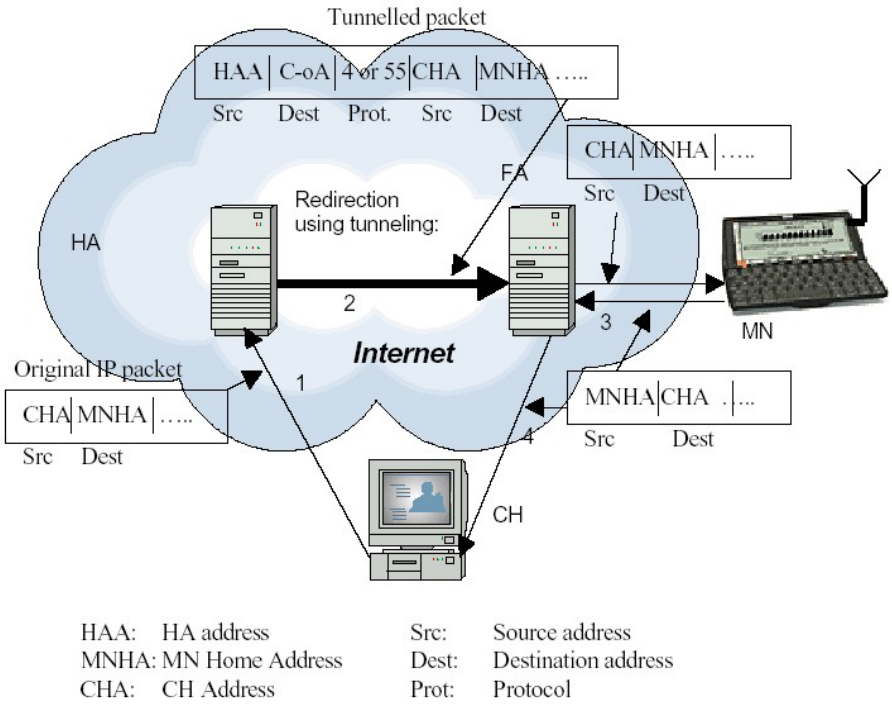

Figure 10: Mobile IP packet flow [Ericsson]

Within Mobile IP, a difference is made between macro mobility (the mobile terminal changes the network or the subnet) and micro mobility (the mobile changes just the base station). While Mobile IP handles well the handover in the macro mobility scenario, the protocol has limitations in the micro mobility [QoSMobileIP]:

- Packet may get lost during handover. There is no mechanism to allow smooth forwarding of packets from the home agent to the new foreign agent.
- Re-negotiation of QoS requirements in the new cell. There is no guarantee of honouring QoS requirements in the new cell due to resource constraint.
- Whenever a mobile terminal moves to the new cell, it needs to send a notification to the home agent, so that incoming packets can be delivered to the new subnet. This results in high signalling overhead.

Since Mobile IP has these limitations and not a very good handover performance, some micro-mobility protocols like Cellular-IP, Hawaii, HMIP and BCMP have been engineered. The micro-mobility protocol is in charge of updating the access network topology so that the mobile node does not need to change its IP address and thus all its sessions are able to continue established [MIND].

Nevertheless, handover is a situation with a high probability of failure, especially in areas with an intensive use of the mobile network.

4.4 Related Work

The goal of this chapter is to list some summaries of other work done in the area of confirmed service delivery. Obviously there aren't yet a lot of solutions or it is also thinkable that they aren't published. But some ideas can be found, especially for the download of Java content.

4.4.1 Confirmed Download

There are different methods for the confirmed download of content. One method for downloading Java content is based on the open standards for Java 2 Micro Edition/Mobile Information Device Profile (J2ME/MIDP). Applications written for J2ME MIDP are called MIDlets.
MIDP OTA (Over the Air) Provisioning (which leverages open standards like HTTP) is an industry-accepted specification that allows Java applications to be delivered over the air and used on Java-enabled phones with delivery acknowledgments [NokiaCOD, MIDPOTA].
There are also solutions for the download of generic content (not only Java MIDlets). In the following the implementation of Nokia will be discussed.

4.4.1.1 Nokia OTA Download for Generic Content

The download of generic content is based on the concept of a Content Object Descriptor (COD), i.e. a file that is downloaded before the actual content object (e.g. an image or a ring tone) is fetched. The mechanism of Nokia is semantically virtually similar to the mechanism defined by the Java Community Process for download of J2ME/MIDP content (download of a JAR File). There the descriptor file is called JAD (Java Application Descriptor).
The COD triggers the download method for generic content in the client device. In the device the download agent is launched and the COD file analysed. The download user agent uses the information in this file to check whether the device is capable of downloading the content object.
The download of the content object is typically performed using HTTP(S). The outcome of the installation is reported back to the network, so it is indicated to the download server, that the content object has been properly received. The installation status is reported by use of an HTTP POST request to a URL, which has been provided by the COD file (InstallNotify attribute).
The content of the body of the POST request includes on the first line a status code and a status message. Examples of this status messages are: “Success”, “Insufficient Memory, “User cancelled”, “Loss of service” or “User aborted” [NokiaOTA].
The implementation of Nokia seems to be ready for several problems during the download. But the question is, if the environment is really ready to behave appropriate. So it is mentioned in [NokiaOTA] that it may be impossible to deliver the status message “Loss of Service” due to the network-service outage.
In response to an installation status report, the server needs only to reply with a “200 OK” response [MIDPOTA].

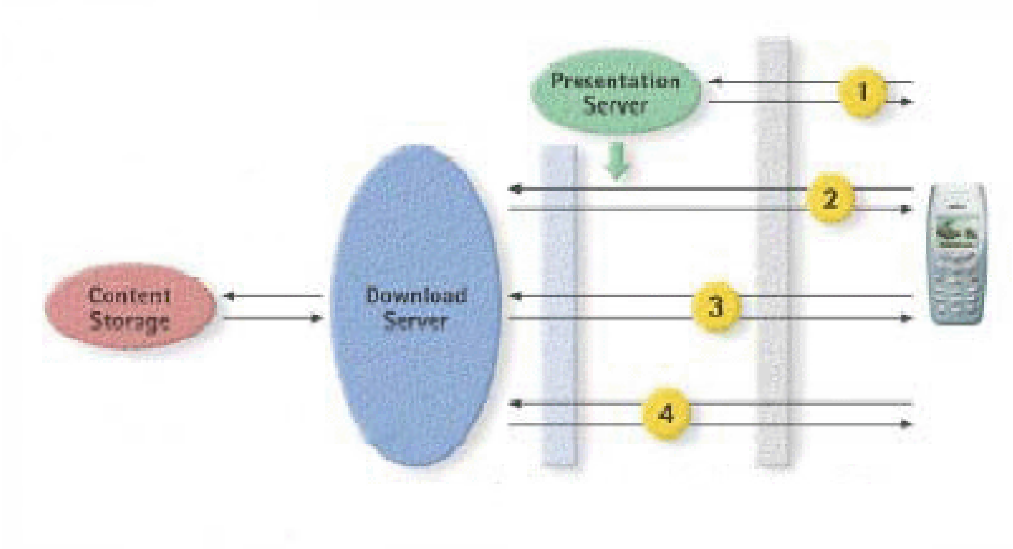

Figure 11: COD Schema [NokiaOTA]

Figure 11 contains the schema of the download process and the following example shows the relevant HTTP requests and responses.

1. The client finds the URL to an interesting service on the presentation server.
2. The client selects this URL that points to the COD file on the download server and downloads the file.
3. The client fetches the content object from the download server.
4. The client reports the status of the download to the download server.

Example:

A possible HTTP Request for the content descriptor:

```
GET http://host.foo.bar/pic-dir/picture.cod?ID=1234
Host: host.foo.bar
Accept: text/x-co-desc
User-Agent: CoolPhone/1.4
Accept-Language: en-US, fi, fr
Accept-Charset: utf-8
```

The response from the server might look as follows:

```
HTTP/1.1 200 OK
Server: CoolServer/1.3.12
Content-Length: 50
Content-Type: text/x-co-desc; charset=utf-8
COD-URL: http://host.foo.bar/pic-dir/picture.gif
COD-Install-Notify: http://foo.bar.com/status
COD-Size: 1345
COD-Type: image/GIF
```

The response contains the Install-Notify URL, to which the status report will be sent after the download.
Then a HTTP-request to the COD-URL follows and the content is sent to the client in the HTTP response.

Finally we have the report of the install status via an HTTP post request:
In case of a successful reception:

```
POST http://foo.bar.com/status
Host: foo.bar.com
Content-Length: 13
900 Success
```

The response from the server then might just be:

```
HTTP/1.1 200 OK
Server: CoolServer/1.3.12
```

4.4.1.2 OMA Generic Content Download Over The Air

The Open Mobile Alliance (OMA) has created a similar approach, which is also a complement to the download of Java applications, described in [MIDPOTA].
The goal was to define a technology for confirmed download that is used to deliver digital content to mobile terminals. The confirmation of a successful installation of a media object typically triggers server side billing actions.
It is mentioned that a download transaction always includes a degree of uncertainty, because the client and the server can't simultaneously know with absolute certainty that a transaction has succeeded. With the OMA Download Installation Notification there may be situations where the server does not know that the client has accepted the media object, but there is no situation where the server would believe, that the client device has accepted the media object when it in fact has been rejected.
Also here we have a file that serves as a download descriptor. It includes an URI that references the media object and may (a confirmation is not always needed or demanded) also contain an InstallNotifyURI to confirm the download. If an InstallNotifyURI has been defined in the download descriptor, then errors during the download process must be reported using the status report mechanism.
An important aspect of the installation notification mechanism in the download descriptor is to prevent a situation where the server has "knowledge" (assumes) that the transaction completed, but the client perceives the transaction as failed.
The implementation of the download agent in the device must make at least one "well-intentioned" (the network connection is known to the extent possible to be present) attempt to send the installation notification to the address defined in the InstallNotifyURI attribute. If such a "well-intentioned" attempt can't be made, despite multiple retries, then the media object must be immediately removed from the device!
This seems to be quite a strict implementation, but so we have a 100% guarantee, that the content has been delivered, when the feedback reaches the server.
In the OMA implementation, the download descriptor is an XML file [OMAOTA]. It has a quite similar form to the COD file from Nokia. One element is the InstallNotifyURI:

```
<media xmlns="http://www.openmobilealliance.org/xmlns/dd">
 <type>image/gif</type>
 <ObjectURI>http:/download.example.com/image.gif</ObjectURI>
 <size>100</size>
 <installNotifyURI>http:/download.example.com/image.gif?id=image
 </installNotifyURI>
</media>
```

The solution of Nokia and also the OMA solution contain notifications in case of problems during the download. But in the [MIDPOTA] document it is written that the network is not likely to be available to report the status of the network.

4.4.2 Other Work

Results of an extensive web research and also of enquiries per email directly at Siemens and Ericsson weren't very successful. Siemens Mobile didn't answer the request. Ericsson stated only, that they see a service specific solution and not a generic one. This means that they think that it makes sense to implement the PoD depending on the kind of service. A conclusion, which has already been made by the author before.

4.4.2.1 Project SOQUET

Last but not least an interesting European research project called SOQUET (System for Quality of Service for 3G Networks) has been found [SOQUET].
This project sets out to utilize the objective measurement of the perceived quality of a multimedia application such as streaming video, for the management of resources in the UMTS radio access network (UTRAN).
A measurement instrument is used, which is capable of measuring the perceived quality of a video stream in real-time and of providing these measurement parameters via a return channel to a quality management system. The so-called Perceived Quality-of-Service (PQoS) can then be used as the determining parameter for the allocation of resources. For this purpose, one or more test probes are located at representative points in the service area, typically in the UMTS cell. They provide information about the PQoS of the received services to the QoS manager [Lauterjung].
The project seems to be interesting from the point of view, how the measurements of the QoS are made. Such measurement could also be used to date for the concerns of the PoD.

The answer to a request to a project participant was that there haven't yet been any investigations, which go in the direction of the PoD. A point is also that the UTRAN is only emulated in the project and there is no access to a real UMTS network.
So, the project doesn't seem to be of further relevance currently.

5 Proof of Delivery – Design

The goal of this chapter is to show, how a PoD can be designed. The intention is to resume all the thoughts and ideas on the way to create the designs, even if some ideas only seemed to make sense at the beginning. But it can be helpful for a reader if he is able to track all the thoughts on the way to find a possible implementation.
Finally, one solution will be discussed in more details in the sixth chapter.

To implement a PoD, there are basically two main aspects to discuss:

- Investigate, when the delivery of the different kinds of services can be reasonable confirmed (after the download, after the service use etc.).
- The technical implementation of the PoD (it is imaginable that the implementation also depends on the kind of service).

5.1 Service Categories

A first decision, which has to be taken, is whether a PoD for mobile services can be implemented by a generic solution, which is applicable to every kind of service. This would mean that the PoD is generated for instance just after the transmission of the last bit belonging to the service.
Such a generic solution would have to cover services such as video streaming, but also services, which contain only a simple (WML)-page or pictures. The form of the PoD would be very general and also very scalable, so that it won't be a problem to apply the solution also to new services. This could be quite a tough undertaking. It seems to be better to have a special solution for each category of service. It is important to stress that this doesn't mean that these different solutions can't be implemented in the same technical way. But it means that this implementation makes differences between the services and is able to decide, when they can be considered as delivered.

Therefore a categorisation of the services has to be made.

5.1.1 Atomic and Non-atomic Services

In a first approach, the services are divided into two main categories: atomic and non-atomic services. The non-atomic services are then in a second step separated into fixed sized and variable sized services:

- Atomic services are services, which are delivered at once. These services are small items with a negligible risk of a failing delivery. Examples can be logos, ring tones, but also small Web pages or emails (without attachments).

- Non-atomic services are services, whose delivery takes a few seconds and therefore the probability of failure is higher. Fixed sized services have a predetermined size (e.g. video streams with a known length), while variable sized services (e.g. live streaming) have a length, which is not yet determined at the beginning of the delivery.

Figure 12 illustrates this idea:

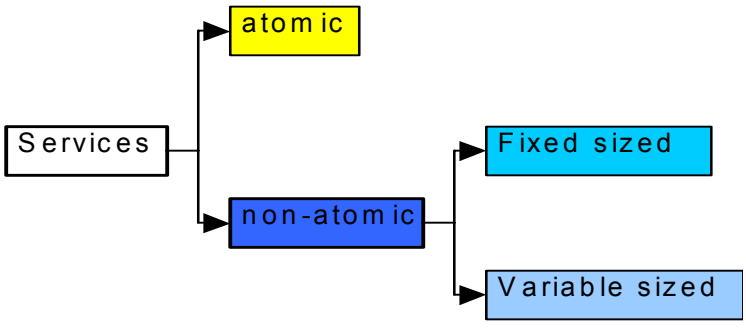

Figure 12: Possible approach for a categorisation of services

The difference between the service categories concerning the PoD is the point of time, when a PoD can be generated. The delivery of an atomic service can be confirmed just immediately after it reached the mobile device. A non-atomic service has a certain duration. The question to discuss is the one about the point in time to generate the PoD. Of course this can be done also only after the data reached completely the device. This is certainly sensible for services, which need a previous download (which takes some seconds or minutes) and then are used locally on the device.
The situation isn't as easy as that for streaming services (variable and fixed size). There it could be inexpressively to generate a confirmation at the end of the stream, because troubles may come up during the time, when the stream is sent and reproduced on the mobile device. These potential occurring troubles should be logged and then influence the meaning of the PoD. The user may even interrupt the stream or the mobile device may cause problems (empty batteries or entering an area without reception).

This approach with such a categorisation of services seems to be a bit too general and not entirely consequent. So, a second one has been created during the design phase of this thesis:

5.1.2 1-phase and 2-phases Services

In this second approach, the services are categorised in two main categories. The principal distinction is not made on the basis of the estimated download time as above, but due to the fact, if the service needs a previous download and is then used afterwards locally on the device or if the download and the use take place in parallel.

5.1.2.1 2-phases Services

The services of this category are downloaded in a first phase and then used in a second phase locally on the device, while there is no further interaction with the server during the usage of the service.
Examples are: Web sites (with premium content), ring tones, but also services with a larger data amount, e.g. videos clips or games.
For such services it makes sense to generate the PoD just after the download (first phase of the service use).

$\Rightarrow$ The end of the download can be defined as being equivalent to the arrival of the last bit belonging to the content at the mobile terminal. Of course this download has to be executed over a reliable connection (e.g. TCP), so that it is sure that it arrives completely at the device.

What the user actually does with the service (delete it immediately, use it several times etc.) is not directly in the interest of the network operator. The provider has done its job, as soon as the download has been successfully (without any errors) completed. The first phase (the download) can be confirmed; the second phase (the use of the service) isn't confirmed.
If further interaction with the server is needed (e.g. another web page has to be fetched), the information from the server is downloaded and then its delivery can be confirmed in the same way again.
It is thinkable that the definition of the point in time, when the service is delivered can be extended. Of course it is simple to take the last bit of the download as a reference. But it could also be possible to demand that the download has to be completed within a certain time frame. The PoD would then only be positive, if the content reaches the device completely within this time window. But within the scope of this thesis it is enough if the download terminates in a conformable time frame. Anyway, the user would certainly abort a download, which seems to be endless due to a very small data rate.

Figure 13 shows the process of the delivery of a 2-phases service. The request (Phase 0) has been drawn only for the sake of completeness and doesn't belong directly to the service delivery. Phase 1a is then the download and before the content is used locally (phase 2), the PoD is generated (1b).
The arrow for the PoD has been drawn purposely in the middle: Depending on the implementation the PoD can be delivered from the device but also from the server or the network (see also the following chapters).

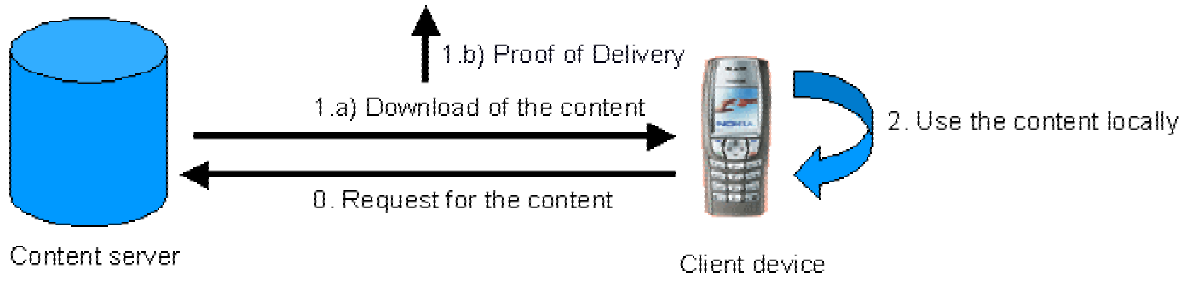

Figure 13: Schema of 2-phases services

5.1.2.2 1-phase Services

The download and the use of the services take place in parallel. The content is replayed continuously on the device. The main examples are (live) video and audio streams. The job of the network operator is to deliver the whole stream. To settle the PoD just after the last bit of the stream doesn't seem to be very clever. Let's imagine the situation of a video stream of 2 minutes, whereas 20 seconds couldn't be displayed (or only in a bad quality) due to problems within the network.
For such services the QoS plays an important role. So it makes probably sense to log periodically the state of the service delivery (count for instance the lost frames) and to send the gathered data immediately or at the end of the service use back to the network carrier.

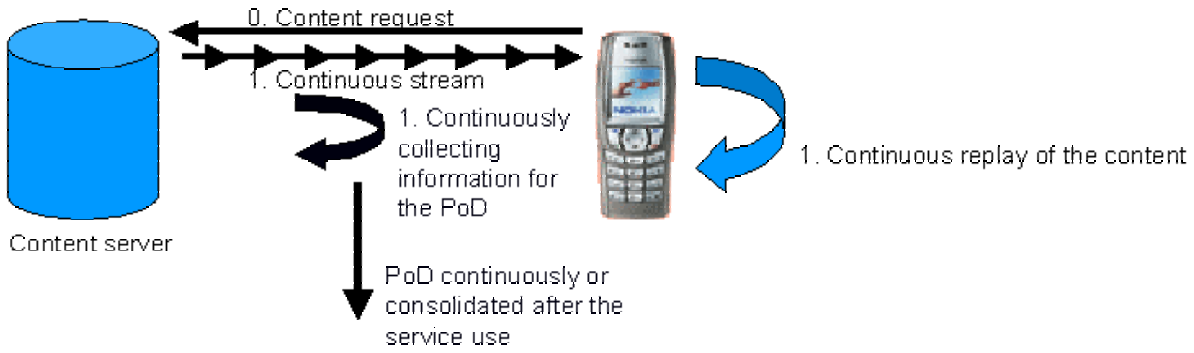

Figure 14: Schema of 1-phase services

Figure 14 contains the illustration of the delivery of 1-phase services. After the request for the service (0), there is the continuous bit stream to the device and the reproduction of the content at the same time. The information for the PoD is collected continuously.

These two service categories cover already most of the services in general and at least the services, which are relevant within this thesis (see chapter 4.2).
Even though it goes beyond the scope of this thesis, below are some thoughts about other kind of services, only to complete this design approach.

5.1.2.3 Special Services

It is thinkable to classify some services in a special category.
Potential candidates for this category are services, which have a special relevance for the user and/or for the service provider. Examples could be financial transactions. These services (M-banking, M-shopping etc.) normally are based on web pages. Hence they could also be viewed as normal "2-phases services". It's a point to discuss, whether such a differentiation could make sense. However, the delivery of such special services can probably be confirmed in the normal way, but it is thinkable, that the PoD of such services is associated with a priority class, because it is of a special importance, whether the PoD arrives successfully and completely at the network operator or not (because it is also thinkable that the PoD gets lost on its way to the network operator).

5.1.2.4 Services with States

Beside the introduced two service categories, there are also services, whose delivery is hardly to prove, because there is not a clear separation of download and service use. Examples are mainly interactive services, such as online gaming.
The delivery of some types of games can be easily proofed:
Example: two users are playing a Battleships game against each other over the mobile network. They probably need to download some data to have the field to play on their devices before they can play. This download can be confirmed and then the users can play and the provider has no further duty (beside providing the network, that the users are able to play against each other) and so no further PoD is needed.
Another situation is, when a customer plays a game, which is hosted on a server, for instance an online gambling game. No separate download is needed; the user just starts the service and can play then as long as he wants to.
The game enters states, which have to be well defined, including one or more final state. Reporting the reached states could then probably provide a Proof of Delivery. It is supposable that an implementation of such a feedback mechanism isn't as simple as that.
In this context it is even worth to think about the question, if really every service delivery has to be confirmed. It can be discussed, if it really makes sense to prove the delivery of every single Web page with premium content.

5.1.2.5 From the Theory to the Practice

The decision, if really every service delivery has to be proven can also depend on the monetary value of the service. If the service only costs for instance -.50 CHF it is probably less relevant for the customer than if the service costs 10.- CHF. But if a reliable working implementation for the PoD can be established it shouldn't be a problem to prove every service delivery, also for comparatively cheap services. For more expensive services, the interest of the content provider in a successful delivery is probably higher. The content provider (third party) may also desire to get a PoD.

Another problem is that a service often doesn't consist just of one page or one picture etc:
A third party could for instance offer a joke service. The user has to agree before the service use that he is billed for the content by the network operator (for instance with 2.- CHF) and can use the joke service then for five hours as often as he wants (implemented probably with a Cookie). The joke service can consist of a portal, where the joke categories and perhaps also cartoons etc. are selected. The user can then read some jokes, browse back to the portal, look at a cartoon, exit the service and re-enter it one hour later and again consume some cartoons etc.
But what is the delivered service in this context? Each joke category is distributed on some linked WML pages (i.e. 1/15, 2/15 etc.). The download of such a page accords every time to a 2-phases service and each page download can be confirmed. This would mean that all the activities of the client during these five hours are logged.
It isn't completely clear, what is "the service" in this context: the user can theoretically look at every joke page and every cartoon (once or several times), but possibly he just wants to read the "joke of the day" (willing to pay 2 CHF for that) and then doesn't use the service for the rest of the five-hour-period.

In the opinion of the author it makes also in this context sense to generate a PoD for every page downloaded by the customer. This data can be logged in the system.
To make things easy it could be defined that if the customer gets the portal of the service (e.g. the menu page for the jokes) the first time and if he then gets just one page linked from this portal, the service is delivered, regardless of the rest of the time of the service use.
If the PoD is only used for statistical reasons and for the CuC to be able to show the customer that the service has been delivered, this definition is less relevant than in the case if the PoD is really used as a proactive instrument to be able to offer discounts to the customers. Then it is thinkable to log the user activities and to offer a discount if for instance more than 30% of his requests to pages belonging to the service have failed.
However, the matters of discounts will be discussed in chapter 7. Of course it should be the goal of the provider to enable a reliable service delivery, so that possible discounts are really an exception and not the common case.

5.2 Implementation Variants of a Proof of Delivery

A Proof of Delivery is an evidence for a network operator that he has done its job.
Since the best evidence for a delivery is a signature from the user and this is not really feasible in the context of a mobile service delivery, methods have to be found, which grant with reasonable certainty to the network operator that he successfully delivered the service.
In the following subsections four ideas are explained, how this could be realised. The possible solutions are presented and also some open questions are mentioned. At the end of the chapter the pros and the cons of the different variants are listed.

5.2.1 Explicit Feedback from the Subscriber

A solution with an explicit feedback from the clients approaches best the mentioned procedure with the signature of the client. The transmission of a signature over a mobile network is technically not (yet) feasible. But if it is possible to get an acknowledgement from the user that he got the service delivered, the provider has something in the hand, even if it doesn't have the same legal weight as a signature.
Such a feedback has the form of a short message, which contains at least the identification of the client, i.e. his Mobile Station ISDN number (MSISDN), an identifier of the service and of the provider and probably a timestamp. These details will be discussed in details later in chapter 7.
The subscriber should send this feedback after he has received the content. Two different ways of implementation force on:

5.2.1.1 Clicking on a Hyperlink

The manual confirmation of the service delivery can be implemented by providing a link, on which the user clicks and which sends the needed information within an HTTP GET request (HTTP POST would be also possible). After the service usage (or after the download has finished) a simple WML page appears on the display of the user, which contains this link.
Such a link could have a form as in the following example:

http://provider.ch/delivery?msisdn=41791234567&tpid=0034&serviceid=647&timestamp=2003-02-28T14-30

After the "?" the MSISDN of the client, the IDs of the third party and of the service and the timestamp are indicated.
This link would have to be delivered together with the service and then it must appear on a WML page on the mobile phone. There must be a mechanism, which is able to determine and build in the MSISDN of the client (probably the MSISDN can already be built in on the server based on the request for the page). The network operator defines the provider ID (each third party has its unique ID) and the service ID. The time stamp needs to be set by an unique time indication. The system time of the device is probably not exact enough. The timestamp must have a well-defined format. In the example a format has been chosen, which is similar to the format used within XML Schema for instance. Probably the indication of the time zone is also needed.

But of course it is easy imaginable that the user may forget to click on such a link. There must be a sequence of the events that he has no choice other than clicking on this link after the service delivery. It is even thinkable that he removes the battery and restarts the device, without clicking on the link. It would be necessary to implement a lock, so that the service really only can be used after the confirmation has been sent by clicking on the link.

It is also important to consider the fact that it is not the user, who has to prove the service delivery; it is the network operator, who wants to prove the delivery! Therefore the implementation has to be as much user-friendly and simple as possible! The delivery of the service has to be confirmed and not the willingness or the ability of the user to send the confirmation!

If the service isn't delivered, there is also no hyperlink appearing on the user's display. So, no confirmation from the user implies that he hasn't got the service. But it is thinkable, that he can also confirm a "non-delivery" by clicking on a second link, which contains for instance a flag "not delivered".
If the user can click on two different links ("delivered", "not delivered") after a stream for instance, there is the danger, that he confirms the benefit of the service or he purposely clicks on the "not delivered" link. Anyway, the solution should be implemented in a way that really only the delivery of the service is confirmed and not the user perception of the benefit of the service (QoS), because this depends too much on the user subjectivity and purpose, how he wants to give the feedback.

Consequently, this implementation is only suited for services with a previous download (2-phases services). After the download, the link is appearing; the user has to click on it and can then afterwards use the service (look at the film, the picture, the page).

5.2.1.2 Activation Code

Another solution could demand to the user to enter a code to unlock the content before he can use it. This method also seems to make sense mainly for services, which primarily need a download of the content and can be used afterwards locally on the device. The code is delivered together with the encoded service and when it is entered not only the service is activated, but also a confirmation sent to the provider at the same time. This could look as it is illustrated in figure 15.

Figure 15: The activation code on the display

The idea behind this solution is quite similar to the one with the hyperlink above. In particular this is a method, which ensures that the service is locked, unless the user enters the code and so is forced to send the PoD. This solution can also go in the direction of Digital Right Management (DRM), kind of server software, which disables illegal distribution of paid content.

5.2.2 Automatic Feedback generated by an Agent on the Device

If we want to free the user from entering a code or clicking on a link, there must be a solution, which generates the PoD automatically.
The specification of MIDP OTA or the confirmed download of Nokia, as described in chapter 4.4, are going in this direction.

An elegant solution to produce a PoD would be to have a software agent on the mobile device. This agent should be able to investigate the state of the service delivery, maybe even to check the QoS, and to send the collected information back to the network operator, without any user interaction.

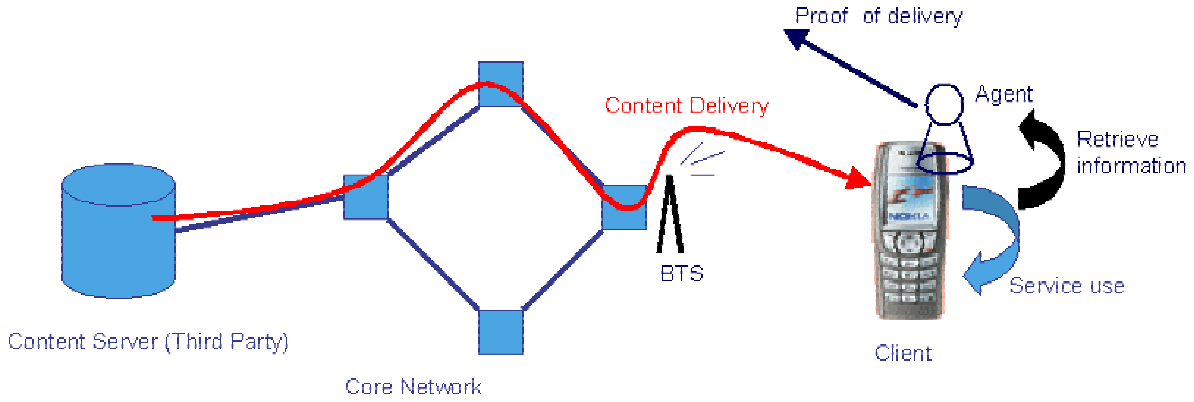

Figure 16: PoD generated by a software agent on the device

Figure 16 illustrates the idea behind an agent-based solution. The content is delivered to the mobile device and the agent, which is hosted there, gets the needed information to send the PoD.

5.2.2.1 Agent Basics

An agent is a software entity, which performs a certain task on a specific host. The agent is an independently running entity.

Software agents are an innovative technology for the efficient realization of complex, distributed and highly interactive heterogeneous systems and applications, such as virtual organisations and virtual mobile teams [SiemensCT].
Agents typically possess several (or all) of the following characteristics [EIF, Javaworld]:

- **Autonomy:**
  An autonomous agent is able to act on its own on behalf of a user. The agent runs (once activated) without any intervention from the user and has the control over its behaviour.
- **Reactivity:**
  A reactive agent has the ability to sense the environment, to process external events and to react to changes.
- **Communication:**
  The agent exchanges messages with other agents or systems (also known as social agent).
- **Cooperation:**
  Agents can temporarily cooperate with other agents to solve tasks efficiently and effectively.
- **Goal-oriented:**
  This characteristic means that the agent not only acts in response to the environment, but towards a goal, that is programmed before the agent is used.
- **Adaptation:**
  An agent may have the ability to learn by experience and modify its future behaviour on past executions. Adaptive and cooperative agents are sometimes also called “intelligent agents”.
- **Stationary:**
  Stationary agents exist as an unique process on one host.
- **Mobility:**
  Mobile agents can migrate to new places. The code and its state are moved to another location. Take care: Mobility normally refers to the code itself, but it can also refer to the nomadic user or device [FIPA].

Agents also tend to be small in size. They do not constitute a complete application. Instead, they form one by working in conjunction with an agent host or other agents. In many ways, agents are of the same scope as applets: small and limited in their functionality.

An agent for the needs of the PoD should have the following characteristics:

- Autonomy (works on his own on the mobile device)
- Reactivity (recognises the service delivery)
- Communication (sends a message with the PoD to the network operator)
- Goal-oriented (gathers the information in order to send a PoD)
- Stationary (doesn’t need to move)
- Possibly adaptation (e.g. recognise often used services)

5.2.2.2 Agent Platform

Agents need resources to act and to communicate. In the specification of FIPA (The Foundation for Intelligent Physical Agents), the run-time support providing such resources is the agent platform. Agents can run only in the scope of an agent platform providing the basic services to support interoperability: a means for sending and receiving messages and a means for finding agents [FIPAsmall].
Due to their limited resources, agents aren’t yet very common on mobile devices, but there are running projects in this context:
The main goal of the LEAP project (Lightweight and Extensible Agent Platform) is to develop a FIPA-compliant agent platform sufficiently lightweight to execute on mobile devices, such as cellular phones or PDAs, but also sufficiently powerful and open to be a first-class choice for enterprise servers.
The development of the project is based on the Java 2 Micro Edition (J2ME) platform. But agents can also be implemented by other languages (.NET-languages or C++) [Laukkanen].

5.2.2.3 Implementation of the PoD

The agent is installed on the device and is responsible to send a PoD within a message (e.g. an HTTP request), when the service has been delivered. The confirmed OTA download (see chapter 4.4) uses a download descriptor and the download agent extracts the needed information to send a delivery notification. This implementation has the advantage that the agent hasn’t to investigate the content of the service do get the needed information. This operation could be too expensive. The resources on

the devices are constricted, so it makes more sense if the information needed by the agent is created on the server side and then delivered separately, so that the agent can directly investigate this descriptor and then knows what kind of service will arrive and what he has to check and to send to the network operator.
The OTA download is (as it is already mentioned by its name) only designed for downloads. An implementation of the PoD with a software agent could also be able to control the delivery of streams, to investigate QoS and to give user specific feedbacks. It would be a more sophisticated implementation than the already existing.

5.2.3 Proof of Delivery using Protocols

Both implementation variants above generate the PoD on the device, either manually by the user or automatically by a software agent. But it is also conceivable that the PoD is created on the server side by investigating the state of protocols. This could make sense especially for streaming (1-phase) services, which seem to be more difficult to prove as already stated above.

5.2.3.1 Transmission Control Protocol (TCP)

A first idea was to implement the PoD using the well-known TCP protocol, if the content is delivered over a TCP connection. But TCP is not really usable for the transport of streaming services, because it retransmits frames depending on its implementation and introduces so further delays. Streams are normally transported over UDP in the Web. Since UDP isn't connection-oriented, it can't be used for a PoD.
This first idea during the conception was that a PoD could be implemented by logging the states of the TCP session on the server side. TCP offers a reliable transmission, so each TCP connection has well-defined states and if there is no error, the connection will be closed properly. The opening and the closing process of a TCP session are executed by a so-called "Three-Way-Handshake". The server starts the closing process, if there isn't any further data to send.
TCP provides a reliable transmission of the data by a mechanism called "positive acknowledgement with retransmission". This means that the system, which sends the data, repeats the transmission of the data, unless the receiver confirms the receipt of the data by means of an acknowledgment.
So it could be possible to look closer at these acknowledgments. This is realisable with a "packet sniffer"-software on the server. On the basis of these acknowledgments and of the fact that the TCP connection(s) are closed properly, it should be possible to determine if the packets belonging to the content have arrived at the client. Of course this would need special software on the server and the network operator, who wants to implement the PoD must have access to the content servers of the third party.

However, after a closer look at the protocols, it became obvious that this solution has restrictions in the context of GSM or GPRS. It could be probably more easily realised in an environment with wired lines.
In the mobile environment, the situation is different: as already showed in chapter 1, there is a WAP gateway, which does the transformation from the HTTP protocol to the WAP protocol. TCP connections from the server are only established to the WAP gateway and not to the mobile device.
If the connection from the mobile phone is not over circuit-switched GSM, but over GPRS and IP, there can be TCP end-to-end connections, but TCP isn't used for every kind of connections and there are still a lot of mobile phones in use, which don't support GPRS.

5.2.3.2 Real Time Streaming Protocol

The RTSP (Real Time Streaming Protocol) could possibly do a good job in this context. Real Networks, Netscape Communications and Columbia University jointly developed RTSP. It is a protocol of the application layer, which enables the control of real-time data. It steers the transmission of media, such as video and audio. It is not the actual task of the RTSP to take care of the transmission of the data, but only to induce their transmission.
The RTSP is extensible. Existing methods can be extended with new parameters, new methods can be added and even a new version of the protocol can be designed.
It supports datagram protocols such as UDP but also connection-oriented protocols such as TCP.
RTSP is a text-based protocol. The RTSP messages consist of a starting line, a header and a body.
The action to execute is indicated in the starting line. The header contains parameters, such as the sequence number, the number of the current session, a content type parameter, which indicates a following body and how it is encoded, or the content length.
The optional body can enclose additional data. Such as in HTTP there are request and response messages, whereas a request and response can both come from the server and from the client. RTSP

is a symmetric protocol in contrast to HTTP. In addition, HTTP is a stateless protocol and a RTSP server (and client) have to maintain “session states” in order to correlate RTSP requests with a stream. But the syntax of RTSP and the operations are similar to HTTP.
The interesting RTSP function in the context of the PoD is the GET_Parameter method. With this method the client can request the value of a parameter at the server and first of all (what really is interesting for a PoD) the server can also request parameters from the client [RTSPRFC].

Example:

```
Server->Client:  GET_PARAMETER rtsp://example.com/fizzle/foo RTSP/1.0
                 CSeq: 431
                 Content-Type: text/parameters
                 Session: 12345678
                 Content-Length: 15

                 packets_received
                 jitter
```

This request demands the value of the received packets and the jitter on the client side. The answer from the client could be:

```
Client->Server:  RTSP/1.0 200 OK
                 CSeq: 431
                 Content-Length: 46
                 Content-Type: text/parameters

                 packets_received: 10
                 jitter: 0.3838
```

If the body is empty, the method can be used to test the availability of the client or the server (such as a “ping” within an IP network).
In the RFC it is mentioned that some servers do not support this method. It is also mentioned that this method is only loosely defined with the intention that the reply content and response content will be defined after further experimentation [RTSPRFC].

It isn’t really obvious, if it is already possible to use this function. There is also nothing written about clients, which support this function. On the Web a lot of documents about RTSP can be found. But within them the examples for the GET_Parameter method are always the same as it is in the RFC. This consolidates the impression that the method can’t yet really be used in a broad context. Nobody seems to have a concrete implementation of it and just copies the example from the RFC.

However, the protocol might theoretically be used to retrieve some information from the client, which could be helpful to implement a PoD. The packets received by the client could be counted and the result afterwards be compared with the number of packets sent from the server.
The problem in this context is probably that only by counting the received packets, we can’t be sure, that this are the “correct” (carrying the service) packets and if they are in the right order (so that for the stream can be replayed without too much delay or jitter).
RTSP supports a session number: A session is a complete RTSP “transaction”, e.g. the transmission of a movie. A session typically consists of a client setting up a transport mechanism for the continuous media stream, starting the stream and closing the stream. The server needs to maintain a “session” state” to be able to correlate RTSP requests with a stream. RTSP methods that contribute to the state use the session header field to identify the RTSP session whose state is being manipulated.
Anyway, in the example above, only the received packets and the jitter are listed. It is imaginable that also other parameters could be fetched like this. So it could perhaps be possible to not only have a PoD, but also a proof of the service quality, which might be of interest in the context of streams.

The newest handsets already support streaming media. The Nokia 7650 for instance allows installing the RealOne Player Mobile. The finish operator Sonera provides on his Web site an emulator to use a stream over RTSP [Sonera]. So it can be assumed that RTSP is a protocol, which will gain importance in the near future with the steady improvement of the handsets.

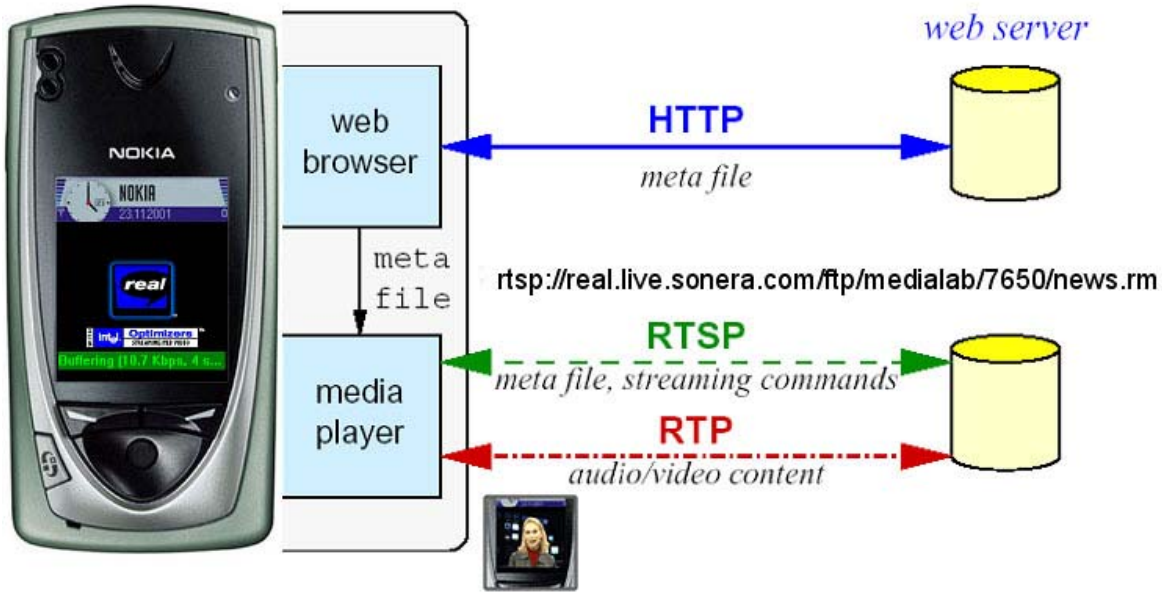

Figure 17: The RTSP in use

Figure 17 illustrates the RTSP in use. The actual content is transmitted over the Real Time Protocol RTP. The media player on the mobile device (in this case a Nokia 7650 with a Real Player) reproduces then the stream. The indicated URL is the one, which is requested from the browser of the handset.
It can be concluded that the newest devices support RTSP. Of course streaming make demands on the bandwidth. But it is thinkable that RTSP could be used for the purposes of a PoD in the near future, if the devices and the server support the needed functions.

5.2.4 Proof of Delivery using Network Probes

So far, three solutions for a PoD have been discussed: Two solutions generate the PoD on the device either by a confirmation carried out by the user or produced by a software agent.
The third method settles the PoD in the protocol stack and tries to get the needed information on the server side. But it may be that the provider has no access to the server of the third party, which acts as the content provider.
But the needed information could also be fetched elsewhere in the network by using probes.

5.2.4.1 Network Probes

A probe is an “on-the-wire” device for real-time monitoring of telecommunications networks. Probes are also known as “analysers”, “network monitors” or “sniffers”. Usually probes are specialised for certain types of analysis, such as fault resolution, session tracking, packet payload analysis, intrusion detection or accounting [IP-value].
The idea behind a possible implementation of a PoD by means of network probes is that the probe sniffs the packets, analyses their content and logs the relevant information, so that afterwards can be proven that the content passed the location of the probe on its way to the client’s device. The problem is that the probe has to be as close as possible to the mobile terminal to cover the most possible of the way of the content to the client. The closest point to the mobile terminal, where it is possible to install a probe, is the BTS. But even if the probe would be there, the air interface wouldn’t be covered, the part of the network, where the most errors can occur.
Another drawback of such a solution is also that the closer to the mobile terminal the probe is installed, the more probes are needed and the more expensive the realisation would be.

Another approach could be that not the data belonging to the content delivered to the customer is logged, but the control flow (if available, e.g. with TCP) on its way from the mobile terminal back to the server. So the probe doesn’t have to be so close to the customer and can be installed anywhere in the network.
Such a solution would also solve the problem that the network provider, who wants to generate the PoD, has probably no access to the content server of the third party.

The current WAP content billing architecture of Swisscom Mobile uses already an implementation for the PoD, which goes in this direction. It is not explicitly a probe, but the confirmations from the handset are logged at the WAP gateway in case of connection-orientated transmissions (see chapter 3.3).

5.2.5 Pros and Cons of the different Solutions

To sum up the different approaches, in the tables below can be found their pros and cons as far as they can be judged at the moment.

5.2.5.1 Explicit Feedback from the Subscriber

Implemented by an HTTP GET request or by an unlock key to be entered before the downloaded content can be used locally.

Pros:	Cons:
• Independent of the device, since hyperlink or code are integrated in a WML page. • Easy to implement (at least the HTTP GET request. The implementation of an unlock key may become more complicated. • Apparent for the subscriber (user awareness) • Significant method as long as the user is forced to send this feedback (no way to use the service without clicking on the link or entering the unlock code).	• Dependent on the user's behaviour (may forget or deny to click or enter the code) if he isn't forced to. • May be annoying always having to enter a code or to click on a link (user-friendliness). • Point of failure, if the service can only be used after the confirmation of the delivery. It is thinkable that the PoD can't be sent due to any reason (no reception area etc.). • Not suitable for every kind of service. Makes only sense for downloads.

Table 2: Explicit feedback from the subscriber

5.2.5.2 Automatic Feedback generated by Agents on the Device

The automatic feedback may be generated by an agent, which is installed on the device and gets information from the device and from a download-/service descriptor, which is delivered together with the service.

Pros:	Cons:
• User independent and user-friendly, because the client doesn't have to care. • May be used arbitrarily and not only to give delivery acknowledgments but probably also to investigate the QoS of the delivered service and to send periodical download reports. Useful for streaming services. • Scalable (only a software update is needed in case of new services) • Fast solution • High significance • The new generation of handset supports broadly MIDP and offers so a usable platform for agents.	• "Big brother". The users don't like if information is sent from their device (the advantages for the users -> possible discounts to their accounts etc. have to be clearly communicated!) • Limited capacity (memory, CPU) of the devices to host agents (will certainly be improved with the time). • Complicated (could depend on the device and on the service) and is probably rather expensive to implement. • The agents need to be installed on the devices. This requires a special procedure, before the services can be used on the device. • Higher stimulation to manipulate the procedure than with other solutions -> possible danger of frauds.

Table 3: Automatic feedback generated by agents

5.2.5.3 Proof of Delivery using Protocols

The information is collected on the server side by means of a protocol such as TCP or RTSP. This solution would mainly be interesting in the context of streaming services.

Pros:	Cons:
• User independent and the users don't have to care of. • Independent of the end device, provided the end device supports the protocol.	• Support of the protocols not warranted by every server or device. • Implementation probably complicated (e.g. only counting of the delivered packets is not enough to be sure, that the service is really delivered). • The network operator has probably no access to the content servers of the third party. So the implementation becomes even more complicated.

Table 4: PoD using protocols

5.2.5.4 Proof of Delivery using Network Probes

Packets are sniffed on their way from the server to the client or vice versa. The information is then extracted from these packets.

Pros:	Cons:
• Independent from user and device • Can also be used if the network operator doesn't have access to the content server of the third party.	• Significance. Depending on the location of the probe. • Expensive solution, if the probes have to be located as close to the client as possible (e.g. at the BTS). Then more probes are needed.

Table 5: PoD using network probes

5.3 Conclusion

Four different implementation variants for the PoD have been presented above. The goal of this fifth chapter was to show, how the delivery of different types of services could be confirmed depending on the point in time, when the confirmation can be generated and from the point of view, how this confirmation can be generated. It has been found, that the PoD can't have the same form for every kind of service. But this fact doesn't exclude that the PoD is technically implemented in the same way for every service. Probably it makes even more sense to have only one technical solution, because it is simpler to maintain.
The four implementation-variants have all their pros and cons. The most interesting (and perhaps also the most challenging) is the one with the software agents on the mobile device:

The user hasn't to care of, it can be used for every kind of service, if the agent can be configured accordingly and it offers the most possibilities to use this technology also for other purposes.
The problems with this solution seem clearly to be in the implementation, considering the limited capacities of the devices.

Therefore the agent-based solution will be discussed in more details within the next chapter and it will also be showed, how it could be implemented.

6 Proof of Delivery with Software Agents

In the fifth chapter several possible solutions for a PoD have been discussed. This chapter now contains a more detailed elaboration of the most interesting and probably also the best solution for a PoD: The solution bases on software agents on the mobile device.

As presented in chapter 4, implementations are already in use, which go in this direction, e.g. Sun's MIDP OTA Provisioning [MIDPOTA] for the download of Java MIDlets or Nokia's confirmed download. They use also some kind of agent on the device, which is able to read the download descriptor, which is sent to the device before a download is executed. But these solutions are only suited for downloads, i.e. for 2-phases services. Unfortunately it is not mentioned in the papers of Nokia or Sun, how this download agent works in detail and how it is implemented. It can be assumed that it is not an agent in the complete sense of the definitions in chapter 5.2.2, but it is only called agent, because it executes autonomously without any user interaction.
A problem is also that solutions like the one of Nokia are manufacturer-specific and don't work with other devices. For a network operator it is simpler if the implementation of the PoD works with all devices (which comply with some technical requirements, see below).

An ulterior motive by implementing a solution based on agents on the mobile device may be that the agents can probably not only be used to generate a PoD, but could even also take on other functions.

6.1 Java 2 Micro Edition

As already mentioned in chapter 5.2.2, an agent needs a platform to be executed. Today this would correspond to a Java Virtual Machine on the mobile device. Thus, this chapter first provides a short introduction into Java within a mobile environment.

Java is a free programming environment developed by Sun Microsystems. The special thing about Java is that it is independent from the platform ("write once, run anywhere"). The compiler doesn't produce directly machine code, but a Java byte code, which will be interpreted by the Java Runtime Environment.
Because the programs aren't executed by the system, they can't cause a system crash. Only the runtime environment may crash. To prevent this, Java has a built-in runtime code checker, the byte code verification.

There exist three different Java versions, developed for different devices:

- Java 2 Enterprise Edition (J2EE) for server applications
- Java 2 Standard Edition (J2SE) for normal PCs
- Java 2 Micro Edition (J2ME) for mobile phones, PDAs etc.

The core of the runtime environment of every Java system is the Virtual Machine. This is the interpreter, which executes the applications by transforming the byte code to machine code. The Virtual Machine has to be developed for every kind of device.

6.1.1 Configuration

A J2ME configuration provides a minimal set of features that all devices in the configuration must support. Two configurations are foreseen:

- **Connected Limited Device Configuration (CLDC)**
  The CLDC configuration is designed for battery-powered devices with a 16bit, 16 MHz-processor and 160-512 KB memory for the whole Java platform (incl. the applications). The network connection has a limited bandwidth.
  Examples are mainly mobile phones, PDAs or Smartphones.
  This configuration has massive restrictions. For instance there is no runtime byte code verification. Instead of that, the compiler inserts a checksum in the byte code. The runtime environment of a CLDC device then just checks this checksum. Of course J2ME is less secure like this, because there can be applications with faked checksums. CLDC uses the so-called KVM (Kilobyte Virtual Machine). The KVM corresponds to the Java Virtual Machine, but it has

been adapted to the needs of small devices. It is called “kilobyte” VM because it needs only 50 Kilobytes and not several megabytes as the standard edition.

- **Connected Device Configuration (CDC)**
  This configuration is designed for line operated devices with a faster CPU and at least 512 KB memory and a faster network connection. CDC uses the classic Virtual Machine.

6.1.2 Mobile Information Device Profile (MIDP)

A profile is an extension to a configuration. It provides the libraries for a developer to write applications for a particular type of device.
The MIDP (current version is MIDP 1.0) defines an application program interface (API) for user interface components, input and event handling, persistent storage, HTTP networking and timers, taking into consideration the screen and memory limitations of mobile devices [Muchow].

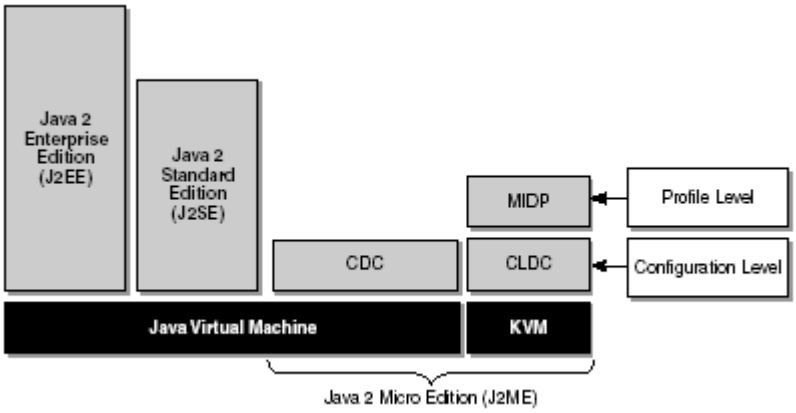

Figure 18: The various Java editions [Muchow]

Applications, which are developed for MIDP, are called MIDlets. In order that MIDlets can be executed on all MIDP devices, MIDP has to be restricted to the lowest common denominator of all target devices (mobile phones, PDAs, etc.). This means that there is no support implemented for SMS, infrared, Bluetooth, ring tones or other special functions of the devices. Access to stored data, for instance to the address book, isn’t possible too.
The access to such data and functions is only possible over proprietary libraries of the manufacturers. But these proprietary libraries can be considered as a danger for the MIDP standard. If the developers fall back on such device specific APIs, there is no way to use the MIDlet also on other devices.

The MIDP supports a subset of the HTTP protocol, which can be implemented using both IP protocols such as TCP/IP and non-IP protocols such as WAP, utilizing a gateway to provide access to HTTP servers on the Internet [MIDPSpec].

MIDP 2.0 will provide solutions to solve the problems of the manufacture specific libraries. MIDP 2.0 will allow network access also over HTTPS (MIDP 1.0 allows only HTTP). Optionally, access on serial ports is integrated. This second version of MIDP will be available in large volumes in Summer 2003 [Knudsen].

MIDlets extend the MIDP classes with specific functions. The MIDlet application manager uses these extended classes to administrate the MIDlets (execute, terminate, interrupt etc.).

6.2 Agent based Solution for the PoD

In order to create a conception, how a PoD based on agents could be implemented, the author of the thesis had contact with a representative of the enterprise Softcom (www.softcomponent.ch) in Freiburg (CH). Softcom has already conducted projects concerning agents on mobile devices. Their perceptions initiated the assumption that an implementation of a PoD should be feasible.
The following subsections contain a short description of these projects and show afterwards, how to go about implementing the PoD.

6.2.1 Agent Projects

6.2.1.1 Project Sniff: Java Mobile Agent for Wireless Devices

Sniff is a mobile agent with a small footprint able to collect data on third generation wireless devices (cellular, Smartphones etc.). Sniff runs on all third generation wireless devices, which have a Java Virtual Machine (J2ME).

This agent moves autonomously from device to device. The strategy is implemented by a Sniff Server, which is installed at the provider and manages the mobile agents gathering data from the devices.
A Sniff is able to collect data relative to the following topics:

- Use of GPRS, Edge or UMTS services
- User's communications
- General mobile device use

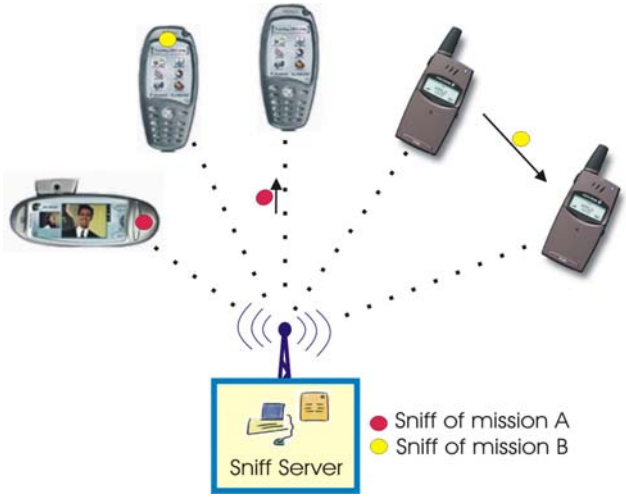

Figure 19: Sniff mobile agent

The agents are hosted within the platform "Grasshopper", which allows the mobility of the agents [Grasshopper].
The conclusion of the project was that stationary agents make more sense for mobile devices with their limited capacities, because the overhead for mobile agents is too large.
For the purposes of the PoD it makes even no concrete sense to use a mobile agent. It isn't necessary that the agents move, they just have to collect the information from one device respectively.

6.2.1.2 Project Smart Agent

The Smart Agent is a stationary agent, which allows executing tasks on mobile devices. This agent charges modules in function of a profile in order to execute specific tasks on the device in collaboration with the server. Thus the user of the mobile device can download functionalities of the agent.

This agent is downloaded with a MIDlet, which allows launching the agent on the mobile device. The modules are downloaded with MIDlets too, which offer graphical interfaces for the configuration and administration of the data.

Softcom has done trials with the Smart Agents in the context of synchronising calendars of events.

The agent can be launched on devices, which have a Java Virtual Machine: MIDP 1.0 / CLDC 1.0 (http://wireless.java.sun.com/device/). The compatibility of the modules on the different devices seems to be a problem, but with MIDP 2.0 this should be improved.

The memory capacity and the CPU of the mobile terminals have an impact on the performance of the agent. But it can be assumed that the process of improving these capacities will continue, so that the devices will steadily have a better performance, but the resource gap between mobile and stationary devices will always be there [Laukkanen].

6.3 PoD with the Smart Agent

A Smart Agent could be used to implement a PoD. The concept is explained below.

6.3.1 Requirements

- The agent introduces only the least possible overhead to the service delivery.
- The performance of the mobile device mustn't be influenced.
- The customer shouldn't be affected by the agent on the device and should also not have to do anything with the agent handling.
- The implementation shouldn't be too expensive.
- The agent has to work reliable, because the PoD can have an influence on the billing process.
- The implementation has to be secure so that the PoD can't be manipulated.

6.3.2 Implementation

Figure 20 illustrates the possible implementation.
Theoretically it is thinkable that a software agent just arrives at the mobile phone immediately before the service is delivered (as the content descriptor in the solution of Nokia mentioned above). The agent would then have to be installed on the device and launched. Of course this takes time. The overhead is also quite considerable, because with every service delivery an agent would have to be sent too, even though it is only used once then.
Therefore the conception plans to equip the agent only with some basic functionality. The service specific information to generate the PoD is then delivered within a small Java class to the handset.

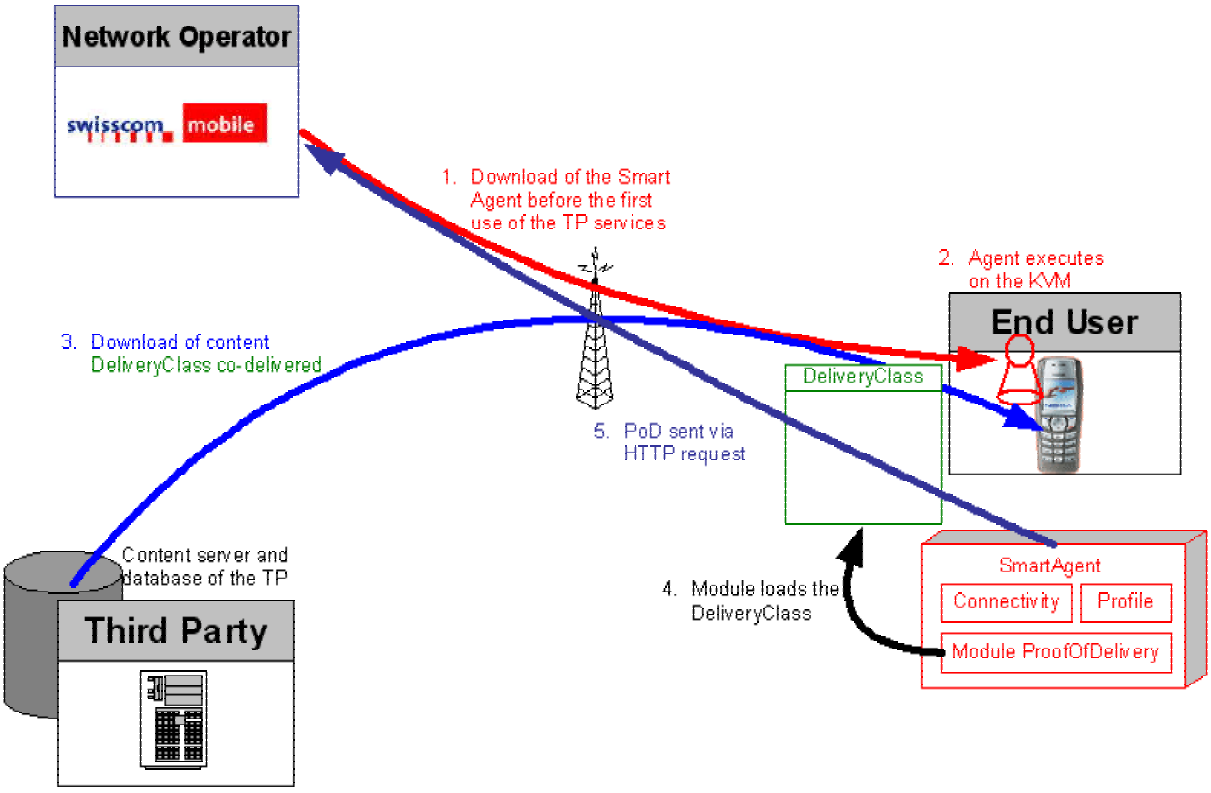

Figure 20: PoD with a Smart Agent

The PoD could be generated with a Smart Agent in the following way:

1. When the customer uses the third party services of the network operator the first time, his mobile device has to be configured. During this process, the Smart Agent is installed on his terminal. The user must agree to this installation. Before, he has to be informed about the installation and the functionality of the agent on his device.

The customer should also have the possibility to decline the installation of the agent and to use the third party services without an agent and therefore also without a PoD. The functionality of the agent has to be explained carefully.

The information for the client could look as follows:
"Your mobile device will now be configured to be ready to use the third party services of Swisscom Mobile. During the installation process a software agent will be installed on your device. This agent is a small piece of software, which will investigate every delivery of content to your device and will automatically send a delivery confirmation and, depending on the service, a feedback about the delivered quality of service to Swisscom Mobile. With this information Swisscom Mobile is able to charge you only for services, which are delivered to you completely and in a satisfactory quality. In addition, the agent helps Swisscom Mobile and the third parties to steadily improve the quality of their network and of their services.
The agent will send no personal information and no location information to Swisscom Mobile and does not influence the performance of your device during the regular use and during the use of third party services. You will also be able to use the third party services, without the installation of the agent, but so you won't benefit from possible discounts. Therefore we recommend you to install the agent on your device.
Do you accept the installation of the software agent? Yes – No."

Maybe the functionality of the agent needs to be communicated even in more details (what he does and what he doesn't). But the text can't be too long (due to the limited display size of the mobile phone it is annoying to scroll too much) and mustn't be too complicated and contain too many technical details, which confuse the clients.
If the customer refuses having the agent installed on his device, the configuration process should continue anyway. Nevertheless the customer should also be able to use the third party services without any restrictions. Such questions have to be examined carefully and also to be checked from the point of view of legal issues.
If the client accepts the installation, the software agent is transmitted to his device, where it is installed. The user is now ready to use all the third party services and the agent is ready to generate the PoD for these services.

2. The Smart Agent executes autonomously on the KVM of the handset. He should execute in the background and only be called by some kind of interrupt handler, when effectively a service delivery is executed.

3. When the client downloads content (can also be a stream), a small Java class (can for instance be called "DeliveryClass.class") is delivered with it. This class is generated automatically and dependently on the kind of service with an algorithm on the server of the third party. The class contains service specific information, for instance the type of the service (download, stream), the length in bits, the number of frames etc.

4. The generic module (called e.g. Module ProofOfDelivery) in the agent loads then this class. With the information out of the class, the module can generate the PoD by investigating the service delivery.

5. The PoD is then packed into a message and sent to an address specified in the module (probably an address of the network operator). The best way would be to use an HTTP request (is supported by MIDP).

6.3.3 Problems and Challenges

The software agent is permanently installed on the mobile device and has to do his job during every service delivery. Therefore the agent has to run always, when the mobile device is turned on (because the user can start a request for a TP service anytime.
The problem in this context is that the agent has to be launched every time, when the user switches on the mobile device. In the trials with the Smart Agents, the agents have been started manually. This wouldn't be very comfortable for the users, i.e. the agent has to be launched automatically, when the user turns on the mobile device or at least, when the user establishes a GPRS connection (modern phones, which support MIDP 1.0 all support GPRS, so it isn't necessary to consider the case of GSM phones, which have to call the number of the WAP gateway for a circuit switched connection).

Another problem is that the content is sent to the device and not to the agent. This means that the agent has to get the information that content has arrived and that he has to execute the module.

Difficulties appear, if not a single bit of the content arrives on the mobile device. If the DeliveryClass arrives, then it is able to determine that the delivery wasn't successful at all. But it is also thinkable, that even the class doesn't arrive on the mobile device.
To the same issues belong also situations, when the user turns off the mobile device during the service delivery (during the download) or when the battery power is too low and the device switches off.
Such situations can be solved if there is a mechanism, which is able to restart the delivery at the point, where it has been interrupted, as soon as the device is available again. Of course this doesn't work for live streams, but it should work for a download of a game for instance.

But if not a single bit of the service and also of the DeliveryClass arrives at the mobile device, the agent isn't aware that there has been an attempt to deliver a service. So the agent can't send a negative PoD to the network operator.
This problem may be solved in the following ways:

- No special implementation. The risk that nothing arrives at the mobile device is accepted. This implies that the billing architecture treats the event "no PoD arrived" accordingly (see chapter 7).
- The DeliveryClass is sent to the device before the service delivery. The reception of the DeliveryClass has then to be confirmed by the device. The content is sent only after this confirmation has arrived at the server. So it is assured that the DeliveryClass is on the device in any case and so also able to send a PoD if not a single bit belonging to the content arrives at the mobile device.
  The drawback of this solution is that there is an overhead before the service delivery takes place. The user has to wait until the confirmation of the reception of the DeliveryClass arrived at the server and the delivery can start. But this should normally only be a question of a few seconds.

An important issues are frauds: It is important that the agent can't be faked and abused.
The implementation with the Smart Agent seems to be quite secure: The class can't be manipulated because it arrives compiled at the device. Somebody would have to know, how the class works to be able to send a faked class to the mobile device, which could manipulate the PoD.
With the upcoming introduction of MIDP 2.0, the PoD message could be sent encrypted over an HTTPS connection back to the server, so that also there is no danger.

7 Integration of the PoD in the Billing Stream

This chapter can be considered separately from the chapters before. It shows, how the PoD can be integrated in the billing process, if it is used as a proactive instrument, which can have an influence on the bill of the customer.
Therefore, it can basically be assumed for the following subchapters that the PoD arrives in form of a short message from anywhere in the network and contains the needed information. There are only some parts, which explicitly mention that an agent on the mobile device generates the PoD, because some assumptions concerning the data flow have to be made.
The chapter is kept as general as possible. The integration of the PoD in the WCB architecture of Swisscom Mobile is treated only as a special case.

7.1 Objectives

The PoD message should contain everything, so that it is applicable for streaming services and for downloads. It contains also enough information that it can be used as a proactive instrument in the billing process.
The PoD can be used proactively to be able to offer discounts to the customers in case of failed or incomplete service deliveries. These potential discounts act as an instrument to increase customer satisfaction.
But nevertheless all efforts should be made to ensure a reliable delivery of the services. This is still the best way to attract the attention of the customers to the third party services and to increase customer satisfaction. Ensuring the QoS should be done in the network and not in the billing. Billing should only help to close a gap, if there are no other ways.
The discounts should only be a means in special cases. Too many discounts are certainly not in the sense of the network operator, because the revenues would then decrease considerably. In the author's opinion, discounts should only be applied in maximal 3-5% of all service delivery cases.
However, possible discounts may arise the motivation of the subscribers to use the TP content services, because they are aware that they only have to pay in case of a successful delivery. On the other hand it may also damage the image of a network operator if the customers believe that a lot of transmissions can fail. The communication of such matters has to be done carefully.

A download (of a Web page, a picture, a ring tone, a video clip etc.) has only a benefit for the customer if it is complete. Thus an incomplete download would entail a discount. Therefore it has to be ensured that interrupted downloads (e.g. due to a tunnel) can be restarted automatically from the beginning or from the point, where they have been interrupted as soon as the client has reception again.

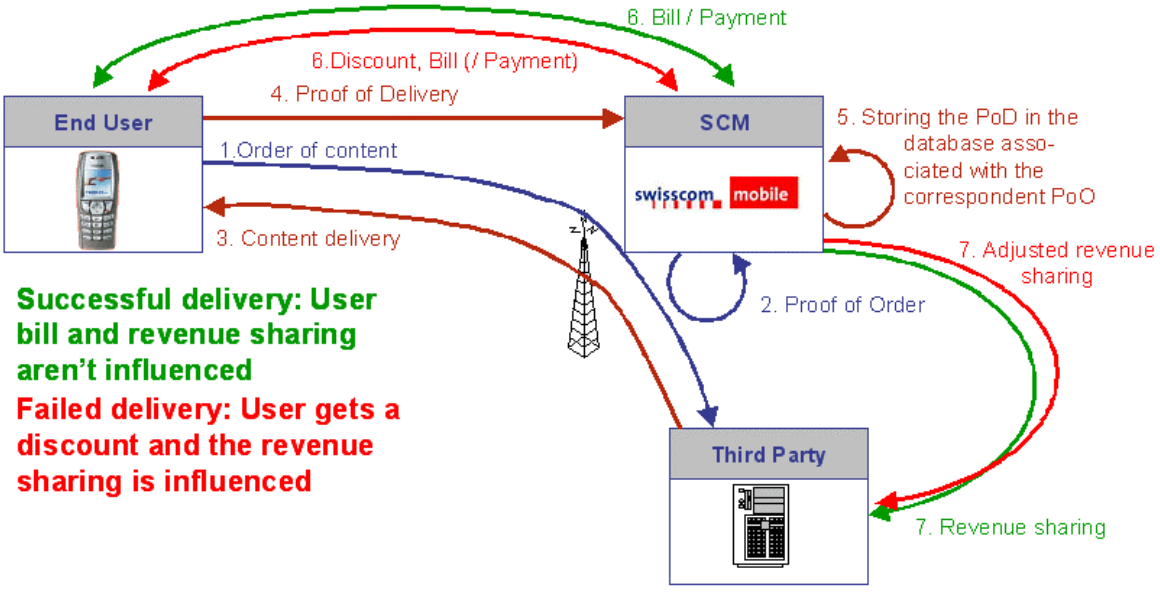

Figure 21: The integration of the PoD in the billing process of Swisscom Mobile

Figure 21 shows the two different situations of a service delivery (successful / failed) and the role of the PoD for the billing and revenue sharing (arrows 6 and 7 respectively in green or red).

7.2 Structure of the PoD

A difference is made between the PoD, which arrives at the system (e.g. encapsulated in an HTTP request) and the one, which is finally stored in the database and used for the re-rating (see also chapter 7.5).
Table 6 gives an overview about these two slightly different structures, while the rest of this subchapter contains the explanation to the fields.

Arriving PoD	Final PoD (stored)
	Identifier
MSISDN of the client	MSISDN of the client
Third Party ID	Third Party ID
Service ID	Service ID
Transmission Medium	Transmission Medium
Delivery Status	Delivery Status
Failure Reason	Failure Reason
Responsibility	Responsibility
Timestamp	Timestamp
Associated Proof of Order	Associated Proof of Order
MoreFollows Flag	
Final Flag	

Table 6: The structure of the PoD

7.2.1 Identifier

Each PoD contains a unique identifier (integer value). The identifier is added in the system, before the PoD is stored.
Example: 5489

7.2.2 MSISDN

The MSISDN identifies each subscriber uniquely and allows finding stored PoDs for each customer and is also needed to offer possible discounts to the right client.
Example: 41791234567

7.2.3 Third Party ID

The ID of the TP is needed because the user can order different services in a short period of time and so the PoD can be associated to the right TP and to the right service of this TP.
Example: 0034

7.2.4 Service ID

A customer may also order different services from the same TP in a short period of time. That's why also the ID of the service is needed.
Example: 647

7.2.5 Transmission Medium

Third party services may be delivered over different medias. It is even also thinkable that they are ordered about another media than the one, which is then actually used for the delivery. E.g. ordering an SMS push service over WAP or ordering a content to the handset from a PC. This field is mainly used for statistical reasons.

7.2.6 Delivery Status

This field contains the main information of the PoD. It has one of the following four different values:

- **Delivery ok**
  The service has been delivered entirely and without any errors.
- **Delivery not ok**
  The service hasn't been delivered (i.e. an incomplete download or a stream with too much interruptions).

- **Delivery partially ok**
  The service has been delivered, but not completely. This value is only relevant in the context of streams, because an incomplete download has no value for the customer (a download can either be ok or not ok, but nothing between), but a stream with some interruptions can still offer a benefit to the user.
- **PoD undefined**
  An error has occurred during the generation or the transmission of the PoD. The PoD can't be stored in the database or be treated in the billing chain.

7.2.7 Failure Reason

This field contains the reason of the failure in case of an unsuccessful delivery.

- **End device**
  The error occurred at the end device (empty battery, user switched off the device, no free memory etc.).
- **Network**
  The error occurred in the network, e.g. loss of service in the air interface.
- **Server**
  The error occurred already at the server (unavailable database etc.).

An even more sophisticated implementation would be, if this field contains a text in the sense of "battery empty" or "network congestion", which finally would also appear on the invoice of the customer in order to inform him about the discount.
To decide about the failure reason seems to be a tough undertaking on the first view. Probably it is also really a problem to implement. An agent on the mobile device could decide if the error occurred on the phone or in the network or at the server. But it seems to be difficult to make the difference between errors in the network or at the server. However, to be able to treat all the possibilities of the PoD in the billing process it is assumed that this information is available.

7.2.8 Responsibility

The failure reason alone doesn't stringent mention the responsibility for an error. An example is if the network operator hosts the server of the third party.

- **End customer**
- **Network operator**
- **Third party**

7.2.9 Timestamp

Each PoD contains a timestamp. It is for instance needed, if a download should have to be completed within a certain time frame.
Example: 2003-02-28T14-30

7.2.10 Associated Proof of Order

Swisscom Mobile has implemented a PoO in the WCB architecture.
Each service order causes the system to store a PoO in a database, where it is kept for 180 days. In order to be able to make the association between the PoO and the corresponding PoD, a number of the PoO should be inserted in the PoD. The bill ID (sequence number) could be used for this purpose. The bill ID is inserted in a header field of the generated page of the TP. The WAP gateway can then read this number, when the page is on its way to the client and it acts so in the today's solution as a kind of PoD. This header isn't sent after the binary encoding of the content at the WAP gateway to the customer. So, the needed information for the association between the PoO and the PoD has to be transmitted otherwise. It is thinkable that it is integrated in the DeliveryClass (see chapter 6.3) for the agent on the handset. This could be a quite tough undertaking, because the PoO and the PoD can come from different sources, if for instance WAP is used to order an SMS service!

7.2.11 Flags

The PoD can be implemented in a way, that more than one PoD arrives at the system for one service. For (longer) streaming services it is for instance thinkable that periodically a PoD message is generated and sent to the system. Therefore the PoD contains a flag "MoreFollows", which can have the

values 0 or 1, and a flag “Final” for the last message, which can also be 0 or 1. The billing architecture has then to collect the messages and to generate one definite PoD to be stored in the database and to be used in the billing process. The generation of the final PoD out of the “intermediary” PoD messages has to be done according to some defined rules.

7.3 Discounts for the End Customer

Above it is mentioned that the agent is able to decide about the delivery (he knows the boarders between “not delivered”, “partially delivered” and “delivered”).
But the agent could also send the raw data to the system and the calculation is made there. But in the following it will be assumed that the agent takes the decision and sends only the result to the billing system.

The PoD should be integrated in the billing process in a way that discounts can be offered proactively. On the other hand it is also thinkable that the PoD is stored like the PoO and only used in case of complaining customers. This would mean that the clients are charged anyway and the PoD isn’t investigated every time, but only for statistical reasons. The customer bears the risk that the service can’t be delivered and he has to pay, unless he complains at the CuC if necessary and there is no positive PoD.
This solution is not very user-friendly. But compared to a service delivery without any PoD it has at least the advantage that the user has the opportunity to complain and it is possible that he isn’t charged, if the CuC can’t find any positive PoD. It is thinkable that a lot of users then try to call the CuC to check if there aren’t missing positive PoDs!

But the assumption for this chapter is that the PoD is used proactively and so only this case is treated. Depending on the delivery status field and the failure reason field of the PoD the decision about the discount for the customer can be made. The responsibility field is for the present not yet considered. It is assumed that the user is responsible for errors on the end device, the network operator for the network and the third party for errors at the content server. The field has been mentioned in the structure above only for the sake of completeness.

Examples for discounting rules:

- **Delivery ok**
  The user has to pay the normal price as it is declared, when he ordered the content and agreed to be billed by the network operator on behalf of the TP.

- **Delivery not ok**
  - **Failure reason End device**
    The customer didn’t get the service delivered due to his own behaviour (not enough battery power or memory etc.). He is charged normally.
  - **Failure reason Network**
    The customer isn’t charged because it isn’t his fault.
  - **Failure reason Server**
    The customer isn’t charged because it isn’t his fault.

- **Delivery partially ok**
  - **Failure reason End device**
    The customer didn’t get the service delivered due to his own behaviour (not enough battery power or memory etc.). He is charged normally.
  - **Failure reason Network**
    The customer is charged, but gets a discount. The agent on the device decides about this partial delivery (knows the boarder between “not delivered”, ”partial delivered”, “delivered”). The discount for the client is fixed at a certain percentage (e.g. 30%).
    On the other hand it is also thinkable that the agent gives back the exact percentage value of the service delivery (would need another field in the PoD message) and the discount for the client can then be offered from the system accordingly to this percentage value.
  - **Failure Reason Server**
    The customer is charged, but gets a discount. The same rules are applied as in the case of “Failure Reason Network”.

- **PoD undefined**
  The PoD can't have an influence on the billing process and therefore the customers have to pay anyway.

It is important that the amount and the reason of each discount are listed on the customer's invoice, in a clear way so that the client isn't confused and the invoices stay plausible.

7.4 Settlement Rules

If the customer gets a discount, there is a reduced money flow towards the network operator and so also the settlement between the network operator and the third party is influenced. The logical consequence is that both parties get the appropriately reduced amount of money.
But if the PoD even contains the reason for a failed or incomplete service delivery, it is possible to subdivide the costs between network operator, third party and customer according to the cause of the failure.
The settlement rules can be applied record per record, i.e. for every service delivery separately. On the other hand the network operator could also guarantee to the third party, that for instance more than 90% of the deliveries are conducted successfully. If the network operator can't fulfil this agreement over a certain period (e.g. one month), he has to pay a fine to the third party in order to compensate for the revenue loss.
Figure 22 explains the revenue sharing by means of three situations.

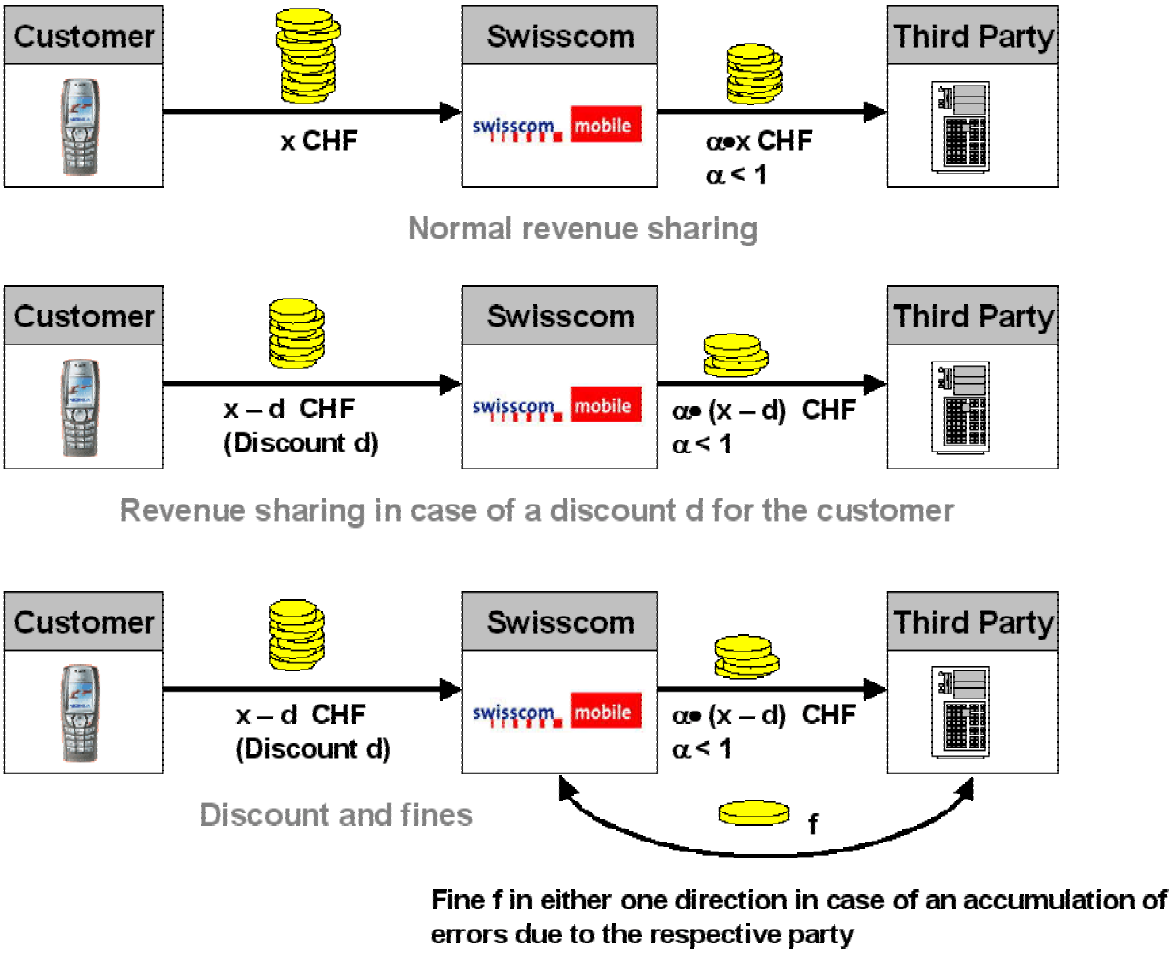

Figure 22: Settlement between the network operator (Swisscom) and the third party

- **Normal revenue sharing**
  The subscriber has to pay x CHF to the network operator. The third party gets its share of this revenue x according to the settlement agreement between the network operator (SCM) and the TP.
  This share is x•α CHF, whereas α <1, e.g. α =0.35.

- **Revenue sharing in case of a discount for the customer**
  If the SCM subscriber gets a discount, the revenue sharing between SCM and the TP is also influenced. The logical consequence is that both parties get the appropriately reduced amount of money. The user only has to pay x-d CHF (whereas d is the amount of the discount). The logical consequence is that the shares between SCM and the TP are also reduced linearly dependent. The TP gets $\alpha \bullet$(x-d) CHF.

- **Discounts and fines**
  But if the PoD even contains the reason for a failed or incomplete service delivery or determines the responsibility for an error, it is possible to assess the revenue sharing. The TP gets $\alpha \bullet$(x-d) CHF as above. In addition, the PoDs are collected and analysed over a certain period (e.g. 1 month). If there is an accumulation of errors in the network or at the server during this period, the respective party has to pay a fine f to the other (e.g. if more than 5% of the deliveries have failed due to errors in the network, SCM has to pay the fine f to the TP).

7.5 Integration of the PoD in the Architecture of SCM

It is assumed that the PoD is sent from the handset to the systems of SCM (e.g. generated by a software agent).

7.5.1 Data Flow

It can be distinguished between three different billing streams: The “Common Billing Stream” contains the generation of the PoD, the correlation with the Proof of Order and the storage of the PoD in the database. The subsequent flows are the “Customer Billing Stream” and the “Partner Billing Stream”. The former treats the influence of the PoD on the invoice of the customer; the latter treats the settlement between SCM and the third party.

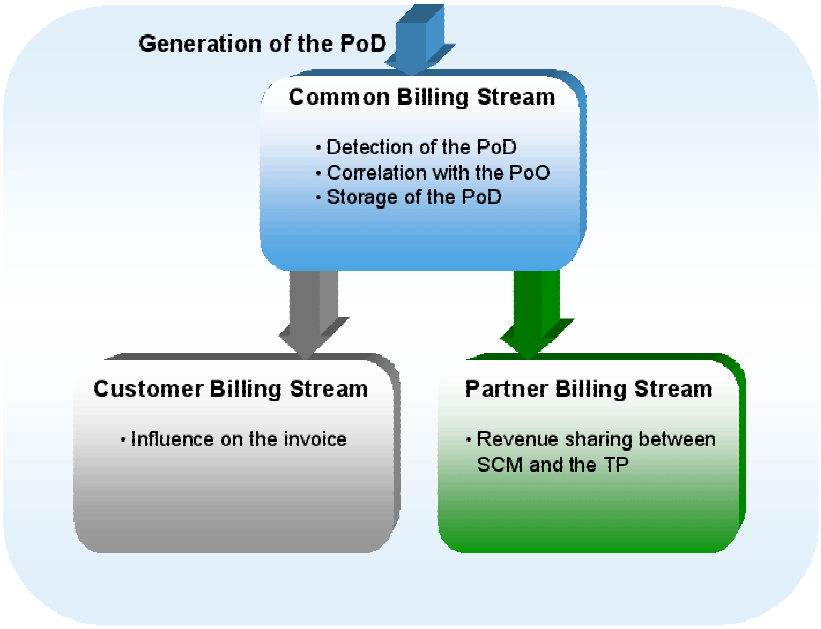

Figure 23: The data flow for the PoD integration in the content billing

7.5.1.1 Common Billing Stream

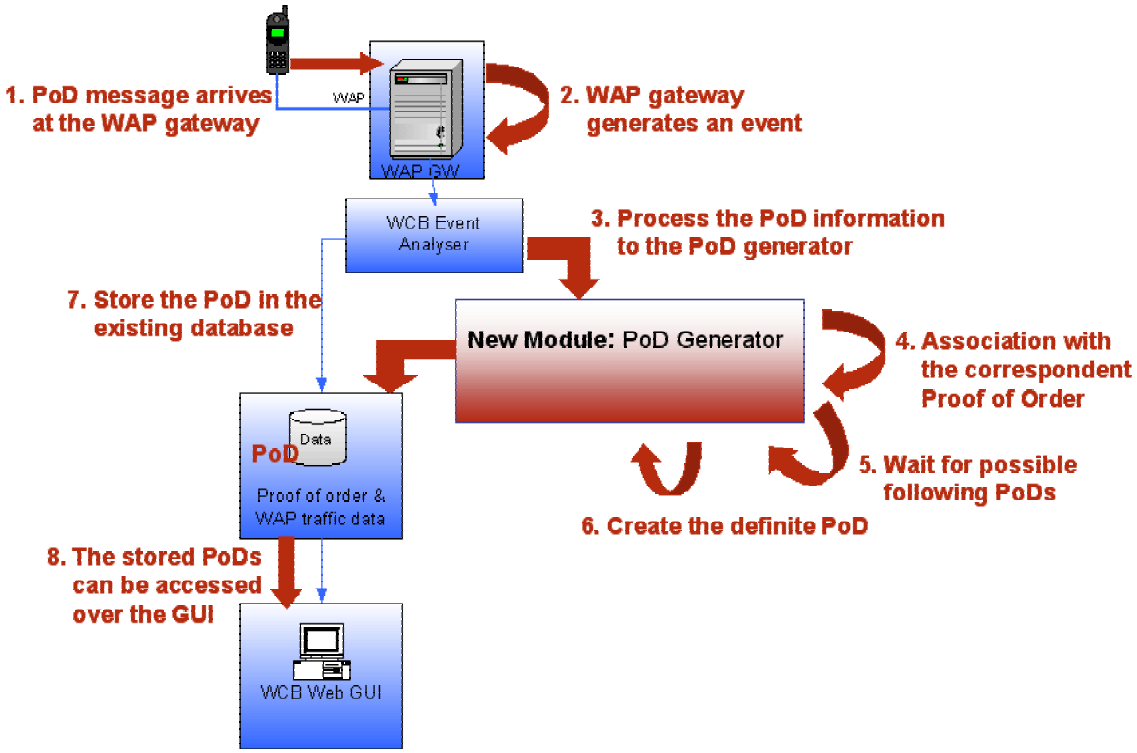

Figure 24: The Common Billing Stream

Figure 24 shows the Common Billing Stream. Only the relevant part of the WCB architecture is visible in order to keep clarity (for details see also chapter 3.3).
The integration of the PoD in the SCM WCB architecture should affect the current architecture only slightly. It is the goal that the PoD can use the current components of the system (possibly adapted or extended). But in this billing stream a new component is needed.

1. The PoD is generated on the mobile device and sent by means of an HTTP GET request to a server of SCM. An HTTP GET request can contain additional information in the URL. In this case the information will be the PoD. A simplified example:

   http://swisscom-mobile.ch/tpb/pod?msisdn=41791234567&tpid=0034&sid=647&timestamp=2003-02-28T13-30&delivery=ok&PoO=5428&final=1

2. The WAP gateway generates an event, when the HTTP request from the handset arrived. One parameter of this event is the URL.
3. The WCB Event Analyser filters the events and processes then the URL to the new module “PoD Generator”. The PoD Generator investigates then the part of the URL after the “?” and forms this information to the PoD structure with the fields as described above. The PoD Generator has also to be able to continue the process, if no PoD arrives or if the PoD contains errors. No customer billing stream (re-rating) is then executed. The erroneous PoD is just stored in the database (with the delivery status set to “PoD undefined” by the PoD Generator.
4. In order to make the association between the PoD and the PoO, both have to be correlated over an identifier. Probably the bill ID can be used for this purpose (this ID would then have to be available on the mobile phone, when the PoD is generated).
5. One PoO can be associated with several PoDs (if for instance during a stream periodically a PoD is generated on the mobile phone and sent to SCM). The PoD Generator waits for the arrival of possible following PoDs. The arriving PoD contains a flag “MoreFollow”. This has the value “1”, if there are more PoDs left. The last PoD of the sequence has then set the “Final” flag to the value “1”. The PoD Generator merges then the information of the different PoDs to one final PoD.
6. The PoD generator produces the final PoD, which is stored in the database.
7. Storage of the PoD in the database for a certain period. The database has possibly to be extended in size to be able to store the PoD data. The PoO is stored for 180 days. Probably

the PoD is stored for a longer period, due to his influence on the billing stream. New relations have to be introduced in the database in order to store the PoD structure.

8. The customer care is able to access and to query the stored PoDs over the WCB Web GUI.

7.5.1.2 Customer Billing Stream

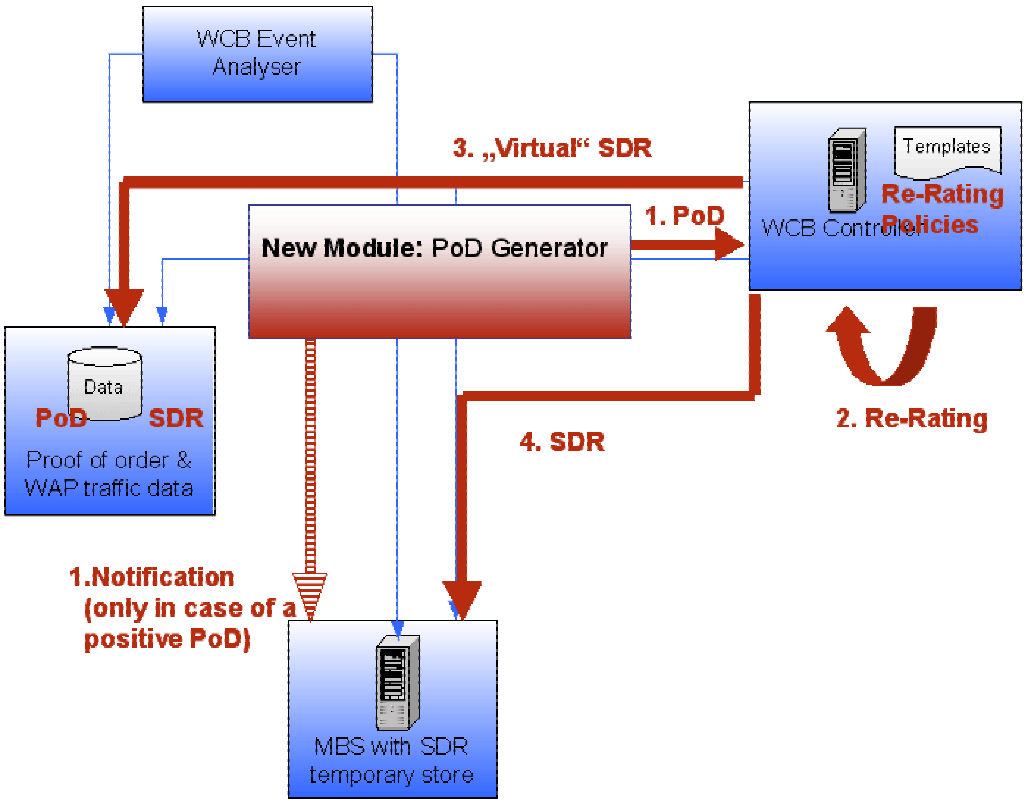

Figure 25: The Customer Billing Stream

1. If the final PoD isn't positive, it is transmitted to the WCB Controller for the re-rating. If it is positive, the MBS is notified that the temporary stored SDR is valid and can be transmitted to the billing system. This procedure corresponds to the already implemented flow of events in the current architecture.
2. The re-rating policies are applied in the WCB controller. The re-rating policies apply rules as they are mentioned in chapter 7.3. The re-rating takes place in the WCB Controller because there is an interface to the third parties. This allows the exchange of information concerning the re-rating policies. A new SDR is generated. It contains besides the new billing amount and the difference to the original one also information about the delivery (ok, not ok, partially ok), about the failure reason (end device, network, server) and about the responsibility (end customer, SCM, TP). This information is afterwards needed for the settlement process between SCM and the TP. In addition, the SDR contains also an identifier of the discounting rule, which has been applied during the re-rating.
   Besides the identifier of the rule, the SDR has also to contain a short text, which will finally appear on the invoice. As already mentioned, the discounts have to be clearly communicated to the customers in order that the invoices are transparent and plausible. This text mustn't be too long. It could for instance have the following form:

   "Network Failure [date/time]: Discount 50%"

   A copy of the generated SDR is then transmitted to the database. It is called "virtual" SDR, because it isn't really used for the billing process, but only stored for statistical reasons in the database. The SDR has to contain the identifier of the PoD in order that the association with the PoD can be made in the database.
3. The "virtual" SDR is stored in the database. The purpose is that SCM is aware about the revenue loss due the re-rating in case of failed service deliveries.

4. The SDR is transmitted to the MBS, where the original SDR is stored. The latter has to be replaced and then transmitted to the billing systems.

The proposal for the flow of events designs the creation of a new SDR. On the other hand it would also be thinkable that not a new SDR is generated, but a “negative” SDR is sent to the billing system, where a correction of the already charged amount can be made. This should also work, but is perhaps a bit more complicated.

7.5.1.3 Partner Billing Stream

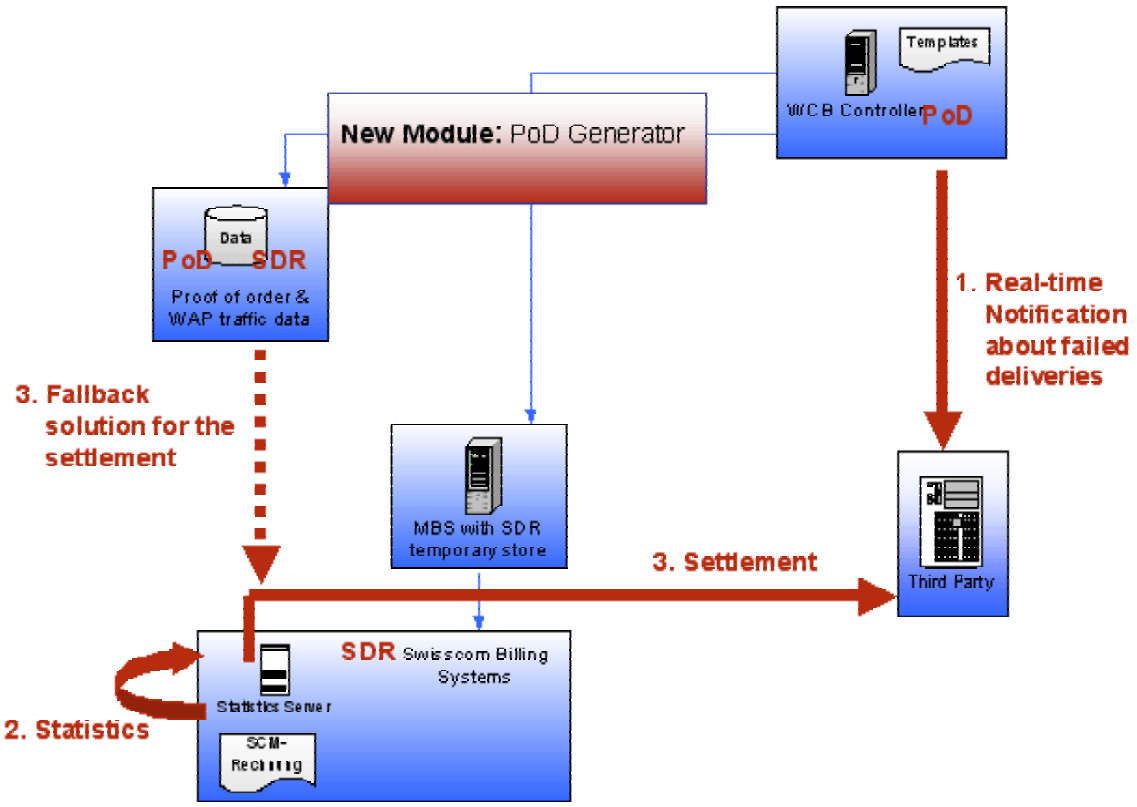

Figure 26: The Partner Billing Stream

1. The TP may desire to get the information about the delivery of the service in real-time and not only during the settlement process. The WCB Controller generates this real-time notification for the TP. The notification contains the PoD fields. If the TP doesn’t get such a message, it can assume that the service has been delivered successfully.
2. The settlement process between SCM and the TP is done manually in the today’s solution based on statistics, which are produced by the Statistics Server.
   With the integration of the PoD, the Statistics Server has to generate further statistics. If the delivery has been successful, there is no change to the data flow. But if a re-rating has taken place, an SDR with additional information arrives at the Statistics Server. This has to create statistics per third party, per service, per failure reason and per responsibility.
3. The revenue sharing between SCM and the TP is executed based on the statistics. An agreement regulates this process. If the statistics aren’t available, the stored data from the database can be used as a fallback solution.

7.5.2 Recommendations

- An appropriate implementation variant has to be found for the association between the PoD and the PoO. The bill ID is probably a solution, but it would have to be available, when the PoD is generated on the mobile device.
- A validation of the proposals concerning the architecture has to be done. It has to be checked, whether the respective parts of the architecture are able to execute the tasks.
- Check, whether all proposed fields of the PoD are necessary and whether they really are available, when the PoD is generated on the handset (technical feasibility).

- Re-rating policies: In which cases a re-rating takes place? What are the influences on the bills (discount in percent etc.)? What is the opinion of the third parties? Some examples of discounting rules are mentioned in chapter 7.3.
- Frauds: It is important that it is for instance impossible to feed or even to spam the system with faked PoD messages.
- A business case could be conducted in order to check, if it would make sense to replace the manual settlement process by an automatic settlement engine.

8 Conclusions and Outlook

This thesis contains some general considerations about an implementation of a Proof of Delivery and about its integration in the billing chain. Because the elaboration has done only on a conceptual basis, some open questions are resting.
The considerations about the PoD in the network (technical implementation) are kept very general. Also the integration in the billing process is investigated on a general level. The WAP content billing architecture of Swisscom Mobile is only considered as an use case. The integration of the PoD also in other delivery platforms would have to be investigated.

8.1 Implementation

The thesis contains a pure conception and no detailed description of an implementation. The best implementation variant seems to be the one with a software agent on the mobile device. At least it seems to have the most potential. From the technical point of view it is interesting and challenging. It would be a nice task to build a prototype and to run tests. The time and also the resources during this thesis were to restrict to implement such a prototype.

8.1.1 Interaction between Agent and Mobile Device

The agent is responsible for collecting the information about the state of the delivery and about the quality of the delivered content and then to send this information in a message to the network operator. To be able to do this, the agent needs to have access to the relevant components of the device. The content is sent to the device and not to the agent. The devices of the different manufacturers have specific implementations. It is necessary that the agent is able to interact with every device. Maybe different versions of agents are needed, so that each kind of device gets its particular agent.
Modern handsets contain a KVM. With MIDP 2.0 the compatibility will even be increased. Therefore an environment for agents on mobile devices seems soon to be broadly available. Of course, agents can also be used for totally different purposes (e.g. to find location based information for the customer).

8.1.2 Reliability

The main task in this context would be to check the reliability of such an agent. Because the PoD has an influence on the billing process, it is very important that the PoD message arrives error-free and with the right content at the systems of the network operator.
But it has also to be assured that the billing process works properly in a situation, when no PoD arrives at the system. It is thinkable that there is some kind of timeout, within the PoD has to arrive to be treated in the billing process.
Beside the reliable transmission of the PoD message it is also important that the content of the message conforms to the actualities. Probably it is reasonable not to pack too much things in the PoD in a first phase. Extensive tests are required anyway. It is imaginable to use the PoD in a first phase only for download services and then after a longer period with satisfying results to build in also the investigations about the QoS and to implement the PoD also for streaming services.

8.1.3 Usability Studies

A challenge is to prove the delivery of streaming services. A download can easily be proofed by checking if the last bit has arrived (over a reliable connection such as TCP). The decision in this context is binary: delivered or not delivered. The user has only a benefit of the service if it has been delivered completely. The client can also consume a stream if it has some interruptions. Of course this isn't really desirable, but if the interruptions or the periods with a reduced QoS aren't too long, the customer can be satisfied anyway. But this level of satisfaction is something very subjective. Each client has its individual perception. Therefore it would make sense to conduct large usability studies to determine the grade of satisfaction depending on the quality of streaming services.
The participants at these studies would have to give a feedback (e.g. insufficient, sufficient, good, very good) about their impression of the quality of the stream. This impression can depend a lot on the price of the stream and also on the kind of the stream (entertainment, news, live stream etc.). Anyway, the goal should be to define thresholds out of the gathered user feedbacks. These thresholds are then the basis for the agent on the device to decide whether the stream has been delivered completely, only partially or not.

A more sophisticated method to decide about the delivery of a stream would be that the expectations of the customer are defined in a Service Level Agreement (SLA). An SLA is a contract between a provider and a client (or between providers) about the QoS (e.g. the availability, the bandwidth, the maximal delay etc.) and sometimes also about other parameters, such as the reaction time of the customer care. This SLA is then the basis for the configuration of the agent on the device of the customer. The agent decides by virtue of the parameters in the SLA if the service is delivered or not. If the agent can measure for instance jitter, bandwidth and the missing frames of the stream, the measured values can be compared with the ones agreed in the SLA and the agent can decide about the delivery. Of course the client would have to pay more for higher expectations to the quality of service. On the other hand he gets also earlier a discount, if the quality of the delivered service doesn't conform to the agreed values.

8.2 PoD in the Billing stream

The PoD allows the network operator offering discounts to his customers. Since the network operator is a normal enterprise, which is interested in revenues as high as possible, discounts should only have to be applied in exceptional cases.
This thesis tries to show the possibilities, which are the results of an introduction of a PoD. If a network operator decides to use such methods and is willing to offer discounts to his customers, he must be ready to offer a reliable service delivery. It is obvious that the subscribers will be eager to get as much discounts as possible and that there will be a lot of claims at the customer care (to prevent this, the rates for calling the customer care could be increased in parallel with the introduction of the PoD…)
If the PoD is used proactively (automatic discounts) it has to be assured by technical measures that the delivery really only fails in exceptional cases.

A client may be attracted to use third party content services, if he is aware that he only has to pay if he gets the service successfully delivered. He feels secured and the motivation is higher to use content services. This could result in a general higher use of such services, which increases on the other hand the revenue of the network operator and should not only compensate, but also clearly exceed the losses due to discounts!
At first sight, this coherence seems to be obvious. But probably carefully conducted business studies will be needed, before a PoD can be introduced.

From the customer's point of view, it seems to be justifiable that he only has to pay for something, which he really also has received. With the implementation of streaming services, which need a minimal bandwidth, this becomes even more important. For the customer it is absolutely unsatisfying, if he has to pay for a jerking live stream with a lot of interruptions.

If UMTS ought to be a success, the customers have to be adapted early to the new service opportunities. Therefore it has to be assured that these services are delivered in a satisfying quality and according to the needs of the customers.
Billing can be used to attract customer's attention to services, but shouldn't be abused in order to hide insufficient quality.

References

[Acterna] Acterna GmBH: GPRS – Das mobile Internet. Grundlagen und Protokolle. 2002

[Aran] Aran technologies: Intelligent Service Assurance in 3G. It's a key differentiator! 2001

[Bill0502] Billing World and OSS Today Magazine: Putting QoS and Customer Service Into the Handset. 2002.

[EIR] Ecole d'ingénieures et d'architectes de Fribourg. InfraWAP & MITech Project: Agent technology applied to the mobile telecommunication. 2002

[FIPA] The Foundation for Intelligent Physical Agents (FIPA). www.fipa.org

[FIPASmall] Adorni, Bergenti, Poggi, Rimassa, G: Enabling FIPA Agents on Small Devices. 2001

[GSMWorld] GSM World: The Website of the GSM Association *www.gsmworld.com*

[Grasshopper] IKV Technologies AG. Grasshopper, the Agent Platform www.grasshopper.de

[IP-value] IP value Technologies. www.ip-value.de

[Javaworld] Javaworld: An introduction to agents. www.javaworld.com

[Knudsen] Knudsen, Jonathan: What's new in MIDP 2.0. 2002

[Laukkanen] Laukkanen: Agents on Mobile Devices. Sonera. 2002

[Lauterjung] Lauterjung: Quality-of-Service for multimedia content over UMTS. 2001

[Loetscher] Loetscher: Business-Models for wireless networks. Semester Thesis, ETH Zurich. 2002

[LucentGPRS] Lucent: The Flexent Gateway GPRS Support Node for UMTS. 2001

[MIDPOTA] Sun Microsystems: Over The Air User Initiated Provisioning. Recommended Practise for the Mobile Information Device Profile. Version 1.0. 2001

[MIDPSpec] Sun Microsystems: Mobile Information Device Profile (JSR-37). JCP Specification. Java 2 Platform, Micro Edition, 1.0a. 2000

[MIND] Ruiz, Mitjana, Burness: Advanced services over future wireless and mobile networks in the framework of the MIND project. 2001

[Muchow] Muchow: Core J2ME Technology and MIDP. Sun Microsystems Press and Prentice Hall. 2002

[NokiaBusiness] Nokia: Open for business. Building the platform for success today. 2002

[NokiaCOD] Nokia: Delivering hot content with confirmation. OTA download for generic content with COD technology. 2002

[NokiaOTA] Nokia: OTA download for generic content. Specification. Version 1.1. 2002

[NokiaE2E] Nokia: Nokia's vision of providing end-to-end Quality of Service in 3G. 2001

[NokiaWAPGPRS] Nokia: Introduction to WAP over GPRS. Forum Nokia. 2001.

[NortelQoS] Nortel Networks: Introduction to Quality of Service (QoS). 2001

[Nortel3GqoS] Nortel Networks: Benefits of Quality of Service (QoS) in 3G wireless Internet. 2001

[OMAOTA] Open Mobile Alliance: Generic Content download Over The Air Specification. Version 1.0. 2002

[QoSMobileIP] Zhang: QoS Issues in Mobile IP: Challenges, Requirements and Solutions. 2001

[ReinBona] Reinbold, Bonaventure: A Comparison of IP mobility protocols. 2001

[RenCasFan] Rendon, Casadevall, Faner: Wireless TCP Options Behaviour in the GPRS Network. Polytechnic University of Catalonia, Barcelona.

[Ringenbach] Ringenbach: IP-Billing and correlated business models. Diploma thesis Fachhochschule beider Basel, Departement Wirtschaft and Swisscom. 2001

[RTSPRFC] Real Time Streaming Protocol (RTSP). IETF RFC 2326

[Siemens3G] Siemens: 3G Wireless Standards for Cellular Mobile Services. The Siemens view. 2002

[SiemensCT] Berger, Bauer, Watzke: A Scalable Agent Infrastructure. Intelligent Autonomous Systems Group, Siemens AG, Corporate Technology. 2001

[SiemensGPRS] Siemens: The full potential of GPRS. New applications and business opportunities. 2001

[Sonera] Sonera: Nokia 7650 optimized streaming media content demos. www.medialab.sonera.fi/demos/mobile/nokia7650.html

[SOQUET] SOQUET: System for Quality of Service for 3G Networks. European Research Project. 2001-2003
www.soquet.leeds.ac.uk

[TagAnz] Tages Anzeiger: Gefragt ist viel Geduld. 03.02.2003

[Transcomm] Transcomm: The emperor has no clothes! Comparing utopian dreams with enterprise reality in the world of wireless data. 2001

[TUBerlin] Technical University Berlin, Telecommunication Networks Group: Current developments and trends in handover design for ALL-IP wireless networks. TKN Report TKN-00-007. 2000

[Uni-x] uni-X Software AG: PocketGuide Billing

[UMTSLink] UMTSlink.at: Mobilfunktechnik transparent.
www.umtslink.at

[Valtakari] Valtakari, K: Billing systems for Internet service providers. Master's thesis. Technische Universität München, Computer Science Departement. 2001

[WAPForum] Wap Forum
www.wapforum.org

[Watrin] Watrin: Leveraging QoS for Content Based Billing. Swisscom. 2001

[Whatis] Whatis?com. Part of the TechTarget Network of Enterprise IT Web Sites.
www.whatis.com

[W&BGPRS] Weckwerth & Bertham, Datenkommunikation und mobile Systeme GMbH: GPRS, Mythen und Fakten.

[X-media] X-media: UMTS Start 2002. Quo vadis Mobilfunk? 2002

[Zona] Zona Market Bulletin: The Need for Speed II. 2001