Róbert Párhonyi

MICRO PAYMENT GATEWAYS

Graduation committee:

Chairman, secretary:	prof.dr.ir. A.J. Mouthaan (University of Twente)
Promoter:	prof.dr.ir. L.J.M. Nieuwenhuis (University of Twente)
Assistant promoter:	dr.ir. A. Pras (University of Twente)
Members:	prof.dr. B. Stiller (University of Zurich and ETH Zurich) prof.dr. R.W. Wagenaar (Technische Universiteit Delft) prof.dr.ir. P.W.P.J. Grefen (Technische Universiteit Eindhoven) prof.dr. G.B. Huitema (Rijksuniversiteit Groningen) prof.dr.ir. B.R.H. Haverkort (University of Twente) dr.ir. M.J. van Sinderen (University of Twente)

The research reported in this thesis is part of the research project "Management of DISH-applications", funded by NWO, the Netherlands Organization for Scientific Research, under grant no. 612.060.002.

Part of this work was sponsored by the Telematica Instituut, the Netherlands, via the "Internet Next Generation" project.

CTIT Ph.D-thesis series, no. 05-72
CTIT, P.O. Box 217, 7500 AE, Enschede, The Netherlands
ISBN 90-365-2239-0
ISSN 1381-3617, no. 05-72

MICRO PAYMENT GATEWAYS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. W.H.M. Zijm,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 20 oktober 2005 om 16.45 uur

door

Róbert Párhonyi

geboren op 28 november 1975
te Cluj-Napoca (Kolozsvár)

Dit proefschrift is goedgekeurd door:

promotor	prof.dr.ir. L.J.M. Nieuwenhuis
assistent-promotor	dr.ir. A. Pras

To my wife, parents and Arthur

Abstract

In the next years, the market for low value online content, like music and videos, is expected to grow substantially. To allow “pay-per-use” of such content, micropayment systems are expected to play an important role. Since there are already many competing micropayment systems on the market, customers and merchants are forced to use multiple systems. To overcome the problems associated with using multiple systems (e.g., learn the usage of several systems, manage multiple accounts and e-wallets, remember multiple passwords, trust different micropayment system operators), in this thesis, we propose a *hybrid payment system* that allows customers and merchants to use their micropayment system of choice, while still being able to pay each other in a seamless manner regardless the choice of the other party. The core component of our system is the *Payment Gateway*, which is responsible for interconnecting the existing (and future) micropayment systems. To become successful, such a system needs to have global acceptance and penetration, a high micropayment volume, high trust level and secure money transfer. The main objective of this thesis is to develop an architecture of the hybrid payment system.

To solve the micropayment system interconnection problem, we propose a generic and systematic interconnection method for existing micropayment systems. This method is to harmonize the payment services of existing systems to a uniform level, called the *uniform payment service*, and interconnect these uniform payment services. We call a system that provides the uniform payment service a *uniform payment system*. Using this method, the number of mapping rules and the amount of information that must be stored will remain limited, which makes this method scalable and the design and realization of the *Payment Gateways* will become much easier. A prerequisite for this method is that the harmonization of existing (and future) micropayment systems to the uniform payment service is possible. We will define the uniform payment service such that the vast majority of existing payment systems can comply with this service without changing their functionality. We will also

prove this fact by presenting two case studies. The compliance of current payment systems with the uniform payment service also means that they can be interconnected without changing their functionality.

The uniform payment service could guide the design of future electronic payment systems such that new systems can be interconnected easily with existing systems. In this way, the uniform payment service, possibly extended with interactions that have only local significance, could become a de facto standard for micropayment systems.

On top of the uniform payment systems, we design a *hybrid payment (or interconnection) protocol*. This protocol bridges the gap between the hybrid and uniform payment services. This protocol will be designed such that (1) the threats for the normal operation and security of the protocol are not considerably bigger than that of the existing systems, (2) hardly any money loss situations will occur, (3) commonly used security techniques can be employed to secure the interactions between the various components of the hybrid payment systems, and (4) that the protocol will be optimized in case no interconnection is needed.

This thesis begins with presenting the research context, problem definition, possible solutions, objective, related research questions and approach followed (Chapter 1). We start our research with studying the payment function within the context of (product) accounting (Chapter 2). We then analyse the structure and functionality of existing electronic payment systems, identify the business roles within these systems, define their main functional characteristics and present an overview of these systems (Chapter 3). Afterwards, the requirements for the hybrid payment system will be derived from the viewpoints of end-users (customer and merchant), stakeholders (operators of micropayment systems and Payment Gateways), legal and regulatory frameworks (Chapter 4).

The main functional characteristics of existing payment systems and the requirements will guide the design of the hybrid payment system. The design will be structured in three phases: (1) formulate the functional requirements for the hybrid system, (2) design the hybrid payment service, (3) discuss the most suitable interconnection method and design the interconnection protocol (Chapter 5 and Chapter 6). The uniform payment systems and interconnection

protocol will be demonstrated to be implementable, which implies that an implementation of the hybrid payment system is achievable. Besides this demonstration, the design of the hybrid payment system will be evaluated to verify whether the hard requirements from Chapter 4 are satisfied (Chapter 7). Finally, the conclusions of our work will be drawn and some research topics for future work will be formulated (Chapter 8).

Acknowledgements

There are a number of people who have contributed to this work in one way or another. Below I present my words of gratitude and appreciation.

First, this work would have not been completed without my promoter Bart Nieuwenhuis. Although, he got involved in a later phase of my research, his support, guidance, enthusiasm, encouragement and motivation have all been of great value. I appreciate his faith in me and his interest in my work.

My assistant-promoter Aiko Pras was involved in my work from the first moment. I would like to thank him his ever-lasting patience, support and guidance to carry on my work in good and difficult times. He was there for me not only as a professional supervisor, but also as an understanding fellow colleague. Our countless and long discussions, his feedback and advice regarding the structure and contents of this thesis were indispensable.

A special word of thanks is due to Dick Quartel, whose contribution had a significant impact on the quality of key chapters and the papers we wrote together. His suggestions and constructive comments eased the design process and made the reasoning throughout the thesis more precise and stronger.

I thank Sander Hille and Henk Jonker of the Telematics Institute for their contribution to this thesis with their expertise in accounting and electronic payment systems.

Many thanks to present and past colleagues at the Network Management, Architecture and ASNA groups for a pleasant and productive environment: Szabi, Ron Sprenkels, Bert, Remco van de Meent, Marten, Luis, Remco Dijkman, Nikolay, Giancarlo, Patricia, Joao-Paulo, Renata, Christian, Aart, Maarten, Tom, Val, Ing, Richard, Hailiang, Boris, Jasper (aka. Hao), Ricardo, Prawin, Kamran and Muhammad. I thank Annelies, Marlous, Helen and Wilma for taking care of all the administrative work.

Furthermore, I thank all my friends who made pleasant my stay in the Netherlands and helped me.

Last but not least, I would like to thank my family. Tímea, my loving partner in life was always there for me, shared all good and difficult moments, and helped me to overcome many difficulties. My parents and brother motivated and supported me from far away throughout the years. *Köszönöm!*

Róbert Párhonyi
Enschede, August 2005

Table of Contents

Chapter 1

Introduction

This chapter introduces the background of the research presented in this thesis (Section 1.1), describes the problem statement (Section 1.2), enumerates alternative solutions that could solve the problem (Section 1.3), defines the objective and research problems (Section 1.4), presents the approach that is followed in this thesis and the structure of this thesis (Section 1.5).

1.1 Background

1.1.1 Research context

This research is performed in a period marked by an astonishing growth of the Internet. One aspect of this growth is the number of Internet users. Indicators of the International Telecommunication Union (ITU, [1]) show that more than 687 million people were using the Internet in 2003 [2]. The same indicators show a moderate growth in the number of personal computers connected to the Internet. In 2003, this number passed 593 million.

The Internet is used for four major activities: content, communications, commerce and search. To measure the activities of Internet users, an Internet Activity Index [3] was launched in July 2004 by the Online Publishers Association, which comprises content providers such as Wall Street Journal, Forbes.com, New Your Times Digital, ESPN.com [19], and Nielsen//NetRatings, an important Internet research firm [4]. This index shows that in 2003 content was the primary reason for people to access the Internet. In other words, 87% of Internet users accessed web sites and Internet applications that

provide news, information and entertainment. What concerns the total time users spent online, content is second after communications.

The quantity of online content cannot be precisely evaluated. On one hand, the latest (January 2005) Internet Domain Survey [5] of the Internet Software Consortium [6] shows that the number of hosts that store online content exceeded 317 million, which is a 36% increase compared to the year before. On the other hand, studies performed at the University of California at Berkley [7] in 2003 tried to estimate how much information is created each year. Their research on the quantity of information or content published on the Internet focuses on two matters. First, the amount of content that consists of static and publicly available web pages ("surface of the web") amounts to 167 terabytes as of summer 2003. Second, the amount of content that consists of web-accessible databases and dynamic web sites ("deep web"), is hundreds of times bigger than the amount of content on the surface of the web, and it is estimated to be between 66 and 91 thousand terabytes [8].

From the beginning, the Internet users considered that content should be for free [9]. That is why nowadays many content providers publish content for free. In our opinion this will change in the future and increasingly more paid content will be published. The content providers realize that customers should pay for valuable content, therefore, an increasingly number of content providers start charging for content. As a consequence, the paid content will become unique, exclusive, and will have an increased quality.

At the moment of writing, paid content is being sold according to different schemes such as subscriptions or pay-per-use. The price of content also varies from a few cents for a music file up to thousands of US Dollars or Euros for a market research report. Online content that is sold for small amounts of money on pay-per-use bases is considered further in this thesis.

Market research companies expect that online content will become an important source of revenues for content providers in the future. These expectations motivate content providers to further enlarge their online content offerings. According to a graph of Gartner [10] published in 2003, not only the quantity of chargeable content is growing, but also the diversity [11]. Multimedia content (e.g., music and movies), online games, and entertainment information

will become the major content types that attract the attention of broadband users [11], [12]. Forrester Research [13] reported that music downloads generated €24 million in Europe in 2003, which will grow to €1,3 billion in 2007 [14]. These revenues will mostly add up from payments for individual downloads rather than subscriptions. In the US, revenues from music download reached US$15 million in 2002. In 2007 the US market of music downloads is expected to grow to US$2 billion. Content providers like Apple iTunes [15], MusicNet [16], BuyMusic [17], or On Demand Distribution (OD2, [18]) have an overwhelming success in selling digital music at low prices (e.g., US$0,79-US$1,14). Music files offered by these providers have their own Digital Rights Management license, and can be individually downloaded and paid.

Low value content revenues are steadily increasing. For instance, the Online Publishers Association reports that among the individual content payments, the share of micropayments increased from 7,4% in 2003 to 17,9% in 2004 [20], [21]. Almost US$50 million was paid with micropayment systems in 2004. Additionally, the share of content subscriptions dropped from 89% in 2003 to 84,6% in 2004.

The acceptance of low value content differs from country to country. Market research performed in Australia and USA in 2002 shows that 57% of Australians do not understand why they should pay for content, and 36% of Americans would stop using content if they have to pay for it [9]. In contrast to these conclusions, Germans are more likely to pay for content. The Association of German Newspaper Publishers (VDZ, [22]) and Sapient, a business consulting and technology advisory firm [23], conducted in Germany a study that questioned 11,238 customers and 15 content providers [24]. This study, published in January 2003, considered both chargeable and free content (e.g., news archives, auto test reports, premium adult entertainment, TV guides). According to this study, about 50% of the interviewed customers would pay for content, especially for database searches, software downloads, archived information, economics and financial content, online banking and brokerage, consumer test reports, etc. Furthermore, the majority of both content providers (52,6%) and customers (51,5%) would prefer individual pay-per-use payments for the consumed content. An interesting finding is that 50% of the web sites accepted micropayments, 18% credit card payments, 16% charged the costs to the customers' Internet Service Provider (ISP) bills and another 16% sent

invoices. Customers favour the micropayment systems, invoice and credit card payments are next.

The VDZ, Sapient, and the Ludwig-Maximilians University of Munich [25] performed in Germany another study [26]. This study provides the prices expected by customers for different content (Table 1.1). Just as the previous study, this is also based on surveys of content providers and customers.

Table 1.1 *Price expectations for online content and services*

	Pay-per-use price	Monthly subscription price
Journalistic content (e.g., archived news, current news, sports, celebrity news, reports, health tips, computer tips, etc.)	€0,29	€1,11
Other content (e.g., photo galleries, e-books, downloads, games, learning for children, etc.)	€0,61	€1,58
Online services (e.g., phone calls, web mail, web search, online lottery service, etc.)	€0,10	€0,52

From these observations can be concluded that content providers expect customers to pay for consuming non-free online content, and the majority of them is willing to pay for it. To support the selling of chargeable content, accounting systems are needed. Such systems calculate the costs of the consumed content, send bills and collect the money from customers.

1.1.2 Accounting

In this thesis, the goal of accounting is to charge customers for their consumption of non-free resources, and to collect the money from them. These resources can, for instance, be content resources, online services or network resources. We note that, these are intangible resources. Besides them, accounting can also be performed for (tangible) products. Such products can, for instance, be public transportation tickets, parking, e-tickets.

Originally, the Open Systems Interconnection (OSI) Management Framework defined accounting as the process of metering, pricing, charging and billing of customers for their service usage [27]. Within the Internet community, the term "*accounting*" is nowadays used to denote only the more restricted metering function [28].

In the context of this thesis, however, the term "*accounting*" is used in its original and broader sense. In this view, accounting integrates the functions of (i) metering resource consumption, (ii) collecting and storing of metering data, (iii) pricing the consumed (unit of) resource, (iv) charging based on collected data and pricing information, (v) billing the resource consumers based on charging information; and performing the payment that clears the bill. Payment is a sub-function of billing.

Depending on the resources being consumed, accounting can be divided into:

- transport accounting and
- product accounting [33].

In case of transport accounting, customers have to pay for using non-free network resources, which transport packets across networks. *Network resources include processors and memory in routers, communication hardware (e.g., modem), links capacity and buffers, etc.* [29]. The measurable quantities of these resources can, for instance, be the amount of data (e.g., number of packets or bytes) that a network delivered to a customer, the used bandwidth, the quality of service class, or the time period while customers were online. Payments in transport accounting are usually made periodically (before or after consuming the resources) and to one organization (ISP). Transport accounting receives significant attention from the networking equipment manufacturers, network operators, standardization bodies and research communities. Current work focuses on defining standardized architectures, protocols, data formats, accounting policies, etc.

In case of product accounting, customers have to pay for the consumption of non-free resources other than network resources. We denote this set of resources by the generic and collective term *products*. We note that, content is the most representative product example, and will be used later in examples.

Content is defined as all means of information that includes text, formatted text, interactive and/or dynamic web pages, images, animation, online games, video and sound files, web-based output of application [30] *that is complemented with metadata.* Metadata is either data describing the information (e.g., title, property rights, keywords), or data that can be derived from the analysis of the information (e.g., index) [32]. The measurable characteristics of these products can, for instance, be the be number of files downloaded, the number of links visited, the number of news articles read, or the time period a movie is watched. Payments in product accounting can be done periodically (e.g., subscriptions) or every time a unit of product is consumed, and depending on the value chain, one or multiple organizations receive these payments. Because the expectations showed that a need for product accounting emerges, the work performed in this area and the results are limited (e.g., lack of product accounting and payment standards), the research of this thesis focuses on product accounting.

Figure 1.1 depicts four common characteristics of the resources: value, risk (i.e., the probability that a customer pays for some product, which will not be available or delivered), delivery and payment. The shaded areas show the characteristics of those products that are in the focus of this thesis.

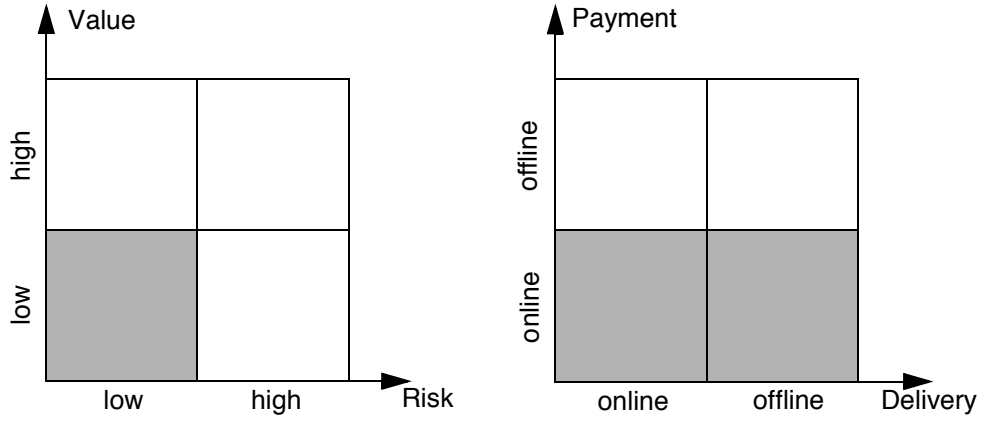


Figure 1.1 *Common characteristics of resources*

1.1.3 Product accounting

This section describes the context of product accounting.

Examples of low value, low risk products that can be sold on the Internet and can be delivered online or offline, and that require product accounting are:

- downloadable music (e.g., mp3 files);
- access to live streaming events (e.g., football matches);
- video on demand (e.g., watching streaming movies);
- adult entertainment (e.g., pictures and videos);
- journalistic content (e.g., local/national/world news, sports, technology);
- entertainment information (e.g., movie listings, tv listings);
- online games (e.g., SpinOff);
- reference information (e.g., Britannica);
- financial information (e.g., stock quotes);
- voice over IP services (e.g., Internet telephony);
- e-tickets (e.g., movie or public transportation tickets).

Generally, two parties are involved in the selling of products: a customer and a merchant. *A customer is an individual person or an organization equipped with an electronic device (e.g., computer, mobile phone, PDA) connected to the Internet that consumes and pays online for products requested from merchants. A merchant is an individual person or an organization that offers products on the Internet and provides these customers, and is being paid for those products.* Between a customer and merchant a business relationship (i.e., consumer-to-business) exists according to which the customer requests products, the merchant delivers these products, and in return, receives money from the customer. A content provider, for instance, is a merchant that offers and provides content.

Between the product request and delivery, product accounting determines the amount of money to be paid and when the payment should be made (Figure 1.2). The payment may take place before the product is delivered (*A* scenario), but also after the delivery (*B* scenario). In the first case, the product is being paid before delivery and on "per-use" bases. Per-use means that only the product that was requested should be paid. In the latter case, the costs of products are calculated for each individual product request and aggregated over a certain

time period (e.g., one month). The products are delivered after each request, while the aggregated costs will be paid in one payment.

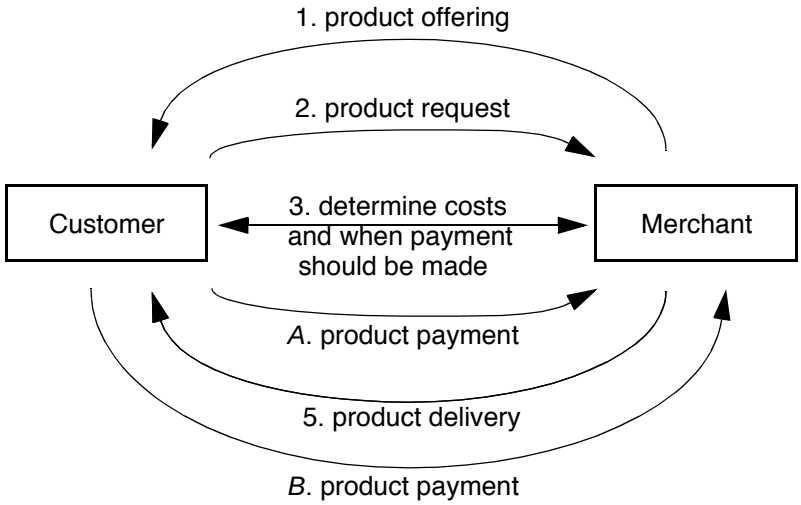


Figure 1.2 *Context of product accounting*

> *Example:* Consider a customer that wants to download some songs of her preferred band. For this, he/she surfs to the web site of an music provider and searches for the songs of that band. We note that, the customer should pay for downloading music files and the prices of the songs are low. After the songs were selected, the customer sends a request to the content provider to deliver the song. The accounting system of the provider calculates the costs for the selected songs, and requests the customer to pay these costs. The customer pays the requested amount of money and then the music files containing the songs can be downloaded.

According to the literature, a distinction is made between

- server based accounting and
- provider based accounting [33], [34].

In case of server based accounting (SBA) merchants carry out the accounting functions themselves using dedicated accounting servers, which send bills to

customers and collect the payments. Such a merchant may also own the servers that handle the product requests and deliveries. Because there is no commonly accepted way to implement the accounting functions, many merchants define and implement the accounting functions in a particular way to fulfil their interests best. This many lead to thousands of product accounting systems. For example, different merchants may apply different definitions of product units, may use different metering strategies, or different formats to store metering data.

In case of provider based accounting (PBA) merchants outsource one or more accounting functions to third parties such as Internet Service Providers (ISPs, network operators, or data transport providers). These parties already have legal agreements with customers, probably perform transport accounting for them, send bills and collect payments from them. ISPs can, for instance, implement all accounting functions. An example for PBA is the accounting practised by NTT Docomo [35], which is an i-mode service and data transport provider in Japan. NTT Docomo performs the metering, data collecting, charging and billing functions on behalf of various i-mode content providers. As a consequence, NTT Docomo sends bills to customers that contain both transport and content charges, collects money from them, and pays the content providers. Other organizations can also provide accounting functions. For instance, Barclays Merchant Services [36] and Streamline [37], which are two major acquirers in the UK. Acquirers are banks or financial institutions that handle the processing of payments on behalf of merchants. These two acquirers provide the payment function for 99,5% of the merchants in the UK [38]. In Ireland 60% of merchants outsourced their payment function. Hence, the payment function is very often outsourced to third parties.

SBA requires additional functionality next to the core business of merchants. This functionality, in turn, needs certain expertise (e.g., for handling payments), which may or may not be available within merchant organizations. We believe that having all expertise needed for product accounting is a requirement that cannot be fulfilled by most merchants. As a consequence, many merchants outsource one or more accounting functions to professional organizations.

1.1.4 Product payment

Product payments can be processed by traditional or electronic payment systems. The traditional payment systems include cash payments, bank transfers, automated withdrawals from bank accounts, paper checks, etc. These payment systems can be used in an electronic marketplace unless low value payments need to be processed. Such payments should be performed the way products are expected to be delivered: online, instantly, very fast and frequently.

Electronic payment systems are favoured for product payments because of the likely electronic nature and delivery of products. Credit card, electronic check (e-check), electronic cash (e-cash), mobile payment and micropayment systems fall in this category. These payment systems handle the transfer of money from customers to merchants using electronic communication channels. Some of these systems became widely accepted systems on the Internet (e.g., credit card systems), others have less success or failed (e.g., micropayment systems [40]). In the operation of electronic payment systems various business organizations are involved, for instance, credit card companies, banks and financial institutions, independent businesses.

Depending on the amount of money to be paid, one or another payment system can be used. Macro payments are those payments that are bigger than US$5 or €5, and micropayments are below this threshold [31]. This threshold, however, varies with the audience. For macro payments credit card, e-check, e-cash or mobile payment systems can be used, while for micropayments mobile payment and micropayment systems can be used.

Ninety-five percent of online payments are performed using credit card systems in 2000. Figure 1.3 depicts the average credit card transaction values for several European countries (in 2002) and the USA (in 2000) [42], [43]. So, credit cards are used to pay larger amounts of money than the price of low value products.

If low value products need to be paid, credit card systems have another drawback: high transaction costs. Table 1.2 gives a few examples of transaction fees charged by credit card processing companies (or acquirers). The columns *Fee*

1 and *Fee 2* contain different transaction fee offers of these companies. In general, acquirers also charge setup, activation and/or monthly fees next to the transaction fees. Information regarding the various fees is provided on the websites of these companies.

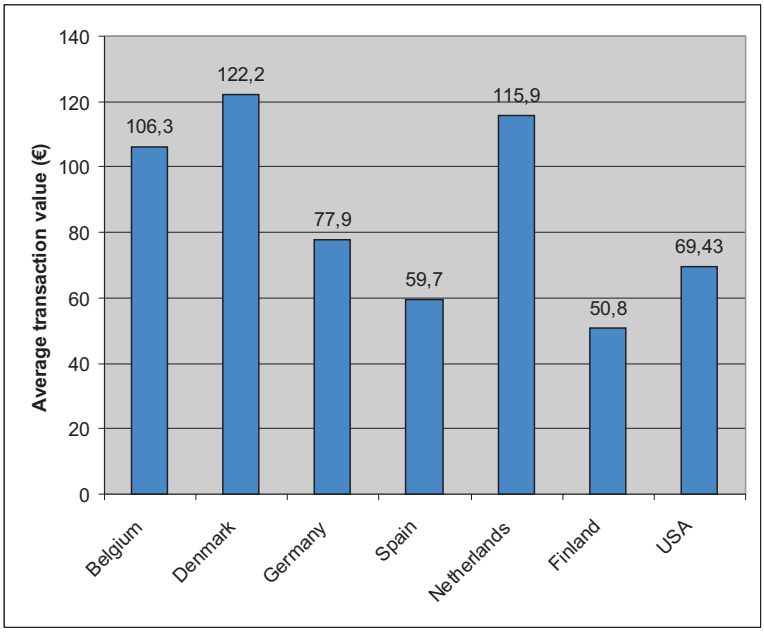


Figure 1.3 *Average credit card transaction values*

Table 1.2 *Examples of credit card transaction fees*

Credit card payment processors	Fee 1	Fee 2
TWYP (ING Bank)	3,5% + €0,43 (Visa)	3,1% + €0,43 (MasterCard)
Ogone B.V.	€0,20-0,89	
PayStone	2,8% + US$0,10 (below US$5)	2,8% + US$0,30 (over US$5)
PayPal	3,9% + €0,35 (receive €)	3,9% + US$0,30 (receive US$)
NetBilling	1,5% + US$0,15	US$0,45
MultiCards	US$0,45	

E-check systems are only popular in those countries in which the paper checks became popular. This means that from a global perspective, their penetration and acceptance is low. The value of e-check payments is similar to the credit card payment values.

Electronic cash systems tried to create the electronic version of cash because this is very popular for retail payments. Although the e-cash systems support low value transactions and allow anonymous and untraceable payments just like cash, they never became successful on a large scale.

Emerging mobile payment systems are very promising because customers could pay any time and anywhere. Such systems support both macro and micropayments, but use other communication channels than the Internet (e.g., GSM).

Many micropayments systems have already been proposed for the Internet. However, until now, no micropayment system managed to reach a dominant position among customers and merchants [44]. Most systems are nowadays used within restricted communities, often within national borders.

1.1.5 Related work

The focus of the research presented in thesis is accounting and payments for low value products. Related work is presented from the perspective of standardization organizations, research projects and commercial products and platforms.

Standardization activities of IETF and W3C for instance, focus mainly on transport accounting, only a limited number of initiatives considered issues related to product accounting (e.g., IPDR, IOTP). In case of electronic payment systems, the standardization work has few results, which did not become widely adopted (e.g., SET, Micropayment Transfer Protocol).

Also most research projects addressed transport accounting (e.g., CATI, SUSIE). Product accounting is addressed in few accounting projects (e.g., GigaABP) and within the context of e-commerce projects (e.g., Opelix, ACTeN). Research on payment systems focuses on issues like what are the

reasons that electronic payment systems failed (e.g., [39]), and what would be the requirements for electronic payment systems that make them successful (e.g., [41], [40]). Researchers developed micropayment systems as well, but many stagnates in a theoretical description phase (e.g., MicroMint, PayWord).

Commercial products and platforms were developed for the complete accounting process (e.g., XACCTusage), but also for one or the other accounting functions. Products and platforms for metering (e.g., NetFlow), billing and payment (e.g., iBill) or only for payments (e.g., Bitpass, Minitix) are available on the market.

1.2 Problem statement

Based on previously presented facts and statements we believe that there is a need for micropayment systems that support low value money transfers between customers and merchants.

Commercial organizations and researchers proposed many micropayment systems for the Internet. However, until now, no micropayment system managed to reach wide acceptance among customers and merchants [44]. Most systems that are still operational are currently being used in restricted communities, mostly within national borders, and do not support cross-border payments. Nevertheless, in the light of globalization, the demand for cross-border payments is growing [44]. Future payment systems should therefore be operated across country borders and preferably be accessible around the world. Another reason for having cross-border payments is the expectation that the cross-border sale of low value content is most likely to grow [45].

Current practice on the Internet shows that different merchants use different payment systems. This practice is driven by the fact that merchants offer the possibility to pay using several payment systems in the hope that an increased number of customers will buy their products. A study performed in Germany, for instance, revealed that 21% of all content providers accepts payments via three payment systems, 17% via two systems, and 55% via a single system [22].

As a consequence of this practice, customers should also use different systems to buy products from different merchants. Customers should be prepared to pay, possibly concurrently, with any required payment system. Figure 1.4 illustrates two customers paying two merchants, which use two different payment systems.

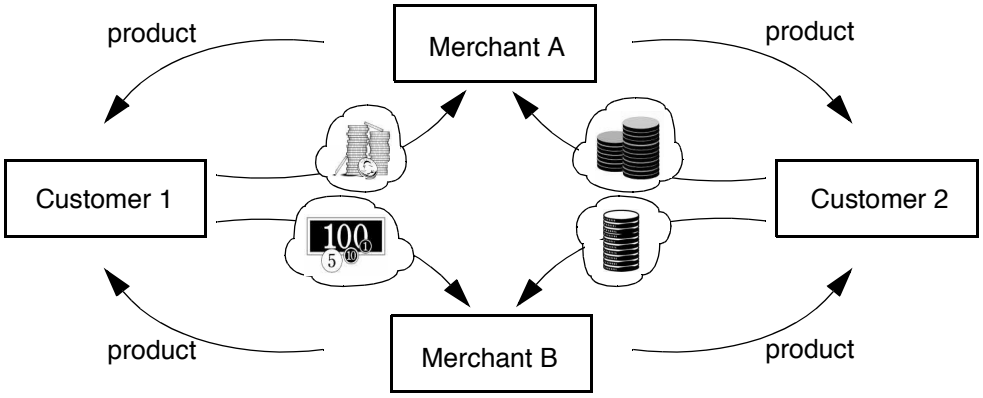


Figure 1.4 *Current payments for products on the Internet*

Merchants unfortunately may encounter different problems when many payment systems are concurrently used. Examples of such problems are: merchants must trust the various organizations that operate these payment systems (payment system operators, PSOs), obtain and install multiple software packages and (sometimes) hardware devices, learn the usage of several systems, adapt their accounting systems and web sites to the needs of each payment system, register on PSO websites and remember passwords, pay for using the various payment systems, contact multiple helpdesks in case of difficulties.

Merchants therefore want to use a single payment system for receiving payments from all customers, regardless the payment systems used by customers (Figure 1.5). Of course, different merchants should be able to choose and use different payment systems.

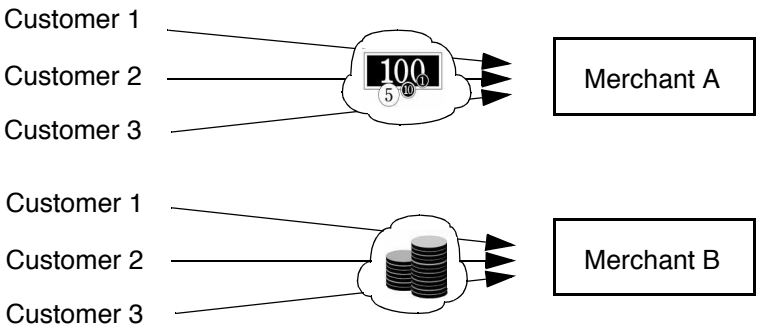


Figure 1.5 *Desired situation from the merchants' point of view*

Customers may also face different problems originating from the usage of multiple payment systems. Examples of their problems are: they must trust the PSOs, learn the usage of several systems, manage multiple accounts and e-wallets, register on PSO websites and remember passwords, obtain and install multiple software packages and (sometimes) hardware devices, contact multiple helpdesks in case of difficulties. For customers this is an inconvenient situation. On the long run the effect of these inconveniences could be that customers will turn away, and not use electronic payment systems frequently. Low value products that need to be sold in large quantities (because of the very low profit obtained per unit of product) will particularly suffer from this situation.

Customers therefore want to use a single payment system for paying all merchants, regardless the payment systems used by merchants (Figure 1.6). Again, different customers should be able to choose and use different payment systems.

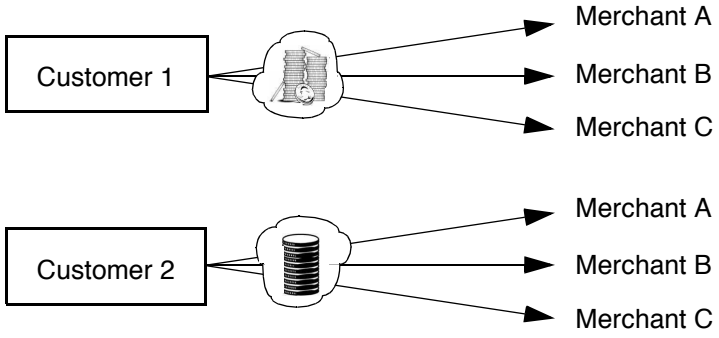


Figure 1.6 *Desired situation from the customers' point of view*

Based on the problems and preferences of customers and merchants, the main research question discussed in this thesis can be formulated as follows:

Can the desired situations of customers and merchants be achieved?

1.3 Alternative solutions

The following three alternatives may be followed to achieve the desired situations of customers and merchants [46]:

- agree on a single existing payment system and introduce it world-wide;
- create a new electronic payment system and introduce it world-wide;
- keep existing payment systems in place and introduce an intermediate system that interconnects the various systems. The system responsible for the interconnection will be called *Payment Gateway* (PG). A consequence of this is that a chain of payments will be performed instead of the traditional direct payments between customers and merchants.

1.3.1 Agree on an existing payment system

The first alternative is that PSOs and users agree on one existing payment system, which will then be introduced world-wide. This alternative, however, will be difficult to realize. The first obstacle is that the payment system operators do no want to give up their market position in favour of another system [47]. They already operate proprietary systems, which meet local (national) needs and regulations (e.g., Micromoney in Germany, w-HA in France, Minitix in The Netherlands, Quick in Austria, Nochex in the United Kingdom). The existing systems operate cost-efficient on a national scale or in broader geographical regions, so any alternative system will have serious competitors [48]. The second obstacle is that this approach violates the free market rules, which encourage competition between PSOs and their systems. The third obstacle is the legislative and regulatory differences, which are likely to occur when a single electronic payment system is being introduced in multiple countries. The fourth obstacle is those customers who may already trust and find convenient their current payment system(s). This trust and convenience may

seriously decrease the customers' willingness to switch over to an alternative system. Customers must therefore be persuaded to adopt the new system, which leads to substantial introduction costs. Hence, the idea of selecting a single payment system has a significant chance of failure. This is also shown by history: many payment systems aimed at global acceptance and domination, but none of them succeeded.

1.3.2 Create a new electronic payment system

The second alternative is to create a new payment system and introduce it world wide. The first step in this process is to define a new standard for electronic payments. In addition to the difficulties listed for the previous alternative, new obstacles arise. One of these is the standardization process. In case of technical standards, like those of the IETF, the standardization process can easily take four to six years. Standards for payment systems, however, also require involvement from financial and legal authorities. This involvement will likely further delay the development of the standard. Additionally, it is not even sure that all legal and regulatory issues can be solved due to the different laws and rules imposed by the financial authorities. For example, there is no common view on whether a payment system operator (e.g., PayPal) needs in every country a banking licence or not. There is no agreement either on the type of organizations that are allowed to issue electronic money, e.g., in the Netherlands e-money is exclusively issued by credit institutions or banks, while in Denmark non-banks are allowed to issue multi-purpose smart cards under special conditions [49].

1.3.3 Payment Gateway

The third alternative is to keep current payment systems in place and make them interoperate by introducing a *Payment Gateway* (PG), which interconnects the various payment systems. In this way, instead of one payment, a chain of payments will be performed. The new system that comprises the PG and existing payment systems is called *Hybrid Payment System*. This third alternative does not suffer from the problems of the previous ones, and is therefore more likely to succeed [46].

Figure 1.7 illustrates the payment systems interconnection. *Customer 1*, *Customer 2*, *Merchant A* and *Merchant B* may all use different payment systems. The PG is placed between the customers and merchants, and provides the interconnection function. The functionality of the PG consists of receiving payments from customers, and then making follow-up payments to merchants.

To realize the interconnection, mapping rules are needed which define how the differences between the various existing payment systems should be bridged. The interconnection of payment systems should be transparent to both the customers and merchants.

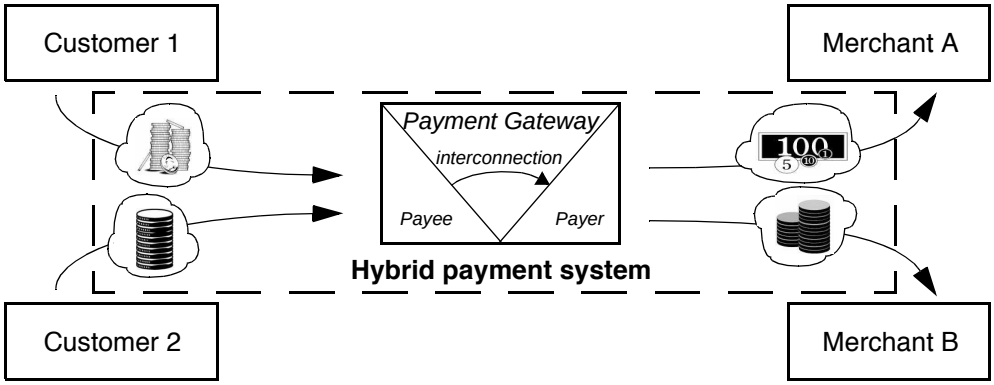


Figure 1.7 *Hybrid payment system incorporating the Payment gateway*

From the merchants' point of view, the advantages of this alternative are:

- Merchants need to trust only one PSO.
- Merchants select and use one payment system provided by the trusted PSO; therefore, they need to obtain and install software and hardware components of just one payment system, and then learn its usage.
- Merchants register only on the web site of the trusted PSO, and contact its helpdesk in case of difficulties.
- Adapt their accounting systems and web sites to the needs of a single payment system.
- Merchants pay only one PSO for using its payment system.

From the customers' point of view, the advantages of this alternative are:

- Customers need to trust only one PSO.
- Customers select and use one payment system provided by the trusted PSO; therefore, they need to obtain and install software and hardware components of just one payment system, and then learn its usage.
- Customers may use one or more accounts or e-wallets within the selected payment system, because the same instructions for use apply for each account or e-wallet of that system. It is likely that in most cases customers will manage only one account or e-wallet, and will receive one bill or financial statement, which contains a list of all product payments.
- Should any difficulty or error occur while using the payment system the customers can contact a single helpdesk.

This alternative does not suffer from the drawbacks of the other two, thus it has additional advantages:

- PSO's present on the market can keep their positions (i.e., their customers) and can operate their proprietary payment systems.
- The free market rules are not violated, PSOs and their systems can compete with each other.
- Investment costs are reduced compared to the first two approaches, and investments are only needed to implement the interconnection function.
- The hybrid payment system can start functioning without any delay as soon as the interconnection function is implemented.
- Legal and regulatory issues are in large measures already addressed, since the payment systems already function, and their operators are already controlled by financial authorities. The PG, however, needs to conform to the applicable legislation.

1.4 Objective and research problems

The previous section proposed the introduction of a Payment Gateway and creation of a hybrid payment system. Since this system does not exist, it has to be designed. A design of this system is an architecture, which models the system in terms of functionality and structure [50]. The objective of the research presented in this thesis is

> *to present an architecture of the hybrid payment system that, in an Internet environment, allows each customer to use a single payment system for paying small amounts of money to all merchants. This architecture also allows each merchant to use a single payment system on the Internet to receive low value payments from all customers.*

Before an architecture of the hybrid payment system is designed, solutions for several research problems should be found. The source of these problems is the introduction of a PG. The following problems should be considered and solved:

- How do payments fit into the accounting process?
- What are the main characteristics of payment systems? What kind of payment systems exist?
- What are the requirements for the targeted hybrid payment system?
- How is the interconnection modelled and realized?
- Which classes of payment systems can be interconnected?

Other issues may arise besides the mentioned ones, e.g., transaction costs, business agreements. The transaction costs may increase for customers and/or merchants, because the service provided by the PG should be paid. But the costs may also decrease, because the PG will probably boost the number of transactions, resulting in a lower cost per transaction. The business agreements between the various operators are also needed beside the technical interoperability of their systems. Such an agreement is, for instance, the usage of a common settlement system, which will require cross-border cooperation

between banks and other parties [47]. It is not in the scope of this thesis to address and solve the latter issues, however.

1.5 Approach and structure

In order to achieve the objective, first the payment function needs to be studied in the broader process of accounting. For this, the state of the art in accounting is studied from the perspective of standardization organizations, research projects, commercial products and platforms (Chapter 2). Because the accounting terminology is very differently used, first the precise accounting terminology is presented, and used throughout this thesis in a consistent way. During the presentation of the terminology, the relationships and information flows between the various accounting functions are also explained.

This thesis mainly focuses on electronic payments systems that transfer low amounts of money, and ensures their interconnection. For this, the electronic systems should be studied first to learn the structure and functionality of the payment systems. Payment related terminology, payment system classification characteristics, and an overview of electronic payment systems are therefore presented (Chapter 3) following a bottom-up approach.

The requirements for the proposed hybrid payment system are derived in a top-down manner from the conclusions of Chapter 2 and 3, and legal and regulatory acts. These requirements are formulated from the perspective of end-users of the new system, payment system and Payment Gateway operators, who are the stakeholders of the new system, and legal and regulatory acts, which constitute the legal framework of the hybrid payment system (Chapter 4).

After that, an architecture of the proposed hybrid payment system is developed in three phases (Chapter 5 and 6). In the first phase, the functional requirements of the hybrid payment system are formulated based on the requirements formulated in Chapter 4 and the characteristics of existing payment systems from Chapter 3. In the second phase, the hybrid payment service (i.e., the behaviour of the system as experienced by the users), is designed based on the functional requirements. In the third phase, the possible interconnection methods for existing micropayment systems are discussed and the hybrid payment protocol

is designed. For the protocol design, the hybrid payment service definition and the selected interconnection method are taken into account.

Demonstrations show that an implementation of the hybrid payment system is achievable. An evaluation verifies whether the design the hybrid payment system satisfies the hard requirements formulated in Chapter 4. For the demonstrations and evaluation, case studies are presented in which the functionality of the various system parts are shown (Chapter 7).

Finally, the conclusions and contributions of this thesis are presented (Chapter 8).

Figure 1.8 depicts the structure of this thesis and the relations between the chapters.

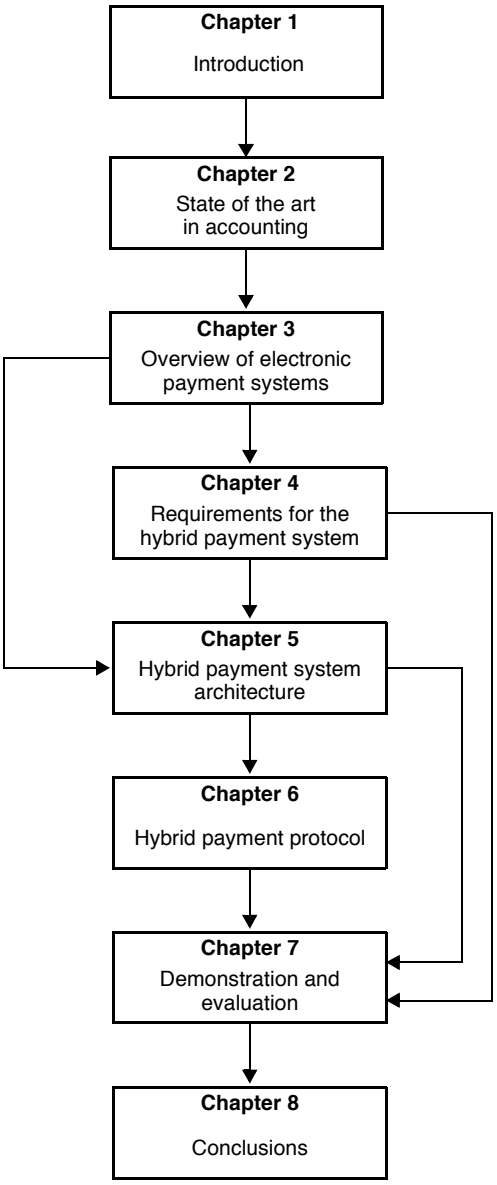


Figure 1.8 *Structure of this thesis*

1.6 References

[1] International Telecommunication Union, http://www.itu.int
[2] International Telecommunication Union: Free statistics home page, http://www.itu.int/ITU-D/ict/statistics/
[3] Internet Activity Index, http://www.online-publishers.org/?pg=activity
[4] Nielsen//NetRatings, http://www.nielsen-netratings.com/
[5] http://www.isc.org/ops/ds/reports/2004-01/
[6] http://www.isc.org/ds
[7] University of California at Berkley, http://www.berkeley.edu/
[8] How much information?, Study of the School of Information Management and Systems, University of California at Berkley, 2003 http://www.sims.berkeley.edu/research/projects/how-much-info-2003
[9] Boumans J., Paid content, E-content report by ACTeN, January 2004
[10] Gartner, Inc., http://www.gartner.com
[11] Leong L., Global Internet offers big opportunities for growth, Report of Gartner, June 2003, http://www4.gartner.com/resources/115900/115902/115902.pdf
[12] Jackson, P. et al., Turning on broadband users, Report of Forrester Research, November 2001, http://www.forrester.com/ER/Press/Release/0,1769,665,00.html
[13] Forrester Research, http://www.forrester.com
[14] Ulph Jennings, R. et al., Downloads: 13% of Europe's music market in 2007, Report of Forrester Research, May 2003, http://www.forrester.com/ER/Press/Release/0,1769,800,00.html
[15] Apple iTunes, http://www.apple.com/itunes/
[16] MusicNet, http://www.musicnet.com
[17] BuyMuic, http://www.buymusic.com
[18] On Demand Distribution, http://www.ondemanddistribution.com
[19] Online Publishers Association, http://www.online-publishers.org/
[20] Online Publishers Association, Online paid content - US market spending report for 2003, May 2004, http://www.online-publishers.org/pdf/opa_paid_content_report_may04.pdf
[21] Online Publishers Association, Online paid content - US market spending report for 2004, March 2005 http://www.online-publishers.org/pdf/paid_content_report_030905.pdf
[22] Verband Deutscher Zeitschriftenverleger, http://www.vdz.de
[23] Sapient, http://www.sapient.de

[24] Verband Deutscher Zeitschriftenverleger and Sapient, Paid content market in Germany, January 2003, http://www.opa-europe.org/mediabase/documents/1_030702_Paid_Content_I_English_mvs.pdf
[25] Ludwig-Maximilians University of Munich, Institut für Unternehmensentwicklung und Organisation, http://www.efoplan.de/
[26] Verband Deutscher Zeitschriftenverleger, Sapient and Ludwig-Maximilians University of München, Study of pricing of paid content and paid services, Berlin, June 2003 http://www.opa-europe.org/mediabase/documents/1_030825_Pricing_Engl._MvS.ppt
[27] Open Systems Interconnection, Basic reference model: Part 4: Management framework, ISO 7498-4, Geneva, 1989
[28] Aboba, B., et al., Introduction to accounting management, IETF RFC 2975, October 2000
[29] Fabrega i Soler, L., A proposal for an admission control method for the assured service in the Internet, Research project of the IITAP Ph.D. Program, University of Girona, November 2002
[30] Glossary of MurdochNet Terminology, Murdoch University, Perth, July 2004
[31] Stiller, B. et al., Charging and accounting for Integrated Internet Services - State of the Art, problems, and trends -, In Proceedings of INET '98, Geneva, July 1998
[32] Biemans, F.P.M., et al., Reference models for networked applications, Lecture notes, Telematics Institute, May 2002
[33] Pras, A. et al., Internet accounting, In IEEE Communications Magazine Vol: 39(5), ISSN 0163-6804, pp. 108-113, May 2001
[34] Párhonyi, R. et al., A provider based accounting architecture, In the Proceedings of IEEE Workshop on IP Operations and Management, ISBN 0-7803-7658-7, pp. 49-53, Dallas, October 2002
[35] NTT Docomo, http://www.nttdocomo.co.jp/english/index.shtml
[36] Barclaycard Merchant Services, http://www.epdq.co.uk/
[37] Streamline, http://www.streamline.co.uk/index_frame.htm
[38] Böhle, K., Integration of electronic payment systems into B2C Internet commerce, Background paper no. 8, Electronic Payment Systems Observatory, April 2002
[39] Øygarden, K., Constructing security - The implementation of the SET technology in Norway, Dissertation, 2001, University of Oslo
[40] Kniberg, K., What makes a micropayment solution succeed, Master's Diploma Project, Department of Applied Information Technology, Kungliga Tekniska Högskolan, Sweden, November 2002

[41] Abrazhevich, D., Electronic payment systems - A user-centered perspective and interaction design, Ph.D. Thesis, Technical University of Eindhoven, April 2004
[42] European Central Bank, Blue Book Addendum, April 2002
[43] Gerdes, G.R., Walton, J.K., Federal Reserve Bulletin, August 2002
[44] Böhle, K. et al., Electronic payment systems, Background paper no. 1, Electronic Payment Systems Observatory, December 2000
[45] Papameletiou, D., Study on e-payments, Volume 1, Institute for Prospective Technological Studies, Seville, May 1999
[46] Párhonyi, R. el al., Collaborative micropayment systems, In the Proceedings of the World Telecommunications Congress (WTC) 2004, ISBN 89 950043-1-2 93560, Seoul, September 2004
[47] Böhle, K. et al., Electronic payment systems - Strategic and technical issues, Background paper No. 1 of the Electronic Payment Systems Observatory, December 2000
[48] Abrazhevich, D. et al., Classification and characteristics of electronic payment systems, In Proceedings of the Electronic Commerce and Web Technologies Conference, Springer-Verlag Berlin Heidelberg, 2001
[49] Hille, S., Legal and regulatory requirements on accounting, billing and payment, Deliverable 1.4 of the GigaABP project of the Telematics Institute, Enschede, November 2000
[50] Vissers, C.A. et al., The architectural design of distributed systems, Reader for The Design of Telematics Systems course, University of Twente, Enschede, November 2000

Chapter 2

State of the art in accounting

Accounting forms the context of this thesis. This chapter, therefore, presents the state of the art in accounting. This overview aims to determine the different interpretations of the term accounting, to investigate the type and amount of work invested in transport and content accounting, and to study which accounting functions can be outsourced and what kind of organizations can provide them.

The term accounting and related terminology are defined and used in many different ways, sometimes in a broader or a more restricted sense. Section 2.1 therefore defines the accounting terminology. We will then use this terminology in a consistent way throughout this thesis.

After that, this chapter presents the results of past and ongoing activities (e.g., standards, RFCs, architectures, protocols) from the perspective of standardization organizations (Section 2.2), research projects (Section 2.3), commercial products and platforms (Section 2.4). This division of the accounting work is inspired from [10]. Section 2.5 presents the comparison of accounting terminology and work presented in this chapter. Finally, Section 2.6 draws the conclusions.

2.1 Terminology

The starting point to define the accounting terminology is the OSI Management Framework [8]. In this framework, accounting is defined as the process of metering the resource consumption, pricing, charging and billing of customers

for their service usage. Payment is a sub-function of billing. Each of these functions generate specific data, which is stepwise processed by a next function until the amount of money to be paid is being determined and the payment has been performed. Figure 2.1 depicts a functional decomposition of an accounting system based on [1].

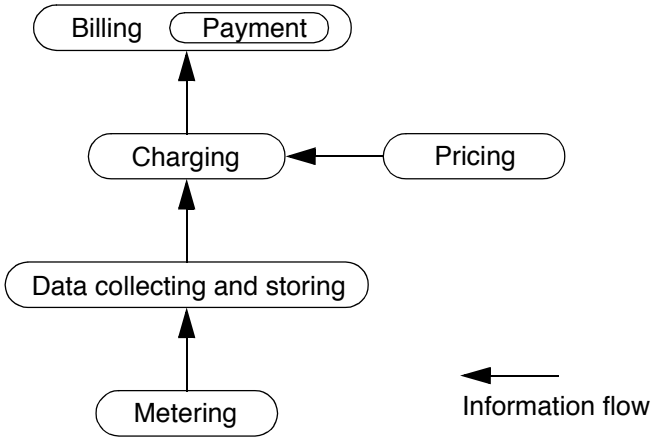


Figure 2.1 *Accounting functions and their relationship*

Accounting is often investigated together with authentication and authorization, because customers should be authenticated to know precisely who consumes the resources and who will pay the bill. Authentication identifies who the customer is. Authorization describes what the customer can do. These three tasks are regarded as AAA (pronounced "triple a").

2.1.1 The metering function

Metering is the function that comprises the information generation and registration activities with respect to the resource consumption of customers [9]. This information is of technical nature and expresses measurable quantities of the consumed resources (e.g., number of bytes sent and received, number of files downloaded, duration of resource consumption, number of links visited and Quality of Service class). These quantities can also be combined. Although this information is primarily used to calculate the charges, it can also be used for other purposes, e.g., for statistical analysis, network dimensioning, or data mining.

Depending on the type of resource, metering can be performed at network or application level. Metering can be performed at the customers' (client) side, or at the merchants' (server) side, or in the network between the two sides. Metering can be intrusive (i.e., there is an interference with the resource consumption) or non-intrusive (i.e., there is no interference).

2.1.2 The data collecting and storing function

Data collecting and storing is the function that comprises the transport and accumulation of metering data in (central) storage facilities [10]. Collection of metering data is required because metering may be performed at distributed places, but the processing of this data can then be done in a central place (e.g., there are situations when metering data originating from different meters must be correlated to create precise resource consumption data). Storing of metering data is required because meters may have a limited storage capacity and the metering data may be needed for a longer period. There is a serious risk of losing metering data when this capacity is insufficient. Since the billing of customers cannot be accurate and reliable. The collected metering data is usually stored in a uniform format. In this way, the processing of data can be more efficiently performed [9]. When different data formats are used, translation of data is necessary, however. In that case, it is important to define syntactical and semantic data conversion rules to avoid the loss of information during the translation. The collecting and translating of the metering data into a uniform format is performed by so-called *mediation systems* [55]. The term mediation is often used in the context of telecommunications accounting systems.

2.1.3 The pricing function

Pricing is the function that sets a price on consuming a (unit of) resource [9]. Pricing translates economical considerations into technical quantities that can be metered. Prices can be determined based on costs/profit basis or on the current market situation [10]. Prices are usually expressed in monetary units. Prices may be influenced by factors like time, date, or duration of content consumption, quality of content, etc. For example, the price of transporting 1MBs of data is €0,05, or the price of 1 minute of streaming video is €0,09.

The prices are mainly static with respect to time. Nevertheless, there can be situations when the price of content changes during the content consumption caused by different circumstances. For instance, pricing may depend on the amount of free bandwidth capacity left. Usually static prices change slower than dynamic prices [11].

Prices are generally collected in pricing schemes. These schemes are a critical part of businesses and closely related to marketing. They are part of the cornerstones of a profitable business [10]. Research has shown that a well-defined pricing scheme is difficult to realize because the prices should reflect the value of the service perceived by customers, and not the costs made to provide the products or service [12]. Take, for instance, the usage-based and flat-fee pricing schemes for telecommunication services. Usage-based pricing can impose a high overhead on telecommunication systems, the operation of a whole network and the provisioning of high-quality end-to-end service is expensive. Flat-fee pricing has proven to be difficult in practice [9]. Note that, the generation or computation of a static pricing scheme is as difficult as of a dynamic [14].

2.1.4 The charging function

Charging is the function that calculates the costs for a given resource consumption [9]. This function uses as input the collected metering data for that particular resource, and the unit price of consuming that resource. The output of this function is the charges that customers will have to pay. In this way, charging translates technical quantities into monetary units. The charges are collected and may then serve multiple purposes. For instance, it may be input for the billing function, auditing, statistical analysis, revenues estimation or financial planning.

Charging policies are those rules that define the frequency of calculating the costs and the granularity of the resulting charging data (depending on its purpose). The costs can, for instance, be calculated every time after collecting and storing the metering data, at regular time intervals (e.g., on a daily or weekly basis), or when requested by the billing function.

Sometimes the charges are distributed over multiple parties (i.e., splitting the costs). The distribution policies (beside the charging policies) define how the charges should be distributed and to whom. In this case, the distribution of charges should be done during the calculation and assigned to each party involved. Such a situation may occur when, for instance, a company has certain interest in subsidizing individual customers for their content purchases. The company supports 25% of the costs made by every customer up to a certain maximum. This means that the charges should be split accordingly between the company and customers.

2.1.5 The billing and payment function

Billing is the function that transforms the charging information into invoices for customers [9], [79]. *This function also includes the transfer and presentation of bills to customers.* A bill may contain the charging information calculated for one resource consumption or collected over a period (e.g., one month) of a customer. For instance, telephone companies collect the charging information for one or two months then send the bills to their customers. Bills can be presented in a traditional (on paper) or electronic (e.g., via email) manner.

Billing policies are rules that configure the billing function. They define among others [10], [40]:

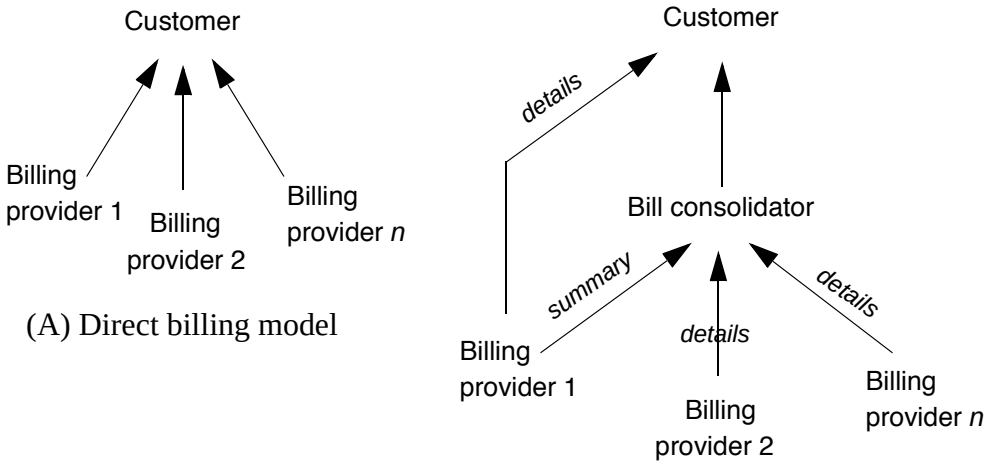
- the type of the bill (e.g., invoice, credit card);
- a time period, called the billing period when the charging data is collected, the bill is created and presented (e.g., one month);
- the outlook and content of bills;
- the date when the bill should be paid;
- possible payment systems to pay the bill.

An aggregated bill presents the total amount of the aggregated charges. An itemized bill contains in detail every separate piece of charging information. The content of the bill depends on the granularity of the charging information.

Payment is the sub-function of billing, and is responsible for the money transfer from the customer to the merchant or service provider. The amount of

money to be paid is indicated on the bill. Bill payments are generally performed with traditional (e.g., cash, paper checks, automatic collection from the bank account of the customer) or electronic payment systems (e.g., credit card systems).

Note that, the billing and payment function is also called Electronic Bill Presentment and Payment (EBPP) if both the bill presentation and payments are performed using the Internet [56]. In literature, two cases are distinguished for Electronic Bill Presentment (EBP) [13]. In the first case, a billing provider presents online (e.g., on its corporate web site) the bill to the customer. This is called direct billing model (Figure 2.2 *A*). This means that the customers need to enroll into the EBP service of every provider. In the second case, the billing providers send their bills to a bill consolidator, which groups and presents the various bills to the intended customers. This is called consolidator billing model (Figure 2.2 *B*). The bill consolidator can act as a portal, which means that the billing provider sends only a summary of the bill and it can still act directly with the customers (called thin consolidation model), or can receive from a billing provider the summary and details of a bill and interacts alone with the customers (called thick consolidation model) [56].



(A) Direct billing model

(B) Consolidator billing model

Figure 2.2 *Direct and consolidator billing models*

2.2 Standardization organizations and activities

This section presents the accounting work and results of the main standardization organizations. These organizations are the Internet Engineering Task Force (IETF, [2]), the Internet Research Task Force (IRTF, [3]), the IP Detail Record (IPDR, [4]), the World Wide Web Consortium (W3C, [5]), the International Standardization Organization (ISO, [6]), the International Telecommunication Union (ITU-T, [15]) and the Interactive Financial eXchange Forum (IFX, [7]).

2.2.1 Internet Engineering Task Force

The IETF is part of the Internet Society [16] and is responsible for the development and standardization of new Internet technologies. It is "a loosely self-organized group of people" who contribute to the evolution of the Internet [17].

The work of the IETF is organized in eight areas and each area consists of several working groups (WG). The areas that consider Internet accounting are the Operations and Management, and Applications areas. The working groups of interest are the Remote Authentication Dial-In User Service WG (RADIUS WG), Authentication, Authorization and Accounting WG (AAA WG), Real-time Traffic Flow Measurements WG (RTFM WG), IP Flow Information Export WG (IPFIX WG), and Internet Open Trading Protocol WG (Trade WG). These different WGs focus on one or more accounting functions, and sometimes continue the work of groups that finished their work. The following sections present the work of these WGs.

Remote Authentication Dial-In User Service (RADIUS) WG

The RADIUS WG was created in 1995. The addressed topic of this group was the communication of authentication and authorization information between a Network Access Server (NAS) and an authentication and authorization server. The WG has finished its work in 1999 and is not active anymore.

The WG had developed a protocol (RADIUS) that carries authentication, authorization, and configuration information between a NAS and a shared Authentication Server (RFC 2856, [18]). RADIUS is mostly used in combina-

tion with modem pools. The protocol collects and stores data regarding the dial-up activity (session) of a user: user identifier and duration of the session, services and protocols used, NAS and port identifiers, addresses, octets transferred and cause of session's termination [3]. Such data is used for auditing and billing purposes.

An extension of RADIUS to enable accounting has been added later (RFC 2866, [19]). The protocol has not primarily been developed for accounting purposes; therefore its support for accounting is limited.

Figure 2.3 depicts two users that request access to network resources. The NAS (RADIUS client) is a device to which users connect. The client requests the RADIUS server to provide authentication status, user profiles and authorizations for the users. The server authenticates the client first and then checks the users' identity and authorization. It returns the users' status (connection authorized or rejected) and configuration information to the client. Transactions between a client and server are authenticated using a shared secret key that is never sent over the network. Furthermore, user passwords included in RADIUS messages are sent encrypted to protect them.

We note that, in the view of this WG, accounting was used as a synonym for metering. A RADIUS client may generate and store metering data about the session of the users. At the beginning of a session, the client sends an "Accounting start" message to the RADIUS server. After the termination of the session, the client sends the collected data (e.g., input/output bytes and packets, duration of the session, etc.) and an "Accounting stop" message to the server. Thus, the protocol is only useful for post processing, not for real-time applications.

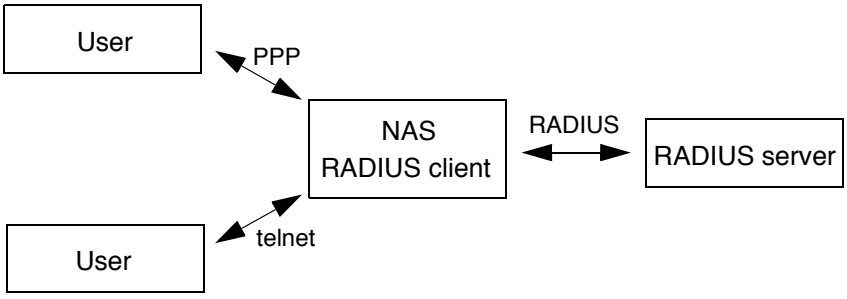


Figure 2.3 *RADIUS architecture*

RADIUS is a widely used protocol; many vendors have included RADIUS support in their equipment (e.g., Lucent Technologies, Cisco Systems). Nowadays, routers and NASs are more complex and can handle a large number of users. The limitations of RADIUS described in the literature ([21], [25]) makes this protocol unsuitable for such environments. The limitations are for instance, scalability and end-to-end security. Additionally, the implementations of RADIUS accounting are vulnerable to packet loss, application and network failures, and device reboots [21].

Authentication, Authorization and Accounting (AAA) WG

The AAA WG was chartered in 1999. The goal of this WG is to specify an AAA protocol that suits the requirements formulated by the NASReq, Mobile IP and ROAMOPS WGs. The work of this WG received lots of interest from a large audience (large number of participants at their IETF meetings and mailing-list discussions). At the time of writing this thesis, the WG was still active.

The contributors of the WG wrote first documents that captured the state of the art in accounting protocols and their design (RFC 2975, [21]), the accounting attributes (e.g., user name, user password, NAS IP address, NAS port) and record formats (e.g., string, address, integer, time) (RFC 2924, [22]). We note that, this WG defines accounting as "the act of collecting information on resource usage for the purpose of capacity planning, auditing, billing or cost allocation" [24]. The collected data can be used for different purposes such as billing, trend analysis, cost allocation or auditing. A study of the AAA base protocol requirements for network access resulted in [23]. In September 2003 an AAA protocol called Diameter is defined (RFC 3588, [24]). This protocol was designed such that it is applicable for several network access models.

Diameter is based on the RADIUS client-server model. Diameter has backwards compatibility with RADIUS and tries to correct its limitations [25]. The purpose of the Diameter base protocol is to provide an AAA framework for the WGs mentioned before, thus it also supports mobility and roaming. The description of the base protocol specifies the message format, transport, error reporting and security services to be used and supported by all Diameter applications.

Diameter has the capability to deliver real-time accounting information. An AAA server can instruct a network device that implements Diameter to generate accounting data, which in fact is metering data. The server also gives instructions on how the accounting data should be delivered (e.g., data transfer strategy and timeliness information). Several fault resilience methods were included to have a robust accounting protocol and to minimize the loss of accounting information.

We note that, the 3rd Generation Partnership Project (3GPP) adopted Diameter as an AAA protocol for 3GPP networks [26]. The details of Diameter-based charging for 3GPP are described in [27].

Real-time Traffic Flow Measurements (RTFM) WG

The RTFM WG [28] started its work in 1996. This WG focused on the metering of network traffic (i.e., traffic flows) and collecting the metered data. Their three main objectives defined in their charter were (1) to review the existing work on traffic flow measurement, (2) to produce an improved traffic flow model, and (3) to develop a standard Flow Meter Management Information Base. The WG has completed its goals and stopped its activities in 1998.

The results of this WG are the:

- Real-time traffic flow measurement (RTFM) architecture for the Internet (RFC 2063, [29]);
- Meter Management Information Based (Meter MIB, RFC 2064, [30]);
- Simple Ruleset Language (SRL, RFC 2723, [31]).

The RTFM architecture (i) describes the entities implementing the metering, data collecting and storing functions, (ii) defines a traffic flow (e.g., a flow can be defined by the attributes of their end-points), and (iii) how to write rules to configure a metering entity to know which data flows are of interest and which are not. The entities of the architecture are a meter, a meter reader and a manager (Figure 2.4). The *meter* measures the traffic flows, generates metering data and stores this data temporarily. The *meter reader* collects the stored data from the meter, i.e., implements the collecting and storing function. This information can be passed to an *analysis application* (which not part of the architec-

ture), and can be further used for billing. The *manager* starts and stops the activity of the meter and downloads to the meter the rulesets (i.e., group of rules) on which the traffic metering is based. The *Meter MIB* is used to control the meter and store the rulesets. Traffic flows are considered bi-directional, and they can be specified at many different levels of aggregation. In general, flows are defined between endpoints: a source endpoint and destination endpoint. These endpoints can be specified between the data link and transport layers (e.g., MAC addresses, IP-addresses or Port-Numbers). The architecture specifies only a pull model for reading the data out of a meter.

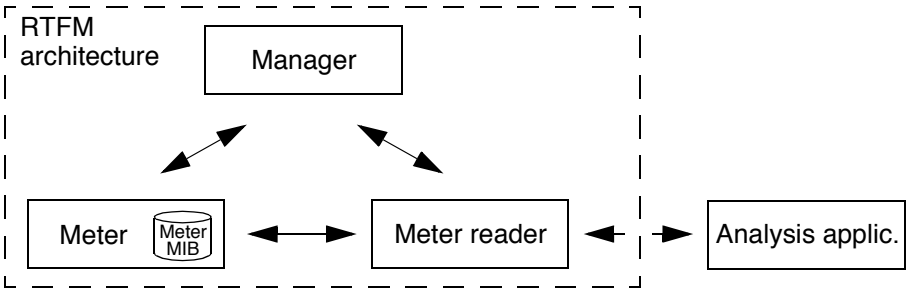


Figure 2.4 *RTFM architecture*

NeTraMet (RFC 2123, [32]) is a public domain implementation of the Meter MIB. NeTraMet provides a Simple Network Management Protocol (SNMP) agent to make the metering data available to meter readers. NeTraMet can also be used for real-time network monitoring and trouble-shooting.

IP Flow Information Export (IPFIX) WG

The IPFIX WG [33] was chartered in 2001. The goal of its work is to develop a standard architecture for IP flow information export. The work of this WG consists of developing a data model that describes the flow information, and a transport protocol, which is used to transfer this information from an exporting entity (implementing the metering function) to collection entities. The information can be used as input for network management systems, billing systems, etc. This means that the WG addresses the data collection and storing accounting function. At the time of writing this thesis, the WG was still active.

This group continues the work of the RTFM WG by revising and extending the previous work, and focuses on the information transport protocol. We note that,

the exporting and collecting process term used by this WG is actually a synonym for the collecting and storing accounting function. Internet-drafts describe the IPFIX reference model, which allows the data collecting and storing entities to collect flow information from one or more metering entities, and to provide the collected information to one or more third parties for processing.

At the beginning of 2004, the work in progress of the WG considered the definition of requirements for the IPFIX protocol, the flow information export architecture and data model, the evaluation of the protocol. The flow export format of NetFlow version 9 developed by Cisco Systems is selected as a basis for the IPFIX standard.

Internet Open Trading Protocol (IOTP) WG

The IOTP WG [34] was chartered in 1998 to continue the engineering work of the IOTP protocol started by the Open Trading Protocol (OTP) Consortium.

The results of the IOTP WG are a framework for trading of products on the Internet (RFC 2801, [35]) and the Electronic Commerce Modelling Language (RFC 3106, [36]). The developers of this framework tried to provide an online trading model, which resembles the every day's trading and which supports current and future mechanisms. The trading process includes the accounting for products (as defined in Section 2.1) and delivery of products. The products are delivered over other channels than the Internet, however.

In the IOTP framework the following roles are identified: customer, merchant, customer care provider (for taking care of disputes with customer), a payment system operator and a delivery provider. The protocol is optimized for the case when the customer and merchant do not have prior relations. Multiple roles can be played by a single organization (e.g., a merchant can also be a customer).

If we assume that no disputes take place (so the customer care provider can be omitted), the following interactions take place (Figure 2.5). First, a customer selects the product(s) from a merchant. The accounting system of the merchant executes the accounting functions, and presents the bill to the customer. To pay the bill, the merchant offers one or more electronic payment systems provided by a payment system operator. The customer selects one system and initiates a

payment, which is processed by the payment system. Usually, these payments have large values (e.g., above €5). After the payment, the delivery provider delivers the selected and paid product(s).

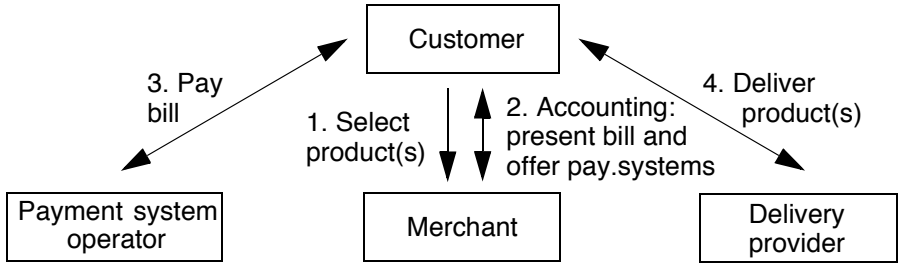


Figure 2.5 *IOTP interactions*

Although, IOTP is an open standard, not many merchants use it. A major reason for this is that there are no incentives to adopt this standard. One problem with IOTP is the inefficient and inconvenient payment message exchange, which is tunnelled through the IOTP protocol. Another problem is that the protocol was designed completely generic and brand-independent, which made it inflexible towards existing and emerging payment systems (e.g., systems that support person-to-person payments). The generic and brand-independent characteristics of standards usually make them powerful, but in this case, it might be a disadvantage. Another problem emerges from the behaviour of merchants: they need payment systems to be integrated into their web shops as soon as possible within a given budget and given time constraints. Payment system developers create proprietary systems for a particular environment and customer group, and are less interested in generic standards [37]. At the time of writing, an improved version of IOTP (IOTP v2.0) was under development.

The Electronic Commerce Modelling Language (ECML) defines a standard set of information fields. The aim of this standard is to ease the process of filling in various fields with customer information on the web sites of merchants. In this way, customers will be less confused by the varying web sites of merchants. For instance, the customer information needed to make payments can be filled in automatically in a standard manner for every merchant from which the customer buys content. ECML may be used with any payment system. At the time of writing, version 2 of ECML is under development.

2.2.2 Internet Research Task Force

The IRTF [3], like the IETF, is part of the Internet Society and is responsible for promoting research for the evolution of the Internet. Its research groups are working on topics related to Internet protocols, architectures and applications. Members of the IRTF are mostly individuals rather than representatives of organizations.

The work related to Internet accounting is carried out within the Authentication, Authorization and Accounting Architecture Research Group (AAAArch RG).

Authentication, Authorization and Accounting Architecture (AAAArch) RG

The AAAArch RG [38] continued the work of the Authorization subgroup of the AAA WG. This RG was chartered to define short-term requirements for a protocol that will support requirements of the NASReq and Mobile IP WGs. The AAAArch RG is not active anymore.

The goal of this RG's work is to allow the realization of authorizations in a scalable manner over a network of interconnected AAA servers, i.e., to allow inter-domain AAA services. The RG developed an AAA architecture that interconnects three types of components [39]:

- generic AAA servers that authenticate users, handle authorizations and collect accounting data;
- application specific modules (ASMs) that manage resources and configure the service equipment to provide an authorized service (Mobile IP, roaming etc.);
- policy and event repositories (PRs) are databases containing information about the available services and resources from which authorization decisions should be made, and the policy rules used to make the decisions. For auditing purposes, these databases also contain the events that occur in the generic AAA servers.

We note that, in the view of this RG, the term accounting defines the collection, transport and storage of resource usage data, hence it is a synonym for the data collecting and storing function defined in Section 2.1.2. Metering data can originate from RTFM or IPFIX meters, logged data, etc.

Recent work focused on policy-based accounting to provide flexibility for data collecting and storing architectures [40]. The policies describe how such architecture can be configured in a standard way. In other words, policies define rules for metering, transport and collection of metering data. These policies can be exchanged between AAA servers to share configuration information.

Figure 2.6 depicts the generic AAA architecture. First, a user requests from a Generic AAA server access to a service. The user specifies a number of parameters of that service. The server sends service specific information (i.e., the service request and the policies of that service) to the ASM, which uses the received information to configure the meter that will measure the service consumption of the user. Then the generated metering data is transported to the AAA server.

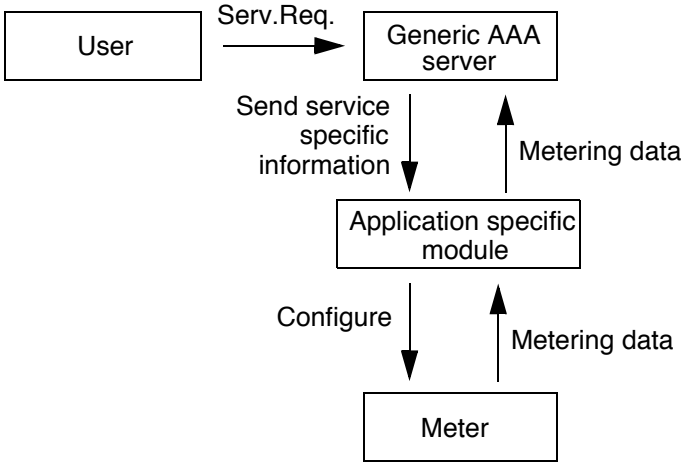


Figure 2.6 *Generic AAA architecture*

2.2.3 Internet Protocol Detail Record Organization (IPDR)

The IPDR Organization [4] is an open consortium that has as objective the reduction of time and costs for metering resource consumption for accounting.

For this, they create ad hoc standards, which are used to define and exchange usage and control data between network and hosting elements, operations and business support systems.

We note that, accounting is defined by IPDR as "the process of collecting and analysing service and resource usage metrics for the purposes of capacity and trend analysis, cost allocation, auditing, and billing, etc."

Particularly, the work of the organization focuses on (i) defining a standard record format that describes interchangeable IP resource and service usage information, and (ii) a protocol that delivers such usage information. This record format is called an IP detail record (IPDR), which is based on the "call detail record" (CDRs) defined in the telecom world. The structure of an IPDR allows the characterization and capturing of any kind of service usage from an IP-based system. This means that metering data (e.g., user identification, usage date and time, usage quantity, resource) can be collected and organized in IPDRs.

The standard specified in [41] contains the followings:

- IPDR reference model, which describes the abstract and operational relationships between metering, data collecting and storing, and other processing entities;
- business requirements, which should be addressed in the protocol definition, and specific scenarios;
- protocol definition, which defines the structure and transfer of IPDRs.

The standard supports IP-based services like Voice-over-IP, wireless Internet access, IP-VPN, video streaming, email, etc.

2.2.4 World Wide Web Consortium (W3C)

W3C [5] began its work in 1994. This work focused on the development of common protocols that contribute to the evolution of the World Wide Web. The members of W3C include technology vendors, merchants, research laboratories, standards bodies and governments.

The activities of W3C are organized into three major groups: Working Groups that are responsible for technical development, Interest Groups that perform more general work, and Coordination Groups that assist the communication between related groups.

The work of the Micropayment Markup Working Group concentrated on the payment function defined in Section 2.1.5. This WG was part of the E-Commerce Interest Group of W3C.

Micropayment Markup (MPM) Working Group

This WG developed a Micropayment Transfer Protocol (MPTP, [42]) and a language called Common Mark-up for Micropayment per-fee-links [43]. The activity of this WG is terminated.

The MPTP specifies how the money transfer is handled using a common broker. This broker has the role of keeping the accounts for both the customer and merchant. MPTP is designed for the transfer of small amounts of money and it provides a high degree of security against fraud.

The development of the Common Mark-up for Micropayment per-fee-links originated from the IBM's standardization efforts. The specification of this language allows information necessary for initiating a micropayment to be embedded in web pages. This embedding permits various micropayment wallets to coexist. This specification is implemented for instance in the NewGenPay micropayment system.

2.2.5 International Standardization Organization (ISO)

ISO [6] is the largest developer of standards. ISO is a non-governmental organization that comprises national standards institutes of 148 countries. The ISO standards are used by various industrial and business organizations, governments, regulatory bodies, etc.

ISO standardized the Open Systems Interconnection reference model (OSI RM, [44]), which provides a framework for developing communication protocol and service standards for interworking of different vendors' equipment.

The OSI RM introduced the notion of the OSI management. The OSI management framework defined in [45] was not generally accepted as a starting point for management, and an additional standard, called Systems Management Overview, was produced [46].

The OSI management framework defined five functional areas. One of them is accounting management. Accounting management is a set of facilities that allows a network manager to determine and allocate costs and charges for the use of network resources [47].

The accounting process is divided into the following functions:

- Usage metering is responsible for monitoring the resource usage, generating and storing metering data. More information about this function can be found in [48].
- Charging is responsible for creating and storing service transaction records or charging data. This function therefore collects metering data and associates pricing information to it.
- Billing is responsible for collecting charging data and producing bills for every particular resource user.

2.2.6 International Telecommunication Union (ITU-T)

The International Telegraph Union was established by 20 founding members in Paris in 1865. The Union’s name was changed into International Telecommunication Union in 1932 to reflect the Union’s responsibilities concerning wireline and wireless communication [15].

Some of the accounting standards were defined by the ITU-T in the M.3000 series and others, the X.700 series were adopted from the OSI standards. The mapping between the accounting related ISO and ITU-T standards is presented in [47].

The term TMN is introduced by the ITU-T as an abbreviation for "Telecommunications Management Network". According to the M.3010 standard, "a TMN is conceptually a separate network that interfaces a telecommunications network at several different points" [49].

Accounting management is one of the management areas supported by a TMN and it is defined as a "set of functions, which enable the use of network services to be measured and the costs for such use to be determined and rendered" [50]. The accounting management in TMN and OSI does not differ much because of the adoption of some OSI standards as a framework for the TMN. The usage metering, collecting metering records, creating charging records and billing are very similar functions. The tariffing function specifies a set of data that is used to determine the prices for the used services [51].

2.2.7 Interactive Financial eXchange (IFX) Forum

The IFX Forum [7] was formed in 1997. Among the members of this forum are financial institutions, software vendors, etc. The goal of their work is to create an open standard for exchanging financial information and instructions independent of network technology or computing platform.

This forum created an XML-based standard that supports message exchange for:

- electronic bill presentment and payment (EPBB);
- business-to-business payments and banking;
- customer-to-business payments and banking;
- Automatic Teller Machine communications [52].

With this standard, the forum tries to provide interoperability between systems that exchange financial information. The IFX standard consists of two parts:

- a specification for business level messages and associated data dictionary, which present the open, interoperable and secure exchange of electronic bills and delivery of payments;
- implementation directions, which provides additional details on how the messages may be represented physically.

2.3 Research projects

Besides standardization efforts, several research projects addressed accounting. The European Commission or national scientific research organizations, for instance, sponsor such projects. These projects were carried out by research centers, institutes or universities. The following sections present the accounting result of the Giga Accounting, Billing and Payment, Charging and Accounting Technologies for the Internet (GigaABP), Charging and Accounting Technologies for the Internet (CATI), Market Managed Multi-service Internet (M3I), Charging for Premium IP Services in the European Information Infrastructure & Service (SUSIE), Internet Quality Measurement and Accounting (IP-QoS), Internet Next Generation (ING), Anticipating Content Technology Needs (ACTeN), and An Open Personalized Electronic Information Commerce System (Opelix) projects. These projects developed, studied new concepts in the context of accounting and tried to assist the accounting related standardization work.

2.3.1 Giga Accounting, Billing and Payment (GigaABP)

The GigaABP project [53] started in January 2000 and has been carried out in The Netherlands at the Telematica Instituut in cooperation with TNO and ATOS Origin. The project focused on the financial exploitation of services and content delivered over the Internet. The goal of the project was to develop knowledge and expertise in systems, services and functions required to enable such services.

We note that, the term accounting was used in this project to denote a sub-function of the data collection and storing function. The result of this sub-function, called accounting data, can be used for information provisioning, as well as for charging, billing and payment.

In this project, a functional architecture was developed for the financial exploitation of network-based services [54]. This architecture presents the accounting functions in an integrated way (Figure 2.7). The information provisioning function is responsible for provisioning of all kinds of information to customers, merchants and service providers, etc., based on the collected and stored data. To study the applicability of the GigaABP architecture, the streaming

video service of a video portal provider was built. As example, in [13] several usage scenarios are presented. Additionally, the Jalda payment system was integrated in the implementation of the architecture.

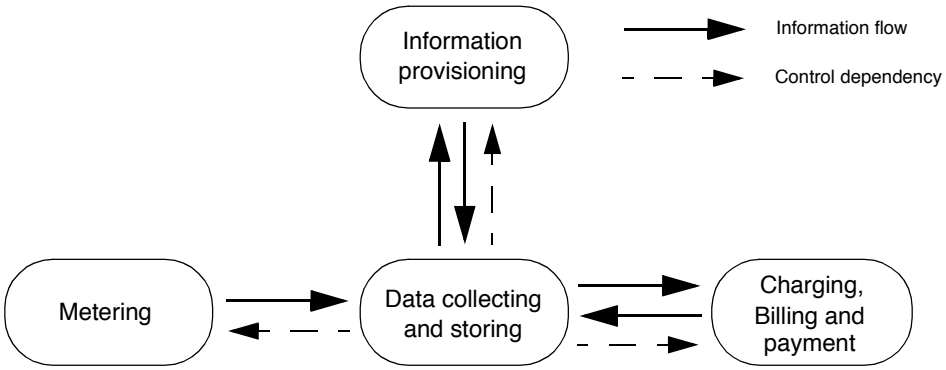


Figure 2.7 *GigaABP functional architecture*

Some accounting functions were considered individually as well. The metering function is presented in [55], where different metering strategies and implementations of the metering function are described. The usage context and various implementations of the billing and payment function are discussed in [56].

2.3.2 Charging and Accounting Technologies for the Internet (CATI)

The CATI project [57] started in July 1998 and it has been carried out at the Swiss Federal Institute of Technology. Its main objective was to design and implement a QoS-enabled, secure and Internet-based Virtual Private Network (VPN) management system, which integrates charging and accounting functionality. The project consists of two sub-projects, one focusing on Internet protocols to support charging and accounting for various Internet services, and the other demonstrating and evaluating these technologies with real-world applications.

We note that, the term accounting was used in this project to denote the data collection and storing function. The metering information is organized into accounting records, which are then collected and used by the charging function.

The main achievements of the CATI project are:

- the development of an architecture that includes the IntServ/DiffServ models and that supports charging, metering and VPN configuration;
- the development and evaluation of pricing and costs models for Internet transport services;
- the design and implementation of charging and accounting extensions for reservations, and their demonstration in case of IP telephony;
- the design and implementation of VPN service management based on a broker hierarchy [58].

To provide a proof of concept and sustain the listed achievements, demonstrators were built. The achievements were complemented by a trust model for providers and customers, security model for business transactions, and secure micropayment systems.

This project demonstrated that it is possible to have a configurable VPN management system that uses efficient charging, integrates an electronic micropayment scheme. This management system interfaces between the IntServ and DiffServ technologies, while preserving standards compatibility. The business models and demonstrators proved that the developed concepts might be realized in the real world.

2.3.3 Market Managed Multi-service Internet (M3I)

M3I [59] is a European Union project in the 5th Framework of the IST-Program that started in 2000. The partners of this project are HP European Laboratories, Athens University of Economics and Business, Darmstadt University of Technology, BT Research, Swiss Federal Institute of Technology and Telenor Research.

We note that, the term accounting was used in this project to denote the data collecting and storing function. This function receives and merges transformed metering data from mediation systems, stores this data and provides it to the charging function when requested.

The M3I project aimed "to design, implement and trial a next-generation system that will enable Internet resource management through market forces, specifically by enabling differential charging for multiple levels of service." This system would allow customers to increase the QoS by accepting different charging rates, to receive real-time feedback and to acknowledge charging information. ISPs would be able to change the prices and to communicate them to the customers, to maintain current QoS levels in case of congestion by changing prices, and to be able to charge for different QoS levels or multicast.

The main results of the M3I project are:

- the development of a pricing framework for price setting, communication and reaction;
- the design and implementation of a charging and accounting system;
- the presentation of new business models for ISPs in a set of scenarios;
- the performing of technical experiments and evaluating customer experience.

2.3.4 Charging for Premium IP Services in the European Information Infrastructures & Services Pilot (SUSIE)

The SUSIE project [60] started in 1999 and was funded by the European Commission under the ACTS programme [61]. The participants of this project were Teltec Ireland, Eurecom, Fraunhofer Institute FOKUS, Martel, Silicon Graphics, Flextel and Waterford Institute of Technology. The main objectives of this project were the usage-based charging of Premium IP services and the integration and validation of accountable IP-based services. Premium IP services are those IP-based services that are enhanced with Quality of Service.

We note that, the term accounting was used in this project to denote the process of pulling together raw metering information and creating accounting information, which can then be used to calculate the charges. This is actually a synonym for the data collecting and storing function defined in Section 2.1.2.

In this project a reference model for the charging and data collecting and storing functions was developed [62]. Figure 2.8 illustrates this model and depicts

the interaction between the different functions. This model allows the five functions to be configured by a policy server. The configuration parameters for these functions are derived from the corresponding policies, and are provided via a configuration plane. The functions can, for instance, be configured differently for IntServ or DiffServ service models.

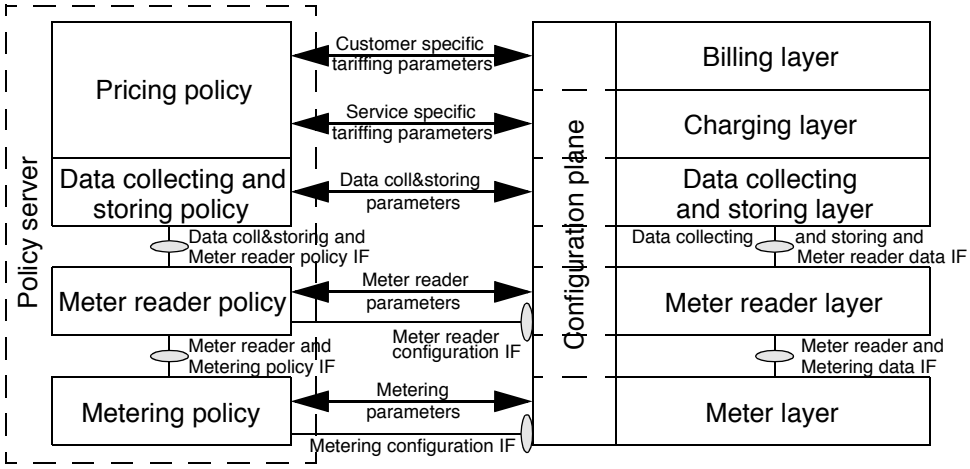


Figure 2.8 *SUSIE reference model for charging and accounting*

The developed reference model can also be used in a multi-provider billing environment. In this case, an IP flow crosses multiple ISP domains, and each ISP may charge adjacent ISPs for the used network resources. ISPs will charge the flow sending and/or receiving users directly. Adjacent ISPs may only exchange bills with each other; it is not needed to exchange charging information as well.

Accounting over multiple service providers (i.e., federated accounting) was considered because this project focused on Premium IP services and these services are delivered within a multi-service provider context. For this, an architecture of the Trade Accounting System was developed [62]. This system may become useful if DiffServ will be deployed on the Internet, and DiffServ providers will compete with each other. In this project, a TINA-based accounting system for Premium IP was developed as well [62].

TINA is a proposed architecture of the Telecommunications Information Networking Architecture Consortium [56]. TINA provides a framework for

billing control and management of dynamic service provisioning platforms [64]. We note that, within TINA, the term accounting denotes the process of "receiving data from a service and calculating charges using prices". This definition is a synonym for both the data collecting and storing, and charging functions.

The advantage of the developed TINA-based accounting system is that it can handle both ATM and IP-based accounting. This means that this system can receive metering data from a Premium IP meter reader and charging data from an ATM accounting system, and creates converged charging information. Hence, this system calculates IP-based charges based on metering information and combines them with related ATM charging information.

SUSIE contributed to the work of the AAA WG and to the NA 8 Working Group of the European Telecommunications Standards Institute. The developed reference model became the basis of the policy-based accounting work [40] of the AAAarch RG.

2.3.5 IP-QoS - Internet Quality Measurement and Accounting

The IP-QoS [65] project started in 2001 and was carried out at the Fraunhofer Institutes FOKUS and IITB. The goals of the project were the design and development of a flexible metering platform to support the deployment of Diff-Serv within IP networks. Such a metering platform is needed to perform usage-based accounting and the validation of guaranteed quality parameters.

We note that, the term accounting was used in this project to denote the process of collecting data about the resource consumption [66]. This definition is therefore a synonym for the metering function described in Section 2.1.1.

The results of the project are the following [66]:

- the identification of the requirements for the metering interface. This identification was needed to be able to develop a communication interface for the configuration of management tasks and data collection;
- the evaluation of existing tools and systems for metering: the evaluation is performed according to the previously identified requirements.

This evaluation concluded that the examined tools and systems do not satisfy the identified requirements.

- the description of the work performed in the metering field by standardization bodies (e.g., IETF), and research bodies (e.g., IRTF).

This project contributed to the work of the IPFIX WG of the IETF.

2.3.6 Internet Next Generation (ING)

The ING project [71] has been performed at the University of Twente (Centre for Telematics and Information Technology) in collaboration with the Telematica Instituut, Ericsson Eurolab Netherlands and KPN Research. The main goals of the project are:

- "to increase the Dutch contribution to the international development of new technologies for transferring and managing Internet traffic; and
- to strengthen the knowledge within the Netherlands on new Internet technologies."

This project studied (i) the development and introduction of QoS mechanisms for the Internet, (ii) the development of new Internet management architectures, (iii) service differentiation, (iv) QoS over wireless and mobile access networks, and (v) Internet accounting.

The work on accounting focused on the development of two architectures: a reverse charging architecture and a provider based accounting architecture (PBAarch). We note that the term accounting is used in this project as defined in Section 2.1.

Charging schemes on the Internet are limited since they do not allow ISPs to charge customers of other ISPs for data that is transferred to those customers. WU5 developed therefore an architecture for reverse charging in the Internet that makes possible such charging [71]. Reverse charging enables new business scenarios for traffic flows in the Internet. Take, for instance, the case of a visual arts college, where students produce lengthy videos within a specific project. These videos are available online, but the college does not want to pay for the traffic generated by viewers, because it has a usage-based contract for

Internet services. In this situation, the reverse charging architecture allows viewers to pay for the costs made by the college for sending the video over the Internet, so the college does not have to pay for the video downloads.

The provider based accounting architecture has a distributed functionality and an innovative view with respect to the payment system that is incorporated [73]. The architecture also enables the outsourcing of the accounting functions. In particular, it is shown how Internet Service Providers can be involved in the billing function.

2.3.7 Anticipating Content Technology Needs (ACTeN)

ACTeN [67] is a 5th Framework European project and is part of the Information Society Technologies project. In this project, 11 partners from 10 countries were involved. The goal of this project was "to build an enlarged business and industry community in the area of multimedia technologies and e-content applications and tools." This project did not only focus on e-content marketing (which requires accounting), but also considered the e-content production in Europe. The project ran from September 2002 until August 2004.

A project report discussed the issue of paid content: history of paid content, characteristics of paid content, categories of content, content technologies and developments in the field of content [68]. This report concludes that paid content is only a temporary issue and paid content services are increasing in quantity. This growth will continue ahead of economic indicators if proper payment and micropayment systems and digital rights management solutions are available.

2.3.8 An Open Personalized Electronic Information Commerce System (Opelix)

Opelix [69] is a 5th Framework European project and is part of the Information Society Technologies project. The objective of this project was the development of an "e-commerce solution that delivers content and services to customers based on their preferences and actions." The project ran from January 2000 until February 2002.

Within Opelix a new business model was developed for information commerce. Additionally, tools were developed, which allow the creation of personalized information considering copyright protection, certification of data, timely delivery of data and payment schemes to pay for the information.

Content (or business offer) transactions are split into two main phases: negotiation and execution of business contracts. The first phase contains three processes: searching, matching and negotiating for content. The second phase contains four processes: the mediation of business offers from several sources, authentication of user, delivery of content, and performing the payment transaction. For these processes, a business offer description language was developed as well.

With respect to the payment process, the project considered payment from a high-level point of view, i.e., it addressed only a superset of conceptual payment models (including pay-per-use, flat fees) and assumed that the used payment system (e.g., credit card systems, bonus point systems, micropayment systems) performs secure payment transactions [70].

2.4 Commercial products and platforms

A large number of accounting related commercial products and platforms were developed and deployed. These products and platforms implement different accounting functions or provide a complete accounting solution. For instance, the metering function is implemented in NetFlow, XACCTusage, LFAP; the billing and payment function is implemented in iBill, NetToll, PayTrust; complete accounting services are provided by iMode, Teleknowledge's Total-e Platform. The commercial products and platforms discussed in this section were selected because they introduced innovative ideas and solutions in the field of accounting, or have a significant market share. Because this thesis mainly focuses on the payment function within accounting, electronic payment systems that are used on the Internet are separately discussed in Chapter 3.

The following products and platforms are presented in this section: NetFlow, XACCTusage, iBill, iMode, NetToll. Descriptions of several other products and platforms can be found in [73].

2.4.1 NetFlow

NetFlow [74] is proprietary metering software developed by Cisco Systems, Inc. NetFlow is part of the Cisco IOS software package and provides the metering function for traffic accounting. The metering data can also be used for network monitoring and planning, application monitoring, Denial of Service monitoring, etc. The metering can be performed for customers, merchants, or ISPs. The data export format of version 9 of this product has been chosen by the IETF as basis for the IPFIX standard.

NetFlow uses the concept of network flow, i.e., a unidirectional sequence of packets between given source and endpoints. During metering only the first packet of a flow needs to be fully processed, the subsequent packets are recognized as being part of the same flow. The quantities of flows that NetFlow can meter are for instance, IP addresses, packet and byte counts, time stamps, Type of Service and application port numbers.

2.4.2 XACCTusage

XACCTusage [75] is a commercial software product developed by XACCT Technologies. This product can meter the network resource consumption of customers in several network elements such as routers, switches, and firewalls; it can collect this data and transform it subsequently into charging and billing information. Therefore, network service providers or ISPs can use XACCTusage for transport accounting. XACCTusage can also be used for network monitoring, and can be connected to any standards-based network management system.

XACCTusage uses the concept of gatherers, i.e., agents that implement the data collecting and storing function. Gatherers collect and store metering data from one or more network elements. Data is stored in relational databases such as Oracle, Microsoft SQL Server and Sybase SQL Server. Gatherers also aggregate and filter the collected data. This data can also be used for traffic engineering, fraud prevention, etc.

2.4.3 iBill

iBill [76] is a billing and payment platform developed by the Internet Billing Company, Ltd. This platform provides the billing and payment function to merchants who sell content, products and services on the Internet. Merchants need to have an iBill account and provide the charging information to iBill; the billing and payment function is outsourced to iBill.

iBill can handle one-time billing or recurring (e.g., monthly) billing of customers. The payment systems that can be used to pay a bill are credit card systems, electronic check systems. After a payment transaction was successfully completed, iBill sends a receipt to the customer and notifies the merchant to deliver the paid products.

2.4.4 NetToll

NetToll is a platform developed by Enition, which stopped its activities. Merchants define the price of their content and charge their customers, while the billing and payment function is outsourced to NetToll. The innovative idea of NetToll was that it also allowed ISPs of the customers to pay for the content received by these customers. This could be done by placing tokens (i.e., a type of electronic money) in the IP-packets. These tokens could be obtained by a NetToll server. Later the ISPs could aggregate the payments made on behalf of a customer, and present them on the monthly bill. For this, the ISP creates so-called toll detail records, which contain the identification information of a customer, value of payment, piece of content being paid, etc.

2.4.5 iMode

iMode [77] is a platform that provides wireless Internet access and services. iMode is developed by NTT DoCoMo, which is a major Japanese telephone operator and ISP. Nowadays, iMode is also available in many European countries.

iMode has a very large customer base and many merchants. The ISP provides a content accounting system to affiliated merchants that publish chargeable content. The ISP charges customers for their network traffic based on subscrip-

tions and per-packet-fees. A central billing relationship exists among the ISP, merchants and customers. According to this relationship, content related payments are collected by the ISP and distributed to merchants. This means that the ISP performs transport accounting for customers, and provides content accounting for merchants. It also aggregates traffic and content related charges of the same customer into a single bill. The ISP pays only 91% of the content related payments to the appropriate merchants, because 9% of the content sales are the fee for providing content accounting.

2.5 Comparison

Table 2.1 presents an overview of the accounting work presented in the previous sections. Each row contains a standardization organization, research project, commercial product or platform, and its accounting definition (marked by X), unless they focused only on one specific function (marked by X*). The last column shows the type of accounting work: transport (T), product (P) or both.

Table 2.1 *Accounting overview*

	Metering	Data collecting and storing	Pricing	Charging	Billing and payment	Type of accounting (T=transport, P=product)
OSI RM	**X**	**X**	**X**	**X**	**X**	T
RADIUS WG	X					T
AAA WG		X				T
RTFM WG	X*					T
IPFIX WG		X				T
IOTP WG					X*	P
AAAArch RG		X				T
IPDR		X				P
MPM WG					X*	T+P
ITU-T	X	X	X	X	X	T
IFX Forum					X*	T+P
GigaABP		X				P
CATI		X				T
M3I		X				T
SUSIE		X				T
IP-QoS		X				T
ING	X	X	X	X	X	T+P
ACTeN					X*	P
Opelix					X*	P
NetFLow	X*					T
XACCTusage	X	X	X	X	X	T
iBill					X*	T+P
NetToll					X*	P
iMode	X	X	X	X	X	T+P
X = definition of accounting			X* = focus on only a specific accounting function			

2.6 Conclusions

Accounting on the Internet is needed to transform a technically focused network to an economically controlled, efficient global network [78]. Accounting systems support the cost recovery of ISPs or network operators (that provide commercial Internet services) and the selling of products and services on the Internet. Accounting is considered by standardization bodies, research bodies and projects, and commercial organizations.

The first conclusion of the accounting overview is that the term accounting has many different interpretations. Mostly it is used as a synonym for only one function (see Table 2.1). The accounting functions are not always investigated or implemented together, i.e., some activities focus only on one or another accounting function (e.g., the AAA WG mainly focuses on the data collecting and storing function, the Micropayment Markup WG investigated only the online transfer of small amounts of money, iBill implemented only the billing and payment function).

Accounting is divided into accounting for network resource usage (transport accounting), and for product or service usage (product accounting). Concurrent accounting for using both resource types is also possible, but only in particular situations (e.g., iMode).

The second conclusion is that most of the accounting work focuses on transport accounting (Table 2.1). An interesting fact is that almost every research project considered transport accounting for different QoS mechanisms. An explanation of this fact is that DiffServ was thought to be widely deployed, and the introduction of different QoS classes required accounting. Product accounting received less attention. Generally, this work is performed in research projects, which propose accounting architectures and protocols for products and services. Currently there is no standardization proposal for product accounting architectures. Standardization efforts are only visible for the billing and payment function. The challenges for product accounting are the definition of metering strategies, the specification of a standard way for storing metering data, and the support of micropayment, credit card, debit card systems, etc. [79].

The third conclusion is that one or more accounting functions can be outsourced to third parties. Accounting functions that need certain expertise (e.g., the billing and payment function) are often outsourced to specialized third parties. For instance, iBill provides the billing and payment function for merchants, NTT Docomo provides a complete content accounting system to all affiliated content providers. In case of outsourcing, data produced by the accounting functions implemented by different organizations will typically cross the administrative boundaries of these organizations [79]. That is why the standardization of the data exchange and interfaces between the accounting functions are fundamental.

An interesting observation is that ISPs can implement various accounting functions. ISPs can play an important role in content accounting (e.g., iMode) or can provide the billing and payment function (e.g., NetToll).

2.7 References

[1] Vissers, C.A. et al., The architectural design of distributed systems, Lecture notes for The Design of Telematics Systems course at the University of Twente, Enschede, November 2000
[2] IETF, http://www.ietf.org
[3] IRTF, http://www.irtf.org
[4] IPDR Organization, http://www.ipdr.org
[5] W3C, http://www.w3c.org
[6] International Standardisation Organisation, http://www.iso.org
[7] IFX Forum, http://www.ifxforum.org
[8] ISO, Open Systems Interconnection, Basic reference model: Part 4: Management framework, ISO 7498-4, Geneva, 1989
[9] Stiller, B. et al., Charging and accounting for Integrated Internet Services - State of the art, problems, and trends -, In proceedings of INET '98, Geneva, July 1998
[10] Stiller, B. et al., Pre-study on customer care, accounting, charging, billing, and pricing, Competence Center for Information and Communication of the Swiss Federal Institute of Technology Zürich, Switzerland, February 1998
[11] Hille, S. et al., Taxonomy of accounting, billing and payment concepts, Deliverable 1.2 of the GigaABP project of the Telematics Institute, TI/RS/2000/021, Enschede, April 2000

[12] Shapiro, C. and Varian, H.R., Information rules. A strategic guide to the network economy, Harvard Business Press, Boston, 1999
[13] Hille, S. et al., A flexible architecture for inter-domain accounting, billing and payment, Deliverable 1.1 of the GigaABP project of the Telematics Institute, TI/RS/2001/083, Enschede, January 2002
[14] Hille, S. et al., State of the art in electronic accounting, billing and payments, Deliverable 1.1 of the GigaABP project of the Telematics Institute, TI/RS/2000/020, Enschede, April 2000
[15] International Telecommunication Union, http://www.itu.int/home/
[16] Internet Society, http://www.isoc.org
[17] Malkin, G., The Tao of IETF; A Guide for New Attendees of the Internet Engineering Task Force, RFC 1539, October 1993
[18] Rigney, C. et al., Remote Authentication Dial In User Service (RADIUS), RFC 2856, IETF, June 2000
[19] Rigney, C., RADIUS Accounting, RFC 2866, IETF, June 2000
[20] AAA WG, http://www.ietf.org/html.charters/aaa-charter.html
[21] Aboba, B. et al., Introduction to Accounting Management, RFC 2975, IETF, October 2000
[22] Brownlee, N. and Blount, A., Accounting Attributes and Record Formats, RFC 2924, IETF, September 2000
[23] Aboba, B. et al., Criteria for Evaluating AAA Protocols for Network Access, RFC 2989, IETF, November 2000
[24] Calhoun, P.R. et al., Diameter Base Protocol, RFC 3588, IETF, November 2001
[25] Metz, C., AAA Protocols: Authentication, Authorization and Accounting for the Internet, IEEE Internet Computing Online, November-December 1999
[26] Bailey, C.C., High Interest Subject:Internet Protocol Over Wireless, ITU-T Global Standards Collaboration Meeting nr. 10, August 2005
[27] 3rd Generation Partnership Project, Technical Specification Group Service and System Aspects, Diameter charging applications, TS 32.299 v.6.3.0, June 2005
[28] RTFM, http://www2.auckland.ac.nz/net//Internet/rtfm/
[29] Brownlee, N. et al., Traffic Flow Measurement: Architecture, RFC 2063, IETF, October 1999
[30] Brownlee, N., Traffic Flow Measurement: Meter MIB, RFC 2064, IETF, October 1999
[31] Brownlee, N., SRL: A Language for Describing Traffic Flows and Specifying Actions for Flow Groups, RFC 2723, IETF, October 1999

[32] Brownlee, N., Traffic Flow Measurement: Experiences with NeTraMet, RFC 2123, IETF, March 1997
[33] IPFIX WG, http://www.ietf.org/html.charters/ipfix-charter.html
[34] IOTP WG, http://www.ietf.org/html.charters/trade-charter.html
[35] Burdett, D., Internet Open Trading Protocol Version 1.0, RFC 2801, IETF, April 2000
[36] Eastlake, D. and Goldstein, T., ECML v1.1: Field Specifications for E-Commerce, RFC 3106, IETF, April 2001
[37] Böhle, K., Integration of electronic payment systems into B2C Internet commerce, Background paper nr. 8, Electronic payment systems observatory, April 2002
[38] AAAarch RG, http://www.irtf.org/charters/aaaarch.html, http://www.aaaarch.org/
[39] de Laat, C. et al., Generic AAA Architecture, RFC 2903, IETF, August 2000
[40] Zseby, T. et al., Policy-based Accounting, RFC 3334, IRTF, October 2002
[41] Network Data Management - Usage (NDM-U) for IP-Based Services v3.1.1, IPDR Organization, October 2002
[42] Micro Payment Transfer Protocol (MPTP) Version 0.1, W3C, November 1995
[43] Michel, T., Common Markup for micropayment per-fee-links, W3C, August 1999
[44] Information processing systems - Open Systems Interconnection - Basic Reference Model, ISO 7498, Geneva, 1984
[45] OSI Management Framework, ISO/IEC 7498-4, 1989
[46] ISO, Information processing systems - Open Systems Interconnection - Systems Management Overview, ISO 10040, Geneva, 1992
[47] Pras, A., Network Management Architectures, Ph.D. Thesis, University of Twente, 1995
[48] OSI System Management: Accounting Meter Function, ISO/IEC 10164-10, 1995
[49] ITU-T, Recommendation M.3010: Principles for a Telecommunication Management Network, Geneva, 1992
[50] ITU-T, Recommendation M.60: Maintainance terminology and definitions, Geneva 1993
[51] ITU-T, Recommendation M.3400, TMN management functions, Geneva, 1992
[52] IFX Forum, Interactive Financial Exchange Business Message Specification version 1.5, January 2004

[53] Giga Accounting, Billing and Payment, http://www.telin.nl/Middleware/GIGAABP/ENindex.htm
[54] Hille, S. et al., A functional architecture for the financial exploitation of network based services, Deliverable 2.1 of the GigaABP project of the Telematics Institute, TI/RS/2000/132, Enschede, January 2001
[55] Jonkers, H. et al., Metering and reporting application usage, Deliverable 1.2 of the GigaABP project of the Telematics Institute, TI/RS/2001/084, Enschede, January 2002
[56] Hille, S. and v.d. Stappen, P., Electronic payments put in context, Deliverable 0.1b of the GigaABP project of the Telematics Institute, TI/RS/2001/081, Enschede, March 2002
[57] Charging and Accounting Technologies for the Internet, http://www.tik.ee.ethz.ch/~cati/
[58] Stiller, B., CATI - Charging and Accounting Technology for the Internet, Public Final Report, Zurich, July 2000
[59] M3I, http://www.m3i.org
[60] SUSIE project, http://www.fokus.gmd.de/research/cc/glone/projects/susie/content.html
[61] ACTS programme, http://www.cordis.lu/acts/home.html
[62] Carle, G. et al., Premium IP services, Deliverable AC320/SUSIE/WP2/N/R/004/B1 of the SUSIE Project, April 1999
[63] TINA: Telecommunications Information Networking Architecture, http://www.tinac.com
[64] Chen, C. et al., Billing service for TINA business model, Paper in the Proceedings of SATNAC 2000, ISBN 0-620-26494
[65] IP-QoS - Internet Quality Measurement and Accounting, http://www.fokus.gmd.de/research//cc/glone/projects/ipqos/content.html
[66] Zseby, T. et al., Internet Quality Measurements and Accounting (IP-QoS), Deliverable 1 of the FhG IITB/Fokus Cooperation project, December 2001
[67] ACTeN, http://www.acten.net
[68] Boumans, J., Paid content, E-Content report of ACTeN, January 2004
[69] Opelix, http://www.ipsi.fraunhofer.de/oasys/projects/opelix/index_e.html
[70] Hauswirth, M. et al., OPELIX: a model and system for information commerce, Technical report TUV-1841-2001-10, Technical University of Vienna, September 2001
[71] Internet Next Generations, http://ing.ctit.utwente.nl
[72] Sprenkels, R. et al., An Architecture for Reverse Charging in the Internet, Proceedings of the IEEE IPOM Workshop, Cracow, September 2000

[73] Párhonyi, R. et al., A Provider Based Accounting Architecture, Proceedings of the IEEE IPOM Workshop, Dallas, October 2002
[74] Cisco NetFlow, http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml
[75] XACCTusage, http://www.xacct.com/PRODUCTS/xacctusage/
[76] iBill, http://www.ibill.com/services/ibillcomplete/default.cfm
[77] iMode, http://www.nttdocomo.co.jp/english/p_s/imode/
[78] Stiller, B. et al., Charging and accounting for Integrated Internet Services - State of the art, problems, and trends, Paper in the Proceedings of INET '98, Geneva, July 1998
[79] Koutsopoulou, M. et al., Charging, accounting and billing management schemes in mobile telecommunication networks and the Internet, IEEE Communications Surveys and Tutorials, January 2004

Chapter 3

Overview of electronic payment systems

The objective of this thesis is to ensure the interoperability between existing payment systems by introducing a Payment Gateway that interconnects these systems into a hybrid payment system. Before the design of this hybrid payment system can be discussed, it is important to get a good understanding of existing payment systems. The purpose of this chapter is to give a bottom-up overview of such payment systems and to identify the fundamental differences between them.

Because the terminology related to payment systems varies in literature, first a consistent terminology will be defined in Section 3.1. We will then use this terminology in a consistent way in the subsequent chapters of this thesis.

To understand existing payment systems, one should know how to analyze them. To guide and ease this analysis, first a characterization model is built in Section 3.2. This model contains two groups of characteristics: business roles and functional characteristics. In the first group the business roles within a payment system are identified. The functional characteristics are of special interest because the functionality of existing payment systems will influence the design of the Payment Gateway.

This chapter concentrates on electronic payment systems that can be used for online product payments in a Consumer-to-Business environment. Section 3.3

presents the evolution and classification of such systems. Based on the characterization model, Section 3.4 analyses representative micropayment systems, which are candidates to be interconnected by the Payment Gateway. Actually, for this research, more systems were studied than those presented in this chapter. Section 3.5 presents the summary of the business roles and functional characteristics of the studied micropayment systems. Finally, Section 3.6 presents the conclusions of this chapter.

3.1 Terminology

Once people engaged in trading exchanged goods (e.g., cattle, grain) with each other, without using any form of money. Nevertheless, barter had several problems, among which the most important was the time constraint. For instance, if somebody wished to trade cattle for wheat, the barter could only take place if both the cattle and wheat were in the same place at the same time, which may be a very short period of time. If other payment means would be available, then cattle could be sold at the "best time" for this means. When the harvest was gathered, wheat could be bought using the received means. This lead to the discovery and introduction of money to be used as payment means. The first form of money was precious metal coins, which were minted from bronze, copper, silver and gold (e.g., 600-300 B.C. in China). These coins were used as a means to make payments, and their value varied according to the weight and purity of the coins. The first paper money was printed in China between 806 and 821 due to copper shortage, and abandoned later around 1455 due to a major inflation. On the American continent paper money appeared in the Massachusetts Bay Colony in 1690. In Europe paper money was issued for the first time in France in 1716. A new development came in the early 1960s, when the Bank of America introduced a new payment instrument, namely the credit card. Another new development came in 1995, when Mondex introduced in various trials the first form of electronic money. In the same year, the Mark Twain Bank deployed DigiCash, an electronic payment system that introduced an anonymous form of electronic money and was developed by D. Chaum [1].

Banking activities, however, were already carried out before any form of money was introduced. Banking originates from Mesopotamia, where between 3000 and 2000 B.C. people deposited in temples and palaces (considered safe places) originally grains, and then later cattle and pieces of precious metals.

The Code of Hammurabi, ruler of Babylon, included laws that regulated banking operations. People of ancient Greece and Roman Empire have founded banks as well. Later, in the middle ages more banks were founded, e.g., Bank of Barcelona in 1401, Bank of St. George in Genoa in 1407 and Fuggers Bank in Augsburg in 1478. England's Royal Exchange was built in the 16th century; it showed the importance of the money business and supported foreign currency exchanges. The first banks in the Netherlands were found in Middelburg and Delft in 1616 and 1621, respectively. The banking activities were extended through time, for instance, Pennsylvania Land Bank started issuing notes in 1723 and the Royal Bank of Scotland allowed certain applicants to loan money. Most banks provided sufficient capital to their founders to be able to trade, run a factory, exploit a mine, etc. The first clearing house was set up by two private bankers in London in 1770.

The relationship of payment related terminology is depicted in Figure 3.1. The following subsections define these terms.

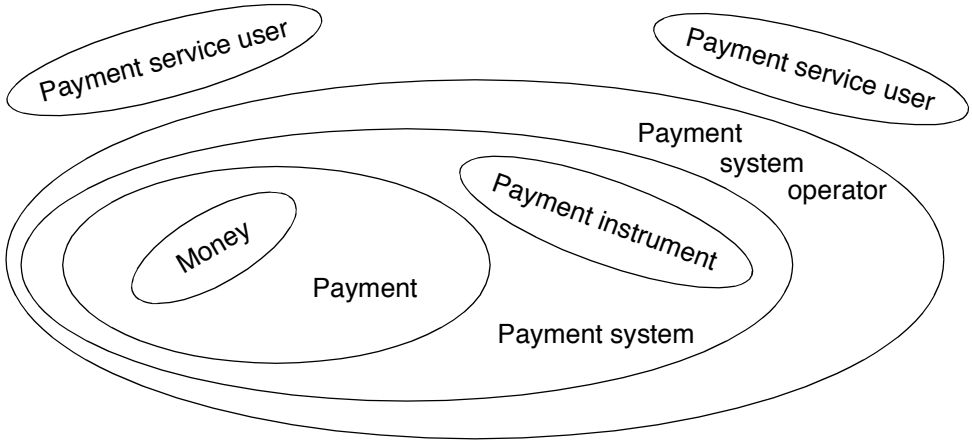


Figure 3.1 *Relation of payment related terminology*

3.1.1 Money

Money is an asset accepted by general consent as a medium of exchange [3].

According to the European Central Bank, money performs three different functions. First, money is a unit of account, i.e., monetary units are used to measure

the value of goods and services. These values are then easily compared. Second, money is a means of payment. For instance, if money is paid for goods, then the paid money can be reused to pay for other goods. Third, money is a store of value, e.g., money can express wealth. This value is defined as a monetary unit and a number (e.g., €10).

Three types of money exist: cash, electronic money (e-money) and bank money [4], [5]. Figure 3.2 depicts the different types of money in a class-diagram. The most abstract concept depicted in the top box is the money; other boxes represent a more specialized type of money.

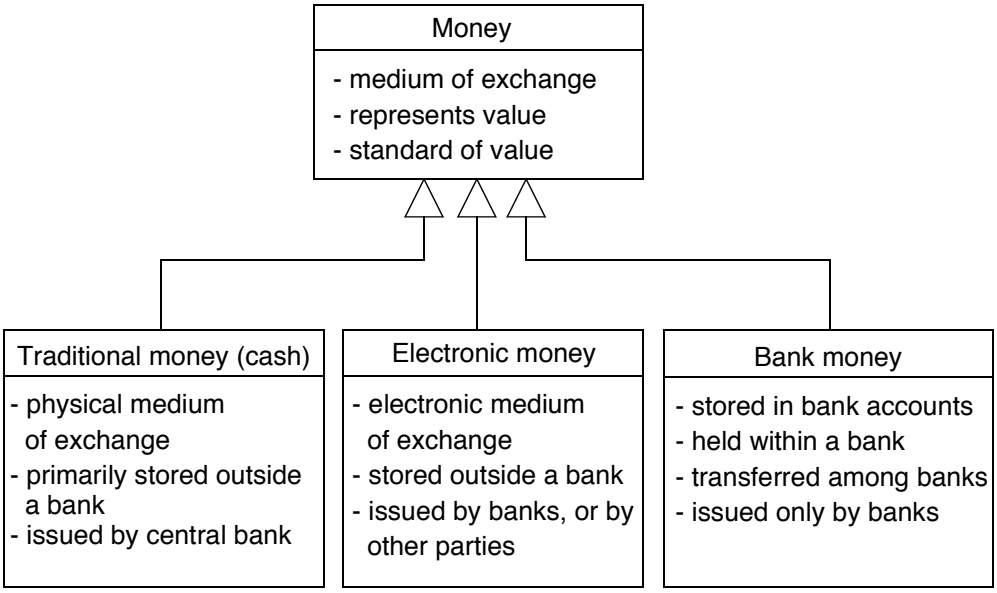


Figure 3.2 *Different types of money and their relationship*

Cash or bank money can be exchanged or converted into electronic money. For instance, Automated Teller Machines (ATM) convert bank money into cash and smart card loading devices can turn bank money into electronic money.

Traditional money (cash)

Cash is *traditional money*, i.e., coins and bank notes. Usually it is created by a national central bank. It is held mostly outside a bank and exchanged in the physical world.

Cash has wide penetration and acceptability, is anonymous, has unlinkability (i.e., separate payments made at a merchant can be linked to each other) and untraceability features, and cash payments are guaranteed (i.e., no risks are involved). From the point of view of customers, cash seems to be for free.

Electronic money

Electronic money is an electronic medium of exchange in an electronic environment [2]. It is issued by banks, but also by other independent and recognized electronic money institutes. In general, the issuer is subject for supervision by financial authorities. Electronic money is stored and exchanged using electronic devices (e.g., computer) and instruments (e.g., smart card).

In the definition of the European Central Bank, e-money is "an electronic store of monetary value on a technical device that may be widely used for making payments to undertakings other than the issuer without necessarily involving bank accounts in the transactions" [3].

E-money differs in many ways from traditional cash. One aspect is the security. Cash is secured by adding particular physical features to it (e.g., special prints and watermarks), which allow on the spot the verification of this money. E-money is secured with cryptographic codes and digital signatures. A second aspect is the way they are exchanged. Cash is physically exchanged in the form of bank notes and coins, while e-money is exchanged in the form of bits and bytes. A third aspect is that cash can be used to make multiple payments, while e-money can be used to make only a single payment.

Bank money

Bank money is immaterial money, which is issued by banks, stored in bank accounts and exchanged between banks. Two types of bank money are distinguished: central bank money and commercial bank money.

Central bank money is monetary value stored in accounts that commercial banks and financial institutions have at a central bank. Commercial bank money is monetary value created when commercial banks offer credit. In a given currency, both bank monies types are convertible into each other at par.

We note that, the safety of both central and commercial bank money is determined by the ability of the central bank to sustain price stability.

More information on bank money can be found in [24].

3.1.2 Payments

In literature and dictionaries, we found two definitions of the term "payment": it is a sum of money paid or it is the act of paying money. In the context of this work the second definition is used, so a payment is a transfer of value (or monetary claim) from a payer to a payee [5], [7], [8].

The payer and payee are the two parties usually involved in payments. Payments are performed based on the initiative of one or both parties.

Cash payments

Cash payments are payments performed using cash.

Cash payments are estimated to 80% of the volume of all payments in 1997, especially to buy low value goods (average payment value was US$11, [20]). In various countries statistics for cash payments differ, however. In the USA, cash payments accounted for 54,2% of all payments in 1997 [33], while in the UK, 72-75% of all payments used cash in 2001 (average value of cash payments in UK was around €15,76 in 2000) [21]. In the Netherlands 85% of all payments were made with cash in 2002, but the volume of cash payments was only 55% of the total volume [22].

Electronic payments

Electronic payments (e-payments) are payments made with value transfers using e-money [8], [16]. E-payments are initiated, processed and acknowledged electronically.

Electronic payments are more limited in volume and value than cash payments. Numbers of the Dutch Central Bank show that in 2002 just 1% of all payments were made with e-money in the Netherlands, which accounts for the 0,20% of

the total volume [22]. The statistics published by the German Central Bank for the year 2003 show that e-money payments are negligible compared to any other type of payments [23].

Electronic payments usually involve value transfers between financial institutions, require that one or both parties (payer and payee), and one or both financial institutions to be specified [26]. A general characteristic of electronic payments is the amount of money being paid. There are situations in which the payer may remain anonymous to the payee. An optional characteristic of electronic payments is the description of the reason or the context in which an amount of money is transferred.

Bank payments

Bank payments are payments performed with bank money. For instance, commercial bank money is used for payments made between accounts stored by the same bank (debiting and crediting of accounts). Commercial bank money is also used in case of credit card payments. Central bank money is used for inter-bank payments.

The volume of bank payments performed in 2002 by the Automated Clearing House (ACH) Network in the USA reached 8,05 billion, while the volume of these payments was valued at US$21,7 trillion. The volume of bank payments performed by the Dutch Clearing House (Interpay) reached €2,96 billion in 2003 [27].

3.1.3 Payment instruments

Payment instruments are the means by which customers pay for services, physical and tangible products [25]. In other words, these instruments enable customers to transfer funds [7]. Generally, the payment instruments have a number, typically an account number, by which the instrument can be identified.

On the payments market various payment instruments can be found because of (i) the diverse nature of payments (e.g., face-to-face, remote, initiated by payer), (ii) the diversity of the involved parties and their relationship (e.g., one-

time payments, recurring payments), and (iii) the variety of payment volumes and values [26].

Payment instruments are generally classified into cash and non-cash payment instruments [26]. The latter class can be divided into checks, debit and credit transfers, and payment cards. Checks instruct the payer's financial institution to debit the account of the payer, and transfer the specified amount to the account of the payee or to pay it out in cash. In case of debit and credit transfers (e.g., giro), the payment is initiated by either the payer or payee, which instructs its financial institution to transfer funds to the payee's account from the payer's account. Payment cards refer to debit and credit cards. Debit cards are usually used at the point of sale for one-time payments, which result into an immediate debit from the payer's account. Credit cards are generally used to make payments against a line of credit established by a financial institution that issued the cards. Both payment cards can also be used to withdraw cash out of ATMs. Other payment instruments that do not fit in the previous classification are, for instance, money orders, wire transfers and travellers' checks.

Figure 3.3 depicts the different types of payment instruments in a class-diagram. The most abstract concept is depicted in the top box, other boxes represent a specific type of instrument.

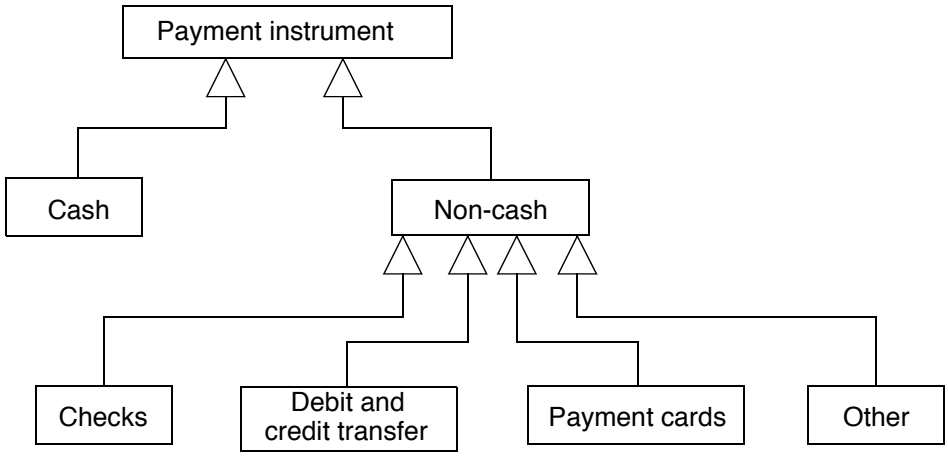


Figure 3.3 *Different types of payment instruments and their relationship*

3.1.4 Payment systems and operators

The Federal Reserve defines a *payment system* as a composition of a legal framework, institutions, people, rules, and technologies [6]. Similarly, the European Central Bank defines a payment system as a composite of "a set of instruments, banking procedures, and typically, inter-bank funds transfer systems, which facilitate the circulation of money" [7]. Bank of England gives a more abstract and simple definition, in which a payment system is "an arrangement that allows the users to transfer money" and which requires:

- agreed technical standards for payment messages and their transmission;
- agreed means of settling claims, which arise among the involved parties;
- a set of common operating procedures and rules [28].

This means that for designing a new payment system these requirements have to be addressed and solved. Such a design, however, is a very complex process, and experts of distinct domains (technical, financial and legal) should be involved. In this technical thesis, we propose a design of the technical part of the hybrid payment system. To avoid introducing new terms to denote this technical part, and to avoid any confusion with respect to the defined terms, we consider a payment system being only a technical system.

An *electronic payment system* is a payment system that makes exclusively use of electronic communication channels (e.g., computer network) to perform payment transactions. Such systems can, for instance, be used to pay for parking, public transportation, vending machines, online content and physical goods. The research presented in this thesis focuses on electronic payment systems that can be used to make payments on the Internet.

Payment system operator (PSO) is an organization or a group of organizations that operate (electronic) payment systems. For instance, a credit card company and affiliated financial institutions (e.g., banks) are operators of a credit card payment system or shortly credit card system.

3.1.5 Payment service and payment service users

The payment service is the external behaviour of a payment system as experienced by the payment service users. In the context of this thesis, two types of service users are distinguished: customers and merchants.

In this thesis, a *customer* is defined as an individual person or organization equipped with an electronic device (e.g., computer, mobile phone, PDA) connected to the Internet that consumes and pays for products requested from merchants. A customer, therefore, comprises two roles: *consumer* and *payer* (Figure 3.4). The consumer wants to spend money, so he/she requests, receives and consumes products, and initiates payments. The payer is part of the payment system, receives the payment initiations and makes sure that money is transferred to the merchant. In literature synonyms for the term "*customer*" are user, end-user, consumer, buyer or purchaser [9], [10], [11].

A customer searches for and selects products before requesting them from a merchant. The products can be consumed in different ways. For instance, one may listen to it (e.g., online radio), view it (e.g., broadcasted sports game or video clip), read it (e.g., scientific articles or information guides), play it (e.g., online games), ride the bus or enter a movie theatre (e.g., e-tickets).

Sometimes a customer is required to register on the web site of a merchant before he/she receives any products. The reason for this is that merchants often like to gather statistical data about their customers, monitor the product preferences of customers or restrict access to certain types of product, etc.

In this thesis, a *merchant* is defined as an individual person or an organization that offers and provides products on the Internet to customers, and is being paid for those products. Therefore, a merchant comprises two roles: *provider* and *payee* (Figure 3.4). The provider receives product requests from consumers, delivers products and receives payment confirmations from the payee. The payee is part of the payment system, receives money and then notifies the provider. Merchants have a B2C relationship with customers. In literature synonyms for the term merchant are seller, retailer or vendor [4], [11].

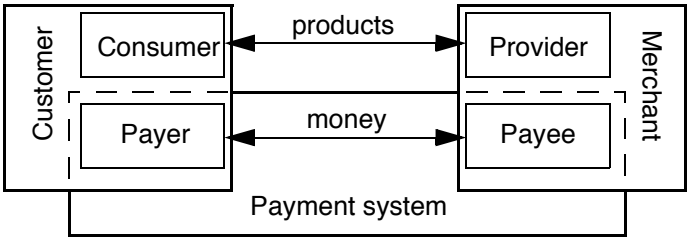


Figure 3.4 *Roles of consumers and merchants*

A content provider is a specific merchant, which offers and sells content on the Internet. It comprises two roles: *content server* and *payee*. The content server receives content requests from consumers and delivers the paid content. We note that, the content offered on the Internet has to be produced first. The content production process includes two sub-processes [12]. In the first sub-process, media assets (e.g., movies, music, pictures, manuscripts, etc.) are created. Then, in the second sub-process, these media assets are transformed into online content, and metadata is added to ease searching, allow digital rights management, etc. These sub-processes may be distributed over several providers (e.g., media producer, digitizer, content producer). However, in the context of this work, only the role of the content server is relevant.

3.2 Characteristics of electronic payment systems

Electronic payment systems can be evaluated based on a number of different criteria. In related literature these systems are classified based on various characteristics such as security, ease of joining and use, pervasiveness, integrity, speed, anonymity, privacy, efficiency, reliability, account-based, token-based, etc. [15], [16], [18].

In this thesis, however, another approach is taken since other criteria are of interest. The next sections define a characterization model for electronic payment systems. First, the business roles that need to be paid in an electronic payment system (e.g., issuer, acquirer) are identified. Second, the functional characteristics (e.g., pre-paid, payment initiations and acknowlegements, supported payment values) are identified.

3.2.1 Business characteristics

On the market of electronic payments a number of actors perform different roles. In order to deploy a payment system these roles have to be played. Figure 3.5 depicts a role model in which various roles (represented by rectangles) have different relationship and interactions with each other [29]. This model was built and validated based on interviews with representatives of banks, financial institutions and payment system operators. The round circle represents the creation point of payments, the arrows represent the exchanged information and money flows. A payment starts when the consumer initiates it and is completed when the provider receives its acknowledgement. After initiation, the payment can only be made if concurrently (i) the consumer and provider are authenticated and identified, (ii) the payer and payee are identified, and (iii) the issuer authorized the e-money transfer from payer to payee.

The *payer* is defined as the buying role of the customer. The *payee* is the selling role of the merchant. In a payment a payer pays an amount of money to a payee.

The *issuer* is defined as the role that contract payers to allow them to use an electronic payment system. The issuer provides the means for making payments to the payers (e.g., issues a payment instrument, electronic money). Generally, the issuer receives money from a payer and issues electronic money of the same value in return. The issuer transfers money via the bank to the acquirer to settle e-payments.

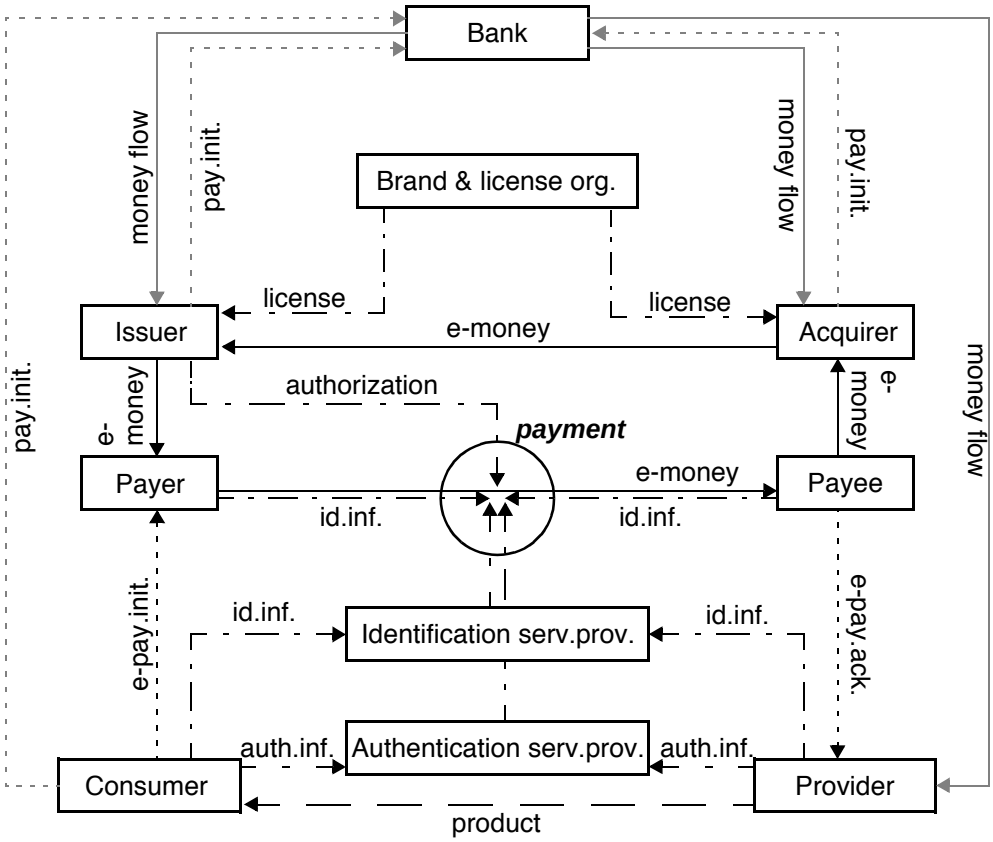
The *acquirer* is defined as the role that contracts payees to allow them to use an electronic payment system. An acquirer holds an account for a payee, and settles the e-payments for that payee.

The *brand and license organization* is the role that defines units of electronic money and associated rules, and then markets them. The brand is the mark, which identifies a particular payment instrument [25]. This organization contracts issuers and acquirers. It also plays the roles of an arbiter (which may be involved in resolving disputes in between the other parties) and of a trusted third party (which is used to enforce trust between the participants).

The *bank* is the role that handles paid money transfers ordered by and on behalf of third parties (e.g., consumer, issuer and acquirer).

The *identification service provider* is the role that distributes identities needed for payments and later recognizes these identities.

The *authentication service provider* is the role that distributes authenticity marks needed for payments and later recognizes these identities. This role includes, for instance, the certification authority, which for certain payment systems, manages the authentication, and symmetric or public key certificates.



pay.init. = money transfer initialization
e-pay.init. = electronic money transfer initialization
e-pay.ack. = electronic payment acknowledgement
auth.inf. = authentication information (e.g., username)
id.inf. = identification information (e.g., password)

Figure 3.5 *Business role model*

Other roles that can be found in electronic payment system and not modelled in the previous figure are:

- The *payment proxy* is the role that contracts payers and payees, and handles money flows between them. Usually the payment proxy has a contract with the acquirer.
- The *transport service provider* is the role that handles the transport of products, payment information and electronic money on behalf of third parties. Usually these parties pay for the transport service.
- The *hardware provider* is the role that provides hardware equipment to all actors present on the market.
- The *software provider* is the role that provides software to all actors present on the market.
- The *product access control service provider* is the role that controls the access to products on behalf of merchants.
- The *attacker* is the role that tries to violate the security and integrity of the system in order to commit fraud [18].

3.2.2 Functional characteristics

Functional characteristics describe the service of a payment system at a high abstraction level. These characteristics consider the interactions between users of the system and the system itself, the conditions that enable these interactions (i.e., allow them to occur), and the information that is exchanged during these interactions. Although there may occur many different interactions between users and the system, for this work only those interactions are of importance that have an end-to-end significance. End-to-end significance means that an interaction that took place between the consumer and the payment system will trigger the occurrence of another interaction between the provider and the system. This can also be valid the other way around: an interaction that occurred between the provider and the system triggers the occurrence of another interaction between the consumer and the system. The interactions that have end-to-end significance are the payment initiations and acknowledgements. It is also relevant which user initiates the payments and which one receives the acknowledgements. For instance, the consumer can interact with the system to initiate a payment, which after completion will be confirmed to

the provider in a different interaction. Other interactions such as registration, login, reviewing of payments history, checking account balance, receive periodically overview of payments) have only local significance for the consumer and provider. These latter interactions are not relevant in studying the interconnection of payment systems, thus they are not considered further. In addition, we abstract from the different ways in which the initiation and acknowledgement of payments can be implemented, because we consider only the results of these interactions important for the interconnection problem. The result of the initiation of a payment is that the payment is accepted by the underlying payment system. The result of an acknowledgement is that a user receives a confirmation that the underlying payment system completed a payment. Generally, an initiation is followed by an acknowledgement. However, due to error situations only an initiation may occur and the payment may not be completed. In these situations, the general rule is that the payer bears the loss of money, but only up to a certain limit, as determined by the European Central Bank [3], for instance. The probability that these situations occur is low since existing payment systems claim high reliability and money loss may only occur rarely (e.g., every one-millionth micropayment fails).

One of the most important pieces of information in the initialization and acknowledgement of a payment are the value and currency of the payment. These characteristics are relevant because range of payments supported by the systems varies from system to system, and can influence the systems' interconnection.

The condition for consumers to use a payment system to pay various providers is to transfer money to the payment system before or after the payments are made. The condition for providers to make use of a payment system is to register. Because this is a general requirement for all providers and the registration hardly differs, we do not have to consider it for the interconnection problem.

We determined the main functional characteristics based on information available on the web sites of these systems, implementation documents, performing payments with the systems, inspecting the source code of provider web sites, and capturing network traffic using Ethereal.

Payment initiations

Payment systems perform payments in the case one or both users initiated the payments and provided them adequate payment information.

Generally, the accounting systems used by merchants generate the largest part of the payment information, because these systems determine the costs (i.e., price of the products) a customer should pay. The generated information usually contains a unique merchant identifier, which that is known by the payment system, and that identifies a destination account where money should be transferred. The payment information also contains the value and currency of the payment, and a (unique) product transaction identifier that is used by the merchant to identify the product transactions, so the paid products can be delivered. Nevertheless, a customer identifier, which is known to the payment system, and that identifies a source account from which money should be transferred, is also needed for each payment. This information is added to the previously generated payment information.

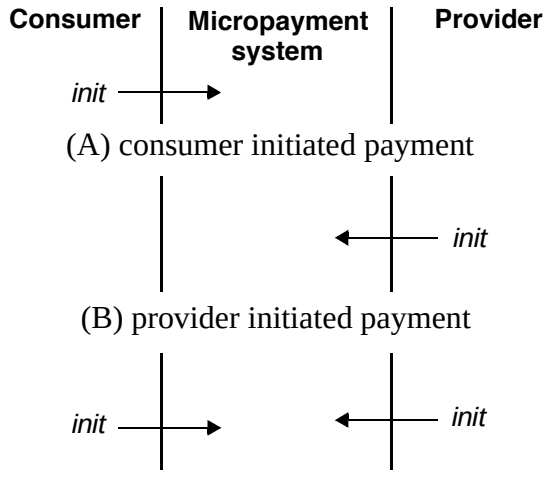


Figure 3.6 *Payment initiations*

We identified three alternative ways to initiate a payment. The vertical lines represent the interaction points or service access points (SAPs) between a micropayment system and its users, arrows represent payment initiations, and the direction of the arrows indicates the direction of the information flow.

First, in Figure 3.6 *A*, the consumer initiates a payment. We call this alterative a "*consumer initiated payment*". Prior to this, the provider provides the payment information to the consumer.

Second, in Figure 3.6 *B*, the provider initiates a payment, and we call this alternative a "*provider initiated payment*". Prior to such an initiation, the provider receives identification information of the consumer.

Third, in Figure 3.6 *C*, both users initiate a payment, which we call a "*jointly initiated payment*". Because both users initiate the same payment, they both have to provide (partial) payment information such that the payment system can correlate the received information, and then make the payment. This means that both users need to interact with each other before the initiation in order to exchange parts of the payment information.

During the analysis of the existing micropayment systems we will group these systems into different categories based on the type of payment initiation, and will also study the payment information needed by these systems to initiate a payment. An overview is presented in Appendix A.

Payment acknowledgements

Completed payments can be acknowledged to the consumers and providers in two ways:

- the payment system provides an acknowledgement to the consumer and/or provider;
- the payment system provides no explicit acknowledgement, instead it retrieves the paid product from the provider, and delivers it to the consumer.

In the first case we can say that the payment system “performs only payments”, while in the second case, the payment system “does more than payments”. Confirmation information of completed payments is only provided in the first case and it differs from system to system.

Three possibilities exist to acknowledge payments. First, in Figure 3.7 *A*, it is acknowledged only the consumer. We call this alternative a "*consumer acknowledged payment*". The consumer will then send further the received confirmation information to the provider in order to get the paid product.

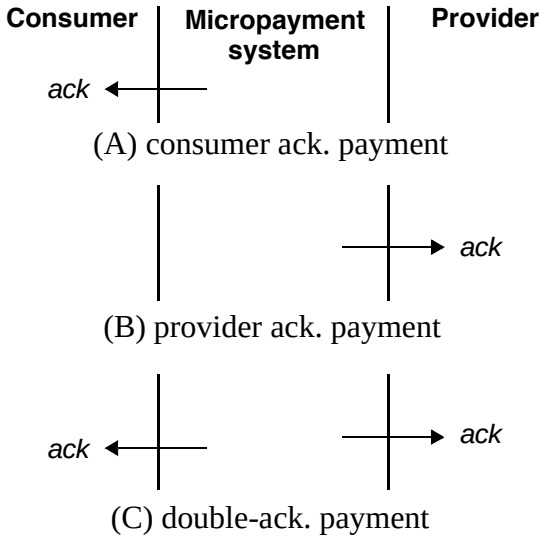


Figure 3.7 *Payment acknowledgements*

Second, in Figure 3.7 *B*, only the provider receives an acknowledgement. We call this a "*provider acknowledged payment*". The provider sends the received confirmation information to the consumer and/or delivers the paid product.

Third, in Figure 3.7 *C*, both the consumer and provider receive acknowledgements. We call this a "*double-acknowledged payment*". These acknowledgements may take place concurrently, but there may also be temporal differences between them. Usually no additional interaction is needed between the consumer and provider, and the latter one will deliver the paid product.

Range of payments

The minimum, maximum values and the currency of money transfers are those functional characteristics that define the range of payments accepted by a micropayment system, and that will be of great importance when it comes to payment systems interconnection. The minimum payment value supported by the payment systems may differ and this may create incompatibility between some systems. The maximum payment value creates fewer problems, because a bigger payment may be broken up into smaller payments. Generally, the payment systems support only one major currency (e.g., US$ or €), but there are a few others, which support multiple currencies.

Usage conditions

The conditions for consumers to make use of a payment system depend on the type of the system. Micropayment systems can be divided into two basic types:

those where no credit occurs or "pre-paid" system, and those where credit does occur or "post-paid" systems [30].

A non-credit system requires consumers to transfer money to the system before they can initiate payments. The amount of money transferred to the system will be stored in form of electronic money. The consumer receives authorization that electronic money is available and can be immediately used to pay products. Such a system is called a pre-paid system (Figure 3.8, [30]).

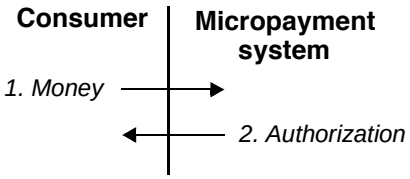


Figure 3.8 *Pre-paid system*

A credit system authorizes consumers to initiate payments before they transfer money to the system. Before consumers receive such an authorization need to present the system a (reliable) money source from which the amounts of money credited over a certain time period can be transferred to the system. If the money source is valid, the consumer receives the authorization to pay the providers. These payment systems are called post-paid systems (Figure 3.9, [30]).

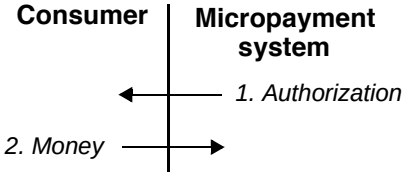


Figure 3.9 *Post-paid system*

3.3 Evolution and classification of electronic payment systems

3.3.1 Evolution of payment systems

The first generation of Internet payment systems started in 1992 [14]. Credit cards were mostly used for making payments online. At that time, payments that contained credit card numbers were transferred through communication channels without any security measures. In parallel, great efforts had started to develop, implement and deploy new electronic payment systems for use on the Internet. These efforts were driven by a number of factors such as the potential of micropayment systems to support online, electronic payments of low values, the necessity of making anonymous payments, and the need for more secure

payment systems for high value payments. Commercial organizations, the banking sector, universities, and research institutes were involved in this work.

In 1998 began the decline of the first generation of payment systems. Their success was very limited, some were only theoretically defined, others were implemented and tested in pilot schemes, many initiatives failed, and some became standards (e.g., SET for credit card payments). Examples of failed initiatives are FirstVirtual, DigiCash, IBM's Micropay, Compaq's MilliCent. Many attempts failed due to the lack of standards for payment systems, being unable to reach a critical mass (i.e., wide penetration among users or reaching a large number of users and transactions), creating merchant specific currencies, or because of difficulties with subscribing and using the system, or due to high transaction costs and low speed [13], [15].

The second generation of Internet payment systems started in 1999 [14]. These systems include characteristics such as pre-paid accounts, virtual accounts for person-to-person or business-to-person payments, email payments.

Mobile payment systems started to appear in the same period. These systems use other communication networks (e.g., GSM) than the Internet. Mobile payment systems allow payments in both real and virtual worlds, and overlap with the new generation of Internet payment systems. Mobile payment systems could become serious competitors to the current electronic payment systems on the Internet, especially in Western Europe [17].

3.3.2 Classification of electronic payment systems

In literature, different classifications of electronic payment systems can be found. Such classifications are based on the type of e-money, type of payment instruments and value of payments.

A classification distinguishes e-cash and account-based systems (Figure 3.10, [16]). The e-cash systems are split into smart card systems (e.g., Mondex, Chipknip) and online cash systems (e.g., NetCash, ECash). The account based systems are split into generic systems (e.g., NetBill, PayPal), specialised systems, and credit and debit systems (e.g., MasterCard, Visa, Cirrus).

A second classification distinguishes token-based and account-based systems [18]. The first group contains direct cash systems such as ECash, MagicMoney and PayMe. The second group is further divided into direct account systems such as CyberCoin, FSTC and NetBill; credit card systems such as SET, Veri-Fone and First Virtual; and push account systems such as AIMP, CheckFree and NetFare. Other classifications presented in [18] are based on the value of payments (i.e., macro payment, small payment and micropayment systems) and on the payment validation method (i.e., online, semi-online and offline systems).

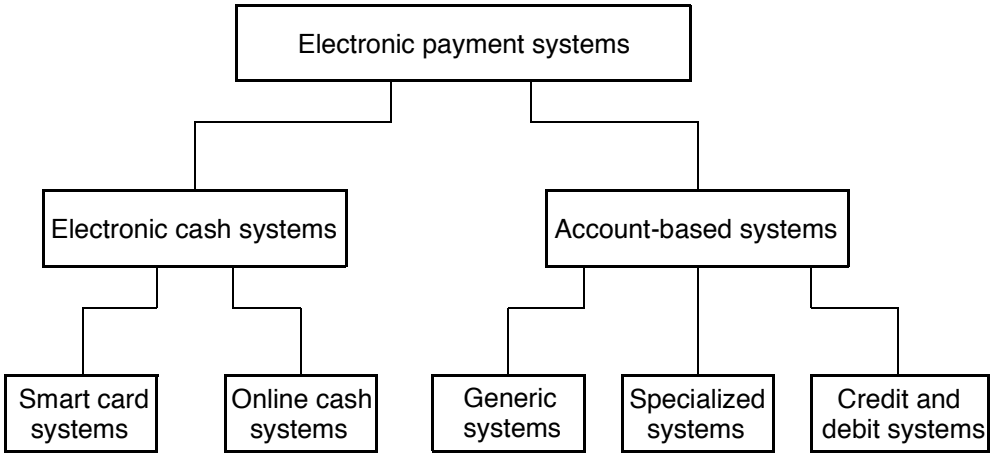


Figure 3.10 *A classification of electronic payment systems*

A third classification makes distinction between credit card, e-check, e-cash and micropayment systems [20]. In the group of credit card systems are included the SET standard, First Virtual, Visa and MasterCard. Examples of e-check systems are NetCheck, ECheck and NetBill. Examples of e-cash systems are ECash, NetCash and CyberCoin. Micropayment systems are, for instance, Wallie, Click&buy, Way2Pay and Bitpass.

In this thesis the following three payment systems are considered and discussed:

- Credit card systems, which are most popular payment system on the Internet;
- Mobile payment systems, which have a great potential for content payments in the future;
- Micropayment systems, which are in the focus in this thesis.

3.3.3 Credit card systems

Online credit card payments make use of the existing credit card payment infrastructure. According to the Encyclopaedia Britannica, credit cards are payment instruments that contain the means to identify a person and authorize this person to charge goods and services to an account, on which this person is billed periodically. Credit cards allow buying on credit, according to the "buy now, pay later" principle.

Credit cards have been used as payment instruments since the early 1960s. At that time, the Bank of America launched a payment system that evolved to the Visa credit card system. By the end of 2001 there were 901 million Visa branded cards issued worldwide [36]. In 2001, MasterCard, another major credit card system had 520 million cards issued worldwide [37].

At the time of writing, the majority of the payments on the Internet (95% of online customer purchases, [38]) are performed using credit cards, making it the most popular payment instrument. A major reason for this is its support for cross border payments. Credit card payments are mostly used to pay for physical goods of a high value. In 2001, the average online transaction was around US$80. In the UK the average value of credit card payments was around €80 in 2000 [21].

Credit cards have several drawbacks. One of them is that credit card payments have a minimum threshold value and a lower value payment cannot be made because of high transaction costs (see Table 1.2). We note that, this threshold has no fixed value. Another disadvantage is that they are not available for everybody (e.g., young people without a stable income or solid financial back-

ground). A third drawback is that online credit card payments have a high fraud risk. The estimated fraudulent online credit card payments will cause a loss of US$2,6 billion in 2004, according to [39], and it will grow up to US$3,6 billion in 2007. Because of the high transaction costs and many fraudulent payments, the number of online merchants who prefer credit card payments slowly declines [40]. As a consequence, many merchants search for alternative payment methods rather than accept credit cards [41].

The most relevant attempts to make secure credit card payments online are:

- **Secure Electronic Transactions** (SET, [42]) was developed by the alliance of Visa International, MasterCard International and technology vendors. SET is a payment protocol that supports only payment cards and relies on the existing credit card infrastructure. SET was considered a practical approach for easy, fast, and secure payments over the Internet [35]. The implementation of the SET technology was unsuccessful, however. SET's failure is mainly caused by the technology's failure to convince the social groups (i.e., private and commercial customers) that it could be implemented without major technical difficulty [43].
- **Secure Payment Application** (SPA, [44]) is an issuer-based authentication mechanism that uses the Universal Cardholder Authentication Field (UCAF) infrastructure of MasterCard. The SPA authenticates the payments of customers and addresses the security of all parties involved in an online payment. SPA can be implemented as an externally hosted service with minimal system impact. SPA does not require the use of PKI, which was one of the negative factors in the failure of SET.
- **3-Domain Secure** ([45], [48]) developed by Visa isolates the responsibilities of the different parties involved in payments. The card issuers have close relationship with card holders (customers), while acquirers have close relationship with merchants. Communication between issuers-card holders and acquirers-merchants takes place during each payment. The three responsibility domains are the Issuer Domain that contains the customers (card holders) and their banks (issuers), the Acquirer Domain: merchants and their banks (acquirers); and the

Interoperability domain: communication between issuing and acquiring parties using Visa's infrastructure.

- **Verified by Visa** is a payment service of Visa that allows customers to pay safe and convenient during online shopping. In this service, the credit card details are protected by a password.

3.3.4 Mobile payment systems

Payments can be made using mobile devices and communication networks. The mobile device and network can play different roles [13]:

- the mobile device as a payment instrument in the physical world: used to make electronic fund transfer for physical goods at the points-of-sale (e.g., paying with mobile phone at gas stations for refuelling a car);
- the mobile device as a payment terminal in the virtual world: used to make electronic transfer for goods delivered over a channel different from the mobile network;
- the mobile network as direct payment channel: electronic products and services are delivered to the mobile device.

Examples of mobile payment systems are:

- **Paiement CB sur mobile** [73] introduced in France is a novel approach to make payments using mobile devices is the combination of payment cards and mobile telephones. These two technologies are well-known by everybody. The idea behind this solution is that mobile handsets can be used as terminals for chip-based payment cards. To enable this payment possibility, mobile devices need to be equipped with a slot and reading device for the card. The French solution uses proprietary (non-European standard) chip-based payment cards [49].
- **Paybox** [74] in Germany and **Moxmo** [75] in The Netherlands and Germany allow payment to be initiated from mobile telephones. These solutions have the advantage of requiring no special device other than the well-known mobile telephone. It is not sure yet whether low payments are also feasible from an economical point of view. The current

operational costs should be reduced to allow such payments to be made [49].

Payments on the Internet using a mobile telephone are currently considered highly promising because customers will be able to pay for products anywhere and any time. Additionally, mobile devices have the potential to facilitate cross-border payments for future payment schemes. The mobile payment service providers will offer the necessary trust that customers require to make the payments, guarantee the payments for merchants, etc. [17]. However, the issues of payment authorization and physical security of the devices need to be solved.

3.4 Micropayment systems

Micropayment systems are potential and efficient alternatives when credit card systems are unable to transfer small amounts of money. The efficiency leads to one definition of micropayment systems:

> *Definition: Micropayment systems are those electronic payment systems that (also) support money transfers, which are smaller than the minimal economically feasible credit card payment.*

Figure 3.11 depicts a micropayment system that maintains long term relationships with the consumers and providers. It is assumed that the relationship between consumers and providers is short-term, irregular or occasional. The micropayment system receives macro payments from consumers. Consumers then make a large number of micropayments to providers. Providers receive pay-outs from the micropayment system in the form of macro payments.

Aggregation is considered to be the key for the micropayment systems: small payments are aggregated until they are settled in a macro payment (e.g., credit card payment, bank transfer) [13], [51].

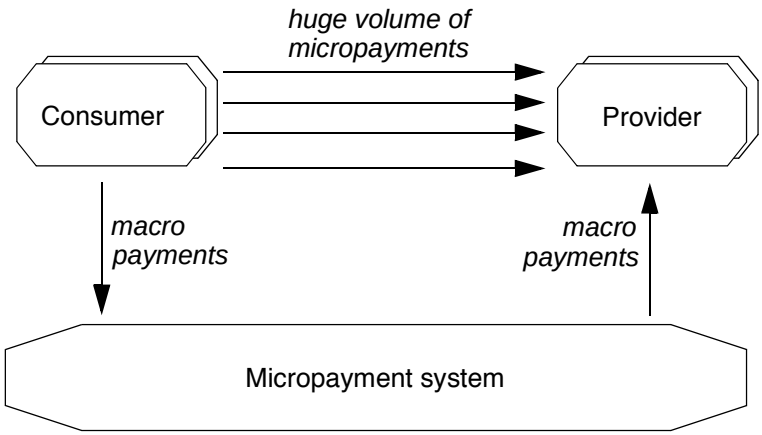


Figure 3.11 *Macro and micropayments*

Micropayment systems have several other advantages over credit card systems. For instance, the risk and scale of loss or fraud for consumers and providers is small. These systems, therefore, do not need features like high security, non-repudiation, money-back guarantee, etc.

We derived other definitions of micropayment systems from the price of products, time differences between completing payments and delivering the paid products.

The consumer price of a product sums up from the production costs, distribution costs, transaction costs and margin. The margin is the profit from the difference between costs and net sales (e.g., 4%). This margin on low value sales will be small. If the margin of such a sale is smaller than the financial transaction costs, the business cannot be profitable. In case of low value payments, the transaction costs can exaggerate the consumer prices. A second definition of micropayment systems is the following:

> *Definition: Micropayment systems are those payment systems that support low value payments at low transaction costs.*

The processing of a micropayment should be fast, generating as little delay as possible because they are convenient during (preferably) anonymous and impulsive product consumption. Consumers expect that low value products are instantly delivered after the payments were completed. Or payments and products delivery can also occur at the same time. Should the provider fail to deliver immediately its products, the consumers can call its helpdesk or customer care

centre. Such a call will cost 10 or 100 times more than the value of the sold products [13]. This situation can affect the provider's long term revenues, can discourage potential consumers and stop returning ones. A third definition of micropayments is derived from this case:

> *Definition: Micropayment systems support small value payments for products with a minimal delay and in exchange the products are instantly delivered*.

In literature, two generations of micropayment system are distinguished [50]. The first generation of micropayment systems began around 1994 and lasted until the end of the 1990's. The developers of these systems primarily aimed at the introduction of the electronic form of cash (called e-cash, e-coins, digital cash or tokens) on the Internet. They focussed on the generation of e-coins or tokens, secure, anonymous and untraceable exchange of them, validation and fraud avoidance. Others developed account-based systems transferring money from customer accounts into merchant accounts similarly to banking systems. Examples of first generation systems are Millicent (developed by Digital Equipment Corporation in 1995), ECash (developed by DigiCash in 1996), MicroMint and PayWord (developed by R.L. Rivest and A. Shamir in 1995-96), SubScrip (developed by Newcastle University, Australia in 1996), NetCash (developed at the University of Southern California in 1996), and *i*KP (developed by IBM in 1997). We also found a few account-based systems: Mondex (developed by MasterCard in 1995), CyberCoin (developed by Cyber-Cash Inc. in 1996), Mini-Pay (developed by A. Herzberg and IBM in 1997).

All first generation systems failed one after the other, stopped after a public trial or remained at a theoretical description level. The most important reasons for their failure are (i) the incapability that these systems are trustworthy, (ii) very low coverage (i.e., the number of customers and merchants using the system) and lack of funding until these systems reached a critical payment volume, (iii) inconvenient usage, (iv) lack of appropriate security mechanisms and (v) lack of anonymity [50].

The second generation (or current) micropayment systems emerged in 1999-2000. These systems are almost without exceptions account-based.

In this thesis, we focus on second generation micropayment systems, which were operational at the time of writing this thesis. The following subsections present a few representative micropayment systems. These payment systems were selected from a report presented to The Dutch Ministry of Economical Affairs [52], payment systems repository of the Electronic Payment Systems Observatory [53], EPayNews.com (a payment news and resource centre, [47]), and the Google directories on payment [54] and micropayment systems [55]. Other systems that are not presented here resemble to a certain extent the selected ones and will be mentioned in the summary (Section 3.5).

3.4.1 Wallie

Wallie [59] is a pre-paid account-based micropayment system introduced in the Netherlands by the Distri Group, the biggest distributor of pre-paid telephone cards. Wallie can only be used in the Netherlands. Wallie includes an issuer that issues pre-paid cards, an acquirer that redeems payments made with the issued cards, an identification service provider that identifies the source and destination accounts of each transaction and a branding organization.

Consumers can buy Wallie cards worth €5, €10, €20, or €50 at one of the 3500 outlet stores. The card contains a pre-paid account number, which consists of 16 digits and 3 letters. The consumer initiates Wallie payments and receives confirmations of the completed payments. Additionally, the consumer can verify the balance of the pre-paid account in a new session by introducing the account number on the web site of Wallie. The history of completed payments is not provided to the consumer, however. The account cannot be used after its balance reached zero. The consumer bears the financial loss, when he/she loses the card. Wallie can be used by consumers for free.

Providers need to register to be able to use the payment service of Wallie. Providers need to set up their web sites to redirect consumers to the payment system. The providers send consumers the payment information (e.g., provider identifier, a product identifier, value of payment expressed in eurocents), which is then sent to Wallie to initiate a payment. After this set up procedure providers will also be able to receive indications from Wallie about the completed payment. Each payment is indicated to the appropriate provider. Providers can also review in real-time the information about completed

payments. Each provider receives periodically (e.g., monthly) a detailed list of all payments, and Wallie transfers the money received from consumers to an account specified during registration. Each provider pays per transaction costs, which are deducted from the total amount of money received by that particular provider. Because Wallie is a pre-paid system, providers do not face the risk of losing moneys.

Wallie payments

Suppose a consumer encounters a content server and wants to buy a piece of content. The consumer decides to pay and clicks on the Wallie icon present on the web site of the content server. With this click, the consumer initiates a Wallie payment. Wallie will open a new window for the consumer to provide his/her identification information. In this window the consumer receives information about the content server and amount of money to be paid, then he/she introduces the account number from the card. If the account number is correct then the consumer is asked to acknowledge the payment. After the consumer gave his/her acknowledgement, the balance of the account is verified by Wallie. If the balance of the account allows the new payment to be made, then Wallie performs the payment and then indicates to both the consumer and content server that the payment was successfully completed. The consumer receives this indication in form of a receipt that can be printed. The receipt contains the content server's name, a unique payment transaction key (generated by Wallie), the date, time, and value of the payment. After that the payment session is automatically closed by Wallie. The content server receives the indication in a so-called "callback-URL". This contains the transaction key (provided to the consumer as well), the content identifier and the value of the payment. After that the content server can send the content to the consumer.

If the balance of the account is lower than the amount to be paid, then the consumer may provide a second account number and open a new session. Maximum 5 account numbers can be used and maximum €150 can be paid in one payment. One acknowledgement is then enough to authorize all payments, and one receipt is received back. After that each session is automatically closed.

In case a consumer did not get the content after it received a confirmation, or the content is other than it was described, the consumer needs to contact the concerned content server and complain. Wallie does not support refunds.

The advantage of this system is that consumers do not have to register and do not have to download and install special software to be able to use Wallie. Additionally, consumers can remain anonymous to providers. A major disadvantage of Wallie is that consumers need to go to a store to buy the cards, online buying of cards is not supported.

Similar systems are, for instance, PaySafeCard [60] introduced in Austria and Germany, Micromoney [61] introduced by Deutsche Telekom in Germany, Centipix [62] that is introduced internationally and uses pre-paid cards in form of JPEG images, Splashplastic [63] in the UK and Mywebcard [64] in Denmark.

3.4.2 Minitix

Minitix [66] is a pre-paid account-based micropayment system introduced in the Netherlands by the RaboBank. Minitix includes a brand organization, an identification and authorization service provider and acts as a payment proxy on behalf of consumers.

Consumers are required to register and provide personal information (e.g., name, address, birthday, gender, identity certificate, email address, etc.) and banking information. All provided information should be correct, otherwise the registration is cancelled. During the registration they also set a user name and password, which will be used later to access the payment system. Consumers also need to agree with the usage terms and conditions. After that, Minitix creates a pre-paid account for consumers who need to deposit money (maximum €100) before they can initiate payments. Consumers can access their accounts by opening a session. During a session they can acknowledge previously initiated payments, review the list of payments, manage their information provided at the registration, or initiate money transfers into their accounts. Consumers can only acknowledge payments if the balance of their account allows those payments. Each completed payment is immediately confirmed to consumers, who will also get a daily email with an overview of their completed

payments for that day. Consumers can request the payment system to terminate the service. In this case Minitix transfers back the unspent amount of money if this is bigger than €1. Consumers that do not use Minitix for more than 3 months will receive back the unspent money and their account will be suspended. The risk of losing money is borne by consumers.

Providers need to register for using Minitix as well. After that they need to set up their web sites to be able to provide payment information to the consumers. Each completed payment is indicated to the appropriate provider, which provides then the paid product. Minitix pays out monthly the providers. At the same time they receive a detailed list of payments. Providers do not run the risk of losing money because Minitix is a pre-paid system. They need to pay a one time entree fee of €180 and then a per transaction fee, which varies between €0,05 and €0,65.

Minitix payments

Suppose a consumer encounters a content server and wants to buy a piece of content. The consumer decides to pay and clicks on the Minitix icon next to the description and price of the selected content. In general, Minitix supports payments between €0,10 and €10, but payments worth €99,99 can also be made under special conditions. The content server provides the payment information to the consumer, which initiates a Minitix payment. Minitix then opens a login window to the consumer. In this window the consumer sees the name of the content server, and the amount to be paid, then he/she introduces the user name and password to access the system. After that Minitix requests the consumer to acknowledge the payment. If the consumer gave his/her consent, Minitix verifies the balance of the account. If the balance allows the new payment, then Minitix performs it. The consumer has no possibility to cancel a payment initiation after it was acknowledged. The consumer and content server receive at the same time a confirmation. Content is delivered after the payment is successfully completed.

If the balance of the account does not allow a payment, the consumer is requested to transfer some money to his/her account, then he/she is requested again to acknowledge the payment.

In case a consumer did not receive the content, or the content is other than it was described, the consumer needs to contact via email the concerned content server and complain. After analysing the complaint, the content server should correct the faulty content delivery.

Similar systems are, for instance, FirePay in the USA, Teletik Safepay [67] and Secoin [68] in the Netherlands.

3.4.3 Click&buy

Firstgate AG introduced a post-paid, account-based micropayment system called Click&buy [65], which is actually a product accounting system in our view. In the beginning the system functioned only in Germany, then it was deployed in several other European countries (e.g., Austria, the Netherlands, France, Spain, Switzerland, UK) and in the USA. Click&buy incorporates a brand organization, an identification and authorization service provider, a payment proxy and a product access control service provider.

Consumers are required to register and provide personal and banking information (e.g., credit card number). All provided information should be correct, otherwise the registration is cancelled. In return, a post-paid account is created for each of them. Consumers can access their accounts by opening a session (i.e., log into the system using a user name-password combination set during the registration). During a session they can acknowledge initiated payments, view their payments, check the balance of the account, and change the information provided at the registration. Consumers can also acknowledge payments to pay for subscriptions. In this case payments will be performed periodically and automatically. Consumers receive periodically (e.g., monthly) an indication that a money transfer took place, which restored the balance of their account. The money transfer is initiated by Click&buy using the banking information provided at the registration. The period can be changed depending on the number and volume of the acknowledged payments. Consumers can end using the system. Before that, however, the balance of their account is restored. Using Click&buy is free for consumers.

Providers also need to register and pay a one time subscription fee for using Click&buy. During the registration the access path (or link), description, price

and availability of the product (measured in time) is provided to Click&buy. Then they need to set up their web sites on which they offer products. Because Click&buy is an accounting system, providers need to register their products, and then protect those products such that only Click&buy will have access to it. In return, the providers receive a premium links for each registered product. These links will be added to their web pages. Products can only be sold in individual units, which means that consumers cannot select multiple products and pay in once. The prices of products vary between €0,05 and €5 because these are the minimum and maximum payment values supported. Providers will be able to see the completed payments in a management environment provided by Click&buy. Click&buy will further handle the payments, retrieve the paid product from the providers and deliver it to the paying consumers. Hence, providers do not receive indications of the successfully completed payments. The money received from consumers is paid out monthly to the providers that also receive a detailed list of completed payments. Because providers need to pay for the payment service, the commission will be deducted from the total amount to be transferred.

Click&buy payments

Suppose a consumer encounters a content server and wants to buy a piece of content. The consumer decides to pay and clicks on the Click&buy icon next to the description and price of that piece of content and initiates a payment. Click&buy opens a login window. In this window the consumer sees the necessary information about the payment, then he/she introduces the user name and password to open a session. After that, Click&buy requests the consumer to acknowledge the payment. Then the payment is performed and Click&buy retrieves the paid content and delivers it to the paying consumer. The consumer does not receive confirmation, instead gains immediately access to the paid content.

In case a consumer did not get the content, or the content is other than it was described, the consumer needs to contact via email the concerned content server or the payment system operator and complain. After analysing the complaint, it is possible to refund the consumer. It is either Click&buy or the content server that initiates the refund.

At the time of writing, 2500 providers and over 3 million consumers used Click&buy. A similar system is WebCent [76] introduced in Germany.

3.4.4 Bitpass

Bitpass [58] is a pre-paid account-based micropayment system developed in 2002 at Stanford University. Bitpass incorporates an issuer of "virtual debit cards" and an acquirer, a brand organization, an identification and authorization service provider, a payment proxy and a software provider.

Consumers need to buy a "virtual debit card" with a specific denomination (e.g., US$3, US$5, US$10). After that they register in order to open a "spender" account. Consumers can register to Bitpass directly from the web site of providers that are already registered to Bitpass. During registration they need to provide information such as email address and the bought card's number. Consumers also need to agree with the usage terms and conditions. The created account will be accessible using the correct email address and password combination set during the registration. Later, consumers are able to use this registration to pay other merchants. Consumers can log into the payment system, acknowledge initiated payments, review the history of their payments, buy more virtual cards and assign them to their account, or change their registration information. Consumers do not pay for using Bitpass.

Providers also need to register first to receive an "earner account", which can be accessed the way consumers access their accounts. Providers also need to agree with the usage terms and conditions. Providers using Bitpass need to follow a set-up procedure and product registration similar to the one described for Click&buy (see Section 3.4.3). Then they will add the resulting premium links to their web sites. Bitpass provides them gateway software, which will receive payment confirmations from Bitpass and will control the product access of paying consumers. Providers are paid out periodically, but they are also allowed requesting a pay-out under certain conditions. Providers will pay Bitpass transaction fees up to 15% of the payment's value. There is no setup or monthly fees.

Bitpass payments

Suppose a consumer encounters a content server and wants to buy a piece of content. A click on the Bitpass icon next to the description and price of the selected content, will initiate a Bitpass payment. Then the consumer needs to provide its email address and password in a Bitpass login window. After that, Bitpass requests the consumer to acknowledge the payment. If the payment has been authorized, Bitpass verifies the consumer's account balance. If the balance allows the new payment, Bitpass sends via the consumer a ticket with the payment confirmation to the gateway of the content server. Otherwise the consumer needs to transfer money to his/her account first. The gateway verifies this ticket and allows or rejects the access to the content.

In case a consumer did not receive the content, or the content is other than it was described, the consumer needs to contact via email the concerned content server and complain. After analysing the complaint, the content server should correct the faulty content delivery.

The trial version of Bitpass began in June 2003. There are already several providers that signed up for this payment service, for instance, www.scottmc-cloud.com sells comic strips (US$0,01-0,99), www.bigfriendlycorpora-tion.com sells mp3 music files (US$0,50/file or US$5/album).

3.4.5 Way2Pay

Way2Pay [72] is a pre-paid account-based payment system developed and introduced by the ING Bank in The Netherlands. Way2Pay includes a brand organization, an identification and authorization service provider and acts as a payment proxy on behalf of consumers.

Consumers are required to register for using Way2Pay. During the registration they need to provide personal, email, and banking information. After that they can open session by logging into the system using their email address and a previously set password. After registration they also need to transfer some money into their account. Consumers are then allowed to acknowledge previ-ously initiated payments, send and request money from other Way2Pay consumers or third parties (who may not be registered at Way2Pay). In other

words, Way2Pay supports person-to-person (P2P) payments as well. Once a payment has been acknowledged, consumers are not allowed to cancel it. Consumers can also change their registration information and review the status of their accounts.

Providers also need to register. Unlike in the case of Click&buy and Bitpass, the providers need only a short set-up of their web sites to allow consumers to pay with Way2Pay. For each piece of product they add information such as the name of the provider, a description of product, a product identifier, price of product (expressed in euros), and two URLs in case the payment is successful or rejected. This information will be provided to Way2Pay in a payment initiation. Providers do not receive payment confirmations or rejections directly from Way2Pay, paying consumers provide them the confirmations or rejections. Based on these indications the providers will provide or not the product to the consumers. Providers can review the history of payments and are allowed to refund consumers.

Way2Pay payments

Suppose a consumer encounters a content server and wants to buy a piece of content. The consumer decides to pay and clicks on the Way2Pay icon next to the description and price of the selected content. With this click he/she initiates a Way2Pay payment. After that Way2Pay opens a login window to the consumer. In this window the consumer sees the name of the content server, and the amount to be paid, then he/she introduces the email address and password to open a payment session. After that, the consumer should to acknowledge the initiated payment. If the consumer gave his/her consent, Way2Pay verifies the balance of the account. If the balance allows the new payment, then Way2Pay performs the payment. The consumer has no possibility to cancel a payment initiation after it was acknowledged. The consumer receives then the confirmation, which is provided to the content server. Also Way2Pay provides a confirmation to the content server in an email. Then the content server sends the content to the consumer. An additional confirmation is received by the consumer via email.

If the balance of the account does not allow the payment, the consumer is requested to transfer some money to his/her account first, then he/she should acknowledge again the payment.

In case a consumer did not get the content, or the content is other than it was described, the consumer needs to contact via email the concerned content server. If the content server does not solve the problem of the consumer, then he/she can contact Way2Pay with his/her complaint. After analysing the complaint, it is possible to refund the consumer.

Similar systems are for instance, Paynova [69], PayPal [70], Paystone [71], BillPoint (used on eBay) and TechnoCash (used on eBay in Australia and New Zealand).

3.4.6 Peppercoin

Peppercoin [56] is a post-paid, account-based micropayment system developed by R.L. Rivest and S. Micali, two professors at MIT. A spin-off company with the same name was founded in 2001 and it is expected to make its commercial debut in late 2003. The name of the system originated from the word "peppercorn", which is defined in the English law as "the smallest amount of money that can be paid in a contract". Peppercoin incorporates a brand organization, an identification and authorization service provider, a payment proxy, a product access control service provider and a software provider.

Providers are required to register for using Peppercoin. Then they need to download an application called PepperMill to create so-called pepperboxes. Pepperboxes are files that contain individual pieces of encrypted products together with product information (e.g., provider, product description, product type, price). Consumers can download these files, but cannot open them. If providers receive payment information from consumers, they will send this information to Peppercoin and a decryption key to the consumers. Providers are periodically paid out. Providers pay per transaction fees for using Peppercoin.

Consumers are also required to register. They need then to download and install an application called PepperPanel. This application will store authoriza-

tions from Peppercoin that the consumers are eligible to pay providers. This application is used to open pepperboxes and pay for them. PepperPanel reads the product information stored in the file and allows consumers to send the payment information to the appropriate providers. After that, the consumers receive the decryption keys to be able to extract and use the products. Consumers receive every now and then a list of completed payments and the total amount spent on products is deducted from their credit card account provided during the registration. Consumers use Peppercoin for free.

Peppercoin payments

Suppose a consumer downloaded a pepperbox and opened it with his/her PepperPanel in order to pay for it. PepperPanel sends the customer's payment information to the content server. The content server sends the information to Peppercoin and provides the consumer the decryption key. No money transfers occur immediately with every payment. Peppercoin transfers the money only on a small fraction of payments of a given content server (e.g., one money transfer occurs out of 100 payments initiated by all consumers).

A statistical method is used to select the payment that will be processed by Peppercoin. This method cannot be controlled by the consumer nor content servers. The selection of a payment can occur in every 100 payments sent to one content server, but it can happen after 91, 105 or 122 payments. Then the value of the selected payment multiplied with the serial number of the payment. E.g., if a consumer pays 20 cents for an mp3 music file and this payment is selected being the 100^{th} consecutive payment received, then the content server will receive US$20 from the Peppercoin. At the same time, the consumers' credit card is charged to pay Peppercoin the aggregated value of the payments made since the last settlement.

In this way, a content server can receive a little bit more or less than what Peppercoin collects from the consumers. The developers of this system proved that the fluctuation of the amounts received by content servers balances out over the time. The statistics and encryption of Peppercoin make sure that the system remains fairy to all parties on the long run [57]. As a result, Peppercoin transfers fewer macro payments than the number of micropayments. This allows an important reduction of transaction fees. Generally, the transaction

fee would be around 27 cents on a 99 cents sale, which can be lowered below 10 cents using Peppercoin.

Among the merchants participating in the trial of Peppercoin were included musicrebellion.com, celebrityrants.com and bigfrankrecords.com.

3.5 Summary

This section presents the summary of the business roles and functional characteristics of the studied micropayment systems. This summary also includes characteristics of other studied micropayment systems, which were not presented in detail in the previous sections.

Additional information about the investigated payment systems can be found in Appendix A of this thesis. This appendix describes time-sequence diagrams of the payments performed by these systems and the parameters that are exchanged in the various interactions.

Table 3.1 presents the business roles identified within the studied micropayment systems. Each X mark implies that the payment system contains the marked role.

Table 3.1 *Business roles*

Micro-payment system	Issuer	Acquirer	Brand and license. org.	Identification serv. prov.	Authorization serv. prov.	Payment proxy	Product access control serv. prov.	Software prov.
Wallie	X	X	X	X	-	-	-	-
Centipix	X	X	X	X	-	-	-	-
PaySafe-Card	X	X	X	X	-	-	-	-
Micro-money	X	X	X	X	-	-	-	-
Microeuro	X	X	X	X	-	-	-	-

Table 3.1 *Business roles (Continued)*

Micro-payment system	Issuer	Acquirer	Brand and license. org.	Identification serv. prov.	Authorization serv. prov.	Payment proxy	Product access control serv. prov.	Software prov.
Minitix	-	-	X	X	X	X	-	-
Secoin	-	-	X	X	X	X	-	-
Teletik	-	-	X	X	X	X	-	-
Softpay	-	-	X	X	X	X	-	-
Click&buy	-	-	X	X	X	X	X	-
Bitpass	X	X	X	X	X	-	X	X
WebCent	-	-	X	X	X	X	X	-
Way2Pay	-	-	X	X	X	X	-	-
PayNova	-	-	X	X	X	X	-	-
PayStone	-	-	X	X	X	X	-	-
PayPal	-	-	X	X	X	X	-	-
Peppercoin	X	X	X	X	X	-	X	X

Table 3.2 presents the functional characteristics of the studied micropayment systems. Because there were no micropayment systems found that support only provider initiated payments in a C2B context, this category was left out. There is no column for the double-confirmed payment, instead an X mark is placed in the columns of consumer and provider confirmed payments.

A plus sign in a cell (e.g., 50+) means that the maximum payment values supported by the appropriate payment system can also be higher under special conditions.

A question mark in a cell (e.g., X?) means that it is not sure whether the given characteristic is correct. In such situations no information about the system was available or the characteristics could not be checked.

Table 3.2 *Functional characteristics*

Micro-payment system	Consumer initiated payments	Jointly initiated payments	Consumer ack. payments	Provider ack. payments	Minimum value	Maximum value	Currency	Consumer costs	Provider costs	Pre-paid	Post-paid
Wallie	X	-	X	X	?	150	€	free	ca. 20% trans. fee	X	-
Centipix	X	-	X	X	0,01	50	US$	0,50/ card	subscr. + trans. fee	X	-
PaySafe-Card	-	X	X	-	0,01	1000	€	free	turnover	X	-
Micro-money	X	-	X	X	?	60	€	free	?	X	-
Microeuro	X	-	-	X?	?	?	€	free	?	X	-
Minitix	X	-	X	X	0,10	10+	€	free	subscr + trans. fee	X	-
Secoin	X	-	X?	X	0,10	10+	€	free	monthly fee + trans. fee	X	-
Teletik	X	-	X	X	0,05	?	€	free	trans. fee	X	-
Softpay	X	-	X	X	0,05	50+	€	€0,15/ transf.	trans. fee	X	-
Click&buy	-	X	-	-	0,05	5	€, US$	free	subscr + trans. fee	-	X
Bitpass	-	X	-	X	0,01	1000	US$	free	trans. fee	X	-
WebCent	-	X	-	X?	0,01	?	€	free	?	X	-
Way2Pay	X	-	X	X	0,01	2500	€	free	trans. fee	X	-
PayNova	-	X	X	X	0,10	1000	€, GBP, SEK, US$	free or €0.50/ transf.	trans. fee	X	-

Table 3.2 *Functional characteristics (Continued)*

Micro-payment system	Consumer initiated payments	Jointly initiated payments	Consumer ack. payments	Provider ack. payments	Minimum value	Maximum value	Currency	Consumer costs	Provider costs	Pre-paid	Post-paid
PayStone	-	X	X?	X	0,25	500	US$, CA$	free	trans fee	X	-
PayPal	X	-	X	X	0,10	1000	US$, €	free	trans fee	-	X
Pepper-coin	X	-	X	-	?	20	US$	free	trans fee	-	X

3.6 Conclusions

Cash-based systems have an enormous success and cash is a very attractive means for payments. Micropayment systems do not have the same properties as cash: widespread acceptability, guaranteed payment, no transaction fees and anonymity. For instance, the majority of current systems are used for free by customers, while merchants pay for using the system; current practice shows that each system lets customers bear the risk of losing their money; and the acceptability of current systems is limited by the fact that many systems function within national borders only.

At the moment of writing, no one of the presented micropayment systems had a breakthrough success. Nevertheless, second generation micropayment systems have much better chance to be successful than their predecessors [50]. In many cases the developers and operators of the new systems learned from the failures of the previous systems.

The accumulated experience (also based on [15], [49], [50] and [52]) implies that the key factors in the success of any payment scheme are:

- the trust of users (customers and merchants) in the payment system and its operator,
- convenience in usage (easy understanding of the functionality and technology, subscription, and use),
- widespread penetration and acceptance by both customers and merchants,
- privacy,
- processing speed,
- anonymity,
- data protection.

The interconnection of payment systems will be influenced by characteristics such payment initiations and acknowledgements, the supported payment values and the usage conditions.

Consumers initiate the payments for most existing systems. Regardless of the initiation type, the most common parameters of initiations are the consumer and provider identifiers, the product transaction (or shopping cart, order) identifier and the amount of money (value and currency). The other parameters of the initiations are related to the implementation of the payment systems.

Providers receive most often payment acknowledgements. The most common parameters of acknowledgements are the product transaction identifier (or context of payment) and payment identifiers (or transaction number, or ticket identifier). The other parameters of the acknowledgements are specific for the implementation of the payment systems.

As shown in Section 3.5, the great majority of micropayment systems are pre-paid and only few micropayment systems are post-paid for which the consumers still need to present a source of money before initiating payments. Among the reasons why PSOs have the tendency to deploy pre-paid systems is to limit the fraud possibilities by guaranteeing the payments to providers. It is also important to notice that post-paid systems require a (long-term) contract with

consumers in which a money source should be provided. This fact makes it more difficult for minors (who have no steady money sources or incomes) to become users of a post-paid system.

As shown in Table 3.2, most systems support a single currency. In case a system supports multiple currencies and the currencies of the customer's account and the amount of money to be paid differ, the customer will pay with the currency specified by the merchant and the payment system (or its operator) will determine the currency exchange rate.

3.7 References

[1] Davies, G., A history of money from ancient times to the present days, University of Wales Press, ISBN 0 7083 1351 5, Cardiff, 1996
[2] Camp, L.J. et al., Token and notational money in electronic commerce, In the Proceedings of the Usenix workshop on Electronic Commerce, New York, July 1995
[3] European Central Bank, Report on electronic money, ISBN 92-9181-012-6, August 1998
[4] Hille, S. and v.d. Stappen, P., Background on the Dutch payment system, Deliverable D0.1a of the GigaABP project of the Telematics Institute, Enschede, February 2002
[5] Lankhorst, M.M. et al., State of the art in e-business services and components, Deliverable 2.1 of the GigaTS project of the Telematics Institute, Enschede, December 2001
[6] Federal Reserve Bank of New York, Glossary, November 2002
[7] European Central Bank, Glossary
[8] Pierce, M., Multi-party electronic payments for mobile communications, Ph.D. Thesis, University of Dublin, October 2000
[9] Online Publishers Association, Online paid content - US market spending report, September 2003
[10] Jonkers, H. et al., Metering and reporting application usage, Deliverable 1.2 of the GigaABP project of the Telematics Institute, Enschede, January 2002
[11] Stiller, B. et al., Charging and accounting for Integrated Internet Services - State of the Art, problems, and trends, Proceedings of INET '98, Geneva, July 1998
[12] Biemans, F.P.M., et al., Reference models for networked applications, Lecture notes, Telematics Institute, Enschede, May 2002

[13] Hille, S., Backgrounds on the Dutch payment system, Deliverable 0.1a of the GigaABP project, Telematics Institute, Enschede, February 2002
[14] Böhle, K., The Innovation Dynamics of Internet Payment Systems Development, Report Nr. 63, Institute for Prospective Technological Studies, April 2002
[15] Kniberg, H., What makes a micropayment solution succeed, Master thesis, Kungliga Tekniska Högskolan, Stockholm, November 2002
[16] Abrazhevich, D., Electronic payment systems - A user-centered perspective and interaction design, Ph.D. Thesis, Technical University of Eindhoven, April 2004
[17] Faber, E. et al., Current innovation in commerce-enabling services, Deliverable 1.1.2 of the BITA Project, Telematics Institute, Enschede, October 2002
[18] Weber, R., Chablis - Market Analysis of Digital Payment Systems, Technical Report TUM-I9819, Technical University of Munich, Munich, August 1998
[19] Böhle, K., Integration of electronic payment systems into B2C Internet-commerce, Background paper No. 8, Electronic Payment Systems Observatory, April 2002
[20] O'Mahoney, D. et al., Electronic payment systems, Artech House, ISBN 0-89006-925-5, 1997
[21] Office for National Statistics of United Kingdom, Non-cash transactions: by method of payment: Social Trends 33, January 2003
[22] De Nederlandsche Bank, Betalen kost geld, Research report, March 2004
[23] Deutsche Bundesbank, Statistics on payment systems in Germany, September 2004
[24] Committee on payment and settlement systems, The role of central bank money in payment systems, Publication of the Bank for International Settlements, Basel, August 2003
[25] Burdett, D., Internet Open Trading Protocol 1.0, RFC 2801, IETF, April 2000
[26] Committee on payment and settlement systems, Retail payments in selected countries - A comparative study, Publication of the Bank for International Settlements, Basel, September 1999
[27] Interpay, Jaarbericht 2003, Utrecht, March 2004
[28] Bank of England, Oversight of payment systems, November 2000
[29] Personal communication with L.J.M. Nieuwenhuis
[30] Chaffey, D., E-business and e-commerce management, Pearson Education Limited, ISBN 0 273 68378 0, 2004

[31] Evans, D.S. and Schmalensee, R., Paying with plastic: The digital revolution in buying and borrowing, 2nd printing, MIT Press, Cambridge, 2000
[32] Peiro, J.A. et al., Designing a generic payment service, IBM Systems Journal, Internet Compunting, Volume 37, Number 1, 1998
[33] Good, B.A., Electronic money, Federal Reserve Bank of Cleveland, August 1997
[34] Geiger, H., Globalization and payment intermediation, In the Proceedings of the 22nd SUERF Colloquium on Adapting to Financial Globalization, Vienna, April 2000
[35] Asokan, N. et al., State of the art in electronic payment systems, IEEE Computer Magazine 30(9), September 1997
[36] VISA, http://www.visaeu.com, Statistics in December 2001
[37] MasterCard, http://www.mastercardintl.com, Corporate statistics in 2001
[38] Kerr, K., Enabling retail payments on the Internet, Research report of Gartner Group, February 2000
[39] Moore, A.M., Taking a bite out of credit card fraud, Research report of Celent Communications, Boston, January 2003
[40] Litan, A., Credit card companies provide little relief for online fraud, Research report of Gartner Group, December 2002
[41] Wills, T. and Favier, J., Retailers Can't Count On Visa To Stop Online Fraud, Research report from Forrester Research, June 2002
[42] SET Secure Electronic Transaction LLC, http://www.setco.org/
[43] Øygarden, K., Constructing security - The implementation of the SET technology in Norway, Dissertation, 2001, University of Oslo
[44] MasterCard, Secure Payment Application
[45] GPayments Pty Ltd., Visa 3-D vs. MasterCard SPA - A comparison of online authentication standards, White paper, March 2002
[46] Hancock, D. and Humphrey, D.B., Payment transactions, instruments, and systems: A survey, Journal of Banking and Finance, Volume 21, Pages 1573-1624, December 1997
[47] EPayNews, http://www.epaynews.com
[48] Visa International Service Association, 3-D Secure - Introduction, v.1.0.2, September 2002
[49] Weber, A. and Rader, M., Mobile Phones as Carriers of Cash and Tickets: The Outlook in Europe, Report Nr. 64, Institute for Prospective Technological Studies, May 2002
[50] Párhonyi R., el al., Second generation micropayment systems, Proceedings of The Fifth IFIP conference on e-Commerce, e-Business, and e-Government (I3E 2005), Poznan, Poland, October 2005

[51] Micali, S. and Rivest, R.L., Micropayments revisited, Proceedings of the Cryptographer's Track at RSA Conference, Springer-Verlag, February 2002
[52] Dutch Ministery of Economical Affairs, Betalen via Nieuwe Media (Pay via new media), Research report, The Hague, October 2003
[53] Inventory of E-Payment Systems Observatory, http://epso.jrc.es/paysys.html
[54] Google, http://directory.google.com/Top/Business/Financial_Services/Merchant_Services/
[55] Google, http://directory.google.com/Top/Business/Financial_Services/Merchant_Services/Other_Payment_Systems/Micropayments/
[56] Peppercoin, http://www.peppercoin.com
[57] Buderi, R. (ed.), Micromanaging money on the web, MIT Technology Insider, June 2003
[58] Bitpass, http://www.bitpass.com
[59] Wallie, http://www.wallie-card.nl
[60] PaySafeCard, http://www.paysafecard.com
[61] Micromoney, http://www.micromoney.de
[62] Centipaid, http://www.centipix.com/
[63] Splashplastic, http://www.splashplatic.com
[64] MyWebCard, http://www.mywebcard.dk
[65] Firstgate Click&buy, http://www.firstgate.com
[66] Minitix, http://www.minitix.nl
[67] Teletik Safepay, http://www.teletik.nl/
[68] Secoin, https://www.secoin.nl
[69] Paynova, http://www.paynova.com
[70] PayPal, http://www.paypal.com
[71] Paystone, http://www.paystone.com
[72] Way2Pay, http://www.way2pay.nl
[73] Itineris, http://www.itineris.com
[74] Paybox, http://www.paybox.de
[75] Moxmo, http://www.moxmo.com
[76] WebCent, http://webcent.web.de

Chapter 4

Requirements for the hybrid payment system

The purpose of this chapter is to identify and formulate the requirements for the hybrid payment system proposed in Chapter 1.

The requirements are derived in a top-down manner from the introduction of the Payment Gateway (and proposal of the hybrid payment system), the summary and conclusions of Chapter 3, legal and regulatory documents of major financial institutions or governments (Figure 4.1).

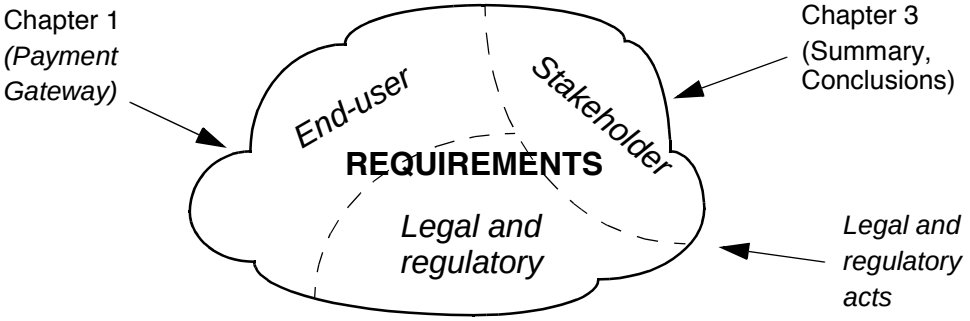


Figure 4.1 *Deriving and categorizing the requirements*

The structuring of the requirements is inspired from the HomeTalk project [1], which is part of the Information Society Technologies Programme [2]. The goal of this project is to create a "truly intelligent user-friendly residential environment that is capable to communicate with the residents via natural voice interface". In this project two target groups are identified: end-users (i.e., the

residents using the system) and stakeholders (i.e., participants building the system), and for these groups two requirement categories are formulated.

We have chosen this structuring technique because the target groups of the hybrid payment system are also the end-users and stakeholders. Customers and merchants (as defined in Section 3.1.5) form the end-users group. The fulfilment of their requirements determines largely the success of the hybrid payment system. The Payment System Operators (PSO) and Payment Gateway Operators (PGO) form the stakeholders group since they provide and operate the "building blocks" of the hybrid system. Their requirements are the expectations of the business units that share the interest in the hybrid payment system.

For the hybrid payment system, however, a third requirement category is formulated. The legal and regulatory requirements define the legal framework wherein the proposed system is expected to function.

Within the requirements categories, we define hard requirements and less important requirements. Because in this thesis, we present one design cycle of the hybrid payment system, the hard requirements will be used for designing the architecture of the hybrid system. This means that, hard requirements will influence the functionality of the this system. The follow-up cycles will address the less important requirements as well. Among these requirements are, for instance, the many legal and regulatory requirements, which give a short insight of the (financial) legislation the hybrid payment system should adjust to.

The structure of this chapter is as follows. The end-user requirements are discussed separately for the customers in Section 4.1 and for the merchants in Section 4.2, since their requirements differ in several aspects. For similar reasons, the requirements of the two stakeholder groups are discussed separately, PSO requirements are defined in Section 4.3 and PGO requirements in Section 4.4. The legal and regulatory requirements are formulated in Section 4.5.

4.1 Customer requirements

The customer requirements focus on the needs and expectations of customers with respect to the operation and characteristics of the hybrid payment system. The weight of each requirement may vary from customer to customer. We identified the following customer requirements:

- use a single payment system;
- make cross-border payments;
- a user-friendly payment system;
- anonymity;
- trust;
- security;
- privacy.

A consequence of these requirements is the costs, which are paid by customers for using the payment system. The degree in which the various requirements are implemented influences the end-user costs. In our view, requirements such as support for cross-border payments, user-friendliness, security and trust influence mostly the costs. For instance, the more secure and trusted a system is experienced by the customers, the more expensive it becomes for them. Customers are prepared to pay for the value they experience while using the system. When setting the customer costs it is important to take into account that they like fixed costs for predictability reasons [11]. Naturally, customers want costs as low as possible. In their perception the use of cash is free (e.g., they do not pay transaction costs when paying for groceries) and they compare using an electronic payment system with using cash [5]. Currently, most payment systems are used by customers for free.

4.1.1 Use a single payment system

Customers want to use a single online payment system on the Internet to pay merchants.

As described in Section 1.2, this is the primary requirement of customers, who want to use a single payment system for paying as many merchants as possible. Customers want to reach the large majority the merchants, so they require from the hybrid payment system a near 100% coverage of the merchant base. This is a hard requirement for the payment system.

This requirement means that as many as possible payment systems should be interconnected in our payment system, so customers can reach most merchants. A customer wants to use one account (stored within an existing payment system) to pay merchants that use the same existing payment system, but also merchants that use different payment systems. Customers also want that in the latter case they pay merchants without noticing that between them and the merchants a chain of payments is performed. This implies that, the payment system should transfer money between the end-users in a transparent manner.

As described in Chapter 1, one of the motivations for the research described in this thesis is the expectation that a significant market for low-value (e.g., ranging between €0,01 and €5,00) electronic content will soon emerge. To support the selling and buying of such content and other products, end-users want to have a payment system that can efficiently transfer small amounts of money. The targeted content types include:

- atomic files: music files, applications, clip art, scientific articles, book chapters;
- streaming content: video on demand, online radio, VoIP, news clips;
- news: sports, technology, local, national or world news;
- weather information services;
- stock quotes and financial services;
- travel information and guides;
- online games;
- dictionary entries;
- consulting or reference information.

4.1.2 Make cross-border payments

Customers want to make cross-border payments, so they want to use a globally accepted payment system.

As described in Chapter 1, it is expected that online content will generate important revenues for content providers in the coming years. Currently, most micropayment systems are deployed and used within national borders, as concluded in Chapter 3.

The need for cross-border payments emerges since the Internet is an international network that knows no country boundaries. Report [9] considers the support for cross-border payments a necessity as it is expected that the cross-border sale of low value products and especially content will most likely grow. Our payment system should therefore support cross-border payments and interconnect (inter)national payment systems. This is again a hard requirement.

4.1.3 A user-friendly payment system

Customers require a user-friendly payment system.

Customers will only accept and use a user-friendly payment system. As concluded in Chapter 3, based on reports ([8], [9]) of the Institute of Prospective Technological Studies (IPTS, [10]), and in a Swedish research report [11] that conducted interviews with PSOs and merchants, end-users consider the ease of use an important characteristic of payment systems.

We decompose the user-friendliness characteristic of a payment system into subscribe and ease of use features. Subscribe is the feature that expresses how easily can end-users acquire access to the system. The subscription is usually performed only once for each end-user. Systems that require a complex subscription procedure must offer significant and visible advantages to new users, otherwise users will be reluctant using them. The ease of use is the feature that expresses how easy and comfortable are the using and learning of the payment system even for an end-user that has never used such a system before. The ease of use is experienced continuously by the customers. They want that our payment system resembles a lot their currently used systems, so

their habits change as little as possible. Otherwise they might be scared away from the complexity of using the payment systems. The ease of use means simple and clear interactions with the system, the possibility of using for our payment system the same authentication methods used for the existing systems, and to change the current usage habits and experiences as little as possible. For instance, customers that have a user name and password combination or an account number to log into the system used so far, should be able to use continuously this combination or number for our payment system. We note that, although, SET was a well engineered payment standard, it failed because the joining and using was too complicated for the end-users [12].

We expect that the ease of use will significantly influence the success of our payment system, and the easier the system is used the more successful becomes. That is why the ease of use is a hard requirement.

4.1.4 Anonymity

Customers require to remain anonymous to PSOs and merchants.

Anonymity has two aspects from the customers' point of view. First, some customers value anonymity to merchants, e.g., when viewing adult content. This type of content is expected to generate between US$5 and US$7 billion within the next five years [18]. Current estimations report 400,000 sites that offer subscriptions for adult content. The number of different customers that visit at least one such sites per week is estimated to 70 million and almost 10 million of them pay for this content. In this context, the success stories of video systems in the Netherlands (VHS, Betamax and Video2000) in the 1980s, showed that the availability of adult content can be an important success factor. The video system that supported such content became the most successful [18], [20]. Analogously, in the context of this thesis, the payment system that will provide customer anonymity in certain situation (e.g., view adult content) will have a better chance to be successful and will be accepted quickly by many customers.

Second, some customers do not want to have each payment "on record" of the PSOs [3]. This means that they do not want to see each payment on the monthly bill of the PSOs.

The anonymity of customers to merchants is a hard requirement, while their anonymity to PSOs is less important. The reason for the latter decision is that PSO's are often required by law to collect customer information.

4.1.5 Trust

Customers should trust the payment system operators.

Because money is involved, customers require that all operators that will be involved in the operation of the hybrid payment system are trustworthy. They also require that they will receive the paid products from the merchants. This means that the operators should contract trustworthy merchants.

Several studies sustain this requirement. Studies [13] of the Telematics Institute (TI, [14]), which is one of the research centres in the Netherlands, reports ([3], [15]) of the IPTS and a research paper [4] consider trust a vital requirement. In these studies trust is seen as a precondition for the success of e-commerce or as a primary condition for the success of a payment system. In addition, a report concluded that trust is more important to end-users than security and end-users are more likely to use a less secure payment system operated by a trusted PSO, rather than using a very secure payment system of a less trusted PSO [11].

We expect that customers trust the PSOs operating the currently used systems, so they will trust the operators involved in our payment system. That is why this is not a hard requirement.

4.1.6 Security

Customers require a secure payment system.

Customers require security, because attempts for misusing a payment system in order to commit fraud on the Internet are common [4]. An IPTS study states that security is a factor that influences the widespread acceptance of the payment system by customers [9]. Accordingly, the security of payment systems should be strong enough to detect attacks or to detect the parties responsible for attacks. Customers also want to be sure that merchants will not

abuse the system and that they will receive the paid product(s) once the payments are completed.

Security is to a certain extent a subjective concept, and felt differently by each end-user ([3], [15]). In this thesis, we adopt the main security concerns formulated by the Mobile Payment Forum [21]:

- Non-repudiation is a property of a payment system that does not allow customers and merchants to deny their actions that resulted in money transfers. Customers want that merchants cannot deny receiving money from them.
- Authentication and authorization are two functions of the payment system that establish the identity of customer and determine his/her right to make payments. These functions are necessary because customers require guarantees that no other user will be buying in their place with their money.
- Data integrity ensures that payment information in not altered after a payment in initiated, and this information can only be accessed by authorized parties. Measures that provide integrity allow the detection of any unauthorized attempt to modify payment information.
- Confidentiality ensures that unauthorized parties cannot access the sensitive payment information that might be used later for fraudulent purposes. Customers want to be sure that their buying transactions remain secret for others.

This payment system will only support transfers of small amounts of money, therefore it only needs limited and lightweight security measures. This is because balance must be kept between the costs of implementing the security measures and the protected value of a payment. In general, a payment is considered to be safe if the costs of breaking its security are higher than the value of that payment.

Security is a hard requirement.

4.1.7 Privacy

Customers require that their privacy is protected by payment system and gateway operators, and merchants.

Privacy is another factor that determines the possible success of a payment system among customers [8]. Customers own their personal information and have a right to privacy, which should not be violated. Studies [17] of Forrester Research state that 60% of online customers fear about the misuse of their personal information (e.g., sending them unsolicited promotional materials).

PSOs and merchants must demand customer information on a need to know basis to be able to perform their activity. They must not abuse the received information, for instance, use it to send promotional material or sell it to third parties. Further, they need to protect is information from external attacks. The safeguards depend on the sensibility of the information. In this thesis, we consider privacy a less important requirement.

4.2 Merchant requirements

The merchant requirements focus on the needs and expectations of merchants with respect to the operation and characteristics of the hybrid payment system. The weight of each requirement may vary among merchants. We identified the following merchant requirements:

- use a single payment system;
- receive cross-border payments;
- a user-friendly payment system;
- trust;
- security.

A result of these requirements is the costs that are paid by merchants for using the system. Similarly to customers, merchants want low usage costs. Currently, most payment systems require merchants to pay for using these systems, for instance, in form of initial or subscribing fees, periodical fees (e.g., monthly,

annual fees) or transaction fees. Hence, costs are relevant from the merchants' viewpoint.

4.2.1 Use a single payment system

Merchants want to use one online payment system to sell products to customers.

As formulated in Section 1.2, this is the primary requirement of merchants that require a single payment system to receive money from as many customers as possible. Providers want to reach the large majority of potential their customers, so they require from the hybrid payment system a near 100% coverage of the customer base. This is a hard requirement for the payment system.

4.2.2 Receive cross-border payments

Merchants want to receive cross-border payments, so they want to use a payment system with global penetration.

This requirement is similar to the one formulated for customers in Section 4.1.2. It is a hard requirement.

4.2.3 User-friendly payment system

Merchants require a user-friendly payment system.

This requirement is rather similar to the one formulated for customers in Section 4.1.3, with the remark that merchants want to focus on the sale of their products and their customers, and not on the complex behaviour of payment system. This is a hard requirement.

With the introduction of our payment system, merchants want that the subscribing and using of this payment system remains user-friendly, so their habits and experiences with the currently used systems changes as little as possible. In other words, they expect (i) simple and clear interactions with the payment system and (ii) the possibility of using their current identification methods for our system as well.

4.2.4 Trust

Merchants should trust the payment system operators.

Trust in the operators of the payment system is a vital requirement for the merchants as well, but trust has a different meaning for them than for customers. Merchants want that payment system operators transfer (periodically or when requested) the collected amounts of money. This money is their income that results from the selling and delivering products to customers.

We expect that merchants trust the PSOs of the currently used systems, and if they keep using these systems, no trust issues will emerge. That is why trust is not a hard demand.

4.2.5 Security

Merchants require a secure payment system.

This requirement is similar to the one formulated for customers in Section 4.1.6 with the remark that the payment system must prevent the non-repudiation of the customers. This is important for merchants because customers may try to abuse the payment system, reverse or cancel the payments, after they received the products. Security is a hard requirement.

4.3 Payment System Operator requirements

In general, PSO requirements include the requirements of the operators of existing and future payment systems such as support for small amounts of money, wide penetration and acceptability of the system, increase revenues and payment volume, etc. The proposed hybrid payment system integrates the existing payment systems and for such an integrated system, we formulate the following requirements in addition to the original PSO requirements:

- minimal changes to their systems;
- availability and performance;
- scalability;
- trust each other and the PGOs.

4.3.1 Minimal changes

PSOs require that the interconnection of their systems is performed such that they need to make minimal changes to their systems.

PSOs prefer that their systems will be interconnected just as they currently function, but if this is not possible, then they accept only minimal changes. These changes should be feasible and achievable for them. This is a hard requirement.

4.3.2 Availability and performance

PSOs require that the availability of the Payment Gateway is very close to 100%, and the performance of the hybrid payment system is comparable to the performance of existing payment systems.

PSOs require the very high availability from the PG because they have contracts with users that want to use the payment system continuously. The PG is the key element in this payment system and its availability determines the availability of our payment system.

PSOs also want to make sure that the overall performance of the hybrid payment system is very high, so it can be really called a hybrid *micro*payment system, just like the systems they operate. The hybrid payment system must be able to perform millions of payments per day, and the PG must not create a bottleneck in the system.

We consider these requirements less important for the design cycle presented in this thesis.

4.3.3 Scalability

PSOs require a scalable hybrid payment system.

This requirement highlights the fact that the components of the hybrid payment system, i.e., the existing payment systems and the PG must scale if the number of users and the volume of hybrid payments increases. PSOs require this

because they have contracts with customers and/or merchants, which need to be fulfilled even if their systems became part of the hybrid payment system.

PSOs built their systems such that these systems will scale if the payment volume of their current customer and merchant base increases up to a certain limit [23]. We expect, however, that the hybrid payment system will put an additional load on current systems because it will have a very large and increasing customer and merchant base. This expectation is realistic because we aim at the global acceptability and high penetration of our payment system. Acceptability is measured by the number of merchants that accept the payments of this system. Penetration is measured by the number of customers that pay using this system. The acceptability and penetration levels of this system show higher rates than those of any individual payment system. This is because our system inherited the end-user base of each individual payment system. New merchants are attracted because there is a significant customer base that will pay them, and vice versa new customers are attracted due to the large merchant base that will provide them products. The idea to expand the user base is shown in Figure 4.2 and can also be found in Metcalfe's Law. This law states that the value of a communications network is proportional to the square of the size of the network. If we consider our payment system, we can declare that the value of this system is proportional with the square of the size of the user base. Odlyzko, however, claims that this proportion is overestimated and suggests that the value of system of user base n grows like is $n\log(n)$ [24].

Scalability is a hard requirement.

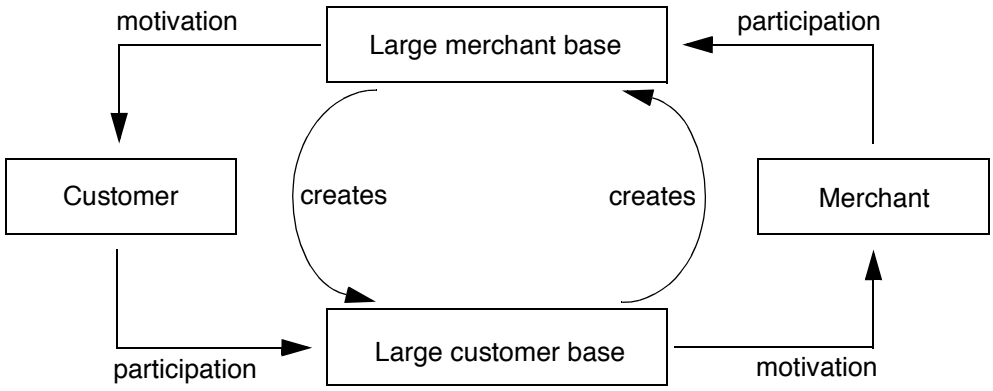


Figure 4.2 *Relation between the customer and merchant bases*

4.3.4 Trust

PSOs must trust each other and the operators of the PG in order to participate in the operation of the payment system.

In our payment system, trust has several aspects as described in Sections 4.1.5 and 4.2.4. However, PSOs can only provide trusted payment systems to their end-users, if they trust the operators of the other payment systems and of the PG.

The organizations involved in this payment system should address the issue of trust between each other. One possibility is to include it in their business agreements, which are required to assure interoperability between them. However, if each PSO needs to sign an agreement with all other PSOs to ensure trust between them, then they may run into scalability problems. Suppose that there are n PSOs ($n > 1$), then $n \times (n-1)$ agreements are needed. In case $n > 100$, the establishment of these agreements will be difficult to achieve.

Similarly to the previous trust requirements, trust is not a hard requirement for this design cycle.

4.4 Payment Gateway Operator requirements

Payment Gateway Operator (PGO) requirements express the needs and expectations of these operators with respect to the operation and characteristics of the hybrid payment system. We identified the following PGO requirements:

- the vast majority of payment systems must be interconnected;
- availability and performance;
- scalability;
- trust in the PSOs.

4.4.1 Vast majority

PGOs require that the PG interconnects the vast majority of payment systems.

PGOs want to attract as much as possible existing and future PSOs to interconnect their systems. The more systems are interconnected the more complete is the PG's role as interconnector and the higher is the penetration and acceptability of our payment system. Only in this way PGOs will be determined to deploy the PG. This is a hard requirement.

4.4.2 Availability and performance

PGOs require that the availability of the interconnected payment systems is very close to 100%, these systems do not create a bottleneck in the payment system.

PGOs require very high availability from the existing (and future) payment systems because only in this way hybrid payments can continuously by processed between any customer and merchant.

PGOs also want to make sure that the performance of the existing systems is sufficient to perform millions of payments per day. These systems therefore must not create a bottleneck and must be scalable if the payment volume increases.

In the design cycle presented in this thesis, this requirement is less important.

4.4.3 Scalability

PGOs require a scalable hybrid payment system.

This requirement is similar to the scalability requirement formulated for PSOs in Section 4.3.3. Scalability is a hard requirement.

4.4.4 Trust

PGOs must trust the PSOs in order to participate in the operation of the payment system.

The PG will play the role of customers and merchants for existing and future payment systems, so the PGOs want to trust the PSOs, as described in Sections

4.1.5 and 4.2.4. Current and future PSOs need to guarantee that their payment systems will perform the payments necessary for hybrid payments.

Similarly to the previous trust related requirements, trust is not a hard requirement for this design cycle.

4.5 Legal and regulatory requirements

Also authorities like the European Central Bank [26], the Federal Reserve [27], national central banks, Institutes of the European Union, or governmental agencies define requirements for electronic payment systems and their operators [25]. These requirements are formulated within legal and regulatory frameworks. Such frameworks contain various acts, directives, recommendations, amendments and guidelines that are concerned with various aspects of electronic payment systems. These frameworks define the roles of the participants in a payment system, their obligations and liabilities, and describe how the electronic payment systems can be used.

This section presents a list of legal and regulatory requirements. This list is based on legal and regulatory frameworks defined by the:

- European Commission (EC): The Data Protection Directive 95/46 [28], and Recommendation 97/489/EEC [29] on the electronic payment instruments;
- European Parliament and Council (EPC): Directive 2000/46/EC [30] on e-money;
- European Central Bank (ECB): Report on electronic money [31];
- US government institutions: Federal Internet Privacy Protection Act (FIPPA, [32]), Uniform Money Service Act (UMSA, [33]), Electronic Funds Transfer Act (EFTA, [34]), and Patriot Act [35].

We note that, these documents are only the most important ones. This list is small, because the detailed analysis of the legal and regulatory frameworks of various countries is outside the scope of this work. Readers who are interested in this subject are referred to a comprehensive study [5], which describes the European legislation in this domain. Moreover, the University of Texas main-

tains an easy accessible collection of e-commerce related legal documents [36], including the domestic and European legislation as well.

Although, the following legal and regulatory requirements are applicable to individual payment systems, PGOs need to consider them as well, because the PG is a new participant in the proposed hybrid payment system. The remainder of this section discuss these following requirements:

- support for audit;
- obligations and liabilities;
- security;
- privacy;
- payments should be irrevocable;
- license and supervision.

4.5.1 Support for audit

Electronic payment systems should have support for audit.

Audit is an independent and professional verification act of the payment records of a payment system performed off-line and after the payments are completed. An audit may also include the examination of a payment system's compliance with applicable laws and regulations. Certified accountants or governmental organizations can perform the auditing.

The ECB requires in its Report [31] that issuers of e-money should provide central banks with "whatever information may be required for the purpose of monetary policy". An electronic payment system "should include adequate accountability and audit trails". Information on the circulation of (electronic) money is relevant for the conduct of monetary policy by the central banks. This information may also be used to create monetary statistics.

Under the UMSA [33], PSOs are also required to report periodically to state legislatures. UMSA also describes the uniformity of the reporting and record keeping requirements. The uniform reporting and licensing requirements serve the prevention of money laundering.

The support for audit is a hard requirement for our payment system.

4.5.2 Obligations and liabilities

Customers should be protected against loss, but they support the loss of money up to a certain limit.

ECB considers in [31] that as a payment system grows, "insurance or loss-sharing schemes" should be adopted to protect end-users against loss and to preserve their confidence in that system. However, customers bear the loss of money in a payment system up to €150. There is no limitation of losses when customers acted fraudulently or with extreme negligence.

The EFTA [34] specifies measures to protect customers of electronic funds transfer systems. These measures include error resolution procedures, limit the liability of customers and require the disclosure of usage terms and conditions.

The implication of this requirement with respect to the hybrid payment system is the providing of a proper payment system interconnection. This means that the new payment initiation should only be accepted by the hybrid payment system, if it is certain that the existing systems involved will be able to make the necessary. We consider that the existing payment systems obey this requirement, since they are already operational and fall under the supervision of financial authorities. The obligations and liabilities requirement is not considered a hard requirement.

4.5.3 Security

Electronic payment systems should be secure.

Not only the end-users require security, but also organizations like the ECB. In its Report [31], the ECB requires that "electronic payment schemes must maintain adequate technical, organisational and procedural safeguards to prevent, contain and detect threats to the security of the scheme, particularly the threat of counterfeits." Similarly to previously formulated security requirements, this is also a hard requirement.

4.5.4 Privacy

End-users have right to privacy, which should be protected.

Not only end-users require privacy, but also governments and organizations like the EC, which issued the Directive 95/46 "on the protection of individuals (end-users) with regard to the processing of personal data and on the free movement of such data" [28]. This Directive is implemented in the national laws of the EU member countries. This Directive is concerned with the fundamental rights of persons, and their right to privacy regarding their personal data. In this context, the Directive provides obligations and responsibilities for organizations involved in collecting, storing and processing personal data, and it regulates how personal data can be transferred to other parties for processing purposes.

Under the FIPPA [32], federal agencies are prohibited to publish information about individuals on the Internet. Individuals may begin a civil action against any agency that made public such information. The personal information include "any item, collection, or grouping of information about an individual" regarding her education, financial transactions, medical history, employment history, or information that specifies "name or the identifying number, symbol or other identifying particular assigned to the individual".

Similarly to the reasoning presented in Section 4.1.7, the privacy is not considered a hard requirement.

4.5.5 Payments should be irrevocable

Payments should be irrevocable, except in the situation when the amount was not determined prior to the payment initiation.

Recommendation 97/489 [29] of the EC states that end-users cannot recall or change their payments unless the amount was not determined before they initiated these payments. This is a less important requirement because current micropayment systems always require the amount of money to be part of the payment information provided to the system in payment initiations (see Section 3.6).

4.5.6 License and supervision

Payment system operators should have licenses, and they will be supervised by financial authorities.

Until 2000, issuing of e-money in Europe was an activity that had different regulations. Some countries regulated it as part of banking activity (e.g., Austria, the Netherlands and Germany), others separately (e.g., Ireland and Denmark). In all cases, however, an e-money issuer needed a license, although the type of the license and the way supervision takes place varies from country to country.

In 2000, Directive 2000/46 [30] of the EPC was issued in order to create a consistent and harmonious European legislation regarding the issuing of e-money. This Directive

- defines which parties (other than banks) can become e-money issuers;
- defines the requirements (i.e., capital, investment and supervision requirements) to become an e-money issuer; and
- determines what activities other than issuing can be conducted (e.g., storing of data on electronic devices).

Accordingly, organizations other than banks such as ISPs are allowed to issue e-money only if they comply with the requirements. Otherwise, these organizations could collaborate with banks to issue e-money.

In most states of the USA, issuing e-money requires a licence, according to the UMSA [33]. Acquirers and other non-bank money service businesses (e.g., that exchange foreign currency or perform money transmitting activities) also require licenses. Additionally, under the Patriot Act [35], it is considered a federal crime to operate as (e-)money issuer without a license. Licenses are issued by competent authorities after a thorough verification of the applicant issuer.

This requirement is less important, because it should be considered when our system will be deployed.

4.6 References

[1] HOMETALK: A voice enabled residential automation & networking platform, HomeTalk System Requirements, Deliverable 1, http://www.hometalk.org/, August 2002
[2] Information Society Technologies (IST), http://www.cordis.lu/ist/
[3] Böhle, K. et al., Electronic payment systems - Strategic and technical issues, Background paper No. 1 of the Electronic Payment Systems Observatory, December 2000
[4] Abrazhevich, D., Classification and characteristics of Electronic Payment Systems, Proceedings of the Electronic Commerce and Web Technologies Conference, Springer-Verlag Berlin Heidelberg, 2001
[5] Hille, S., Legal and regulatory requirements on accounting, billing and payment, Deliverable 1.4 of the GigaABP project of the Telematics Institute, Enschede, November 2000
[6] Weber, R., Chablis - Market Analysis of Digital Payment Systems, Technical Report TUM-I9819, Technical University of Munich, August 1998
[7] Camp, L.J. et al., Token and notational money in electronic commerce, In Usenix Workshop on Electronic Commerce, July 1995
[8] Weber, A. and Rader, M., Mobile phones as carriers of cash and tickets: The outlook in Europe, Paper in the IPTS Report No. 44, Institute for Prospective Technological Studies of the Joint Research Centre of the European Commission, May 2000
[9] Papameletiou, D., Study on electronic payment systems for the Committee on Economical and Monetary affairs and Industrial Policy of the European Parliament, Volume I: Main report, Institute of Prospective Technological Studies of the Joint Research Centre of the European Commission, Seville, May 1999
[10] Institute of Prospective Technological Studies, Research institute of the Joint Research Centre of the European Commission, http://www.jrc.es/home/index.html
[11] Kniberg, H., What makes a micropayment system succeed, Master's Diploma Project, Kungliga Tekniska Högskolan, Stockholm, November 2002
[12] Øygarden, K., Constructing security - The implementation of the SET technology in Norway, Dissertation, 2001, University of Oslo
[13] Hille, S. and v.d. Stappen, P., Backgrounds on the Dutch payment system, Deliverable 0.1a of the GigaABP project of the Telematics Institute, Enschede, February 2002

[14] Telematics Institute, http://www.telin.nl
[15] Centeno, C., Building security and consumer trust in Internet payment - The potential of "soft" measures, Background paper No. 7 of the Electronic Payment Systems Observatory, April 2002
[16] Pierce, M., Multi-party electronic payments for mobile communications, Ph.D. Thesis, University of Dublin, Dublin, October 2000
[17] Kelley, C.M. el al, Privacy concerns cost eCommerce $15 billion, Research report of Forrester Research, September 2001
[18] Thornburgh, D. and Lin, H. S. (eds.), Youth, pornography and the Internet, National Academy Press, 2002, ISBN 0-309-08274-9, http://bob.nap.edu/html/youth_internet/
[19] Wikipedia, http://en.wikipedia.org/wiki/Betamax, http://en.wikipedia.org/wiki/VCR
[20] Wasko, J., Hollywood in the information age: Beyond the silver screen, University of Texas Press, ISBN 0292790945, February 1995
[21] Mobile Payment Forum, Enabling secure, interoperable, and user-friendly mobile systems, White paper, December 2002
[22] Faber, E. et al., Innovation in payment services: the case of mobile payment, In the Proceedings of BITA project of the Telematics Institute, Enschede, March 2002
[23] Párhonyi R., el al., Second generation micropayment systems, Proceedings of The Fifth IFIP conference on e-Commerce, e-Business, and e-Government (I3E 2005), Poznan, Poland, October 2005
[24] Odlyzko, A. and Tilly, B., A refutation of Metcalfe's law and a better estimate for the value of networks and network interconnections, March 2005
[25] Bargh, M. et al., State of the art in e-business services and components, Deliverable 2.1 of the GigaTS Project of the Telematics Institute, Enschede, December 2001
[26] European Central Bank, http://www.ecb.int
[27] Federal Reserve, http://www.federalreserve.gov
[28] European Parliament and Council Directive 95/46/EC, October 1995 http://europa.eu.int/comm/internal_market/en/dataprot/law/index.htm
[29] European Commission Recommendation 97/489/EC, July 1997 http://europa.eu.int/ISPO/ecommerce/legal/documents/recpay.zip
[30] European Parliament and Council Directive 2000/46/EC, July 2000 http://www.utexas.edu/law/faculty/ecommerce/Statutes/ElectronicMoneyDirective.pdf
[31] European Central Bank, Report on electronic money, August 1998 http://www.dnb.nl/betalingsverkeer/pdf/emoney.pdf

[32] Federal Internet Privacy Protection Act, April 1997 http://thomas.loc.gov/cgi-bin/query/z?c105:H.R.1367:
[33] National Conference of Commissioners on Uniform State Laws, Uniform Money Service Act, 2000, http://www.nccusl.org
[34] Electronic Fund Transfer Act, Published in the Code of Federal Regulations, Title 12, Volume 2, Part 205, January 2001
[35] US Department of the Treasury, USA Patriot Act, 2001 http://www.fincen.gov/hr3162.pdf
[36] University of Texas, http://www.utexas.edu/law/faculty/ecommerce/statutes.htm
[37] Hille, S. and v.d. Stappen, P., Electronic payment put in context, Deliverable 0.1b of the GigaABP project of the Telematics Institute, Enschede, March 2002

Chapter 5

Hybrid payment system architecture

This chapter presents the design of an architecture of the hybrid payment system. To structure this design process we use the methodology and concepts introduced in [1]. This methodology has three phases, which are illustrated in Figure 5.1.

In the first phase (Section 5.1), we formulate the functional requirements based on the main functional characteristics of existing payment systems (presented in Chapter 3) and the hard requirements of end-users, stakeholders, financial authorities and governments (formulated in Chapter 4). The less important requirements are not taken into account in the current design cycle.

In the second phase (Section 5.2), we design the hybrid payment service, which represents the external behaviour of our system as experiences by the end-users.

In the third phase, we discuss possible approaches for interconnecting existing micropayment systems and design the hybrid payment protocol. Sections 5.3 through 5.6 introduce and compare two approaches to interconnect micropayment systems, and present in detail the most suitable approach. The design of the hybrid payment protocol built on to top of the services of existing payment systems will be presented in Chapter 6.

Section 5.7 presents the conclusions of this chapter.

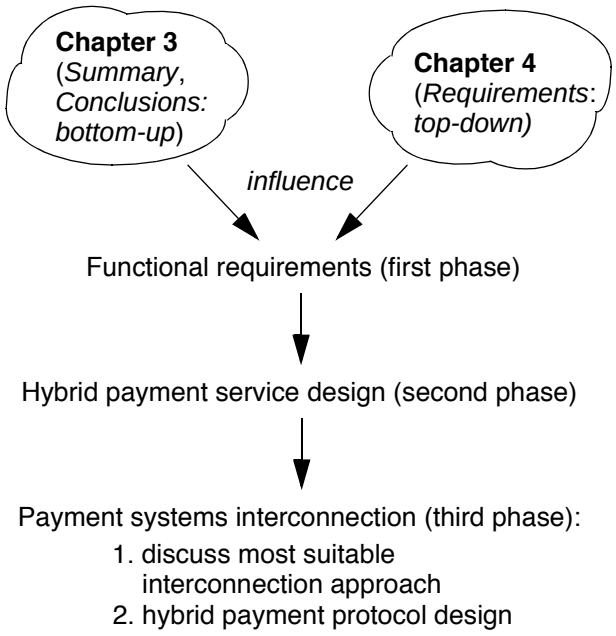


Figure 5.1 *Design methodology*

5.1 Functional requirements of the hybrid payment system

The functional requirements define the main characteristics and functionality of the hybrid payment system as expected by end-users, stakeholders and financial authorities. We define those characteristics of the hybrid payment system that are part of the payment system characterization model defined in Section 3.2.2. These characteristics are the way payments are initiated and acknowledged, the supported payment values and currencies, and the conditions for using the payment system.

To formulate the functional requirements we consider the functional characteristics of existing payment systems discussed in Chapter 3 and the hard requirements formulated in Chapter 4. These functional characteristics influence the characteristics of the hybrid system because end-users require minimal changes in their habits and PSOs for their systems, and our goal is to interconnect as many existing systems as possible.

5.1.1 Payment initiations

Requirement: *The hybrid payment system requires consumers to initiate the payments.*

Three arguments motivate this requirement: (1) the observed common practice of existing payment systems, (2) a customer requirement and (3) the drawback of the jointly initiated payments.

First, the majority of existing payment systems requires consumers to initiate the payments (see Section 3.5). Consumers are always involved in payment initiations because they take the final decision whether to allow the payment system to transfer money from their account.

Second, customers require and appreciate a user-friendly payment system, whose functionality resembles the functionality of their currently used systems, so their payment habits do not have to change (see Section 4.1.3).

Third, jointly initiated payments, which is the only alternative to consumer initiated payments used in practice (see Section 3.5), may be impractical. In this case, the initiations need to be correlated within the hybrid payment system, which will be difficult considering the likelihood that consumers and providers will use different payment systems. The initiation correlation requires a more complex interconnection method from the hybrid system than the consumer initiated payments. We note that, this argument does not exclude from the hybrid system those systems that require jointly initiated payments.

5.1.2 Payment acknowledgements

Requirement: *The hybrid payment system acknowledges payments to providers.*

Three arguments motivate this requirement: (1) the observed common practice of existing payment systems, (2) a merchant requirements and (3) the delay introduced by consumer confirmed payments.

First, the majority of existing payment systems provides acknowledgements to providers (see Section 3.5). Providers need acknowledgements from the hybrid system immediately after the payment is completed, so they can deliver the paid products. Customers consider the acknowledgements as evidences of completed money transfers, and we consider that the delivered products are sufficient evidences that the payments were successfully processed.

Second, merchants require and appreciate a user-friendly payment system, whose functionality resembles the functionality of their currently used systems, so their payment habits do not have to change (see Section 4.2.3).

Third, in case of systems that provide only consumer acknowledgements, the consumers need to send this to the providers, so the providers can deliver the paid product(s). This alternative introduces additional delay in the product transaction, and besides this is less common in current practice.

We need to address here a hard requirement of the consumers concerning the payment acknowledgements, namely that they want to remain anonymous with respect to providers (see Section 4.1.4), so *payment confirmations must not reveal the identity of consumers*.

5.1.3 Supported payment values and currencies

Requirement: *The hybrid system needs to determine in the chain of payments the highest minimum payment value for each hybrid payment, and determine whether the payment value to be transferred is supported or not.*

We do not define a global minimum payment value for the hybrid payment system to avoid unnecessary limitations of this system. A minimum always exists, however, and this depends on the payment systems involved in the chain of payments. This minimum is the highest of all minimums supported by the involved systems, and may change for each hybrid payment.

As showed in Section 3.5, existing payment systems support different minimum payment values. This fact may create interoperability problems if one of the interconnected systems does not support a certain minimum payment value. Payment systems used by merchants will always support the amount of money that was determined by the product accounting systems, because these systems know the minimum payment values. The payment systems used by customers, however, may not be able to transfer such an amount of money, so the customer may not be able to buy the selected product(s). Nevertheless, because the supported minimum payment values range approximately between US$0,01 up to €0,25 and because we expect that the product prices are rarely that low, we think that the mentioned interoperability problem will rarely occur.

Requirement: *The hybrid system needs to determine in the chain of payments the lowest maximum payment value for each hybrid payment, and determine whether the payment value to be transferred is supported or not.*

We do not define a global maximum payment value to avoid unnecessary limitations of the hybrid system. A maximum always exists, however, and this is the lowest of all maximums supported by the interconnected systems. This maximum, therefore, may change for each hybrid payment.

As showed in Section 3.5, existing payment systems support different maximum payment values. This fact may create interoperability problems if one of the involved payment systems cannot process a particular payment. The payment systems used by the merchants will support each payment value determined by the product accounting systems. The payment systems used by customers, however, may not be able to make such payments, so the selected product(s) may not be bought.

Requirement: *The hybrid payment system supports multiple currencies, and exchanges the currencies, if necessary.*

Two arguments motivate this requirement: (1) the facts that existing payment systems support different (sets of) currencies (see Section 3.5) and customers always need to pay with the currency given by merchants, and (2) the requirement of customers and merchants for cross-border payments (see Sections 4.1.2 and 4.2.2).

The difference among payment currencies supported by the existing payment systems shows less variance than the payment values; the Euro and US Dollar are the most frequently supported currencies. The hybrid payment system supports the currencies of the incorporated existing payment systems such that (i) consumers will be able to pay all providers regardless of the currency the providers use, and (ii) providers will be able to receive money from all consumers even if the consumers use different currencies. To provide this support, the hybrid payment system needs to exchange currencies.

5.1.4 Usage conditions

Requirement: *The hybrid payment system functions as both a pre-paid and a post-paid system.*

This requirement is motivated by the main idea behind the hybrid payment system, i.e., existing payment systems need to be interconnected. These systems function as pre-paid or post-paid systems (see Section 3.5), but if they will be interconnected, the hybrid payment system needs to function as pre-paid and as post-paid at the same time, otherwise a main characteristic of the system will change for the customers. Moreover, such a change would modify the contracts customers have with their systems, the risk bearer in case of problems, etc., and these modifications should be avoided. Hence, those consumers who used a pre-paid existing system will use the hybrid system as a pre-paid, and those who used a post-paid system will use the hybrid system as a post-paid system.

5.1.5 Summary

Table 5.1 presents the functional characteristics of the hybrid payment system.

Table 5.1 *Functional characteristics of the hybrid payment system*

Micro-payment system	Consumer initiated payments	Jointly initiated payments	Consumer ack. payments	Provider ack. payments	Minimum value	Maximum value	Currency	Pre-paid	Post-paid
HPS	X	-	-	X	variable	variable	multi	X	X

5.2 Hybrid payment service design

The hybrid payment service represents the *external behaviour* of the hybrid payment system (HPS) experienced by consumers and providers. The following sub-sections present a stepwise definition of this service using the formu-

lated functional requirements. We design the hybrid payment service in terms of (i) interactions between the service users and the hybrid payment system, and (ii) the relationships between these interactions [1].

To represent the interactions between the end-users and the hybrid system, we introduce service primitives (SP), which are defined by stating their purpose and parameters. The parameters of a SP represent the information exchanged an interaction. SPs occur at service access points (SAPs) between consumers or providers, and the hybrid payment system. SAPs are specific interaction points and represent a mechanism that enables the interactions. To determine the relationships between the primitives, we introduce constraints that control the occurring of the interactions. We model these constraints using the *Interaction System Design Language* (ISDL), which "supports the design of distributed systems by providing generic design concept and a notation to model the structure and behaviour of distributed systems" [2]. ISDL was developed at the University of Twente.

5.2.1 Hybrid payment service users

The hybrid payment service users, which form the environment of the hybrid payment system, are divided in two groups. The first user group consists of consumers (as defined in Section 3.1.5), who use the hybrid payment system to pay small amounts of money to providers. The second group consists of providers (as defined in Section 3.1.5) that use the hybrid payment system to receive small amounts of money from consumers. Consumers and providers that use the hybrid payment service are from now on called *HConsumers* and *HProviders* to clearly indicate that they are users of the hybrid payment service.

We distinguish two alternatives for HConsumers and HProviders to interact with the hybrid payment system. One alternative is to use the interfaces of the existing payment systems. This means that the payment system used by the HConsumer would receive payment initiation information in many different formats, because this information is generated together with different HProviders that probably use other payment systems. These HProviders can only generate payment information that is specific for their payment systems. This information may be too much, not enough or unacceptable for the HConsumers'

systems due to the different semantics of the information. To overcome such problems and be able to initiate payments with all kind of information, the HPS needs a rather complex functionality that includes the definition of mappings between all kinds of payment initiation information. Due to the large number of existing payment systems and the high probability that they all require different payment initiation information, this alternative seems to be difficult to achieve.

The other alternative is to regulate (or standardize) the interfaces between service users and the HPS, and to make these interfaces system independent. In this way HConsumers and HProviders will always interact with the HPS in the same way and will provide the payment information in the same format, regardless of their specific payment systems. This alternative has a higher chance of success than the first one because it has no drawbacks or problems, so we will follow it.

We note that, the first two functional requirements (Section 5.1.1 and 5.1.2) already regulate a bit the interactions between the users and the HPS by deciding which user initiates payments and which receives payment acknowledgements.

5.2.2 Hybrid payment service primitives

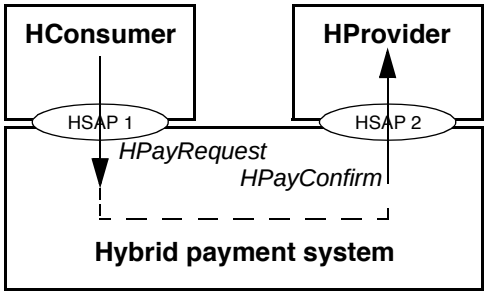


Figure 5.2 *Entity structure and service primitives*

As consequences of the first two functional requirements (Section 5.1.1 and 5.1.2), we introduce an *HPayRequest* service primitive (SP) at the SAP between the HConsumer and HPS (HSAP1), and an *HPayConfirm* SP at the SAP between the HProvider and HPS (HSAP2) (Figure 5.2).

HPayRequest service primitive

The HConsumer executes the *HPayRequest* SP first to initiate a hybrid payment.

The payment initiation indicates the determination of the HConsumer to pay a given amount of money to a specified HProvider. Once the *HPayRequest* SP has been performed, it is not possible for the HConsumer to reverse, cancel or deny its effect. If this SP is implemented reliably then non-repudiation is avoided (see Section 4.2.5). Similarly to existing payment systems, an accepted payment initiation means in general that the payment is successfully performed, and only in very rare situations an accepted initiation does not result into a money transfer. Because only the result of an initiation is relevant (see Section 3.2.2), we abstract from other interactions that are part of the initiations for existing payment systems (e.g., log in to the system, confirmation of the payment initiation, etc.). The interconnected payment systems will process a number of payments for a single hybrid payment, so the execution of a payment initiation may take a longer period of time (e.g., a few seconds) than the processing of a payment by a single existing system.

The parameters of the *HPayRequest* SP provide the necessary information to the HPS to perform the payment. In practice, different payment systems use different sets of parameters in payment initiations (see Section 3.6). Generally, four parameters are identified in all payment initiations: consumer identifier, provider identifier, product transaction identifier and amount of money. Other parameters such as the provider name and URL, product description and URL, success and failure URLs, are implementation specific parameters, so we do not consider them. Table 5.2 summarizes the parameters of the *HPayRequest* SP:

Table 5.2 *HPayRequest parameters*

Parameter	Description
HConsumer ID	Unique identifier (within the HPS) issued by the HPS to the HConsumer. This ID is used to authenticate the HConsumer and to determine the source account of the payment.

Table 5.2 *HPayRequest parameters (Continued)*

Parameter	Description
HProvider ID	Unique identifier (within the HPS) issued by the HPS to the HProvider. This ID is used to identify the HProvider, to determine the destination account of the payment and the HSAP where the payment will be confirmed. The HProvider provides this ID to the HConsumer prior to each payment initiation.
HProductTrans ID	Unique identifier (within the HProvider) of the product transaction generated by the product accounting system of the HProvider. The HProvider provides this ID to the HConsumer prior each payment initiation.
Amount of money	Specifies the amount of money that will be paid in terms of value and currency. It is determined by the product accounting system of the HProvider. The HProvider provides the amount of money to the HConsumer prior to each payment initiation.

The account of the HConsumer and HProvider are stored within their chosen existing payment system, and the HConsumer ID and HProvider ID identify uniquely their accounts within the HPS. The HConsumer has for the existing system an identifier that is used for authentication and this also identifies his/her account. Usually this identifier consists of a <*username, password*> couple. For example: <*johndoe, my@password75*>. Preferably, this identifier should not change with the introduction of the HPS, so consumers will use a user-friendly system (as demanded in Section 4.1.3). This identifier must be unique within the HPS, however. Similarly, the HProvider has for the existing payment system an identifier that is used to identify its account. Usually this identifier consists of a single <*providerid*> field. For example: <*808*>. Preferably, this identifier should not change with the introduction of the HPS, so providers will use a user-friendly system (as requested in Section 4.1.3). This identifier must be unique within the HPS, however.

We propose the following simple method for creating unique HConsumer and HProvider IDs based on existing identifiers:

1. Let n denote the number of existing payment systems, and Ω the set of existing payment systems: $\Omega = \{ PS_i \mid i = 1, n \}$.

2. The set of all consumer identifiers of a payment system PS_i is: $C_i = \{ c_j \mid c_j$ is a consumer identifier of $PS_i, j = 1, \#PS_i$ *consumers* $\}$.

3. The set of all provider identifiers of a payment system PS_i is: $P_i = \{ p_j \mid p_j$ is a provider identifier of $PS_i, j = 1, \#PS_i$ *providers* $\}$.

4. Suppose each PS_i system has a globally unique name or identifier[1], and Δ denotes the set of these identifiers: $\Delta = \{ ID_i \mid ID_i$ is the unique identifier of $PS_i, PS_i \in \Omega, i = 1, n \}$.

5. The unique consumer identifiers for the HPS are created by combining the unique identifier of a PS_i with the consumer identifiers from C_i of this PS_i. The set of HConsumer identifiers is: $HCons = \{ < ID_i, c > \mid ID_i \in \Delta, c \in C_i, i = 1, n \}$.

6. Similarly, the unique provider identifiers for the HPS are created by combining the unique identifier of a PS_i with the provider identifiers from P_i of this PS_i. The set of HProvider identifiers is: $HProv = \{ < ID_i, p > \mid ID_i \in \Delta, p \in P_i, i = 1, n \}$.

Using this method, consumers and providers keep on using their identifiers, which are extended with a unique identifier of their system. Although, there might be two consumers or providers that use identical identifiers for two different systems, this solution guarantees the uniqueness of the HConsumer and HProvider IDs, respectively, because there are no two existing payment systems with the identifiers or names. *An ID identifies uniquely an HConsumer or HProvider and the account of this HConsumer or HProvider.*

1. This is a realistic presumption, because also in the banking world banks have unique SWIFT codes.

Example: Suppose a consumer used so far a payment system called *DigiCoinX*, which was accessible online through the web page of the system by using a <*consumer name*, *password*> combination. After the introduction of the Hybrid Payment System, the PSO of *DigiCoinX* offers two new options for the consumers to initiate hybrid payments: download and install an application (called *DigiCoinX4Hybrid*) or surf to *www.digicoinX4hybrid.com*, fill in the payment information and click on the *Initiate payment* button. In this implementation, the click on this button corresponds to initiating a hybrid payment (executing the HPayRequest SP), and the provided information is sent to the HPS. Figure 5.3 depicts the payment initiation window of the DigiCoinX4Hybrid application in which the consumer filled in the hybrid payment information. The interface available on the www.digicoinX4hybrid.com is designed similarly.

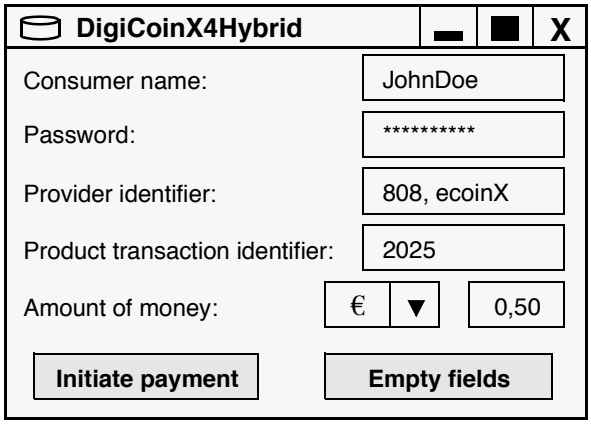


Figure 5.3 *DigiCoinX4Hybrid example*

The conditions on the occurrence of the HPayRequest SP are the following:

- The HConsumer and HProvider IDs must exist and be known to the HPS, so the HPS can authenticate and identify the users (and their accounts). This is one of the security requirements formulated in Section 4.1.6.
- The amount of money must be in the range of the highest minimum and lowest maximum values supported by the systems that are

involved in a hybrid payment, as formulated in the third functional requirement in Section 5.1.3.

- Because the hybrid payment system must act as both pre-paid and post-paid system (see the fourth functional requirement in Section 5.1.4), the verification of the source account can be performed in two ways. In case the HPS is a pre-paid system, the account balance of the HConsumer is verified to determine whether it allows the new payment. If the account balance is too low, the HConsumer needs to transfer some money into its account and initiate the payment again. How the money transfer is performed does not need to be defined in this stage of the design, however. In case the HPS is a post-paid system, then the credit limit of the HConsumer is verified. If this limit is not yet reached or not set at all, the HPS accepts the request. Otherwise, the balance of the credit account must be restored, and then the payment can be initiated again.

HPayConfirm service primitive

The HPS executes the *HPayConfirm* SP after an *HPayRequest* SP has occurred. In this interaction the HPS indicates the HProvider that a payment is completed. To identify the SAP where the SP will be executed, the HPS uses the *HProvider ID* specified in the *HPayRequest* SP. The HProvider is not allowed to refuse or deny a confirmation and uses the confirmation information to (i) make product usage statistics, (ii) trend analysis, (iii) handle HConsumer complaints, (iv) financial accounting (or tax payment), and (v) offer support for audit. Some of these reasons require that the HProvider stores the confirmation information. Because of the low amounts of money transferred by the HPS, the storage period should be relatively short (e.g., one or two months). If this SP is implemented reliably then non-repudiation is prevented (see Section 4.1.6).

Parameters of the *HPayConfirm* SP provide information to the HProvider to be able to serve the paying HConsumer. Again, in practice, many different payment confirmation solutions are applied (see Section 3.6). Generally, two parameters are identified in the confirmations of the studied systems: product transaction identifier and a payment identifier. We can argue whether or not the transferred amount of money should be indicated to the HProvider. Because the payment is initiated by the HConsumer, the HConsumer may modify (reduce)

the amount of money to cheat on the HProvider. If the amount of money is also indicated to the HProvider, it will be sure that the correct amount of money is transferred. Other parameters such as provider identifier, product description, date and time of payment, are implementation specific parameters, so we do not consider them. Table 5.3 summarizes the parameters of the *HPayConfirm* SP considered in this design phase.

Table 5.3 *HPayConfirm parameters*

Parameter	Description
HProductTrans ID	Unique identifier (within the HProvider) of the product transaction. The HProvider supplied the ID to the HConsumer prior to each payment initiation.
Amount of money	Specifies the amount of money that was paid in terms of value and currency. The HProvider supplied the amount of money to the HConsumer prior to each payment initiation.
HPayment ID	Unique payment identifier (within the HPS) generated by the HPS. This ID is also stored in the HPS. It can be used to trace back hybrid payments (e.g., in conflict situations between HConsumer and HProvider) or to offer support for audit procedures (see legal requirement in Section 4.5.1).

Example: Suppose a provider called *scientificlibrary.org* used so far a payment system called *eCoinX*, which sent acknowledgement messages (e.g., using Secure HTTP Post messages) to the *www.scientificlibrary.org:808* address managed by the web server of the provider. These acknowledgements indicated the date and time of the payments, the paid content transaction identifiers, the paid amount of money, and a key that uniquely identified the payments within *eCoinX*. After *eCoinX* became part of the Hybrid Payment System, the PSO of *eCoinX* and the provider agree that acknowledgements of hybrid payments will be sent to the *www.scientificlybrary.org:6870* and will contain 3 parameters: the paid content transaction identifier, the paid amount of money, and a unique payment identifier. The latter parameter identifies the payment in the HPS. The provider's account within *eCoinX* keeps on storing the paid amounts of money. Before a hybrid payment, the provider specifies to the consumer its account identifier, the

content transaction identifier, and the amount of money to be paid (e.g., <1980, eCoinX>, 950102, US$0,95).

5.2.3 Local service interfaces and remote interaction functions

We distinguish between local and remote constraints that control the occurrence of the service primitives. Local service interfaces (LSI) define constraints that control the interactions and exchanged information at each service access point. Remote interaction functions (RIF) define constraints that control the interactions and exchanged information between different service access points.

Figure 5.4 models the local and remote constraints for the occurrence of the SPs by defining the behaviour of HConsumers, HProviders and the HPS for initiating and confirming hybrid payments, and indicating the relationship between an initiation and acknowledgement. The notations used for behaviour modelling are explained in Appendix B of this thesis, and in more details in [1].

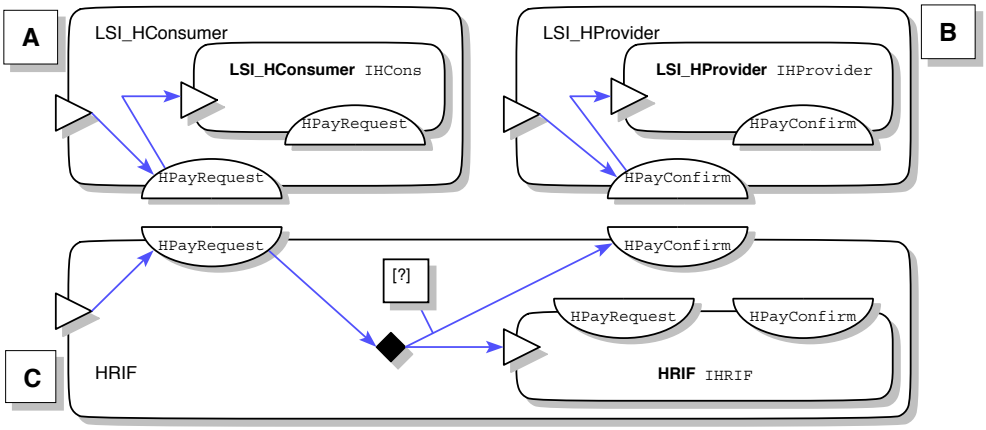


Figure 5.4 *LSIs and RIFs of the hybrid payment service*

Figure 5.4 *A* and *B* depict the behaviours *LSI_HConsumer* and *LSI_HProvider*, respectively. Because the HConsumer needs to initiate the hybrid payments, the behaviour *LSI_HConsumer* contains one interaction contribution in which the HConsumer executes the *HPayRequest* SP. After that the HConsumer may initiate new payments, so the behaviour is recurrently instantiated (*IHCons*). Because the HProvider receives payment acknowledgements, the behaviour

LSI_HProvider contains one interaction contribution in which the HPS acknowledges a completed payment by executing the *HPayConfirm* SP. After that the HProvider may receive new confirmations, so the behaviour is recurrently instantiated (*IHProv*). We note that, an interaction will only occur if both parties contribute to the interaction.

Figure 5.4 *C* depicts the behaviour *HRIF*. This behaviour consists of an interaction contribution to receive a payment initiation first. After that, independently of each other, the initiated and completed payment is confirmed to the HProvider in a second interaction contribution, and the behaviour is recurrently instantiated (*IHRIF*) because the HConsumer may request new payments. The question mark in the *HRIF* behaviour indicates that an initiation is not always followed by an acknowledgement, so the hybrid payment system is not totally reliable. This happens in very rare situations. The reliability and robustness of the HPS (not modelled in the figure) is similar to the reliability and robustness of existing payment systems. These systems claim that no money loss situations can occur, error situations are tolerated, traced back and corrected. In rare events in which a payment initiation is accepted by a HPS, but no acknowledgement follows, the consumer bears the loss of money (see legal requirement in Section 4.5.2).

The relationships between the parameters (not modelled in the figure) of the service primitives at different SAPs are the following:

- HPayRequest.HProductTrans ID = HPayConfirm.HProductTrans ID;
- HPayRequest.Amount of money = HPayConfirm.Amount of money.

5.2.4 Hybrid payments example

Figure 5.5 depicts an example of three successive hybrid payments in a time-sequence diagram. The vertical lines represent the service access points between the users and HPS. Payment initiations can occur one after the other if the HPS allows them, and a confirmation follows each accepted request. Two of the initiated payments are acknowledged to the specified HProvider, while

one is unsuccessful. This figure also indicates that the time difference between the initialization and acknowledgment of the payments may differ.

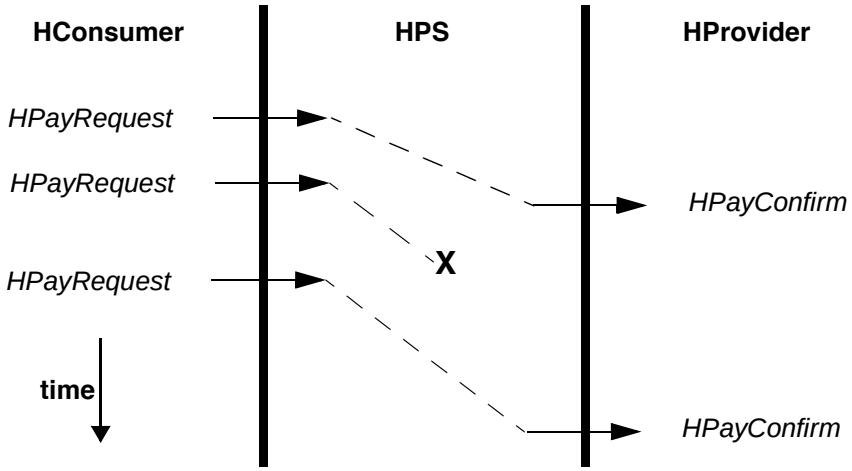


Figure 5.5 *Hybrid payments*

5.3 Interconnection approaches

The question on how to interconnect existing (sub-)systems to create a larger or global system is a traditional one and has been studied in the literature before. Within the context of computer networks, for example, the embedding of heterogeneous, especially incompatible sub-networks (e.g., a reliable and an unreliable sub-network) into a global network was such a question. Two approaches have been identified to solve this question in [3]. One of them is to interconnect available sub-networks just as they are, so without building additional functionality on top of these sub-networks. The other one is to use the sub-networks as building blocks, and adapt their functionality by applying "wrapping" functions such that their services become compatible with each other (e.g., both sub-networks became reliable). The motivation behind this harmonization approach is that interconnection will be easier if all sub-networks exhibit a uniform service level. The latter approach is also followed by the ISO 8648:1988 standard that defines how "real networks" should be interconnected [4].

Inspired by the approaches in [3], we discuss two similar approaches to solve the micropayment system interconnection problem:

- ad-hoc interconnection of payment systems;
- interconnection of uniform payment systems.

5.3.1 Ad-hoc interconnection of payment systems

The first interconnection approach is to take the different payment systems and interconnect them in an ad-hoc manner.

Figure 5.6 depicts the ad-hoc interconnection of two different payment systems. The *HConsumer* is represented within the *HPS* by the *Payer* entity, which uses *payment system A*. The *HProvider* is represented within the *HPS* by the *Payee* entity, which uses *payment system B*. Hence, *Payer* and *Payee* are the actual users of the existing payment systems on behalf of the hybrid system users, and their functionality depends on the underlying payment system. The *Payer* and *Payee* entities realise the mapping between the hybrid payment service and the existing payment services. Payment systems *A* and *B* are interconnected via a *Payment Gateway* (PG) that incorporates the *Payee* and *Payer* entities specific to system *A* and *B*, respectively.

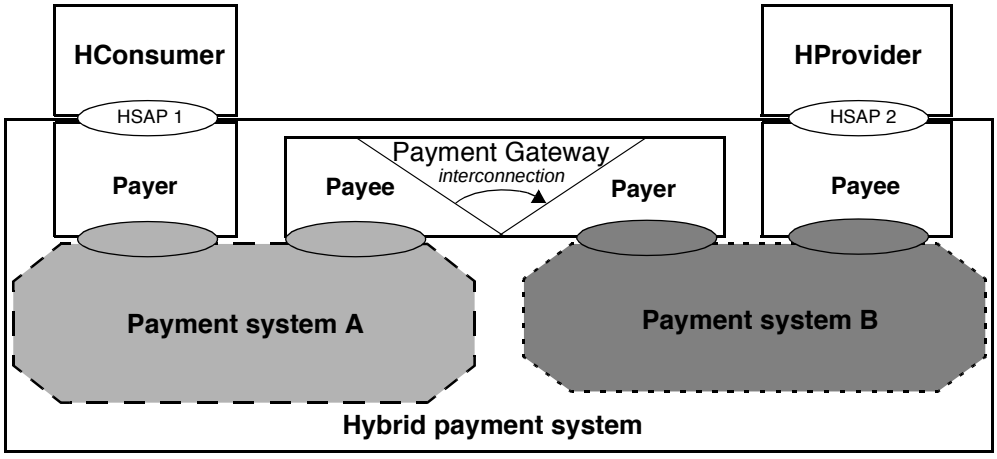


Figure 5.6 *Ad-hoc interconnection of payment systems*

This approach requires the definition of mappings between each pair of interconnected payment systems. This actually means that two mapping rules must be defined for each pair of interconnected payment systems, because the interconnection must be bi-directional and the *PG* should be able to function as payer and payee for each system. The *PG* must determine, generate for and receive from each underlying payment system information structured in different formats. Additionally, the *PG* must also store this information to provide support for auditing (see Section 4.5.1). These operations require a complex functionality and huge data storage capacity from the *PG*.

In case a new payment system is incorporated in the hybrid system built on an ad-hoc manner, mapping rules between the new and already incorporated systems, and the hybrid system need to be defined.

Looking at the number of current micropayment systems and at their relatively short longevity (i.e., their availability on the electronic payments market), and considering future developments (e.g., appearance of new systems, increasing volume of micropayments), the ad-hoc interconnection would only work if the number of payment systems is small, so interconnections could be developed on an individual basis. Despite of the strong point of this approach that allows existing systems to be interconnected without building new functionality on top of them (except the *PG*), a more generic approach is required which can be applied in case of many payment systems.

5.3.2 Interconnection of uniform payment systems

The second interconnection method is to harmonize or (de-)enhance the payment services of existing systems to a uniform level, which we call the *uniform payment service* and interconnect these uniform payment services [5]. A prerequisite for this method is that the harmonization or (de-)enhancement of existing (and future) micropayment systems to the uniform payment service is possible. We call a system that provides the uniform payment service a *uniform payment system*, and a money transfer that is performed by such a system a *uniform payment*.

Figure 5.7 depicts the interconnection of two uniform payment systems that *wrap* the existing micropayment systems *A* and *B* from Figure 5.6. The *Payer*

entity is decomposed into (i) an *HPayer* entity, which is a user of a *Uniform payment system* (1 or 2) and provides the hybrid payment service to the *HConsumer* or supports the *PG*'s interconnection function, and (ii) a *UPayer* entity, which is a user of an *existing payment system* (A or B) and provides the uniform payment service to the *HPayer*. Similarly, the *Payee* is decomposed into entities *HPayee* and *UPayee* such that the *HPayee* provides the hybrid payment service to the *HProvider* or supports the *PG*'s interconnection function, and *UPayee*, which is a user of an existing payment system provides the uniform payment service to the *HPayee*. The composition of internal entities *HPayee*, *HPayer* and *interconnection* of the Payment Gateway is called the *Hybrid Payment Gateway* (*HPG*) in the sequel. The HPG also coordinates the processing of hybrid payments by coordinating and contributing to the initiation of uniform payments.

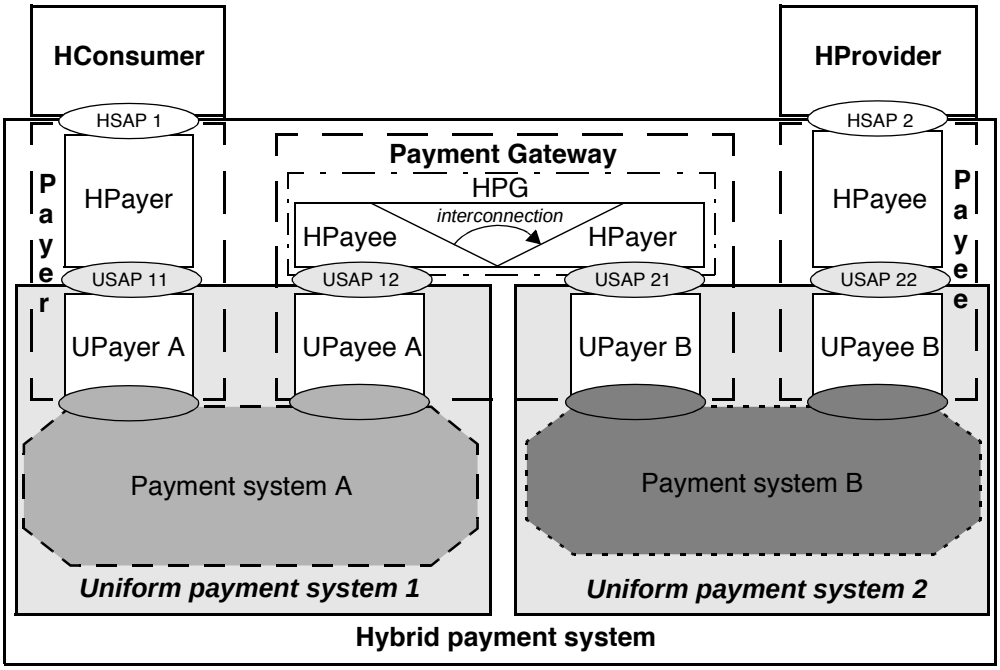


Figure 5.7 *Interconnecting uniform payment systems*

An advantage of this approach over the first one is that only one set of mapping rules needs to be defined for each existing payment system. Another advantage is that the mapping rules between the services of the uniform and the hybrid payment systems (realized by the *HPayer* and *HPayee*) have to be defined only

once. Moreover, if a new payment system is incorporated in the hybrid payment system, only two mapping rules need to be defined. In conclusion, in this interconnection approach, we have seriously reduced the number of mapping rules compared to the previous one.

A generally applicable uniform payment service, which can be provided by existing payment systems, enables a generic and systematic method to interconnect micropayment systems [5]. Moreover, the hybrid system seems to be easier to be designed and realized. Therefore, in the remainder of this thesis, we consider the interconnection of uniform payment systems.

5.4 Functional requirements of the uniform payment system

In this section, the functional requirements for the uniform payment system are formulated. These requirements define how uniform payments are initiated and acknowledged, what currencies and payment values are supported, and what the usage conditions are.

The chosen interconnection method requires the (de-)enhancement of existing payment systems to a uniform payment service level, followed by the interconnection of uniform payment systems afterwards. To achieve the first requirement, we need to find an intermediary payment service level (i.e., the uniform payment service) situated between the hybrid and existing payment services, such that it can be provided by as many existing payment systems as possible. The challenge is now whether this uniform payment service should be closer to or resemble more the hybrid payment service or the existing payment services? In the first option, the uniform payment service is based on the most common characteristics of existing payment systems, all payment system (implementation) specific characteristics are hidden within the uniform payment systems (i.e., are handled during the (de-)enhancement), and the protocol layer on top of the uniform payment systems handles the interconnection only. In the second option, only some of the payment specific characteristics are handled within the uniform payment systems, and the protocol layer needs to deal with the remaining different characteristics and handle the interconnection. The first option allows a clear separation of concerns and a systematic design approach, while the second option offers only a partial solution. In conclusion, we follow the first option, so the uniform payment service is

designed based on the most common practices or characteristics of existing payment systems.

For the functional requirements we also consider some of the hard requirements formulated in Chapter 4.

5.4.1 Uniform payment initiations

Requirement: *The uniform payment service requires consumers to initiate payments (in our case an HPayer plays the role of a consumer).*

The motivation for this requirement is the requirement of PSOs that existing payment systems should change as little as possible (see Section 4.3.1), and the fact that most payment systems support consumer initiated payments (see Section 3.5). Furthermore, the consequence of supporting jointly initiated payments would not only be that the majority of existing payment systems must be changed, but the interconnection of systems that support jointly initiated payments is more complicated than for systems, which support consumer initiated payments.

5.4.2 Uniform payment acknowledgements

Requirement: *The uniform payment service acknowledges completed payments to providers (in our case an HPayee plays the role of a provider).*

The motivation for this requirement is similar to the one on payment initiations: existing payment systems should change as little as possible, and most systems support this acknowledgement type. HPayees need the acknowledgements to continue processing an initiated hybrid payment: the HPG needs to initiate the subsequent uniform payment, and the HPayees need to confirm the hybrid payment to the HProviders.

One of the requirements of the consumers is to remain anonymous with respect to providers (see Section 4.1.4), so *payment confirmations must not reveal the consumers' identity to providers.*

5.4.3 Supported payment values and currencies

Requirement: *The uniform payment service does not prescribe a fixed minimum or maximum payment value, nor prescribes a fixed currency.*

Fixed limits on payment values are not imposed, because of the variance in minimum and maximum values (and currencies) supported by existing systems. In case global minimum and maximum payment values would be imposed, the range of supported hybrid payment values would be reduced significantly. Instead, the uniform payment service inherits the minimum and maximum payment values and the supported currencies from the underlying payment systems.

5.4.4 Usage conditions

Requirement: *The uniform payment service supports pre-paid or post-paid payments.*

From the consumer's (in our case HPayer's) viewpoint, one of the most important differences between pre-paid and post-paid systems, is how it obtains the authorization to use a payment system. Because of that difference, the financial risks and the responsibilities of users and systems involved, it is unlikely that a pre-paid system can function as a post-paid system, or the other way around. That is why we define two types of uniform payment systems: pre-paid (denoted as UPS_{pre}) and post-paid (denoted as UPS_{post}). A UPS inherits this characteristic from an underlying payment system. In the sequel, the term *UPS* is used in case the pre-paid or post-paid characteristic of a UPS is considered irrelevant.

5.4.5 Summary

Table 5.4 summarizes the uniform payment systems' functional characteristics.

Table 5.4 *Functional characteristics of the uniform payment systems*

Micro-payment system	Consumer (HPayer) initiated payments	Jointly initiated payments	Consumer (HPayer) ack. payments	Provider (HPayee) ack. payments	Minimum value	Maximum value	Currency	Pre-paid	Post-paid
UPS_{pre}	X	-	-	X	variable	variable	variable	X	-
UPS_{post}	X	-	-	X	variable	variable	variable	-	X

5.5 Uniform payment service design

Because two types of uniform payment systems are introduced in Section 5.4.4, two payment services should be defined. In this section, however, one uniform payment service will be defined due to the many similarities of these uniform systems. The pre-paid and post-paid characteristics of the uniform payment systems will be taken into account and mentioned where they are relevant.

The users of the uniform payment services (i.e., HPayers and HPayees) are already introduced in Section 5.3.2, so the service definition consists of defining the service primitives, their parameters, and the constraints that control their occurrences.

5.5.1 Uniform payment service primitives

According to the first two functional requirements, the uniform payments are initiated by HPayers and are acknowledged to HPayees. To model these payment interactions, we introduce two service primitives (SPs): the *UPayRequest* primitive, which occurs at the SAP between the HPayer and the UPS, and the *UPayConfirm* primitive, which occurs at the SAP between the HPayee and the UPS.

UPayRequest service primitive

The HPayer executes the *UPayRequest* SP first to initiate a uniform payment. Prior accepting an initiation, the UPS needs to authenticate the HPayer and decide whether to authorize the payment or not. The authorization procedure depends on the type of the UPS, which is pre-paid or post-paid.

The payment initiation indicates the determination of the HPayer to pay a certain amount of money to a specified HPayee. In case the UPS accepts the request, this cannot be reversed or cancelled, and the amount of money is transferred. In other words, an accepted initiation means that the payment was successfully completed. We note, that the *UPayRequest* SP abstracts from a number of interactions that are commonly found in implementations of this service primitive, e.g., log in to the system and confirmation of the payment initiation.

Parameters of the *UPayRequest* SP provide the necessary information to the UPS to be able to perform the initiated payment. We observed before that generally four parameters are identified in each payment initiation (see Section 3.6): the consumer identifier, provider identifier, product transaction identifier and amount of money. Because there are no products exchanged between the HPayer and HPG, or between the HPG and HPayee, we call the product transaction identifier a context identifier, because this allows the HPG or HPayee to identify the context or reason of the payment. Other parameters could be the provider name and URL, product description and URL, success and failure URLs, etc. But since this information is not vital for performing payments, we abstract from it. Table 5.5 presents the definitions of the parameters needed for each uniform payment initiation.

The accounts of HPayers and HPayees are stored and maintained by the existing payment systems, and the HPayer ID and HPayee ID identify these accounts uniquely within the UPS. HPayers and HPayees will inherit the consumer and provider identifiers, respectively from the existing payment systems. This means that the HPayer and HPayee IDs for a given PS_i system will be taken from the C_i and P_i sets, respectively (see Section 5.2.2). We note that, this is only an example of how the account identifiers can be created, other information can also be added to the account identifiers.

Table 5.5 *UPayRequest parameters*

Parameter	Description
HPayer ID	Unique identifier (within the UPS) issued by the UPS to the HPayer. This ID is used to authenticate the HPayer, and to determine the source account of the payment.
HPayee ID	Unique identifier (within the UPS) issued by the UPS to the HPayee. This ID is used to identify the HPayee, to determine the destination account of the payment and the address where the payment will be confirmed.
Context ID	Unique identifier (within the HPayee) that characterizes the context of the payment. Its role is similar to the HProduct-Trans ID, but in case of uniform payment systems no product transactions take place between HPayers and HPayees. This ID is generated by the HPayer or HPayee.
Amount of money	Specifies the amount of money that will be paid in terms of value and currency.

The conditions on the execution of the *UPayRequest* SP are the following:

- the HPayer and HPayee IDs must exist and be known to the UPS, so the UPS can authenticate and identify the users and their accounts;
- the amount of money must be between the minimum and maximum payment values supported by the UPS, and the currency must also be supported by the UPS (see functional requirement in Section 5.4.3).
- an UPS_{pre} verifies the source account balance of the HPayer to determine whether it allows the new payment. In case the account balance is too low, the HPayer needs to transfer some money into its accounts and initiates the payment again. How the money transfer is performed does not need to be defined in this stage of the design (see functional requirement in Section 5.4.4);
- an UPS_{post} verifies the credit limit of the HPayer. If this limit is not reached yet or not set at all, the UPS_{post} accepts the initiation. Otherwise, the balance of HPayer's account must be restored first, and then the payment can be initiated again (see functional requirement in Section 5.4.4).

UPayConfirm service primitive

The UPS executes the *UPayConfirm* SP after an *UPayRequest* SP has occurred. In this interaction the UPS indicates to the HPayee the completion of a uniform payment. To identify the SAP where the SP will be executed, the *UPS* uses the *HPayee ID* specified in the *UPayRequest* SP. The HPayee is not allowed to refuse nor deny a payment acknowledgement to prevent non-repudiation.

Parameters of the *UPayConfirm* SP provide information to the HPayee to be able to initiate a second uniform payment or to acknowledge the hybrid payment to the Provider. We observed before that the product transaction and payment identifiers are always provided in acknowledgements. Again, we call the product transaction identifier a context identifier. We can argue whether or not the transferred amount of money should be indicated to the HPayee. Because the payment is initiated by the HPayer, the HPayer could modify (decrease) the amount of money to cheat on the HPayee. In case the amount of money is also indicated to the HPayee, the HPayee is able to verify that the correct amount of money is transferred. Other parameters could be the payee identifier, product description, date and time of payment, etc. Since this information is not essential for the HPayee, we abstract from it. The parameters of payment confirmations are defined in Table 5.6.

Table 5.6 *UPayConfirm parameters*

Parameter	Description
Context ID	Unique identifier (within the HPayee) that characterizes the context of the payment. This ID is generated by the HPayer or HPayee.
Amount of money	Specifies the amount of money that was paid in terms of value and currency.
UPayment ID	Unique payment identifier (within the UPS) generated by the UPS. This ID must also be stored in the UPS. It can be used to trace back hybrid payments (e.g., in conflict situations between HPayer and HPayee) or to offer support for audit procedures (see legal requirement in Section 4.5.1).

The condition on the execution of this service primitive is the completion of the money transfer from the given source account to the destination account.

5.5.2 Local service interfaces and remote interaction functions

We distinguish between local (LSI) and remote (RIF) constraints that control the occurrence of the service primitives. Figure 5.8 models the local and remote constraints for the occurrence of the SPs by defining how *HPayers* initiate, *HPayer* receive acknowledgements of uniform payment, and what the relationship between an initiation and acknowledgement is. The notations used for behaviour modelling are explained in Appendix B of this thesis, and in more details in [1].

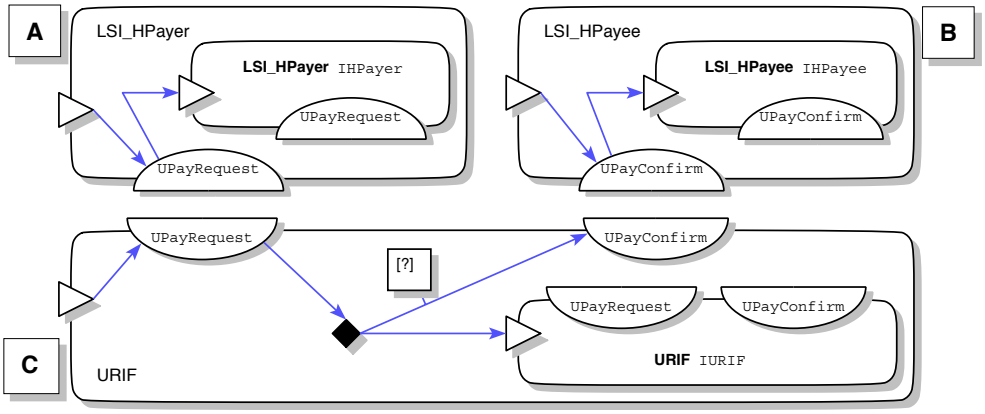


Figure 5.8 *LSIs and RIFs of the uniform payment service*

Figure 5.8 *A* and *B* depict the *LSI_HPayer* and *LSI_HPayee* behaviours, respectively. Because the HPayer needs to initiate the hybrid payments, the behaviour *LSI_HPayer* contains one interaction contribution in which the HPayer executes the *UPayRequest* SP. After that, the HPayer may initiate new payments, so the behaviour is recurrently instantiated (*IHPayer*). Because the HPayee receives payment acknowledgements, the behaviour *LSI_HPayee* contains one interaction contribution in which the UPS acknowledges a completed payment by executing the *UPayConfirm* SP. After that the HPayee may receive new confirmations, so the behaviour is recurrently instantiated (*IHPayee*). We note that, an interaction will only occur if all involved participants contribute to the interaction.

Figure 5.8 *C* depicts the behaviour URIF that defines the relationship between an initiation and acknowledgement. The behaviour of the UPS consists of an interaction contribution with the HPayer first. After that, independently of each other, the requested and completed payment is confirmed to the HPayee in a second interaction contribution, and the behaviour is recurrently instantiated (*IURIF*) because the HPayer is allowed to initiate a new payment. The question mark in the *URIF* behaviour indicates that an acknowledgement does not always follow an initiation, so the hybrid payment system is not totally reliable. This happens in very rare situations.

The reliability and robustness of the *UPSs* (not modelled in the figure) is determined by the reliability and robustness of existing payment systems. These systems claim that payments do not fail, error situations are tolerated, traced back and corrected, so no money loss situations can occur. In the rare event that a payment initiation is accepted by a *UPS* but no acknowledgement follows, the *HPayer* bears the loss of money. This is symbolized by the question mark in the *URIF* behaviour indicates.

The relationships between the parameters (not modelled in the figure) of the service primitives at different SAPs are the following:

- UPayRequest.Context ID = UPayConfirm.Context ID;
- UPayRequest.Amount of money = UPayConfirm.Amount of money.

5.5.3 Uniform payment examples

Figure 5.9 depicts an example of three successive uniform payments. Payment initiations can occur one after the other, if the UPS allows them, and a confirmation follows each accepted initiation. Two of the initiated payments are acknowledged to the specified *HPayee*, while the third payment failed. This figure also indicates that the time difference between the initialization and acknowledgment of the payments may differ.

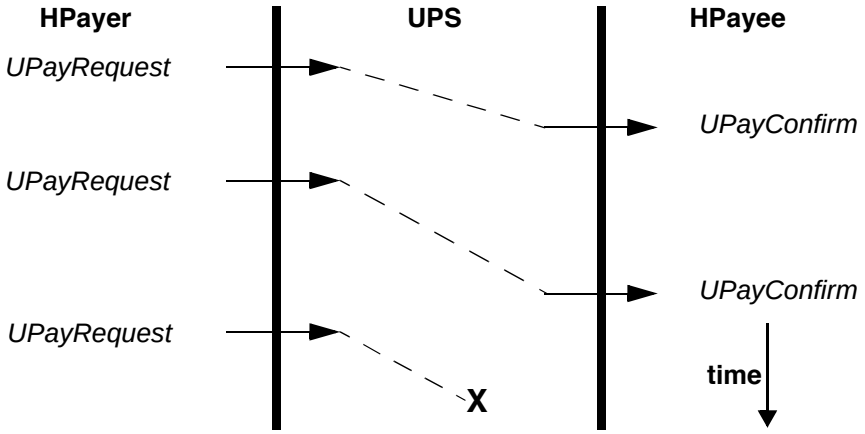


Figure 5.9 *Uniform payment examples*

5.6 Compliance with the uniform payment service

The uniform payment service is defined based on the most common characteristics and practices of existing payment systems, so most of these systems should be able to provide the uniform payment service. But because of the big variety experienced among existing systems, the doubt arises whether all these systems can provide or can comply with the uniform payment service or not. To answer this question, the following subsections present a compliance discussion. They will compare the payment initiations and acknowledgements in case of the uniform and existing payment services, and propose modifications for the existing services, if there are differences noticed and they cannot comply with the uniform service. We note that, the supported payment values and currencies, and the usage conditions do not create compliance problems as the uniform payment services inherit these characteristics from the existing payment services.

Payment initiations

Existing payment systems require consumer (in our case *UPayer*) or jointly (*UPayer* and *UPayee*) initiated payments. The UPS performs consumer (in our case *HPayer*) initiated payments, so according to the summary presented in Section 3.5 most payment systems will be able to provide the uniform payment service.

We distinguish two cases for the systems that require jointly initiated payments based on the time difference between the initiations. In the first case, both involved users initiate the payment within a few seconds. This case can easily be adapted to support consumer initiated payments, because the *UPayer* can interact with the *UPayee* (via a different underlying system than the UPS) to determine the payment initiation information based on the information received from the *HPayer*. Subsequently, the *UPayer* and *UPayee* can initiate jointly the payment for the existing system. This is achievable because the *UPayer* and *UPayee* provide the (de-)enhancement of the existing payment services to the uniform payment service (see Figure 5.7).

In the second case, the *UPayer* may initiate a payment weeks or months after the *UPayee* has initiated the payment. Typically, the *UPayee* initiates only one payment for every product the provider sells. A payment is processed each time the *UPayer* provides the remaining part of the payment information. Such systems (e.g., Bitpass, Click&buy) cannot be (de-)enhanced to provide the uniform payment service because these systems store fixed product transaction identifiers and prices, while in our case the product transaction, context identifiers, and the amount of money are generated in real-time immediately after consumers requested the products. Additionally, these systems support payments for individual (atomic) pieces of content, more than one piece of content cannot be paid with one payment. These systems need, therefore, to change such that they support consumer initiated payments or fall into the first case of jointly initiated payments. For this, they could apply one of the following two modification proposals.

One proposal is to allow both the *UPayer* and *UPayee* to initiate in real-time together payments after the *UPayer* and UPayee determine the payment initiation information. The other proposal is to allow only *UPayers* to initiate payments, so they provide alone the necessary payment information to the payment system. In either case, it is necessary to be able to provide the complete payment information in real-time, so this information must not be in advance partially stored in the payment systems.

> *Example*: Suppose a payment system requires jointly initiated payments. Because the PSO wants its system to become part of the hybrid payment system, it opens a new port on the web server of

the system, where the web server will receive payment initiations (e.g., secure HTTP Request messages with four parameters) from both users. In this way, the PSO's system can be enhanced to provide the uniform payment service, thus it can become part of the hybrid payment system.

Payment acknowledgements

In Section 3.2.2 we identified two groups of payment systems: systems that perform only payments and systems that do more than payments. In the first group, payment acknowledgements are provided predominantly to both users or only to providers. The UPS, however, provides acknowledgements only to providers, i.e., *HPayees*. Accordingly, systems that provide acknowledgements only to *UPayees* comply automatically with the uniform service. Other systems that provide double-acknowledgements also comply with the uniform service, because *UPayers* will ignore the acknowledgements and not forward it to the *HPayers*. This is possible because *UPayers* provide the mapping between the service of existing systems and the uniform service. Finally, systems that provide acknowledgements only to *UPayers*, can also comply with the uniform service, because *UPayers* can forward the acknowledgement over to the *UPayees* that acknowledge the payment to the *HPayees*.

In the second group, however, there are systems that provide absolutely no acknowledgements to their users (e.g., Click&buy) and provide the paid product to the consumers. This means that these systems cannot provide the uniform service unless their functionality is modified. A possible modification would be to provide an acknowledgement to the *UPayer* or *UPayee*. If acknowledgements are provided to the *UPayees*, then they can forward it to the *HPayees*. If acknowledgements are provided to the *UPayers*, then they can forward them to the *UPayees*, which will confirm the payments to the *HPayees*.

Example: Suppose a payment system provides double payment acknowledgements. Because the PSO wants its system to become part of the hybrid payment system, it makes an agreement with the providers that from now the providers will receive an acknowledgement. The acknowledgements can, for instance, be a secure

sent via HTTP Post messages, which carry the three parameters of the acknowledgement.

Summary

We found in our studies fourteen systems out of seventeen (i.e., 82%) that comply with the uniform payment service, because *UPayers* and *UPayees* are able to provide the mapping between the existing and uniform payment services. Three systems (i.e., Click&buy, Bitpass and WebCent) need to make modifications.

An interesting observation is that existing payment systems have more often compliance problems with the way uniform payments are initiated rather than with the way they are acknowledged.

5.7 Conclusions

This chapter developed the hybrid payment service, i.e., the external behaviour of the hybrid payment system and an interconnection method for payment systems.

The design of the hybrid payment service was guided by the functional characteristics of existing payment systems and hard requirements, so this service resembles current existing payment services, but it also addresses to the hard requirements of end-users, stakeholders, financial authorities and governments.

This chapter also introduced a generic interconnection method that supports the systematic interconnection of existing and possible future payment systems. This method requires the de-enhancement of micropayment systems towards a minimal uniform service level. In this way, the number of mapping rules and the amount of information that must be stored is limited, which makes this method highly scalable, and the design and realization of the Hybrid Payment Gateway becomes much easier.

The design of the uniform payment service was based on the common functional characteristics of existing payment systems. After a compliance analysis

of existing systems, we concluded that the majority of the systems can provide the uniform payment service. We proposed functional modifications for the other systems.

The uniform payment service could lead the design of future electronic payment systems such that new systems can be interconnected easily with existing systems. In this way, the uniform payment service, possibly extended with interactions that have only local significance, could become a de facto standard for micropayment systems.

The hybrid and uniform payment services are rather similar, which has two benefits. First, the impact of the payment services on their users and on the existing payment systems is minimal, so little changes are needed in their habits or functionality. Second, the design of the hybrid payment protocol becomes much easier because the gap between these services that needs to be bridged by this protocol is not very large.

5.8 References

[1] Vissers, C.A. et al., The architectural design of distributed systems, Lecture notes for The Design of Telematics Systems course at the University of Twente, Enschede, November 2000
[2] Interaction System Design Language, http://isdl.ctit.utwente.nl/
[3] van Sinderen, M. and Vissers, C.A., An architectural model for network interconnection, In the Proceedings of the EUTECO - European Teleinformatics Conference, 1983
[4] ISO 8648:1988, Information processing systems -- Open Systems Interconnection -- Internal organization of the Network Layer, May 1999
[5] Párhonyi, R. et al., An interconnection architecture for micropayment systems, In the Proceedings of the 7th International Conference on E-Commerce (ICEC 2005), Xi'an, China, August 2005
[6] Weber, R., Chablis - Market Analysis of Digital Payment Systems, Technical Report TUM-I9819, Technical University of Munich, Munich, August 1998

Chapter 6

Hybrid payment protocol

This chapter presents the hybrid payment protocol design.

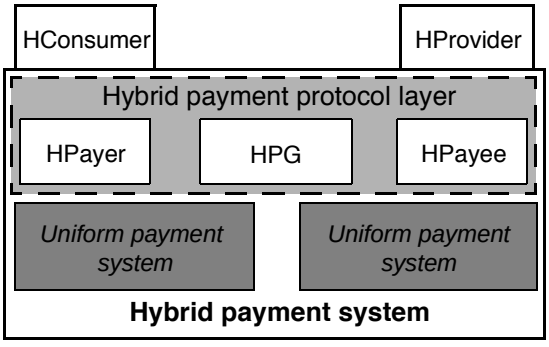


Figure 6.1 *Hybrid payment system architecture*

Figure 6.1 illustrates the architecture of the hybrid payment system (HPS). The HConsumer and HProvider are the users of the hybrid payment system. The protocol layer interconnects the underlying uniform payment systems (UPS) and consists of the HPayer, Hybrid Payment Gateway (HPG), and HPayee protocol entities. The HPayer and HPayee protocol entities also interact with the HConsumer and HProvider to provide them the hybrid payment service.

The starting points for the protocol design are the hybrid and uniform payment services, as defined in Section 5.2 and 5.5. Section 6.1 discusses the differences between these services and identifies the protocol functions that are needed to bridge the gap between these services. These functions are then grouped into protocol elements. Section 6.2 designs these elements and assigns them to protocol entities. Section 6.3 presents the protocol messages, i.e., the protocol data units that are exchanged. Sections 6.1 through 6.3 consider the normal (error-free) protocol behaviour. Section 6.4 discusses the robustness, security

and optimization aspects of this protocol. Section 6.5 discusses whether a single or multiple HPGs should be used. Section 6.6 concludes this chapter.

6.1 Identification and grouping of protocol functions

According to the design methodology followed in this thesis, the functionality of the hybrid payment protocol will be defined from an external perspective. This means that it will be defined in terms of (i) the hybrid payment service primitives, (ii) the uniform payment service primitives and (iii) the relationships between these primitives [1]. The service primitives of the two payment services were already defined in Sections 5.2.2 and 5.5.1. To understand the relationships between the primitives of both services, this section analyses the differences between the services.

At first glance, the services of the hybrid and the uniform payment services are rather similar. Similarities are, for instance, the consumer initiated and provider acknowledged payments, the semantics of the corresponding service primitives, and the number and semantics of the parameters of the corresponding service primitives (e.g., the *HPayRequest* and *UPayRequest* service primitives have each four parameters with similar semantics).

Figure 6.2 focuses on the part of Figure 6.1 that shows the hybrid payment protocol layer including the service access points, the service primitives (SPs) of the hybrid and uniform payment services and the parameters of these primitives.

HSAP1 and HSAP2 are the two access points where an HConsumer and an HProvider (not shown in the figure) interact with the hybrid payment system. At HSAP1 the *HPayRequest* and at HSAP2 the *HPayConfirm* service primitive occurs. USAP11 and USAP12 are two access points of the *C-UPS* (i.e., the UPS on the HConsumer's side), where the *UPayRequest* and *UPayConfirm* primitives occur. Similarly, USAP21 and USAP22 are two access points of the *P-UPS* (i.e., the UPS on the HProvider's side), where the *UPayRequest* and *UPayConfirm* primitives occur.

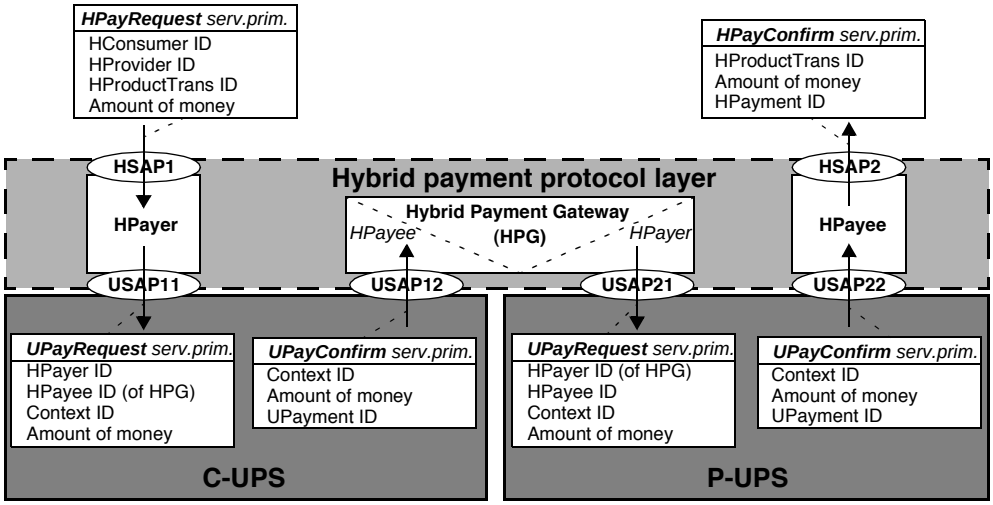


Figure 6.2 *Service primitives, service access points and UPSs*

The differences between the related parameters of the primitives of the two payment services shown in Figure 6.2 are the following:

- the identifiers of the source accounts from which money will be transferred: the HConsumer ID and HPayer ID;
- the identifiers of the destination accounts to which money will be transferred: the HProvider ID and HPayee ID;
- the identifiers of the product transaction and the context of uniform payments: HProductTrans ID and Context ID;
- the amounts of money specified in the *HPayRequest* and *UPayRequest*, and in two consecutive *UPayRequest* service primitives;
- the payment identifiers of the hybrid and uniform payment systems: HPayment ID and UPayment IDs.

These differences will be further discussed in the following subsections. The discussion results in the identification and description of the protocol functions that bridge the gap between (or map) the hybrid and uniform payment services.

We note that, this approach focuses on functional aspects of the protocol. Other aspects, such as robustness, security and optimization will be discussed in Section 6.4.

6.1.1 Source account identifiers

Difference: *the source account identifiers of the HPayRequest and UPayRequest SPs are different.*

The HConsumer ID parameter of the *HPayRequest* SP is a globally unique identifier, which identifies the account of an HConsumer within the HPS. In fact this account is stored within the existing payment system chosen by the HConsumer.

The HPayer ID parameter of the *UPayRequest* SPs is unique for a specific UPS, and identifies a source account within this specific UPS. This account is actually stored within the existing system wrapped by this UPS.

The entities of the protocol layer need to determine the source account identifiers for the UPSs:

1. The HPayer ID for the C-UPS is determined based on the HConsumer ID, because there is a 1:1 relationship between these IDs and they identify the same source account (see Section 5.2.2).
2. The HPayer ID for the P-UPS is the source account identifier of the HPG. This identifier is determined depending on the payment system (P-UPS) used by the HProvider. As explained in Section 5.2.2, this system is identified by the HProvider ID.

The HConsumer ID is also used for another purpose: authentication (or the checking the HConsumer's authenticity). One of the conditions, mentioned in Section 5.2.2, for accepting a payment initiation is that the HConsumer is authenticated.

The following functions are identified to map the source account identifiers:

- determine HPayer IDs for the UPSs;
- authenticate HConsumer.

6.1.2 Destination account identifiers

Difference: *the destination account identifiers of the HPayRequest and UPayRequest SPs are different.*

The HProvider ID parameter of the *HPayRequest* SP is a globally unique identifier, which identifies the account of an HProvider within the HPS. In fact this account is stored within the existing payment system used by the HProvider.

The HPayee ID parameter of the *UPayRequest* SPs is unique for a specific UPS, and identifies a destination account within that UPS. Actually, this account is stored within the existing payment system wrapped by that UPS.

The entities of the protocol layer need to determine the destination account identifiers for the underlying UPSs:

1. The HPayee ID for the C-UPS is the destination account identifier of the HPG. This identifier is determined depending on the payment system (C-UPS) used by the HConsumer. As explained in Section 5.2.2, this system is identified by the HConsumer ID.
2. The HPayee ID for the P-UPS is determined based on the HProvider ID, because there is a 1:1 relationship between these IDs and they identify the same destination account (see Section 5.2.2).

The HProvider ID is also used for another purpose: identification (or the checking whether the HProvider's account exists). One of the conditions, mentioned in Section 5.2.2, for accepting a payment initiation is that the HProvider is identified.

The following functions are identified to map the destination account identifiers:

- determine HPayee IDs for the UPSs;
- identify HProvider.

6.1.3 Product transaction and context identifiers

Difference: *the product transaction identifier of the HPayRequest and HPay-Confirm, and the context identifiers of the UPayRequest and UPayConfirm SPs have different semantics.*

Prior to a payment, the HConsumer receives from the HProvider an identifier of the product transaction (HProductTrans ID). The HConsumer initiates the hybrid payment using this identifier. The hybrid payment system returns this identifier in a confirmation to the HProvider (Figure 6.2). The HProvider will then know which product has been paid by the HConsumer and will deliver it.

To transfer the HProductTrans ID through the hybrid payment system, the Context ID parameters of the UPSs will be used. A function of the protocol layer is responsible for mapping the HProductTrans ID to the Context IDs in the HPayer and HPayee PEs. This function also needs to relate in the HPG PE the Context ID transferred by the C-UPS and to the Context ID, which will be further transferred by the P-UPS.

The following function is identified to bridge the difference between the product transaction and context identifiers:

- relate the HProductTrans ID onto the Context IDs for the UPSs.

6.1.4 Amounts of money

Difference: *the amount of money specified in an HPayRequest and a UPayRequest SP, and of two consecutive UPayRequest SPs may differ.*

In the design of the hybrid payment system was decided that there is no global minimum and maximum payment values, nor fixed set of currencies supported (Section 5.1.3). A similar decision was taken in the design of the uniform payment system, which inherit the supported payment values and currencies from the existing payment systems they wrap (Section 5.4.3). As a consequence, the HPS and UPSs may differ with respect to the supported amounts of money.

The protocol layer needs to solve the differences between the amounts of money. This means that it needs to verify whether an amount of money specified in the hybrid payment initiation can be transferred by the C-UPS and P-UPS. If one of the UPSs does not support the specified amount of money, then the HPS cannot transfer this amount of money and the hybrid payment initiation is rejected. If the currency is not supported, the protocol layer needs to exchange the currency and again verify whether the exchanged amount of money is supported.

The following function is identified to bridge the difference between the amounts of money to be transferred:

- verify whether the UPSs support the specified amount of money and exchange the currency, if necessary.

6.1.5 Payment identifiers

Difference: *the payment identifiers of the HPayConfirm and UPayConfirm SPs have different semantics.*

The HPayment ID is a unique identifier generated by the HPS that allows an HProvider to trace back hybrid payments within the HPS and provides support for auditing (see requirement in Section 4.5.1). As explained in Section 5.2.2, the HProvider stores this HPayment ID. The UPayment ID serves a similar purpose, but is generated by a specific UPS to audit and trace back payments for this specific UPS.

The protocol layer receives UPayment IDs in payment confirmations from the UPSs. The protocol layer then needs to generate the unique HPayment ID and provide it to the HProvider in confirmations. To enable the trace back and auditing of payments, the HPayment ID and the UPayment ID of the P-UPS should be related to each other or stored together in the protocol layer. The storage of the two identifiers in the protocol layer is unnecessary since the HProvider stores the HPayment ID and the P-UPS the UPayment ID. The consequence of this is that the HPayment ID should be generated such that it is related to the UPayment ID provided by the P-UPS. Additionally, for the tracing back of payments and auditing purposes, payment information that relates

two uniform payments (performed for a hybrid payment) to each other need to be stored by the protocol layer.

Since the volume of payments may be high, the protocol entities may need to store large amounts of payment information. Because of the low payment values, however, the storage period could be relatively short (e.g., one or two months)[1].

The following functions are identified to map the payment identifiers:

- generate HPayment ID;
- store payment information.

6.1.6 Grouping of protocol functions into protocol elements

To improve the readability and understandability of the protocol design, the identified protocol functions will now be grouped into protocol elements and organized in a systematic way. Protocol elements are functional building blocks that contain closely related protocol functions [1].

The following protocol elements are identified:

- *Account identifier determination*: contains the functions that determine the HPayer and HPayee identifiers;
- *HConsumer authentication*: contains the function responsible for the authentication of the HConsumer;
- *HProvider identification*: contains the function responsible for the identification of the HProvider;
- *Amount of money verification*: contains the function that verifies whether the UPSs support the amount of money specified in the *HPayRequest* SP, and exchanges the currency, if necessary;

1. It is important to notice that payment systems that transfer amounts of money larger than micropayments need to store the payment information for several years.

- *Payment information transfer*: contains the functions responsible for relating the HProductTrans ID to the Context IDs and generating the HPayment ID;
- *Payment information storage*: contains the functions responsible for storing payment information.

6.2 Design and assignment of protocol elements

This section designs the previously identified protocol elements, specifies their relationship as some elements are mutually dependent, and assigns them to protocol entities. The design consists of identifying, describing, comparing different alternative solutions for realizing these protocol elements, and selecting the most appropriate solution. The assignment of the elements to protocol entities should be made such that the functionality of the HPayer and HPayee PEs is kept as simple as possible to ease the implementation of the consumer and provider software. These two PEs should not be overloaded with new functionality originating from the introduction of the HPG (e.g., identification of HProviders).

Throughout the design of the protocol elements a number of assumptions will be made. An initial assumption is that the HPG PE maintains a database table called *UPS Table* that contains information about UPSs. The structure of this table is built up stepwise throughout the design. A second assumption is that a new underlying system called *Data Transfer System* (DTS) will be involved besides the UPSs to support information exchange between the protocol entities. The DTS is an alternative if the UPSs cannot support the required information exchange. The Internet is a good example for the DTS. The information that needs to be exchanged will be defined step-wise throughout the design as well. The precise behaviour of the DTS will be defined in Section 6.3.2. Other assumptions will be formulated in the design.

Figure 6.3 illustrates the mentioned two initial assumptions.

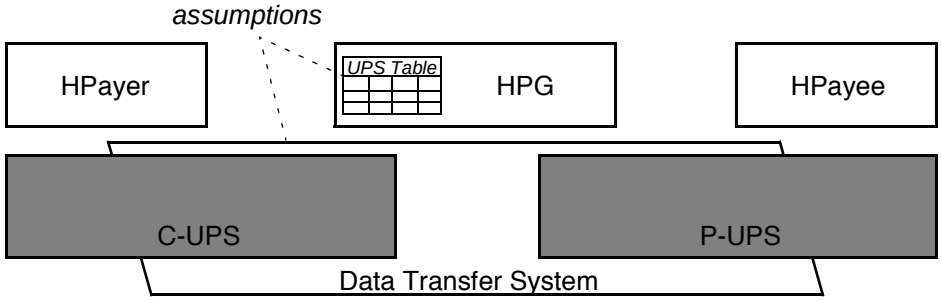


Figure 6.3 *Initial assumptions*

6.2.1 Account identifier determination

This protocol element is responsible for determining the HPayer and HPayee IDs for the C-UPS and P-UPS. It actually determines the source and destination accounts of the *UPayRequest* SPs based on the source and destination accounts specified in the *HPayRequest* SP. If any of the account identifiers cannot be determined, the hybrid payment cannot be processed and the hybrid payment initiation will be rejected.

Because of the way the HConsumer and HProvider IDs are created (Section 5.2.2), the mapping between these IDs and the HPayer and HPayee IDs (Section 5.5.1) is simple:

- The HPayer and HPayee IDs will be the same as the consumer and provider identifiers used by the existing payment systems. Therefore, they take the following forms: $< c >$ or $< p >$, where $c \in C$, $p \in P$, C and P represents the set of consumer and provider identifiers, respectively, of a payment system.
- Since the HConsumer and HProvider IDs are created by combining the unique identifier of the existing payment system (*ID*) with the identifier of the consumer or provider, they take the following forms: $< ID_c, c >$ and $< ID_p, p >$, where ID_c and ID_p symbolize the identifiers of the consumer's and provider's payment system, respectively,
  $< ID_c, c > \in HCons$, which is the set of HConsumer IDs and
  $< ID_p, p > \in HProv$, which is the set of HProvider IDs.

- The mapping functions between the account identifiers are:
  f: *HCons* $\rightarrow$ *C* such that f (< $ID_{C\text{-}UPS}$, *c* >) = < *c* >, where *C* denotes the set of HPayer IDs for the C-UPS.
  g: *HProv* $\rightarrow$ *P* such that g (< $ID_{P\text{-}UPS}$, *p* >) = < *p* >, where *P* denotes the set of HPayee IDs for the P-UPS.

In conclusion, the account identifiers are determined as follows:

1. for the *UPayRequest* SP of the C-UPS:
   * HPayer ID can be easily determined from the HConsumer ID;
   * HPayee ID (of the HPG) depends on the C-UPS ID and is stored in the *UPS Table* of the HPG;
2. for the *UPayRequest* SP of the P-UPS:
   * HPayer ID (of the HPG) depends on the P-UPS ID and is stored in the *UPS Table* of the HPG;
   * HPayee ID can be easily determined from the HProvider ID.

The assignment of this protocol element is as follows. The HPayer PE determines the HPayer ID for the C-UPS to preserve the anonymity of the HConsumer, as formulated in Section 4.1.4. The HPG PE determines its own account identifiers. The HPG PE also determines the HPayee ID for the P-UPS, because the HPG PE needs the P-UPS ID (to find its source account identifier) and the HPayee ID for the P-UPS (to initiate the uniform payment). This assignment of the protocol element requires that the HPayer PE sends the C-UPS ID and the HProvider ID to the HPG PE. In return, the HPG needs to send to the HPayer PE the HPayee ID for the C-UPS, which is required for the uniform payment initiation.

One alternative for the information exchange between the HPayer and HPG PEs is the C-UPS. This alternative requires the extension of uniform payment service such that it supports the necessary information transfer in both directions. Because the UPSs wrap existing payment systems, which do not provide information transfer services, the UPSs needs to incorporate another system for the information transfer. Another alternative is the usage of the DTS. Both alternatives require the extension of the functionality of the HPayer and HPG

PEs. The realization of the second alternative seems to be easier and more likely, especially if the DTS already exists. Figure 6.4 illustrates the account identifier determination performed jointly by the HPayer and HPG PEs, which use the DTS.

The way the account identifiers of the HPG are determined requires that the *UPS Table* contains an HPayer and HPayee ID for the UPSs in which the HPG plays an HPayer and HPayee role, respectively.

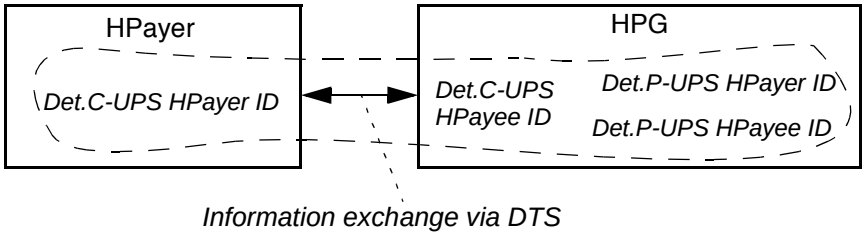


Figure 6.4 *Assignment of the Account identifier determination protocol element*

6.2.2 HConsumer authentication

This protocol element validates the authenticity of an HConsumer based on the HConsumer ID. A hybrid payment of the HConsumer will only be processed if the HConsumer ID is known by the HPS. If the verification fails, the hybrid payment initiation will be rejected.

We identify two alternatives to authenticate an HConsumer (Figure 6.5). One alternative is to authenticate it in the protocol layer. The other alternative is to hand over the authentication to the C-UPS.

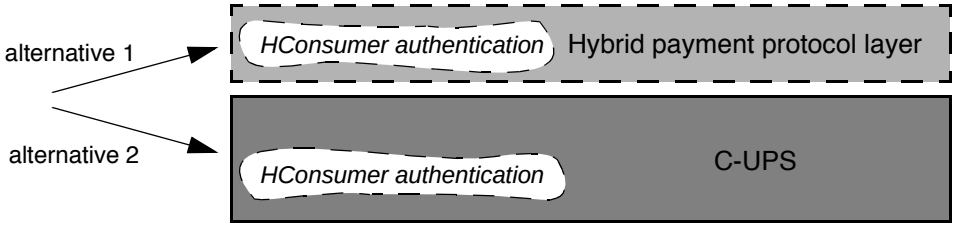


Figure 6.5 *HConsumer authentication alternatives*

For the first alternative, the protocol layer needs to maintain an administration of all HConsumers. This means that all HConsumer IDs must be stored in a (central) database. Because the HPS aims at global acceptance, this database needs to store tens of millions of IDs, search and provide information for hundreds of millions of hybrid payments every day (e.g., 500 million payments a day needs approximately 5800 evenly distributed searches per second). The stored information is valuable, because it allows access to money, so proper security mechanisms must be applied to protect it. An example is the access control of the PEs retrieving information from this database. Due to the fact that UPSs also keep an administration of HPayers for authentication purposes, the administration performed in the protocol layer would lead to (unnecessary) redundancy. We note that, one of the failure reasons of the ECash micropayment system in the 1990s was the bottleneck created by a central database that registered each spent e-coin and was used to prevent double-spending.

In the second alternative, the C-UPS authenticates the HConsumer based on the HPayer ID determined from the HConsumer ID. UPSs always authenticate the HPayers during payment initiations. If the authentication of an HPayer fails, then the authentication of the HConsumer (situated on top of the HPayer PE) fails too.

We decided to follow the second alternative, because this one does not require any additional functionality and information storage in the protocol layer (compared to the first alternative), does not run into scalability problems, security threats are avoided and it is common that UPSs authenticate the HPayers (existing systems also operate in this way). In conclusion, the authentication of HConsumers is delegated to the C-UPS and becomes part of the initiation of the uniform payment performed.

6.2.3 HProvider identification

This protocol element establishes whether the identity of an HProvider is known to the HPS or not. In other words, this element determines whether the destination account of the hybrid payment exists or not. If the identification fails, (i.e., the destination account does not exist), the hybrid payment initiation will be rejected.

Similarly to the HConsumer authentication, we identify two alternatives to identify an HProvider (Figure 6.6). One alternative is to identify it in the protocol layer. The other alternative is to request the P-UPS to identify it, because it keeps the identification information of HProviders.

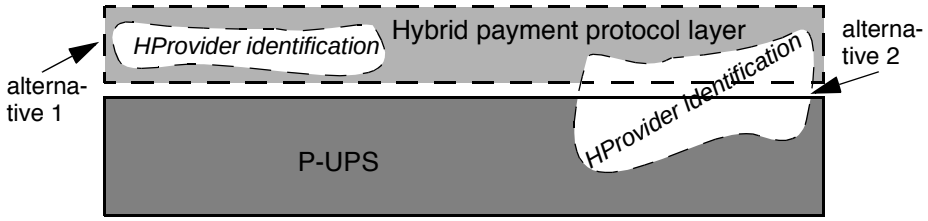


Figure 6.6 *HProvider identification alternatives*

The first alternative requires the administration of all HProviders in the protocol layer. This means that the HProvider IDs must be stored in a (central) database, which is then accessed for every hybrid payment. This alternative, similarly to the first alternative presented for the HConsumers authentication (Section 6.2.2), requires a double administration of HProviders.

The second alternative is to request the P-UPS to identify the HProvider. This UPS administers the HPayee IDs and always identifies the HPayees during payment initiations.

We decided to follow the second alternative, because this one does not require any additional functionality and information storage in the protocol layer (compared to the first alternative) and does not run into scalability problems. As a consequence, the identification of HProviders is made by means of interactions between the protocol layer and the P-UPS.

In the second alternative, the HPayer or the HPG PE could interact with the P-UPS. Both entities received the HProvider ID, but the HPG is known to this UPS and interacts regularly with it to initiate uniform payments. For the latter reason, the best solution is to let the HPG interact with the P-UPS.

The HProvider identification could be part of the uniform payment initiation for the P-UPS. If the initiation is accepted, then the HProvider is identified, so its destination account exists. We distinguish two cases depending on the order of the payment initiations for the C-UPS and P-UPS:

1. If the uniform payment initiation for P-UPS occurs first, the HPG faces the risk of losing money because the other initiation may fail due to reasons such as authentication, low account balance or exceeded credit limit.
2. If the uniform payment initiation for C-UPS occurs first, then the HPayer faces the risk of losing money because the initiation of the other payment may fail, e.g., the HProvider could not be identified, so the destination account does not exist.

Mechanisms to rollback a completed payment could be built into the hybrid payment system, but would be rather expensive considering the size of micropayments.

To avoid the risk of losing money, the identification needs to take place before any uniform payment is initiated. For this, the uniform payment service could be extended with a new service primitive that occurs at the USAP between the HPG PE and the P-UPS. The HPG PE executes this SP to request this UPS to identify the HProvider, whose ID is provided to the P-UPS as the only parameter of this primitive. Although, such an interaction may have a request-response nature, we abstract from this detail and consider that the identification is successful if this service primitive has occurred. Existing payment systems can comply with such an extended uniform payment service without any problems, because they all identify the providers before processing the payments.

In conclusion, the HPG PE executes a service primitive called *HProvIdentifyRequest* to request the P-UPS to identify the HProvider before any uniform payments are initiated. The HPayer PE sends the HProvider ID to the HPG PE, so information exchange is needed between these two protocol entities.

6.2.4 Amount of money verification

This protocol element checks whether the two underlying UPSs support the amount of money specified in the hybrid payment initiation (*HPayRequest* SP), and changes the currency, if necessary. If an amount of money is not supported or the payment currency cannot be exchanged into a supported currency, then the hybrid payment initiation will be rejected.

The checking algorithm depicted in Figure 6.7 needs to be performed for both UPSs to verify first whether the currency is supported. If it is supported and the value of the payment is also supported, then the amount of money for the uniform payment is equal to the specified amount of money. In this case the UPS will accept a payment initiation with respect to the amount of money. If the currency is supported but the value not, then the hybrid payment initiation will be rejected. Otherwise, if the currency is not supported, then it is converted[1] to a supported currency. After that, the payment value is checked. If the value is supported, the amount of money for the uniform payment is equal to the converted amount of money. Otherwise, the hybrid payment initiation will be rejected.

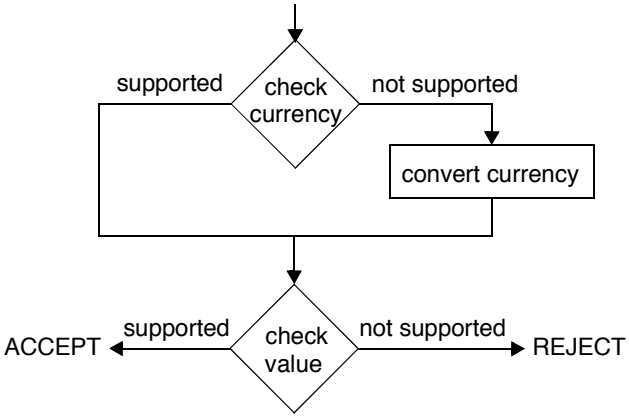


Figure 6.7 *Checking amount of money and converting currencies*

This protocol element could be performed by the HPayer and/or HPG PEs. The HPG PE, however, is the better choice, because it knows the range of payments of each interconnected UPS, this element needs special expertise and license for the currency exchange activities (see Section 4.5.6).

To perform the checking, the HPG PE needs (1) the currency exchange rates and (2) the identifiers of the existing payment systems. First, we assume that the HPG PE maintains a so-called *Currency Exchange Table* with the exchange rates. These rates should be updated regularly (e.g., every few hours) based on

1. The currency exchange handling of the hybrid payment system is very similar to the practice of banks. Banks allow their customers to initiate payments with other currencies than the one of their bank accounts, and exchange the currencies based on the exchange rates of these banks.

exchange information provided by banks, stock exchanges, etc. Second, we assume that, the *UPS Table* also contains fields for the minimum and maximum payment values, the default currency and set of currencies supported by a given payment system. These are the *Min, Max, Default currency* and *Set of currencies* fields. The HPG PE can retrieve the values stored in fields from the *UPS Table* based on the identifier of an existing payment system.

6.2.5 Payment information transfer

This protocol element is responsible for transferring the payment information (i.e., HProductTrans ID and Amount of money specified in the *HPayRequest* SP) from the HPayer PE to the HPayee PE, and generating the hybrid payment identifier.

The specified HProductTrans ID and Amount of money could be mapped onto the Context IDs and Amount of money parameters of the UPayRequest SPs and transferred from one PE to another via the underlying UPSs.

Payment information transfer to the HPG PE

In case of the C-UPS the problems with this approach would be that (1) it is not guaranteed that the HProductTrans ID is unique within the HPG PE, as the service definition required in Section 5.5.1, and (2) the Amount of money may not be supported, as described in Section 6.1.4. The consequences of these problems are that the HPG PE cannot uniquely recognize the payment confirmed by the C-UPS and relate it to the other uniform payment nor to the hybrid payment being processed. Hence, the HProductTrans ID and Amount of money should be transferred to the HPG PE via the DTS alternative system and the HPG PE needs a mechanism that allows recognizing the payment performed by the C-UPS and relate it to the other two payments. This mechanism is the Context ID and works as follows.

Suppose the HPG PE received the HProductTrans ID and Amount of money via the alternative DTS. The HPG PE generates next a unique Context ID and associates it to the HProductTrans ID. When this Context ID is provided to the HPG PE in a confirmation by the C-UPS, the HPG can recognize the processed payment and relate it to the others. For this, however, the HPG PE needs to

send this Context ID to the HPayer PE via the DTS. The HPayer PE will then use it to initiate the uniform payment. This means that the information transfer to the HPG PE and then to the HPayer PE occurs before the uniform payments are initiated.

Payment information transfer to the HPayee PE

In case of the P-UPS, the HPG PE can assign the HProductTrans ID and Amount of money to the Context ID and Amount of money parameters, respectively, so the P-UPS will transfer them to the HPayee PE. This ID will be unique within the HPayee PE because it was generated such by the HProvider, and the Amount of money is supported by the P-UPS assumed that the amount of money is calculated correctly by the HProvider's accounting system and the HConsumer provided correctly the amount to the hybrid payment system.

Payment information transfer sequence

The execution of the *Payment information transfer* protocol element begins with the HPayer PE receiving the HProductTrans ID and Amount of money parameters and ends with the HPayee PE confirming the hybrid payment to the HProvider. This protocol element uses the new underlying system, the C-UPS and P-UPS to transfer the payment information. The execution of this element depends and intertwines with the execution of the previously specified protocol elements, as these elements contribute with information necessary to initiate the uniform payments.

We identified two alternatives in the sequence in which the payment information transfer from one PE to another is achieved (after the information exchange between the HPayer and HPG PEs occurred and the previously specified four protocol elements are successfully performed):

1. The HPayer PE initiates first the uniform payment for C-UPS, and when this is acknowledged, the HPG PE initiates the uniform payment for the P-UPS.
2. The HPG PE initiates first the uniform payment for the P-UPS, and the HPG PE initiates the uniform payment for the C-UPS immediately after receiving the Context ID.

In the first alternative, the processing of the hybrid payment depends only on the HPayer PE and has a very high chance for success if the C-UPS accepts the initiation, because all necessary payment initiation information is available and we can fairly assume that the HPG's account at the P-UPS supports the new payment. The two successive uniform payments create the "*chain of payments*" introduced in Chapter 1. If the C-UPS rejects the initiation, then no money transfers occur at all and the hybrid payment initiation is also rejected.

In the second alternative, the P-UPS processes first a uniform payment. If the C-UPS rejects the initiation (e.g., HPayer is not authenticated, account balance too low), then the HPG PE lost the transferred money.

The decision is to let the payment information transfer occur according to the first alternative to avoid money-loss situations.

Figure 6.8 illustrates the transfer of the payment information from the HPayer PE to the HPayee PE via the HPG PE and the underlying systems. First, the HPayer PE sends the HProductTrans ID and Amount of money to the HPG PE via a new system. The HPG generates then the Context ID, associates it to the HProductTrans ID, and sends it to the HPayer PE. The HPayer PE transfers the Context ID and (the supported) Amount of money to the HPG PE via the C-UPS. Subsequently, the HPG PE transfers the HProductTrans ID as a Context ID and the Amount of money to the HPayee PE via the P-UPS. In this way, all three PEs are involved in transferring the payment information.

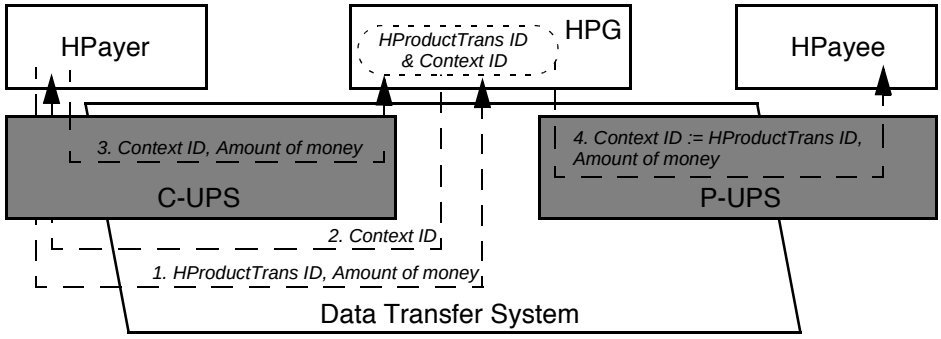


Figure 6.8 *Transferring the payment information*

HPayment ID generation

The generation of the HPayment ID is the last function before the hybrid payment is acknowledged to the HProvider. Both the HPG and HPayee PEs could generate this identifier. The HPayee PE, however, is a better choice because (i) it represents the HProvider in the hybrid payment system and knows the HSAP where the hybrid payment needs to be confirmed, and (ii) the HPG PE would need to send this identifier to the HPayee PE via another underlying system than the P-UPS.

We identify two alternatives that guarantee the uniqueness of this identifier: to use an increasing global index or to use the information available in the HPayee PE. The first alternative could create a bottleneck in performing the hybrid payments and limit the efficiency of the HPS because each HPayee needs to access this index for every hybrid payment.

In the second alternative, the HPayee could combine the uniform payment identifier and the identifier of the existing payment system, since the uniform payment identifier is unique within a UPS and the identifier of this system is also globally unique. Hence, their combination results in a unique hybrid payment identifier. In this way, the relation between the hybrid and uniform payments is also solved. Because of the obvious advantages of the second option, we decided to follow it.

6.2.6 Payment information storage

This protocol element is responsible for storing hybrid and uniform payment information in the protocol layer such that hybrid and uniform payments can be traced back and sufficient information is maintained to support auditing.

Besides of the protocol layer, payment information is stored by the HProvider (see Section 5.2.2), C-UPS and P-UPS as required by law (see Section 4.5.1). Within the protocol layer, the HPG PE is the place where two uniform payments performed for a hybrid payment can be related to each other. Additionally, because of its role in the hybrid payment system, the HPG PE falls under the supervision of legal and regulatory authorities, so it needs to store payment information. This means that if the P-UPS rejects the payment initia-

tion, the HPG still needs to store information about the payment performed by the C-UPS. In conclusion, the HPG PE stores payment information about the uniform payments, while the HPayer and HPayee PEs do not need to store any information.

We note that, the uniform payment confirmed by the P-UPS is related to the hybrid payment in the way the HPayment ID is generated (see Section 6.2.5). This ID is part of the hybrid payment information that is stored by the HProvider.

Figure 6.9 illustrates the payment information available within the HPG PE:

- Context ID, amount of money, and UPayment ID from the acknowledgement of the C-UPS and the destination account HPayee ID;
- HPayer ID, HPayee ID, Context ID and amount of money from the uniform payment initiation for the P-UPS.

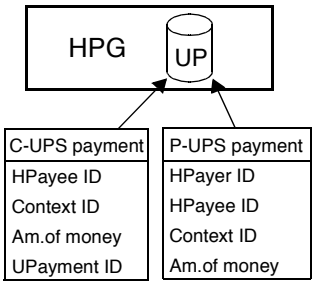


Figure 6.9 *Payment information storage*

To store the payment information, we assume that the HPG PE maintains a database table called *UniformPayments*. One record (or row) of this table contains the payment information of the two subsequent uniform payments performed for a hybrid payment as listed above.

6.2.7 Summary and examples

This section summarizes first the assignment of protocol elements to PEs and the relationship of these elements. Afterwards, it describes the proposed extension of the uniform payment system, the assumptions, and gives examples for the introduced database tables.

Figure 6.10 illustrates the assignment of the protocol elements and the order in which they occur.

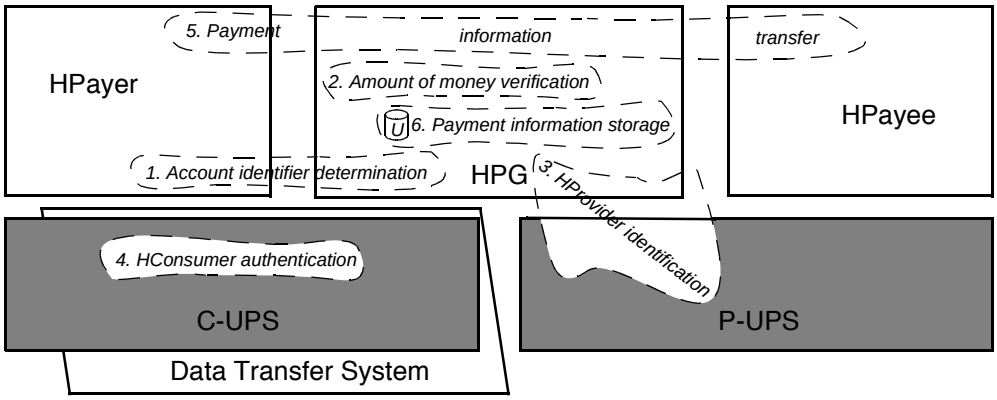


Figure 6.10 *Assignment of protocol elements to PEs and UPSs*

The processing of a hybrid payment starts after the hybrid payment initiation information is received from an HConsumer and ends when it is acknowledged to the appropriate HProvider. Between these two actions the protocol elements occur in the following order:

1. The account identifiers are determined by the HPayer and HPG PEs independently of each other.
2. The amounts of money of the two uniform payments are verified by the HPG PE, which also changes the currency, if necessary.
3. The HProvider is identified by the HPG PE and the P-UPS.
4. The HConsumer authentication is delegated to the C-UPS and intertwines with the payment information transfer protocol element.

* *Remark*: The first three protocol elements require information exchange between the HPayer and HPG PEs via the DTS. This exchange is part of the payment information transfer protocol element, which also depends on the execution of the first four elements.

5. The payment information is transferred from the HPayer to the HPayee via the HPG PE. From this point on, the transfer is realized using the UPSs by means of executing the UPayRequest and UPayConfirm service primitives. The transfer ends after the HPayee acknowledged the hybrid payment to the HProvider.

6. The payment information storage is performed by the HPG PE. The execution of this element also intertwines with the payment information transfer as the transfer is in progress through P-UPS.

The extension of the uniform payment service introduced a new service primitive. This service primitive (called *HProvIdentifyRequest*) requests the underlying UPS to identify an HProvider. The HProvider ID is the only parameter of the service primitive. If this primitive occurs the HPayee is identified successfully.

One assumption throughout the design was that the HPG PE maintains a *UPS Table*, which contains a record for each underlying UPS. A record consists of: the existing payment system identifier (in this case the name), HPayer ID, HPayee ID, minimum and maximum payment values, default currency, and set of currencies. All fields must be filled in in order to enable a proper interconnection. This table stores mainly static information, because the UPSs and their characteristics do not change often (e.g., weekly or monthly). Table 6.1 presents two example records of this table.

Table 6.1 *UPS Table example*

Ex. pay. sys. id.	HPayer ID	HPayee ID	Min	Max	Default currency	Set of currencies
Minitix	<gatex, tom&jerry>	164	0,10	10	€	{€}
PayStone	<gatex@xmail.com, l0ck3d,oddries>	951753	0,25	500	CA$	{CA$, US$}

Another assumption throughout the design was that the HPG PE maintains a *Currency Exchange Table*, which contains the exchange rates between different currencies. A record of this table consists of: a source currency followed by the exchange rates to other currencies. This table is updated periodically (e.g., daily) because of the frequent fluctuation of the exchange rates. A sample of the *Currency Exchange Table* is presented in Table 6.2.

Table 6.2 *Currency Exchange Table example*

Source currency	Euro (€)	GBP (£)	USD ($)
Euro (€)	1	1,45205	0,7814
GBP (£)	0,6886	1	0,5381
USD ($)	1,2797	1.8582	1

The last assumption throughout the design was that the HPG PE maintains a *UniformPayments* table, which contains a record for each hybrid payment. A record consists of: the HPayee ID, Context ID, Amount of money and UPayment ID of the payment performed by the C-UPS, and the HPayer ID, HPayee ID, Context ID and Amount of money of the payment performed by the P-UPS. Should the P-UPS rejects the payment initiation, the last four fields of a record are empty. Table 6.3 presents an example of this table with two records (the indices in the name of the fields show whether the information is related to the firstly or secondly performed uniform payment).

Table 6.3 *UniformPayments example*

HPayee ID1	Context ID1	Amount 1	UPayment ID1	HPayer ID2	HPayee ID2	Context ID2	Amount 2
164	10010096	€0,50	2811# 1975	<pgx, pwdx>	2648	20251980	€0,50
6824	23121943	US$1,21	701--411	46972967-UZL	9	10111213	€0,95

6.3 Hybrid payment protocol messages

The assignment of the protocol elements requires information exchange between the protocol entities. Protocol data units (PDUs) are units of information exchanged and make possible the cooperation of these entities necessary for the processing of hybrid payments [1]. This section defines the abstract[1]

1. Concrete PDUs (i.e., the representation of abstract PDUs in terms of in bits and bytes) are omitted in this design phase, so the coding of abstract PDUs is not considered.

PDUs and their exchange and presents the normal (error-free) protocol behaviour in terms of service primitives and PDUs.

The information exchanged via the different underlying systems (DTS and UPSs) is represented by different PDUs. The DTS transfers Data-PDUs between the HPayer and HPG PEs. The UPSs transfer Pay-PDUs between the HPayer and HPG PEs, and between the HPG and HPayee PEs, respectively.

6.3.1 Data-PDUs

The Data-PDUs contain the information exchanged between the HPayer and HPG PEs. We identify two types of Data-PDUs. One type of PDUs is transferred from the HPayer PE to the HPG PE and is called Data2HPG-PDU. The other type of PDU is transferred from the HPG PE to the HPayer PE and is called Data2-HPayer PDU. To be able to send each other Data-PDUs, the HPayer and HPG PEs need to contain protocol elements that create, send and receive the Data-PDUs.

The structure of the Data2HPG-PDU is the following (Figure 6.11):

- C-UPS ID (see *Account identifier determination* protocol element);
- HProvider ID (see *Account identifier determination* protocol element);
- Amount of money (see *Amount of money verification* and *Payment information transfer* protocol elements);
- HProductTrans ID (see *Payment information transfer* protocol element).

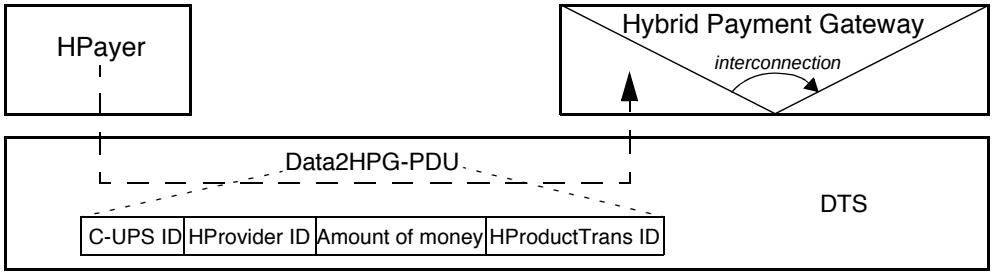


Figure 6.11 *Data2HPG-PDU exchange*

The structure of the Data2HPG-PDU is the following (Figure 6.12):

- HPayee ID (see *Account identifier determination* protocol element);
- Context ID (see *Account identifier determination* protocol element);
- Amount of money (see *Amount of money verification* protocol element).

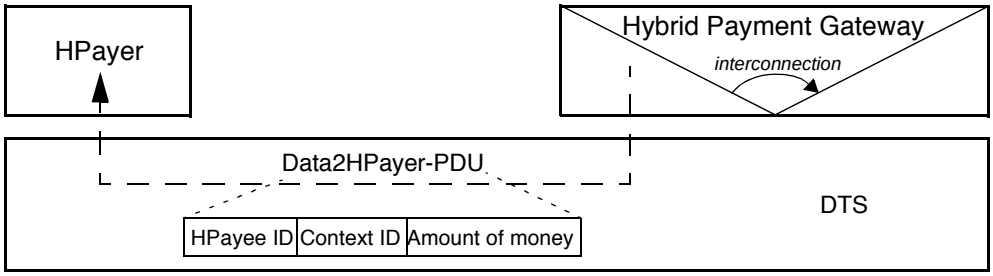


Figure 6.12 *Data2HPayer-PDU exchange*

In case the HPG decided that a hybrid payment cannot be performed (e.g., the amount of money is not supported by the C-UPS or the HProvider is not identified) the Data2HPayer-PDU contains an error message in place of the Context ID, and the Amount of money is set to zero or null value, so the HPayer PE will reject the hybrid payment initiation.

6.3.2 Data Transfer System

The DTS provides the transfer of the Data2HPG-PDU and Data2HPayer-PDU. We introduce two service primitives that represent the interactions between the HPayer, HPG and DTS: one for sending a PDU (*DataReq* SP) and another for receiving one (*DataInd* SP).

The *DataReq* SP allows a user of the DTS to send another user a Data-PDU. The HPayer executes this SP to send a Data2HPG-PDU to the HPG, which in turn executes this SP to send a Data2HPayer-PDU to the HPayer. This SP is executed at a SAP between a user and the DTS. The parameters of this SP are the address of the receiving user and the PDU itself.

The *DataInd* SP indicates a DTS user that the DTS transferred a Data-PDU. If the DTS executes this SP at a SAP of the HPG, then it transferred a Data2HPG-

PDU from the HPayer. If the DTS executes this SP at a SAP of the HPayer, then it transferred a Data2HPayer-PDU from the HPG. The parameters of this SP are the address of the sending user and the PDU itself.

A *DataInd* SP always follows a *DataReq* SP, so the DTS must be a *reliable* system, which cannot lose PDUs due to errors such as network failures. The execution of the *DataReq* SP is successful if the subsequent *DataInd* SP occurred. Each hybrid payment requires a Data2HPG- and a Data2HPayer-PDU exchange.

Other errors such as data corruption, data duplication, random data generation and misdelivery are not allowed to occur and the DTS needs to provide a *reliable data transfer* service. Otherwise the functionality of the hybrid payment system is compromised, which in turn will compromise the users' acceptance of and their trust in the hybrid system.

Figure 6.13 illustrates the HPayer and HPG using the DTS. We assume that the HPayer knows the DTS address of the HPG (as default and constant information stored by the HPayer), so it executes the *DataReq* SP to send the Data2HPG-PDU to the HPG. The DTS executes next the *DataInd* SP to deliver the PDU. After that, the HPG determines the necessary information and executes the *DataReq* SP to send the HPayer the Data2HPayer-PDU. The HPayer's DTS address is provided to the HPG in the *DataInd* SP. Finally, the DTS delivers this PDU.

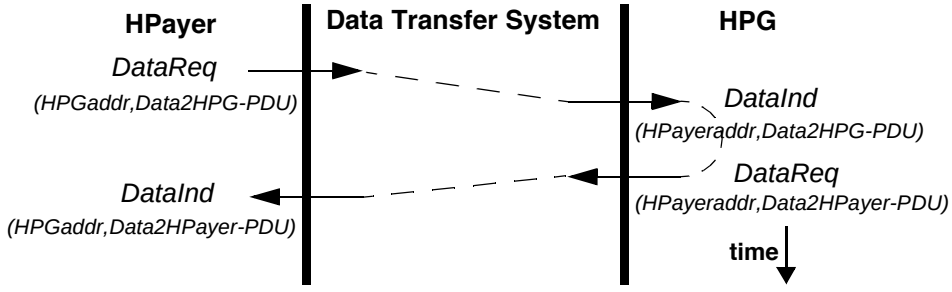


Figure 6.13 *Data Transfer System behaviour instance*

6.3.3 Pay-PDUs

We define a single abstract payment PDU (denoted as Pay-PDU) that is transferred via the underlying UPSs from an HPayer to the HPG, and from the HPG to an HPayee (Figure 6.14). The purpose of this PDU is to transfer payment information. The Pay-PDUs contain the Context ID and the Amount of money (see *Payment information transfer* protocol element).

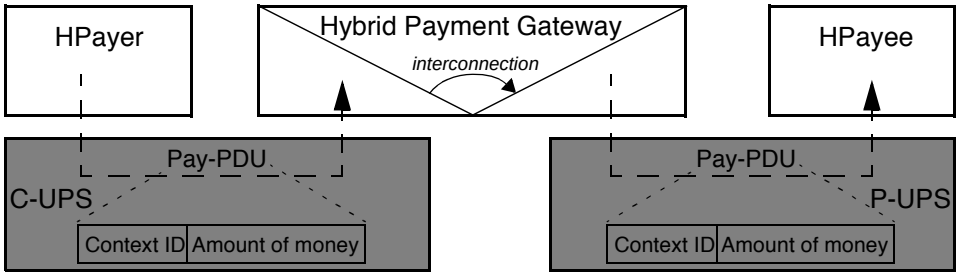


Figure 6.14 *Transferring Pay-PDUs*

6.3.4 Normal protocol behaviour

Figure 6.15 illustrates the normal behaviour of the hybrid payment protocol in a time sequence diagram. An HPayer, an HPG and an HPayee PEs are using two UPSs and the DTS. The functions performed internally by these protocol entities are not shown.

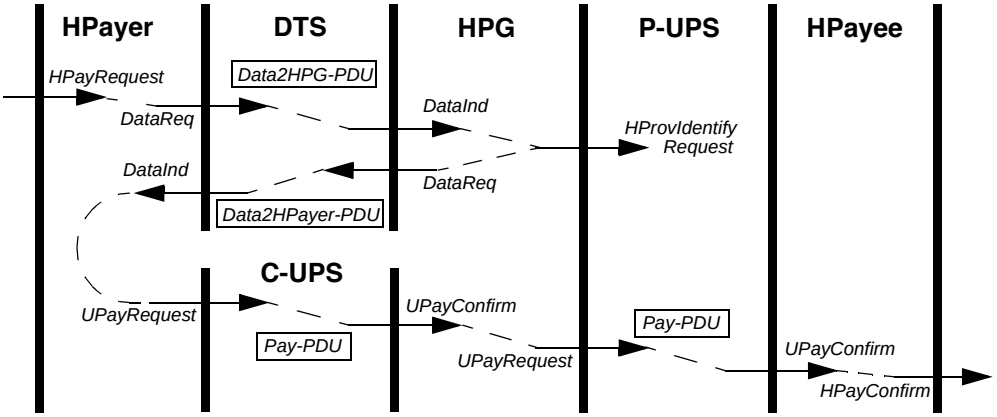


Figure 6.15 *Normal hybrid payment protocol behaviour*

The normal behaviour is as follows:

1. The HPayer receives the hybrid payment information (*HPayRequest* SP).
2. The HPayer creates and sends a Data2HPG-PDU via the DTS to the HPG (*DataReq* SP).
3. The HPG receives the Data2HPG-PDU (*DataInd* SP).
4. The HPG identifies successfully the HProvider (*HProvIdentifyRequest* SP), verifies whether the amounts of money are supported by C-UPS and P-UPS, looks up its HPayee ID for P-UPS, and generates the Context ID.
5. The HPG creates and sends a Data2HPayer-PDU via the DTS to the HPayer (*DataReq* SP).
6. The HPayer receives the payment information necessary to initiate the first uniform payment (*DataInd* SP).
7. The HPayer initiates the *first* uniform payment and sends a Pay-PDU (with the Context ID and Amount of money) to the HPG via the C-UPS (*UPayRequest* SP).
8. The HPG receives the Pay-PDU (*UPayConfirm* SP), recognizes the Context ID and relates them to the payment information received earlier.
9. The HPG initiates the *second* uniform payment and sends a Pay-PDU (with the HProductTrans ID and Amount of money) to the HPayee via the P-UPS (*UPayRequest* SP). The HPG also stores the necessary payment information in the *UniformPayments Table*.
10. The HPayee receives the Pay-PDU (*UPayConfirm* SP).
11. The HPayee generates the HPayment ID and confirms the hybrid payment to the HProvider (*HPayConfirm* SP).

6.3.5 PDU summary

Table 6.4 summarizes the defined PDUs, their parameters and the underlying systems that will transfer them.

Table 6.4 *Protocol data units*

PDU name	Parameters	Transferred by
Data2HPG-PDU	C-UPS ID HProvider ID Amount of money HProductTrans ID	Data Transfer System
Data2HPayer-PDU	HPayee ID Context ID Amount of money	Data Transfer System
Pay-PDU	Context ID Amount of money	Uniform Payment System

6.4 Robustness, security and optimization

So far the expected (normal) behaviour of the protocol was presented. This section discusses three aspects of the hybrid payment protocol: robustness, security and optimization.

6.4.1 Robustness

Robustness is the resilience of the hybrid payment protocol to error situations that can occur as a consequence of unexpected behaviour or activity of the hybrid payment system components. This section discusses several error situations, shows what are the consequences of these situations and indicates how the protocol behaves in such situations and tolerates them.

This section discusses a restricted class of error situations that may occur in the protocol elements and DTS as they are the new components of the HPS. We assume that certain error situations (e.g., malfunction of a database during payment information storage) do not occur and that in the environment of the hybrid payment system (HConsumers and HProviders) no errors will occur. Additionally, we can assume that error situations will hardly occur within UPSs, if they accept a payment initiation, because the reliability and robustness of the UPSs and existing payment systems are comparable (see Section 5.5.2), and the latter systems claim high reliability and robustness. This also means that if the payment initialization information for the P-UPS is determined and

the C-UPS processed the payment, the hybrid payment can almost surely be processed and the chance that the HPayer will lose money is very low.

Two error situations are identified and discussed: (1) one of the protocol entities behaves unexpectedly while the underlying systems function normally; (2) the DTS is offline or unreachable.

Error situations of protocol entities

Suppose the underlying systems function normally (as expected), the following error situations may occur in the three protocol entities:

- The HPayer PE sends the Data2HPG-PDU but does not receive the expected Data2HPayer-PDU from the HPG (e.g., the HPG crashes before sending the PDU): without the information from the HPG the HPayer cannot initiate the uniform payment. The HPayer PE waits for a certain time (e.g., the time-out period could be a few seconds[1]) for the HPG's answer and then it will reject the hybrid payment initiation, so no hybrid payment will be processed. If the HPayer PE receives the Data2HPayer-PDU after the time-out period, it will not consider it. Considering the role of the HPG in the hybrid payment system, we consider that this error situation will hardly occur.
- The HPayer PE receives the Data2HPayer-PDU but does not send the Pay-PDU (e.g., it crashes or the C-UPS rejects the payment initiation): the HPG PE waits for the expected Pay-PDU until a certain time-out period elapses. After this period, the HPG will store the data received in the Data2HPG-PDU in the *UniformPayments* table as follows: all fields will be filled with the appropriate information in except the *UPayment ID1*, which indicates that the C-UPS confirmed no payment. In this way, if the Pay-PDU is delivered after this period, the HPG can recognize it based on the Context ID and continue the processing of the hybrid payment. If the HPayer PE does not send the PDU, no hybrid payment will be processed and it will reject the hybrid payment initiation.

1. The length of the time-out period needs to be set carefully such that if it is too long then the performance of the hybrid payment system will be reduced and if it is too short then the PDUs may frequently arrive late.

- The HPG PE receives the Pay-PDU but fails sending the subsequent Pay-PDU (e.g., it crashes or the P-UPS rejects the payment initiation): the hybrid payment failed since is not processed completely. The consequence is that the HPayer lost the money transferred by the C-UPS. The HPayer, however, may get its money back because some existing payment systems allow the rollback of payments. Even though the HPG is responsible for the money loss, considering its role in the hybrid payment system, we can assume that such a situation will hardly occur.
- The HPayee PE receives the Pay-PDU but does not acknowledge the hybrid payment (e.g., it crashes): one of the following situations may occur (i) the HPayer PE loses money; (ii) the HPayer PE loses money, the HConsumer complains to the HProvider about the loss, and the HProvider investigates the payment at the HPayee PE and eventually deliver the paid product(s), (iii) the HPayee PE may return the money to the HPG EP, which in turn may return the money to the HPayer PE. Considering the role of the HPayee PE in the hybrid payment system and its relation with the HProvider, we can assume that this error will hardly ever occur.

Among the above discussed error situations the hybrid payment system does not get blocked, only in the last two situations money may get lost and the amount of money lost is relatively low. None of these situations are likely to occur, however.

Error situation of the Data Transfer System

As explained in Section 6.3.2, the DTS provides a *reliable data transfer* service. Suppose the protocol entities are functioning normally, the DTS may still be offline or unavailable. This situation will cause a communication error between the HPayer and HPG PEs and payment information cannot be transferred. If the HPayer PE cannot send the Data2HPG-PDU to the HPG, then it will reject the hybrid payment. If the HPG PE cannot send the Data2HPayer-PDU to the HPayer, then the HPayer will wait for some time and then reject the hybrid payment initiation.

Conclusions

This section discussed several error situations and showed that such situations occur rarely and the hybrid payment protocol tolerates them. If a money loss situation still occurs, the HPayer (and HConsumer) bears the loss as explained in Section 4.5.2. None of the discussed error situations is a serious threat to the normal operation of the hybrid payment system.

6.4.2 Security

Because the hybrid payment system is an online system, it is likely to be attacked by malicious users (attackers) in order to commit fraud, steal money or misuse the system. The hybrid payment system must therefore be secure enough to prevent and detect such attacks. Other attacks such as denial-of-service on the components of the hybrid payment system may still occur, but are independent of the system.

Payer and payee protocol entities of existing payment systems can be attacked in order to retrieve fraudulently information or money. Because the functionality of the HPayer and HPayee PEs is similar to the existing ones, their implementation does not create a higher security risk and the chance that they will be attacked will not be higher.

This section concentrates on fraud attempts of the HPG and on so-called "*man-in-the-middle*" attacks. These are attacks on the interactions between PEs and the underlying systems. The interactions where potential attacks may occur are the HPayer-HPG communication via the DTS and the payment initiation and confirmation interactions.

Fraud attempts of the HPG

The HPG may try intentionally to commit fraud and misuse its position within the hybrid payment system. We identified the following three fraud attempts: (1) the HPG increases its accounts balance without actually transferring money to these accounts, (2) the HPG requests more money from the UPSs as pay-out than what was actually transferred to its accounts, and (3) receive money from C-UPSs but not transferring any money via the P-UPSs.

In the first attempt the HPG needs to access the UPSs and increase its accounts balances without being detected by the UPSs. This attempt is rather difficult to realize since the account balances are stored in a closed system under heavy security mechanisms. Besides this, payer account balances are usually increased when money is transferred to the payment system and this transfer is realized via specialized banking systems. Hence, the chance that the HPG manages to commit fraud in this way is small.

In the second attempt the HPG presents fake payment confirmations and demand from the UPSs to pay-out more money than what its account balances indicate. UPSs are required by law to generate and store audit information (see Section 4.5.1), so in such a situation they can investigate the claims of the HPG and trace back all payments in question. Even external audit authorities or organizations (e.g., national banks) could be involved in solving such claims. Hence, an audit can easily demonstrate the cheating attempt of the HPG, which will be discouraged to make such demands.

The third attempt is probably the HPG's easiest way to commit fraud because the HPG does not have to deal directly with the UPSs. In this case, the HPG receives payment confirmations from C-UPSs but every now and then "fail" or "forget" to initiate the payments for the P-UPSs. In this way, the HPG makes significant profits while HPayers lose money. Cheated HPayers in turn could demand from their UPSs investigations of their payments. So, the operation of the HPG, which has payer and payee contracts with the UPSs, is verified based on these contracts. At the same time, the HPG is subject to audit by financial authorities, and such fraud attempts can easily be detected. These arguments explain why the chance that the HPG will misuse its position in the hybrid payment system is small.

Communication via the Data Transfer System

An attacker could observe the Data-PDU exchange between an HPayer and HPG (eavesdropping), and try to commit fraud by modifying the content of these PDUs. For instance, it could modify the HProvider ID and increase the Amount of money in the Data2HPG-PDU to re-direct a (bigger) money transfer to its own destination account. Or it could modify the HPayee ID and

increase the Amount of money in the Data2HPayer-PDU to let the HPayer pay a bigger amount of money into its own destination account.

To prevent or detect such attacks, the DTS needs to secure these PDUs and be able to detect any modification that occurred during their transmission. To secure the communication channel the Secure Socket Layer (SSL 3.0) or the Transport Layer Security (TLS 1.0, [2]) could be used. The SSL, for instance, provides the authentication of the end-points and communication privacy over the Internet using cryptographic algorithms like RSA, DES, MD5 and SHA, and prevents eavesdropping, altering or forgery of Data-PDUs. Because of the authentication, the HPayer will always know that the HPG is really the party it claims to be or not.

Communication via the Uniform Payment Systems

We consider that the security of UPSs is similar to the security of existing systems, which are closed systems and resistant to malicious attacks. This means that UPSs do not represent a higher security risk than existing payment systems.

One option for an attacker is to capture and modify the payment information of a Pay-PDU when it is sent to the UPS such that it steals the HPayer ID (which provides access to the account), changes the HPayee ID (to re-direct the money to its own account), or modifies the amount of money or the Context ID. In this way, the attacker can steal the HPayer's money, the HPayer or the HPG will pay for the product(s) of the attacker.

Another option for an attacker is to capture and modify the Pay-PDU when it is delivered such that the HPG or HPayee will receive a confirmation for the product(s) of the attacker.

The mentioned potential attacks exist in case of existing payment systems, which commonly use the HTTP communication protocol over the earlier mentioned SSL or TLS security protocols. This solution could also be used to secure the uniform payment initiation and acknowledgement interactions.

Conclusion

A first conclusion is that the chance that the HPG will try to commit fraud is small. A second conclusion is that the SSL or TLS chosen to be used by the DTS and UPSs to secure the communication with the protocol entities can prevent "man-in-the-middle" attacks. The consequence of utilizing the chosen security mechanisms is that the protocol elements responsible for the information exchange need to support them.

6.4.3 Optimization

So far the hybrid payment protocol was designed for the general case in which two UPSs are interconnected by the HPG to process a hybrid payment. The HPG PE may not be needed for each hybrid payment, however. This is the case when both the HConsumer and HProvider have their accounts within the same existing payment system. Involving the HPG in the processing of a payment in such a case would unnecessarily reduce the performance of the hybrid payment protocol. This section discusses the protocol operation in case no interconnection is needed.

To be able to determine whether interconnection is needed or not, the HPayer PE needs a protocol element that compares the identifiers of the two existing payment systems. After the hybrid payment information is received by the HPayer PE, this element obtains these system identifiers from the HConsumer and HProvider IDs (see Section 5.2.2). If they are different, then interconnection is needed and the protocol elements are used as described in this chapter so far. Otherwise, interconnection is needed and the operation of the protocol is as follows (Figure 6.16):

- the HPayer PE determines the HPayer and HPayee IDs from the HConsumer and HProvider IDs, respectively;
- the UPS performs the HProvider identification and HConsumer authentication when the uniform payment is initiated (the Pay-PDU will be created with the HProvider ID and Amount of money parameters of the *HPayRequest* SP);
- the payment information transfer protocol element begins with the initiation of the uniform payment, ends with the confirmation of this pay-

ment, and intertwines with the identification and authentication protocol elements.

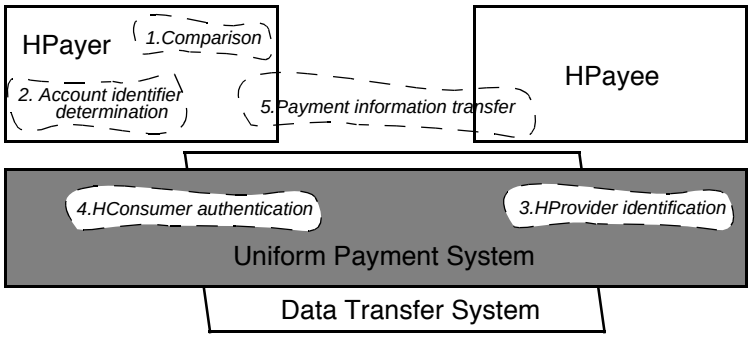


Figure 6.16 *Re-assignment of protocol elements*

We note that, (i) additional interactions with the UPS to identify the HProvider are not necessary; (ii) the HProductTrans ID will be unique within the HPayee PE and it can be mapped onto the Context ID; (iii) the verification of the amount of money is not necessary, and if the UPS does not support the amount of money, then it rejects the uniform payment initiation; (iv) if the uniform payment initiation fails, then the hybrid payment initiation also fails and no money loss situations can occur, and (v) payment information is stored by the UPS and HProvider.

6.5 Multiple Hybrid Payment Gateways

This section discusses the question whether a single HPG will be sufficient to interconnect all UPSs or multiple HPGs are needed. In this discussion the cases of a single and multiple HPGs are compared based on various arguments such as business, reliability and investments. Several arguments will sustain the introduction of multiple HPGs. In this case, however, cooperation between the HPGs is needed. We also identify and discuss two alternatives for inter-HPG cooperation.

6.5.1 Discussion on single vs. multiple HPGs

Using a single HPG for interconnection is a simple and therefore probably the ideal solution. The real world, however, is far from ideal and the idea of a

single HPG may turn out to be difficult to accomplish. Throughout the discussion we assume that a customer is advised by the operator (PSO) of his/her chosen micropayment system to use a certain default HPG. The HPayer PE as part of the customer always uses this HPG, if interconnection is needed.

Business argument

A single HPG and its operator create a monopolistic situation, which violates the free market rules. To avoid this, multiple, independent gateways may be operated, which compete with each other with respect to transaction fees, the number of interconnected systems, currencies exchange capabilities, reliability, etc. Then, the PSOs can advise the customers and providers to switch over to another HPG if this provides a better or cheaper interconnection.

Reliability argument

A single HPG is a single point of failure in the hybrid payment system, and vulnerable to fraud, errors, DOS attacks, etc. Suppose the only HPG stops providing the interconnection, then the whole hybrid payment system stops functioning. If multiple HPGs provide the interconnection then the hybrid payment system can function with a higher degree of availability. In this case, the PSOs can advise the customers to switch to another HPG if the default one stops functioning or the default HPG can transparently re-direct the customers to other HPGs and delegate the interconnection to other HPGs. This proposal, however, requires the existence of a register with all HPGs and cooperation between the HPGs.

Investments argument

The HPG initiates payments processed by pre-paid and post-paid UPSs. For this, the HPG needs to make sure that the balance of pre-paid accounts or the credit limit of post-paid accounts will support new payments at any time. Otherwise, the interconnection of UPSs is compromised. A single HPG needs considerable investments because most existing payment systems are pre-paid. We note that, in practice, the pay-outs from UPS (i.e., the money transfers from the UPSs to the HPG and HPayees) occur much later, generally between 2-4 weeks or 1 month after the uniform payment confirmations (Figure 6.3). An

advantage of multiple HPGs is that they can share the initial investments because each of them interconnects and invests money in only a subset of all UPSs, so the investments can be significantly reduced.

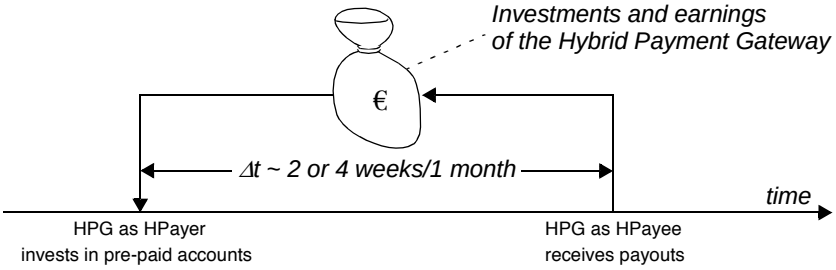


Figure 6.17 *Time difference between the investments and earnings of HPG*

> *Remark:* In the case, the HPG interconnects a post-paid C-UPS with a pre-paid P-UPS, then the HPG plays the role of a credit institution, since it provides credit to the HPayer. In other words, the HPG invests money into the pre-paid P-UPS to pay the HPayee on behalf of the HPayer, which transfers money into the post-paid UPS after the investment of the HPG. A consequence of this is that the HPG may need a credit institution or banking license for its activities. This issue is outside of the scope of this thesis, however.

Other arguments

A single HPG may not be able to interconnect all UPSs because of different monetary legislation applied in different countries, political differences or conflicts between countries, and may not be able to exchange any pair of existing currencies. To avoid these problems, multiple HPGs may be operated which cooperate with each other. For instance, suppose that an HPayer wants to pay an HPayee and they use UPSs that are not interconnected by the default HPG. This HPG can transparently re-direct the HPayer to another HPG that interconnects those UPSs. Or if there is no other HPG that interconnects those UPSs, two HPGs, one connected to the C-UPS and the other to the P-UPS, can work together to process the hybrid payment. Multiple HPGs, however, require cooperation alternatives and the existence of a register, which records all available HPGs and the roles they can play for the interconnected UPSs (payer or payee).

Summary and conclusion

Table 6.5 summarizes the arguments in favour and against a single HPG. Multiple HPGs, however, require an HPG register and cooperation between each other.

Table 6.5 *Single vs. multiple HPGs*

Single HPG	Multiple HPGs
simplistic solution	realistic situation
monopolistic situation	competition between them
single point of failure, vulnerability to fraud and errors	higher degree of availability
may not be able to interconnect all systems	are likely to able to interconnect all systems
significant initial monetary investments	reduced initial monetary investments

The hybrid payment system in which multiple HPGs provide the interconnection is more realistic, robust and easier to be realized than in the case of a single HPG. Multiple HPGs also increase the hybrid payment system's chance to succeed.

6.5.2 Inter-Hybrid Payment Gateway cooperation

This section presents the design of cooperation alternatives between HPGs. Throughout the design, we assume that the HPayer uses the default HPG as advised by the operator of the C-UPS. The default HPG is connected to the C-UPS, but cannot interconnect the C-UPS with the P-UPS. We identify two cooperation alternatives: (1) there is another HPG that can interconnect the C-UPS with the P-UPS and the default HPG delegates the HPayer to the other HPG, (2) there is no HPG that can interconnect the C-UPS with the P-UPS, but there is another HPG that is connected to the P-UPS.

We note that, the involvement of two HPGs will increase the processing time of the hybrid payment. This will affect the time-out period settings mentioned in Section 6.4.

Indirect HPG-cooperation (re-direction)

The first alternative is called *indirect HPG-cooperation* because the default HPG needs to find another HPG that can perform the interconnection and transparently re-directs (or delegates) the HPayer to this HPG. In this way there is no direct interaction between the two HPGs and the default one delegates the interconnection to the other HPG.

This alternative requires a register with all available HPGs, so the default HPG can find a suitable HPG. For instance, a so-called *Payment Gateway Register* (PGR) could contain information about each HPG together with the payment systems they interconnect, the role(s) they can perform in the interconnection, and their Data Transfer System[1] addresses.

Table 6.6 illustrates the registration of two HPGs in the PGR in case the Internet is used as a DTS. HPGs should register themselves after they become operational, update their registration information, if necessary, and try to find other HPGs, when cannot perform a particular interconnection.

Table 6.6 *Payment Gateway Register example*

HPG Name	Existing pay.sys.	DTS SAP (www address)	HPayer	HPayee
Dutch Payment Gateway	Minitix	www.intergate.nl:9090	X	X
Dutch Payment Gateway	Way2Pay	www.intergate.nl:9090	X	-
Dutch Payment Gateway	Wallie	www.intergate.nl:9090	X	X
Global Gateway Services	PayNova	www.globalgateway-services.com:1024	X	X
Global Gateway Services	PayStone	www.globalgateway-services.com:1024	X	X
Global Gateway Services	Way2Pay	www.globalgateway-services.com:1024	X	X

1. The DTS functions as described in Section 6.3.1 and supplemented with the robustness and security discussions in Sections 6.4.1 and 6.4.2.

The indirect cooperation requires that HPGs and HPayers incorporate protocol elements that handle the following functions:

- HPG: register and maintain registration information, search for an alternative HPG, send re-direction information to an HPayer;
- HPayer: accept the re-direction from the default HPG and process it.

Figure 6.18 illustrates an indirect HPG-cooperation scenario in which the default HPG (def-HPG) of the HPayer cannot perform the interconnection of the two specified UPSs, and redirects the HPayer to another HPG (other-HPG). The HPayer then restarts the processing of the hybrid payment with the new HPG.

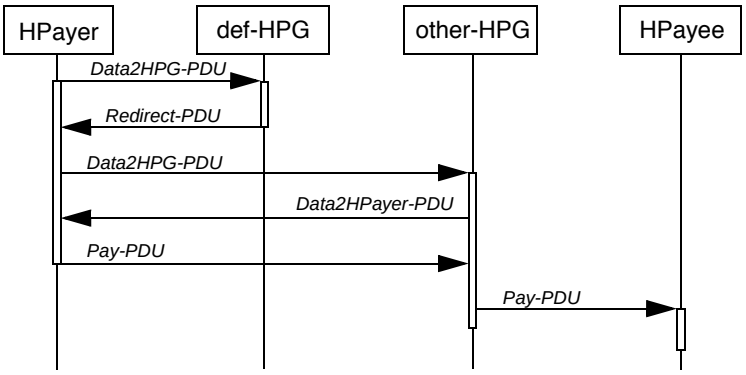


Figure 6.18 *Indirect HPG-cooperation scenario*

Direct HPG-cooperation

The second alternative is called *direct HPG-cooperation* since the default HPG needs to find another HPG with which it will directly interact to perform the hybrid payment. The default HPG remains to be involved in the uniform payment for the C-UPS, while the other HPG in the uniform payment for the P-UPS.

We identify two underlying systems that can be used between the HPGs to realize the direct cooperation: a UPS or the DTS.

A UPS between the two HPGs is one solution for the direct HPG-cooperation. This means that a third UPS will be involved in the processing of the hybrid

payment besides the C-UPS and P-UPS. Using the third UPS, the default HPG transfers money to the other HPG. In this way, a chain of payments is created all the way from an HPayer to an HPayee. This solution is hardly profitable for the HPGs because they pay for using the UPSs if they play the role of the HPayee and may need to pay as HPayers. Another disadvantage of this solution is that the time needed for completing hybrid payments would increase. If three (or more) uniform payments need to be performed, then the completion time of a hybrid payment would take more than three times longer than the completion time of a uniform payment. As a consequence, HConsumers would be reluctant to use a slow HPS to pay micropayments to HProviders, and HPG operators would consider this system as too expensive.

The DTS between the two HPGs is another solution for the direct HPG-cooperation. This means that HPGs interact with each other by sending and receiving the payment information necessary to initiate payments for the UPSs. Before payment information is sent to another HPG, both gateways need to authenticate each other to make sure that they are the parties they claim to be. When a HPG sends payment information to another HPG, it requests the other gateway to initiate a uniform payment and promises that the specified amount of money will be transferred to the other gateway later in a settlement. If the other HPG trusts the default one, it accepts the participation in the processing of the hybrid payment. The inter-gateway settlement would occur at the end of a certain time-period, when the total amount of money specified in previously sent payment information is transferred to the other HPG (e.g., in monthly wholesale payments). The money transfer may also occur when a HPG reaches a certain debt towards another HPG, and needs to level the payment balance. The money transfer could be completed using banking networks. Depending on the location of the HPGs, the bank transfers may be cross-border.

To realize the second alternative, HPGs need to use the *Payment Gateway Register* introduced earlier. HPGs that want to send payment information to other HPGs would check the PGR to find a HPG that uses the UPS of the specified HPayee. Another requirement for the realization of the second approach is the need for authentication and trust-enabling mechanisms that enforce the trust relationship between the HPGs.

Figure 6.19 illustrates a direct HPG-cooperation scenario. The default HPG (def-HPG), finds an auxiliary HPG (aux-HPG), which can play the HPayer role for the P-UPS. The def-HPG sends a modified Data2HPG-PDU denoted as Data2HPG*-PDU to the aux-HPG for requesting it to participate in the processing of the hybrid payment. From this PDU the identifier of the HPayer's existing payment system is omitted because it is not relevant for the aux-HPG. The aux-HPG performs next the verification of the amount of money and the identification of the HProvider with the P-UPS. It sends then an Accept-PDU to the def-HPG to indicate that the amount of money is supported, the HProvider is identified and it accepted the participation request. The def-HPG sends next the Data2HPayer-PDU to the HPayer, which replies with the Pay-PDU. Subsequently, the def-HPG sends a Confirm-PDU[1] to the aux-HPG to indicate that the first uniform payment has been completed. Finally, the add-HPG sends the Pay-PDU to the HPayee, which will complete the processing of the hybrid payment by confirming it to the HProvider (not shown in the figure).

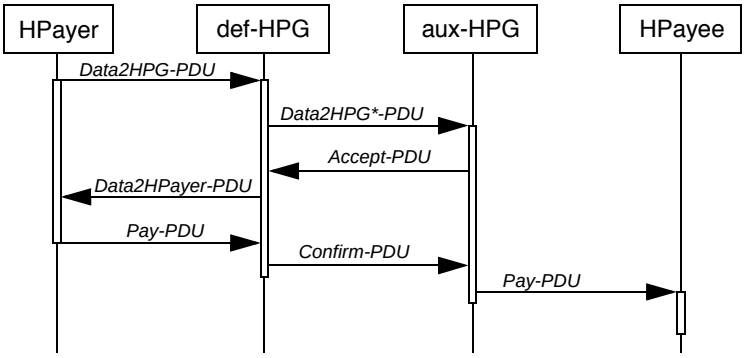


Figure 6.19 *Direct HPG-cooperation scenario*

The directly cooperating HPGs cannot store information about *two* uniform payments in the *UniformPayments* table. Nevertheless, to provide support for audit and tracking back of hybrid payments:

- the default HPG should store in place of the P-UPS information: (i) the name of the other HPG in the HPayer ID2 field, (ii) the HProvider ID

1. The structure of the Redirect-, Accept- and Confirm abstract PDUs is not defined because at this design phase only their semantics is relevant.

in the HPayee ID2 field, (iii) the HProductTrans ID and Amount of money in the Context ID2 and Amount2 fields;

- the auxiliary HPG should store in place of the C-UPS payment information: the name of the default gateway in the HPayee ID1 field and leave the other three fields empty (see Table 6.3).

We expect that maximal two HPGs will be used to process hybrid payments, although theoretically more HPGs could be involved.

6.6 Conclusions

In this chapter, we designed a hybrid payment protocol. The focus throughout the design was on the functional aspects of the protocol. Alternative solutions for realizing the different protocol elements were identified, described, compared and the most appropriate solutions were selected. Most of the protocol elements are assigned to the Hybrid Payment Gateway to keep the payer and payee functionality as simple as possible. Additionally, the robustness, security, and optimization of the protocol were discussed. These discussions showed that (1) the threats for the normal operation and security of the protocol are not considerably bigger than that of the existing systems, (2) hardly any money loss situations can occur, (3) commonly used security techniques can be employed to secure the interactions between the various components of the hybrid payment systems, and (4) that the protocol is optimized in case no interconnection is needed.

We also proposed the usage of multiple Hybrid Payment Gateways because of business, reliability, legal and risks reasons. HPGs can interoperate with each other using UPSs or other underlying service providers. Because the usage of UPSs may be too expensive and slow, it is preferred that the HPGs will interact with each other using the defined Data Transfer System.

6.7 References

[1] Ferreira Pires L. and Quartel, D.A.C., Protocol engineering, Lecture notes for The Protocol Engineering course at the University of Twente, Enschede, August 2001

[2] Dierks, T. and Allen, C., The TLS Protocol version 1.0, IETF RFC 2246, January 1999

Chapter 7

Demonstration and evaluation

The purpose of this chapter is to investigate whether an implementation of the hybrid payment system is achievable. The hybrid payment system can be implemented if the implementations of its constituent parts, i.e., the uniform payment systems and hybrid payment protocol are achievable. Because we cannot prove the implementability of these parts for all possible usage scenarios, we describe a typical payment scenario and use it in the subsequent studies (Section 7.1). To demonstrate that the uniform payment systems can be implemented, two case studies describe how existing payment systems can be (de)enhanced to provide the uniform payment service and prove that there are available underlying systems (Section 7.2). To demonstrate that the protocol is implementable, another two case studies describe the functionality of the protocol entities (Section 7.3). The studies give directions to realize an implementation. After the demonstrations, we draw the conclusions (Section 7.4).

This chapter also evaluates whether the design of the hybrid payment system satisfies the hard requirements formulated in Chapter 4 (Section 7.5). The role of the hard requirements was to guide the design the hybrid payment system.

7.1 Payment scenario

This section describes a typical payment scenario to show how the users experience and pay via the hybrid payment system for online videos. In this scenario, we suppose that a customer named *John Doe* found on the Internet an online merchant called *clipcollection.org*, which sells low priced music videos.

Assume *John Doe* is a user of the *Minitix*[1] micropayment system (see Section 3.4.2 for more details), and pays through a small software application called *PayAll*. This application is provided by the *RaboBank*, the operator of the *Minitix*. Actually, *PayAll* is an implementation of the *HPayer* protocol entity. *John Doe* uses this application for paying small amounts of money to various merchants.

Assume *clipcollection.org* has a contract with and uses the *Way2Pay* micropayment system. It receives monthly the amount of money paid by consumers and pays transaction fees in return. To receive payment acknowledgements, *clipcollection.org* runs a web server. This server receives and processes HTTPS[2] request messages from an application called *OneReceive* supplied by the *ING Bank*, the operator of *Way2Pay*. This application is in fact an implementation of the *HPayee* protocol entity. The information received in the payment acknowledgements is stored in a database table called *IncomingPayments*, so *clipcollection.org* knows which content transactions have been paid.

Both *Minitix* and *Way2Pay* are part of the hybrid payment system, so their users can pay each other regardless the system used by the other party. *PayAll* and *OneReceive* represent the two users in the hybrid payment system.

Figure 7.1 depicts the interactions that occur between *John Doe* and *clipcollection.org*, and between these users and the hybrid payment system, respectively. In interaction *A John Doe*, a fan of the band called *Xperience* selects from the Top 10 videos of *clipcollection.org* the latest video clip of this band. His browser sends an HTTPS request message to the web server of *clipcollection.org*.

In interaction *B* (an HTTPS response message), *clipcollection.org* sends *John* a web page that indicates *John* that he must pay for that video clip €0,75. This page also contains the account identifier of *clipcollection.org*, the unique identifier of the product (video) transaction and a DownloadURL. The identifiers are needed for John to initiate a payment. It is interesting to notice that the account identifier resembles the SWIFT code and the bank account number (or

1. All names, email addresses, URLs, etc. are used for illustrative purposes only.
2. Security is needed to prevent the modification of the merchant account identifier, amount of money, product transaction identifier or URL, and the theft of the DownloadURL.

IBAN number) used for international bank transfers. The DownloadURL provides access to the video clip once the payment is completed.

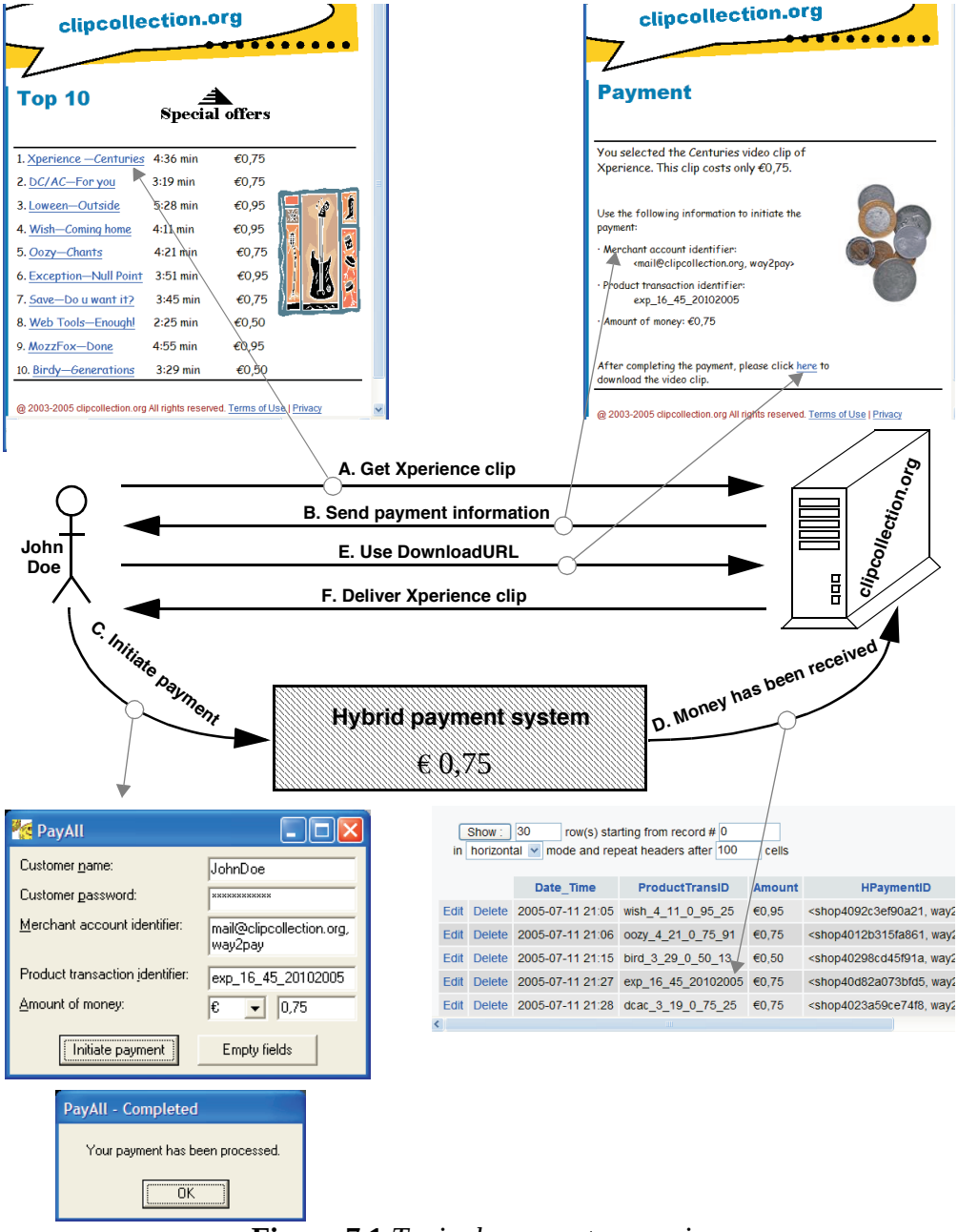


Figure 7.1 *Typical payment scenario*

John Doe next starts up the *PayAll* application, which is the interface to the hybrid payment system and fills in his user name and password, and the information received from *clipcollection.org*. When he clicks on the *Initiate payment* button, the payment information is submitted to the hybrid payment system in an HTTPS request message and the processing of the payment begins (Figure 7.1 interaction *C*). The initiation is accepted by the hybrid payment system because:

- *John*'s user name and password are correct, so he is authenticated by the system and authorized to use it;
- the account identifier of *clipcollection.org* exists and is in use;
- *Way2Pay* and *Minitix* support the transfer of €0,75;
- *John* has enough money in his account at the moment of the initiation.

Within a couple of seconds after the initiation, he receives a confirmation message[1] from his application. If the payment initiation could not accepted by the hybrid payment system, he would receive an error message instead.

In the meanwhile, in interaction *D*, the web server of *clipcollection.org* receives from *OneReceive* a HTTPS request message with the payment acknowledgement information stored in three parameters: the transaction identifier exp_16_45_20102005, the transferred amount of money €0,75 and the unique hybrid payment identifier <shop40d82a073bfd5, Way2Pay>. The server inserts this information with the current date and time into the *IncomingPayments* table.

Finally, *John* clicks on the DownloadURL specified on the payment page and sends an HTTPS request message to download the clip (*E*). *Clipcollection.org* receives the request for the content and checks its database to see whether the payment is confirmed for that product transaction. Since this transaction is found in the database, it sends the clip in an HTTP response message (*F*).

1. Although, in Section 5.2.2, only one interaction between a consumer and the hybrid payment system is defined, in an implementation such an abstract interaction needs to be decomposed into a couple of interactions. That is why, the payment initiation is followed by either a completion (or acknowledgement) message or by an error message.

If *John* clicks on the DownloadURL before the server received the payment confirmation, he would receive an error message, which lets him know that the payment is not completely processed yet. He can try to use this URL a few seconds later. The DownloadURL remains valid for 24 hours after the payment confirmation to make sure that the video can be downloaded.

7.2 Uniform payment service

The following two case studies describe how the payment services of two existing micropayment systems can be (de)enhanced such that they will provide the uniform payment service. The purpose of these studies is to demonstrate that implementations of the uniform payment systems are achievable, so there are available underlying payment systems.

For these case studies we selected two existing micropayment systems for which detailed information about their functionality was available and could be tested. These systems are *Minitix* and *Way2Pay* and were operational in the Netherlands at the time of writing this thesis. For further descriptions see Sections 3.4.2, 3.4.5 and Appendix A of this thesis.

Minitix is wrapped by a uniform payment system, which we call *UneePay*. *UneePay* plays the role of the C-UPS within the hybrid payment system (see Figure 6.2). *Way2Pay* is wrapped by another uniform payment system, which we will call *Uni4mSys*. *Uni4mSys* plays the role of the P-UPS.

7.2.1 Minitix

Figure 7.2 illustrates two software modules (or applications), which we call *Payer-Hancer* and *Payee-Hancer*. These modules perform the mapping between the payment services of *Minitix* and *UneePay* on the payer and payee side, respectively (Figure 5.7).

Payer-Hancer provides the uniform payment service to a HPayer, in our case to *PayAll* and interacts with *Minitix* to make the payment. *Payee-Hancer* provides the uniform payment service to a HPayee, in our case to *InterGate* and interacts with *Minitix* to receive confirmations of processed payments. Two

arrows represent six service primitives that occur between *Payer-Hancer* and *Minitix*, the numbering of the primitives indicates their occurrence sequence, and the direction of the arrows the main information flow. The parameters of these primitives (not shown in the figure) are determined based on the *UPay-Request* SP parameters, provided or generated by *Payer-Hancer* and *Minitix*. Another arrow represents the single service primitive that occurs between *Minitix* and *Payee-Hancer* and that confirms a processed payment to *Payee-Hancer*. The parameters of this primitive will be used to confirm the uniform payment to *InterGate*.

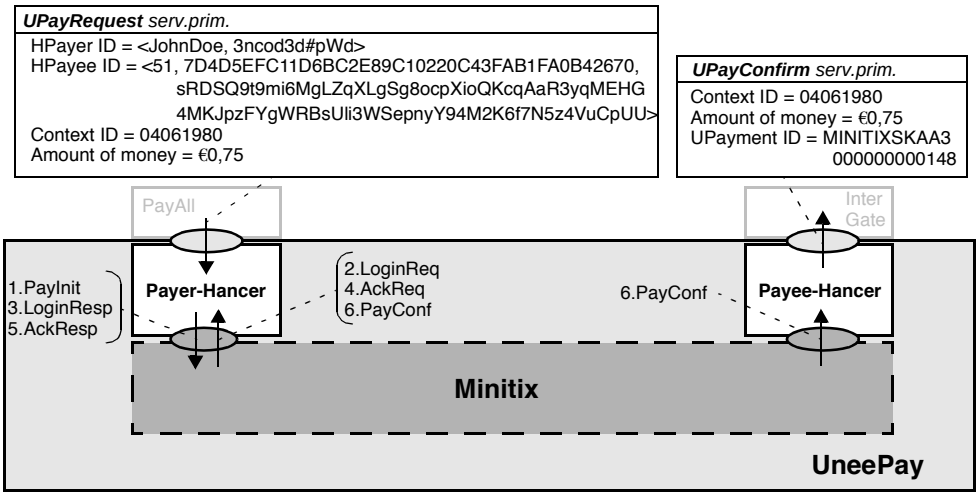


Figure 7.2 *Minitix wrapped by UneePay*

Payer-Hancer

Payer-Hancer solves the *payer* side differences between the uniform and *Minitix* payment services by (i) mapping the parameters of the *UPayRequest* SP onto the *Minitix* service primitive parameters and generating two other parameters, and (ii) contributing to the execution of the subsequent Minitix service primitives.

After receiving the parameters of the *UPayRequest* SP, *Payer-Hancer* maps these parameters onto the parameters of the *Minitix* SPs as follows:

- the HPayer ID parameter is decomposed and assigned to the Username and Password parameters;
- the HPayee ID parameter is decomposed and assigned to the VID, CertID and Sig parameters;
- the Context ID parameter is assigned to the OrdID parameter;
- the Amount of money parameter is decomposed and assigned to the Amount (payment value) and Currency (payment currency) parameters.

Payer-Hancer stores (i) the MType and MVer Minitix specific parameters, which describe the type of interaction and version of *Minitix*, and (ii) the Desc1, Desc2 and Desc3 product description parameters. None of these parameters plays an important role in the processing of the payment within UneePay, however, and can have constant (or null) values.

Additionally, *Payer-Hancer* generates (i) the MTime and Until parameters by assigning the current date and time to MTime and MTime + 1 hour to Until (so the payment needs to be completed within one hour after initiation); (ii) the VName and VSite (name and web site of the provider) provider specific parameters using the provider identifier; (iii) the ReturnURL and ErrorURL (two URLs where the consumer will be re-directed if the payment is successful or failed) parameters using the VSite parameter. Neither these parameters play an important role in the processing of the payment within *UneePay*, however.

Table 7.1 summarizes the parameters and their values that *Payer-Hancer* will provide to Minitix in the successive interactions. Section A.1 of the Appendix describes these interactions in detail. After the *PayConf* SP occurred, *Payer-Hancer* received a confirmation of the processed payment and the processing of the uniform payment ends for it. We note that, *Payer-Hancer* does not need to invoke the ReturnURL (or ErrorURL) received in the confirmation, as regular *Minitix* users need to do, because also *Payee-Hancer* receives a payment confirmation.

Table 7.1 *Minitix PayInit SP parameters*

Parameter name	Parameter value	Description
MType	MPS_PAYREQ	type of interaction (stored constant value)
MVer	0010	minitix version (stored constant value)
MTime	2005-07-11T21:26	date and time of initiation (generated value)
VID	51	provider identifier
VName	name51	provider name (generated value using VID)
VSite	site51.com	provider web site (generated value using VID)
OrdID	04061980	transaction id of provider
Amount	0,75	payment value
CurrencyCode	EUR	payment currency
DESC1	product	product description 1 (stored constant value)
DESC2		product description 2 (stored constant value)
DESC3		product description 3 (stored constant value)
Until	2005-07-11T22:26	date and time until the offer remains valid (generated)
ReturnURL	www.site51.com/result.php?res=1	success URL (generated value using VID)
ErrorURL	www.site51.com/result.php?res=0	failure URL (generated value using VID)
CertID	7D4D5EFC11D6BC2E89C10220C43FAB1FA0B42670	provider certificate issued by Minitix
Signature	sRDSQ9t9mi6MgLZqXLgSg8ocpXioQKcqAaR3yqMEHG4MKJpzFYgW.....	provider digital signature

Payee-Hancer

Payee-Hancer solves the *payee* side differences between the uniform and *Minitix* payment services by (i) mapping the parameters of the *PayConf* SP onto the *UPayConfirm* SP parameters, and (ii) contributing to the execution of the *PayConf* and *UPayConfirm* SPs.

Table 7.2 summarizes the *PayConf* SP parameters that *Payee-Hancer* receives from *Minitix*. *Payee-Hancer* performs the mapping between the parameters of the *PayConf* and *UPayConfirm* SPs as follows:

- the OrdID parameter is assigned to the Context ID parameter;
- the CurrencyCode and Amount parameters are combined and assigned to the Amount of money parameter;
- the TicketID and StackID parameters are combined and assigned to the UPayment ID parameter.

The VID parameter is not relevant to the uniform payment, so *Payee-Hancer* ignores (or discards) it. After the mapping, *Payee-Hancer* confirms the uniform payment to *InterGate* and completes the processing of the uniform payment.

Table 7.2 *Minitix PayConf SP parameters*

Parameter name	Parameter value	Description
VID	51	provider identifier
OrdID	04061980	transaction id of provider
Amount	0,75	payment value
CurrencyCode	EUR	payment currency
StackID	000000000148	identifier of the ticket roll from where the ticket originates
TicketID	MINITIXSKAA3	serial number of the ticket

7.2.2 Way2Pay

Figure 7.3 illustrates two software modules (or applications), which we call *Payer-Mapper* and *Payee-Mapper*. These modules perform the mapping between the payment services of *Way2Pay* and *Uni4mSys* on the payer and payee side, respectively.

Payer-Mapper and *Payee-Mapper* function similarly to the *Minitix* modules presented in the previous section. To avoid repetition, the rest of this section presents only those aspects of *Payer-Mapper* and *Payee-Mapper* which differ from the *Minitix* two modules.

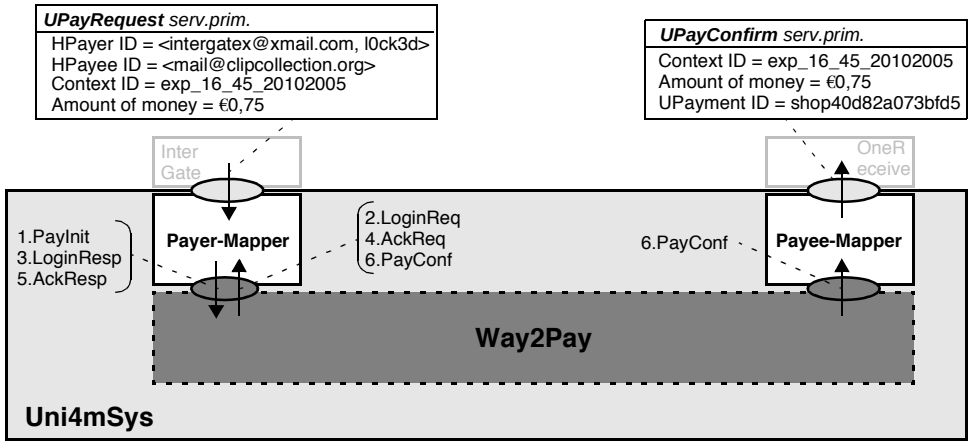


Figure 7.3 *Way2Pay wrapped by Uni4mSys*

Payer-Mapper

Payer-Mapper provides the uniform payment service to a HPayer, in our case to *InterGate* and interacts with *Way2Pay* to make a payment.

After receiving the parameters of the *UPayRequest* SP, *Payer-Mapper* maps these parameters onto the parameters of the *Way2Pay* SPs as follows:

- the HPayer ID parameter is decomposed and assigned to the Emailaddress and Password parameters;
- the HPayee ID parameter is assigned to the MerID parameter;

- the Context ID parameter and the payment value (without the currency) are combined together[1] (using the "::" marker) and assigned to the TID parameter. This combination is possible because Way2Pay transfers the TID transparently to the payee and does not process or verifies it.
- the Amount of money parameter is assigned without the currency to the AMT parameter.

Payer-Mapper stores three *Way2Pay* specific parameters: ItemName, which is a short description of the product, the SURL and FURL, which are two URLs where the consumer will be re-directed if the payment is successful or failed, respectively. None of these parameters plays an important role in the processing of the uniform payment within *Uni4mSys* and can have constant values.

Table 7.3 summarizes the parameters and their values that *Payer-Mapper* will provide to *Way2Pay* in the successive interactions. Section A.3 of the Appendix describes these interactions in detail. After the *PayConf* SP occurred, *Payer-Mapper* received a confirmation of the processed payment and the processing of the uniform payment ends for it. We note that, *Payer-Mapper* does not need to invoke the SURL or FURL received in the confirmation, as regular *Way2Pay* users need to do, because *Payee-Mapper* also receives a confirmation.

Table 7.3 *Way2Pay PayInit SP parameters*

Parameter name	Parameter value	Description
Emailaddress	intergatex@xmail.com	login name of consumer
Password	l0ck3d	password of consumer
MerID	mail@clipcollection.org	unique provider identifier
MName	name clipcollection.org	provider name (generated value using MerID)
TID	exp_16_45_20102005::0,75	transaction identifier combined with the payment value

1. Way2Pay uses only the transaction identifier from all payment initiation information to confirm the payment to the provider. The transaction identifier is combined with the payment value, so the provider will know the payment value. Payee-Mapper needs then to decompose the TID.

Table 7.3 *Way2Pay PayInit SP parameters (Continued)*

Parameter name	Parameter value	Description
AMT	0,75	payment value (in Euros)
ItemName	product description	short product description
SURL	http://www.confimapp.nl/ result.php?res=1	success URL
FURL	http://www.confimapp.nl/ result.php?res=0	failure URL

Payee-Mapper

Payee-Mapper provides the uniform payment service to a HPayee, in our case to *OneReceive* and interacts with *Way2Pay* to receive the confirmation of a processed payment. When it became operational, *Payee-Mapper* requested *Way2Pay* to email the confirmation of each processed payment.

Table 7.4 summarizes the *PayConf* SP parameters that *Payee-Mapper* receives from *Way2Pay*. Some of these parameters will be used to confirm the uniform payment to *OneReceive*.

Table 7.4 *Way2Pay PayConf SP parameters*

Parameter name	Parameter value	Description
TID	exp_16_45_20102005::0,75	transaction identifier combined with the payment value
MPurID	shop40d82a073bfd5	unique payment identifier of Way2Pay
ErrNo	0	error code (is 0 if the payment is successful)
ErrDesc	Thank you for your shopping by <provider name>	error description

If ErrNo equals zero then *Payee-Mapper* maps two of the parameters as follows:

- the first part of the TID parameter is assigned to the Context ID parameter (the "::" marker is recognized);
- the second part of the TID parameter and the Euro currency are assigned to the Amount of money parameter;
- the MPurID parameter is assigned to the UPayment ID parameter.

If ErrNo differs from zero, no mapping nor confirmation of the uniform payment will be performed.

The ErrNo and ErrDesc parameters are not relevant to the uniform payment, so *Payee-Mapper* ignores them. After the mapping, *Payee-Mapper* confirms the uniform payment to *OneReceive* and completes the processing of the uniform payment.

7.3 Hybrid payment protocol

The following two case studies describe how the hybrid payment protocol can be implemented. These studies are based on the usage scenario presented in Section 7.1 and describe the processing of hybrid payments:

- in a basic interconnection scenario in which one HPG is involved in the processing of a hybrid payment; and
- in an interconnection scenario in which two directly cooperating HPGs are involved in the processing of a hybrid payment.

These cases illustrate step-by-step (or chronological) the operation of the hybrid payment protocol, i.e., how the protocol elements designed in Chapter 6 are performed. The *Transfer product and payment information* protocol element is decomposed into several functions (e.g., data transfer, context identifier generation, etc.) because the performing of these functions intertwines with the performing of other protocol elements. For these cases, we assume that there are a reliable and secure *Data Transfer System* (DTS) and two UPSs introduced earlier, *UneePay* and *Uni4mSys*, which wrap *Minitix* and *Way2Pay*.

7.3.1 Basic interconnection scenario

This scenario is based on the design of the protocol elements and the definition of protocol messages in Sections 6.2 and 6.3. The state of the hybrid payment system at the moment John Doe clicked the "*Initiate payment*" button of *PayAll* is the following:

- *PayAll*: has received the payment information (see Figure 7.1 interaction *C*) and begins processing the hybrid payment;
- *InterGate*: is an HPG that interconnects the *UneePay* and *Uni4mSys* systems, and has a *UPS_Table, UniformPayments and CurrencyExchange* Oracle database table in which information about the uniform payment systems, the performed uniform payments, and the currency exchange rates, respectively, are stored;
- *OneReceive*: receives payment confirmations from *Uni4mSys* and confirms hybrid payments to *clicollection.org*.

Comparison

PayAll knows that *John Doe* has an account at *Minitix* (because the application is built for *Minitix* users only), and determines based on merchant account identifier the that *clipcollection.org* has an account at *Way2Pay*. Because these two systems differ, *InterGate* must be involved in the processing of the hybrid payment. *PayAll* then sends the necessary information to *InterGate*.

Data transfer to InterGate

PayAll sends the data to InterGate in order to determine the uniform payment initiation information for *UneePay*. Figure 7.4 illustrates the Data2HPG-PDU transferred via the Internet to *InterGate*. In this case the Internet is the Data Transfer System.

The 64.37.99.10 IP-address of *InterGate*, where this PDU needs to be transferred was provided to *John Doe* when he subscribed to use the *Minitix* system and was advised to use *InterGate* to pay merchants that use payment systems other than *Minitix*. This address is stored by *PayAll* and used if interconnection is needed.

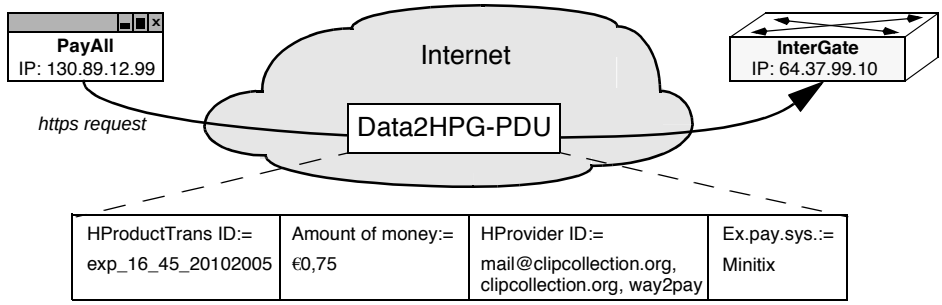


Figure 7.4 *Transfer of Data2HPG-PDU*

PayAll sends the Data2HPG-PDU to *InterGate* in an "HTTPS request" message. The type of this message implies that *InterGate* needs to sends back an answer in a "HTTPS response" message. This latter message will be used when *InterGate* sends the Data2HPayer-PDU to *PayAll*.

InterGate also receives with this PDU the 130.89.12.99 IP-address of *PayAll*.

Account identifier determination

PayAll and *InterGate* are responsible to determine the source and destination accounts for both uniform payment systems.

PayAll knows the source account identifier (i.e., *John*'s username and password) for *UneePay*. *InterGate* needs to provide the destination account identifier for *UneePay*, so it looks-up this identifier in the *UPS_Table*. The source account identifier for Uni4mSys is also looked-up in this table, while the destination account identifier is determined from the merchant account identifier. The SELECT SQL look-up commands executed by *InterGate* are:

```
SELECT HPAYEE_ID FROM UPS_TABLE
WHERE Ex_pay_sys="Minitix" INTO dest1;

SELECT HPAYER_ID FROM UPS_TABLE
WHERE Ex_pay_sys="Way2Pay" INTO src2;
```

The determined source and destination account identifiers are the following:

- *UneePay*: <JohnDoe, 3ncod3d#pWd> and <51, 7D4D5EFC11D6BC2E89C10220C43FAB1FA0B42670, sRDSQ9t9mi6MgLZqXLgSg8ocpXioQKcqAaR3yqMEHG4MKJpzFYgWRBsUli3WSepnyY94M2K6f7N5z4VuCpUU>;
- *Uni4mSys*: <intergatex@xmail.com, l0ck3d> and <mail@clipcollection.org>.

clipcollection.org identification

InterGate determines in this step whether the *clipcollection.org* is known by *Uni4mSys*. For this, it requests *Uni4mSys* to check whether the <mail@clipcollection.org> account identifier is know. The reply of *Uni4mSys* is positive, thus the account exists and is in use.

The interaction between *InterGate* and *Uni4mSys* could be implemented via the HTTP or HTTPS web protocols, which use a simple request-response type message exchange. Actually, the usage of HTTPS is not necessary because there are no security concerns for the identification.

Amount of money verification

InterGate verifies in this step whether *UneePay* and *Uni4mSys* can transfer €0,75. These systems inherit the supported payment values and currencies from the wrapped systems. *InterGate*, therefore, looks-up in the *UPS_Table* the supported minimal and maximal payment values, the default currency and set of currencies for *Minitix* and *Way2Pay*:

```
SELECT MIN,MAX,DEFAULT_CURR,SET_CURR
FROM UPS_TABLE WHERE Ex_pay_sys="Minitix"
INTO min1, max1, defcurr1, setcurr1;

SELECT MIN,MAX,DEFAULT_CURR,SET_CURR
FROM UPS_TABLE WHERE Ex_pay_sys="Way2Pay"
INTO min2, max2, defcurr2, setcurr2;
```

Afterwards, it runs the verification algorithm for each payment system, as explained in Section 6.2.4. Currency exchange is not needed. Because the 0,75

is between 0,10 and 10 for Minitix, and between 0,01 and 2500 for Way2Pay, *InterGate* concludes that the amount of money is supported by both payment systems. In conclusion, the amounts of money to be transferred are:

- *UneePay*: €0,75;
- *Uni4mSys*: €0,75.

Context ID generation

InterGate generates a unique Context ID that is related to the HProductTrans ID. When *InterGate* will receive this Context ID in a Pay-PDU from *PayAll*, it will know which hybrid payment is being processed and which uniform payment needs to be initiated next. *InterGate* will use the HProductTrans ID as Context ID in another Pay-PDU that will be transferred by *Uni4mSys* to *OneReceive*. The two context identifiers are:

- *UneePay*: 04061980 (generated);
- *Uni4mSys*: exp_16_45_20102005 (equals to the HProductTrans ID).

Data transfer to PayAll

All information needed for the uniform payments has been successfully determined and verified, so *InterGate* sends the uniform payment initiation information to *PayAll* in a Data2HPayer-PDU (Figure 7.5). This PDU is sent in an "HTTPS response" message via the Internet to the 130.89.12.99 IP-address of *PayAll*.

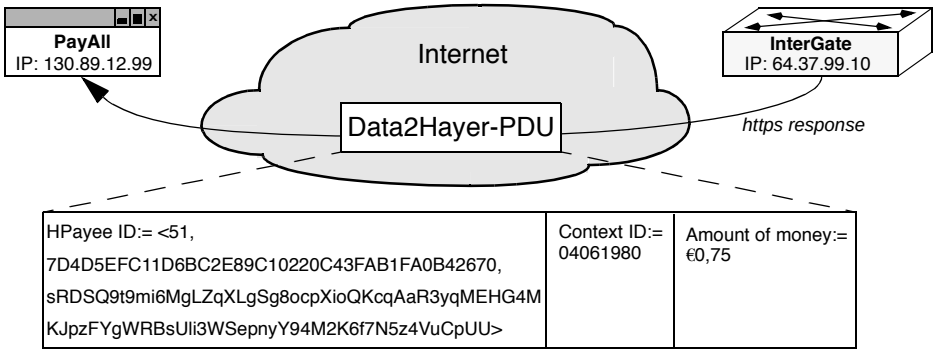


Figure 7.5 *Transfer of Data2HPayer-PDU*

Payment information transfer via UneePay

PayAll received the Data2HPayer-PDU and initiates the uniform payment for *UneePay* with the following information:

- Source account identifier: <JohnDoe, 3ncod3d#pWd>;
- Destination account identifier: <51, 7D4D5EFC11D6BC2E89C10220C43FAB1FA0B42670, sRDSQ9t9mi6MgLZqXLgSg8ocpXioQKcqAaR3yqMEHG4MKJpzFYgWRBsUli3WSepnyY94M2K6f7N5z4VuCpUU>;
- Context identifier: 04061980;
- Amount of money: €0,75.

The initiation is accepted by *UneePay* because the (i) John Doe is authorized to use the system, (ii) the destination account identifier is known to the system, (iii) the amount of money is supported, and (iv) John Doe's account supports this payment. After processing the payment, *UneePay* delivers a Pay-PDU to *InterGate* containing the following information:

- Context identifier: 04061980;
- Amount of money: €0,75;
- UPayment identifier: MINITIXSKAA3000000000148.

Although, for the hybrid payment, the subsequent uniform payment still needs to be processed, *PayAll* confirms the payment to John (Figure 7.1 interaction *C*). This can be done because in the design of the hybrid payment protocol is assumed that if the initialization information for all uniform payments is determined and the first payment is completed then the second uniform payment will also be completed and confirmed to *clipcollection.org*. Under these conditions the chance that the processing of the hybrid payment will fail is very small.

Both the uniform payment initiation and confirmation interactions between the users and *UneePay* could be implemented via the HTTPS protocol using request and response messages, as it is done in current practice by most existing micropayment systems.

Payment information transfer via Uni4mSys

After receiving the Pay-PDU, *InterGate* relates it to the hybrid payment and initiates the subsequent uniform payment for *Uni4mSys* with the following information:

- Source account identifier: <intergatex@xmail.com, l0ck3d>;
- Destination account identifier: <mail@clipcollection.org>;
- Context identifier: exp_16_45_20102005;
- Amount of money: €0,75.

The initiation is accepted by *Uni4mSys*, because (i) InterGate is authorized to use the system, (ii) the <mail@clipcollection.org> account is known to the system, (iii) the amount of money is supported and (iv) *InterGate* has enough money in its account. After processing the payment, *Uni4mSys* delivers a Pay-PDU to *OneReceive* with the following information:

- Context identifier: exp_16_45_20102005;
- Amount of money: €0,75;
- UPayment identifier: shop40d82a073bfd5.

Again, both the uniform payment initiation and confirmation interactions between the users and *Uni4mSys* could be implemented via the HTTPS protocol.

To complete the processing of the hybrid payment, *OneReceive* generates the HPayment ID (unique identifier of the hybrid payment) using the UPayment ID and the name of the existing payment system wrapped by Uni4mSys: <shop40d82a073bfd5, Way2Pay>. *OneReceive* then confirms the payment to *clipcollection.org*.

Payment information storage

InterGate stores the payment information in the *UniformPayments* table in order to provide audit support and to be able to trace back payments. The information storage is realized in two phases: first, the confirmation information

received from *UneePay* is stored; second, payment initiation information provided to the *Uni4mSys* is stored. Table 7.5 illustrates the record saved by *InterGate*.

Table 7.5 *UniformPayments record*

HPayee ID1	Context ID1	Amount 1	UPayment ID1	HPayer ID2	HPayee ID2	Context ID2	Amount 2
<51, 7D4D5..., sRDS...>	04061980	€0,75	MINITIXSKAA3000000000148	<intergatex@xmail.com, lock3d>	<mail@clipcollection.org>	exp_16_45_20102005	€0,75

The SQL commands that store the information in the *UniformPayments* table are:

```
INSERT INTO UNIFORMPAYMENTS
(HPAYEE_ID1, CONTEXT_ID1, AMOUNT1, UPAY_ID1
VALUES "<51,7D4D5EFC11D6BC2E89C10220C43FAB1FA0B42670,
sRDSQ9t9mi6MgLZqXLgSg8ocpXioQKcqAaR3yqMEHG4MKJpz
FYgWRBsUli3WSepnyY94M2K6f7N5z4VuCpUU>", "04061980",
"€0,75", "MINITIXSKAA3000000000148";

UPDATE UNIFORMPAYMENTS SET
HPAYER_ID2 = "<intergatex@xmail.com, lock3d>",
HPAYEE_ID2 = "<mail@clipcollection.org>",
CONTEXT_ID2 = "exp_16_45_20102005",
AMOUNT2 = "€0,75"
WHERE CONTEXT_ID1 = "04061980";
```

Summary

Figure 7.6 summarizes the PDU exchange sequence and a possible implementation of this exchange. First the Data-PDUs are exchanged between *PayAll* and *InterGate*, then one Pay-PDU is transferred from *PayAll* to *InterGate*, and another from *InterGate* to *OneReceive*.

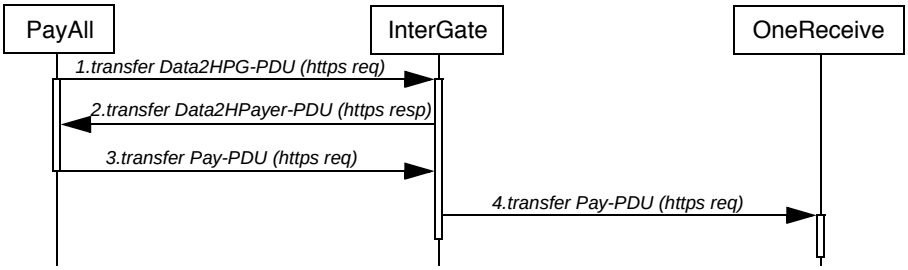


Figure 7.6 *PDU exchange*

Figure 7.7 summarizes the functionality of the hybrid payment protocol in terms of executed service primitives. These primitives occur between the interactions *C* and *D* from Figure 7.1. The direction of the arrows indicate the direction of the information flow. The internal actions of the various components are not shown.

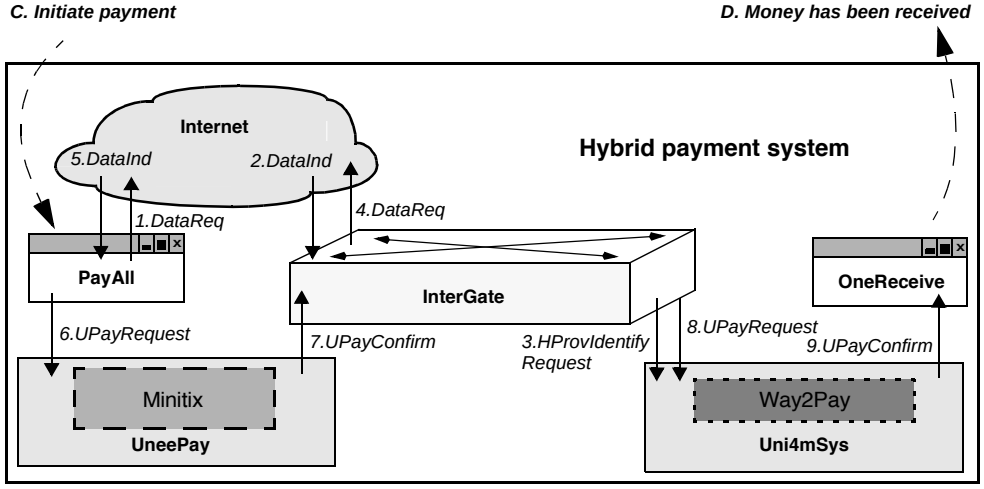


Figure 7.7 *Interconnection with one HPG*

7.3.2 Interconnection with two Hybrid Payment Gateways

This scenario is based on the direct cooperation between two HPGs, as specified in Section 6.5.2. The state of the hybrid payment system at the moment John Doe clicked the "*Initiate payment*" button is the following:

- *PayAll*: has received the payment information (see Figure 7.1 interaction *C*) and begins processing the hybrid payment;
- *InterGate*: is a HPG connected only to *UneePay* and will cooperate with *DutchGate*, which is connected to *Uni4mSys*; additionally, *InterGate* has a *UPS_Table, UniformPayments and CurrencyExchange* Oracle database table in which information about the uniform payment systems, the performed uniform payments, and the currency exchange rates, respectively, are stored;
- *DutchGate* is another Hybrid Payment Gateway, its information can be found in the *Payment Gateway Register* (i.e., a global register for all gateways); additionally, *DutchGate* has its own *UPS_Table, UniformPayments and CurrencyExchange* Oracle database tables in which information about the uniform payment systems, the performed uniform payments, and the currency exchange rates, respectively, are stored;
- *OneReceive*: receives payment confirmations from *Uni4mSys* and confirms hybrid payments to *clicollection.org*.

To avoid repetition, this scenario begins after *PayAll* compared the two payment systems, decided that interconnection is needed and sent the Data2HPG-PDU to the default gateway, *InterGate* (see Figure 7.4).

Account identifier determination (by PayAll and InterGate)

PayAll knows the source account identifier for *UneePay*. *InterGate* looks-up its destination account identifier for *UneePay* in the *UPS_Table*:

- *UneePay*: <JohnDoe, 3ncod3d#pWd> and <51, 7D4D5EFC11D6BC2E89C10220C43FAB1FA0B42670, sRDSQ9t9mi6MgLZqXLgSg8ocpXioQKcqAaR3yqMEHG4MKJpzF YgWRBsUli3WSepnyY94M2K6f7N5z4VuCpUU>.

As explained earlier, *InterGate* does not have a source account at *Uni4mSys*. But *InterGate* finds in the *Payment Gateway Register* another gateway, called *DutchGate* that has an account at *Uni4mSys*. *InterGate* needs, therefore, to send the payment information to *DutchGate*, which will perform the account

identifier determination for *Uni4mSys* later. Before the data transfer *InterGate* verifies the amount of money for *UneePay*, because if *UneePay* does not support the amount of money then contacting *DutchGate* is senseless.

Amount of money verification (by InterGate)

InterGate verifies whether UneePay supports the transfer of €0,75. So, it looks-up in the *UPS_Table* the supported minimal and maximal payment values, the default currency and set of currencies for *Minitix*:

```
SELECT MIN,MAX,DEFAULT_CURR,SET_CURR
FROM UPS_TABLE WHERE Ex_pay_sys="Minitix"
INTO min, max, defcurr, setcurr;
```

Afterwards, *InterGate* runs the verification algorithm, which specifies that €0,75 is supported.

Data transfer to DutchGate

InterGate sends a Data2HPG*-PDU to *DuchtGate* to request it whether it can pay *clipcollection.org* via *Uni4mSys*. This PDU contains information necessary for initiating the payment for *Uni4mSys*. Also this PDU is transferred via the Internet in an "HTTPS request" message. The 80.60.40.25 IP-address of *DutchGate* is available in the *Payment Gateway Register*. Figure 7.8 illustrates the structure and transfer of this PDU between the two gateways.

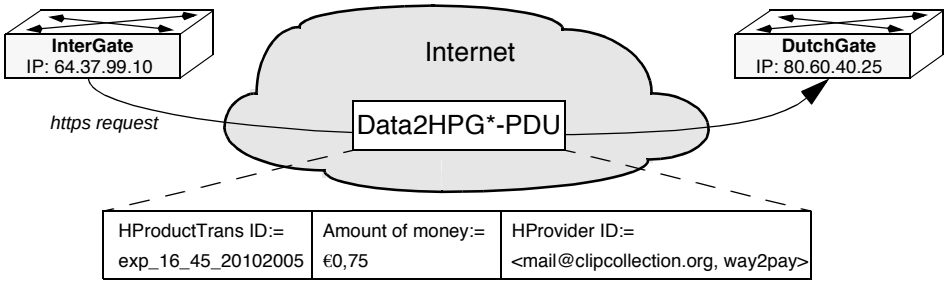


Figure 7.8 *Transfer of Data2HPG*-PDU*

DutchGate also receives with this PDU the 64.37.99.10 IP-address of *InterGate*.

Account identifier determination (by DutchGate)

DutchGate looks-up its source account identifier for *Uni4mPay* in the *UPS_Table* and determines the destination account identifier based on the information received from *InterGate*. The two account identifiers are the following:

- *Uni4mSys*: <dutchgate@dgate.nl, n3v3rL@nd> and <mail@clipcollection.org>.

***clipcollection.org* identification (by DutchGate)**

DutchGate sends the <mail@clipcollection.org> account identifier to *Uni4mSys* and requests this system to check whether the account exists and is in use. The answer of *Uni4mSys* is positive, the account exists, so *clipcollection.org* is identified.

The interaction between *DutchGate* and *Uni4mSys* could be implemented via the HTTP or HTTPS web protocols. Because there are no security concerns for the identification, the usage of HTTPS is not necessary.

Amount of money verification (by DutchGate)

DutchGate verifies whether *Uni4mSys* can transfer €0,75. First, it retrieves from the *UPS_Table* the minimal and maximal payment values, the default currency and set of currencies for Way2Pay, which is wrapped by *Uni4mSys*:

```
SELECT MIN,MAX,DEFAULT_CURR,SET_CURR
FROM UPS_TABLE WHERE Ex_pay_sys="Way2Pay"
INTO min, max, defcurr, setcurr;
```

Afterwards, *DutchGate* runs the verification algorithm, which determines that no currency exchange is needed and the amount of money is supported.

Data transfer to InterGate

Because the account identifier determination, account identification and the amount of money verification were successful, *DutchGate* is able to pay *clip-*

collection.org via *Uni4mSys*, so it sends an answer to *InterGate* that accepts paying *clipcollection.org*. Figure 7.9 illustrates the content of the Accept-PDU sent via the Internet in an "HTTPS response" message. The HProductTrans ID and HProvider ID included in the response will allow *InterGate* to identify uniquely which payment request was accepted by *DutchGate*.

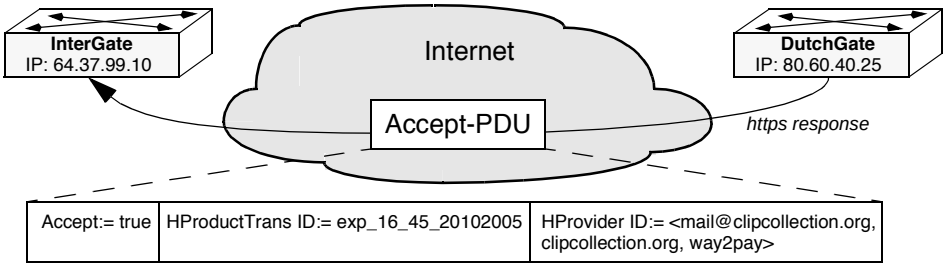


Figure 7.9 *Transfer of Accept-PDU*

Context ID generation (by InterGate), Data transfer to PayAll, Payment information transfer via UneePay (first uniform payment)

These three steps are performed as described in the first scenario in Section 7.3.1. At the end, *UneePay* will process the uniform payment initiated by *PayAll* and deliver a Pay-PDU to *InterGate* with the following information:

- Context identifier: 04061980;
- Amount of money: €0,75;
- UPayment identifier: shop40d82a073bfd5.

Data transfer to DutchGate

InterGate confirms next to *DutchGate* that the previously accepted payment can be initiated. For this, *InterGate* sends a Confirm-PDU via the Internet in an "HTTPS request" message (Figure 7.10).

The message type implies that InterGate is not expecting an answer from *DutchGate*, so considers that the subsequent uniform payment will be processed and then the hybrid payment will be confirmed to *clipcollection.org*. The HProductTrans ID and HProvider ID will allow *DutchGate* to identify uniquely the previously accepted payment request.

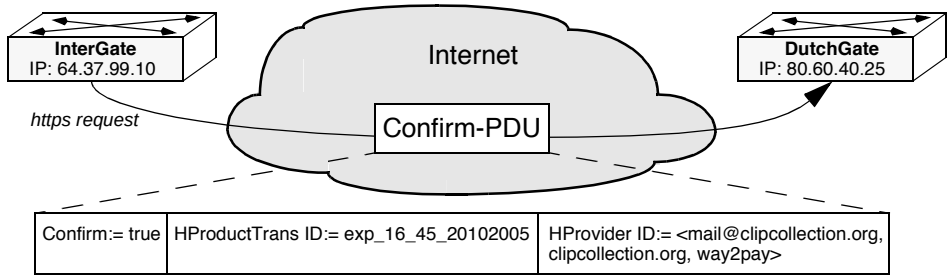


Figure 7.10 *Transfer of Confirm-PDU*

Information storage (by InterGate)

InterGate stores the available payment information in the *UniformPayments* table after sending the Confirm-PDU. This information consists of its destination account identifier (where the money was received), confirmation information from *UneePay*, the name of the other gateway as the HPayer ID2, and the information sent to it. Then the processing of the hybrid payment ends for *InterGate.*

Table 7.6 *UniformPayments record (InterGate)*

HPayee ID1	Context ID1	Amount 1	UPayment ID1	HPayer ID2	HPayee ID2	Context ID2	Amount 2
<51, 7D4D5..., sRDS...>	04061980	€0,75	MINITIXSKAA30000000000148	DutchGate	<mail@clipcollection.org>	exp_16_45_20102005	€0,75

The SQL command that stores the information in this table is:

```
INSERT INTO UNIFORMPAYMENTS
(HPAYEE_ID1, CONTEXT_ID1, AMOUNT1, UPAY_ID1,
HPAYER_ID2, HPAYEE_ID2, CONTEXT_ID2, AMOUNT2)
VALUES 164, "04061980", "€0,75",
"MINITIXSKAA3000000000148", "DutchGate",
<mail@clipcollection.org>, exp_16_45_20102005,
"€0,75";
```

Payment information transfer via Uni4mSys (second uniform payment)

DutchGate initiates the subsequent uniform payment after receiving the Confirm-PDU from *InterGate* with the following information:

- Source account identifier: <dutchgate@dgate.nl, n3v3rL@nd>;
- Destination account identifier: <mail@clipcollection.org>;
- Context identifier: exp_16_45_20102005;
- Amount of money: €0,75.

The initiation is accepted by *Uni4mSys*, because (i) *DutchGate* is authorized to use the system, (ii) has enough money in its account, (iii) the destination account exists and (iv) the amount of money is supported. As a consequence, *Uni4mSys* delivers a Pay-PDU to *OneReceive* with the following information:

- Context identifier: exp_16_45_20102005;
- Amount of money: €0,75;
- UPayment identifier: shop40d82a073bfd5.

OneReceive generates next the HPayment ID and confirms the hybrid payment similarly to the previous case study.

Information storage (by DutchGate)

DutchGate stores the payment information in the *UniformPayments* table in order to provide audit support and to be able to trace back the payments (Table 7.5). The storage occurs after the payment initialization is accepted. The processing of the hybrid payment ends for *DutchGate* after the information storage.

Table 7.7 *UniformPayments record (DutchGate)*

HPayee ID1	Context ID1	Amount 1	UPayment ID1	HPayer ID2	HPayee ID2	Context ID2	Amount 2
Inter-Gate				<dutchgate @dgate.nl, n3v3rL@nd>	<mail@clip-collec-tion.org>	exp_16_45 _20102005	€0,75

The SQL command that stores the information in this table is:

```
INSERT INTO UNIFORMPAYMENTS
(HPAYEE_ID1, CONTEXT_ID1, AMOUNT1, UPAY_ID1,
HPAYER_ID2, HPAYEE_ID2, CONTEXT_ID2, AMOUNT2)
VALUES "InterGate", "", "", "", "<dutchgate@dgate.nl,
n3v3rL@nd>",<mail@clipcollection.org>,
exp_16_45_20102005, "€0,75";
```

Summary

Figure 7.11 summarizes the functionality of the hybrid payment protocol in terms of executed service primitives in case two gateways are used. These primitives occur between the interactions *C* and *D* from Figure 7.1. The direction of the arrows indicate the direction of the information flow. The internal actions of the various components are not shown in this figure.

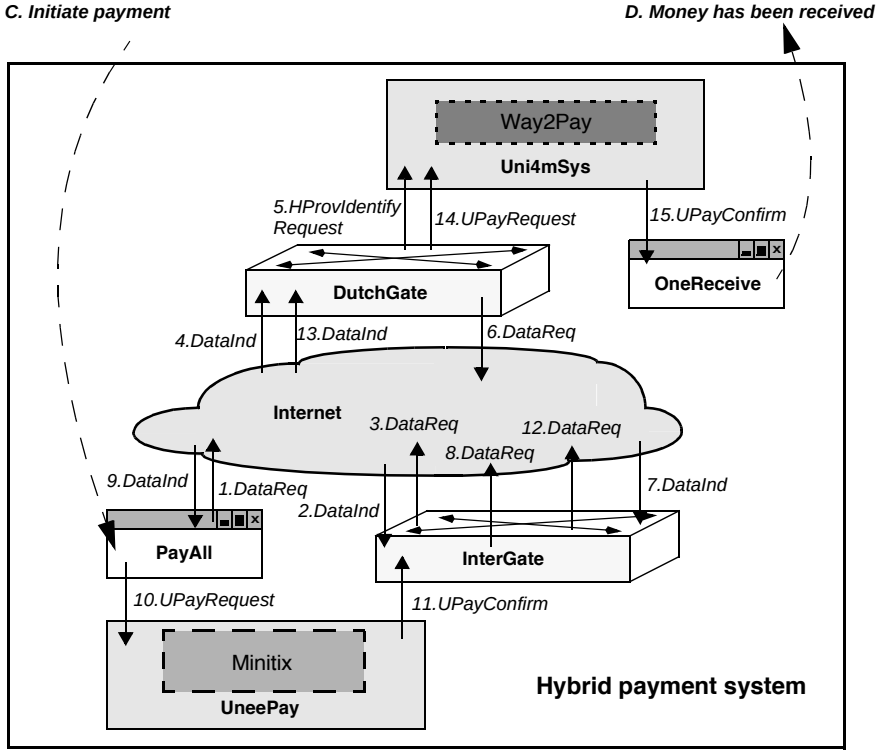


Figure 7.11 *Interconnection with two gateways*

7.4 Conclusions on the case studies

Based on the presented case studies, we conclude that an implementation of the hybrid payment system is indeed achievable.

Uniform payment systems

The first two case studies demonstrated that the differences between the existing and uniform payment services can be bridged, so implementations of the uniform payment systems are possible.

The bridging is realized without modifying the existing payment services, which is one of the requirements of micropayment system operators. The uniform payment initiation information is precise enough for the *payer* side modules to initiate a payment for the existing payment systems. The parameters of the *UPayRequest* SPs are mapped onto one or more parameters of the *PayInit* SPs. Additional parameters are stored or generated by the mapping modules. The confirmation information provided to *payee* side modules is also enough to be able to confirm the uniform payment, as required by the uniform payment service definition. The information needed to be transferred via the existing payment systems was in both cases possible.

These two case studies also show that the functionality of *Minitix* and *Way2Pay* are very similar. The only difference is the information they use.

Hybrid payment protocol

The other two case studies demonstrated that an implementation of the hybrid payment protocol is achievable because only common techniques (e.g., HTTP and HTTPS standard protocols for secure communication, Oracle, MS SQL Server or IBM DB2 database systems for searching and storing information) need to be used. The interactions in which sensitive payment information are exchanged are secured, thus so-called "*man-in-the-middle*" attacks are prevented or detected. Implementing and securing the *PayAll*, *InterGate* and *OneReceive* components is not a difficult task.

The involvement of the second gateway is totally transparent to *PayAll* and *OneReceive*. This involvement requires more Internet traffic (or Data Transfer System), but only 6 more interactions compared to the single gateway case (see Figure 7.7). Additionally, it does not affect the auditability of the hybrid payment system nor the tracing back of payments. The time-out period used by the system components (to avoid the blockage of the hybrid payment system) should be set such that the latency of the additional communication and processing is taken into account. Additionally, both gateways need to maintain a so-called "session" and store related information until their contribution to the processing of the hybrid payments ends. Bi-directional payments are also possible, provided that the gateways have the necessary source and destination accounts.

7.5 Evaluation

Chapter 4 formulated a number of hard requirements grouped in five categories: customer, merchant, Payment System Operator (PSO), Payment Gateway Operator (PGO), legal and regulatory. The next sections discuss whether the design of the hybrid payment system satisfies these requirements. Requirements that appear in several categories (e.g., security) are discussed together. Although, trust was defined as a less important requirement, its satisfaction can also be discussed.

7.5.1 Use a single payment system

This requirement was the primary requirement for customers and merchants. The hybrid payment system introduced and designed in this thesis allows customers and merchants to use their chosen micropayment system to pay or be paid, regardless the system of the other party. The only condition for an existing micropayment system to become part of the hybrid payment system is to be able to provide the uniform payment service. Section 5.6 concluded that the vast majority (82%) of the investigated payment systems comply with the uniform payment service. This means that users of these systems can use the hybrid payment system, so the requirement is satisfied.

7.5.2 Cross-border payments

This requirement was relevant for customers and merchants. The hybrid payment system supports cross-border payments, provided that the micropayment systems chosen by customers and merchants are part of the hybrid system. The cross-border potential of the hybrid payment system is high because the vast majority of the investigated existing payment systems can be incorporated in the hybrid payment system (see Section 5.6).

7.5.3 User-friendly payment system

This requirement was relevant for customers and merchants. This requirement is taken into account during designing the hybrid and uniform payment services (see Sections 5.2.2 and 5.5.1). Additionally, the payment scenario described in Section 7.1 showed that the interactions between a customer and a merchant, and between them and the hybrid payment system, are very similar to the ones described in Section 3.4. Small differences for the customers are the manual filling in of the payment information (which generally is hidden from customers in payment pages such as the one depicted in Figure 7.1), the usage of the small payment application (which resembles the payment interfaces of existing systems, but influences the portability of the hybrid system), and the manually performed request for the paid content after the payment was completed (which is in general solved with HTTP redirections). We note that, instead of this application a web-based interface that includes a Java-applet, for instance, could also be used. Merchants have to change only their web pages such that after a customer selects a piece of content, he/she receives the payment page with the necessary payment information. Such modifications, however, are necessary for any payment system they want to use. Their interactions with the hybrid payment system remains very similar to the interactions with their current systems.

In conclusion, the payment habits of the users do not really have to change, so the user-friendliness and convenience of the hybrid payment system are comparable with that of existing payment systems and did not decrease as a consequence of the interconnection with Payment Gateways.

7.5.4 Anonymity

This requirement was only relevant for customers. In this thesis, customers combine a consumer and payer role, and merchants a provider and payee role. The hybrid and uniform payment services are designed such that the identities of consumers and payers are not revealed to the providers or payees, respectively (see Sections 5.2.2 and 5.5.1). In this way, customers remain fully anonymous to the merchants. In case customers can remain anonymous with respect to the operators of their chosen payment systems, they will also remain anonymous to the operators of the hybrid payment system.

This conclusion is also demonstrated in this chapter in the payment scenario and the case studies, which showed that the identifier of the *John Doe* is kept within his *PayAll* payment application, which only uses it to initiate the uniform payment. *InterGate* finds out only the IP-address of *PayAll*. In this way, *John Doe* remains fully anonymous to the gateway and to the payment application of *clipcollection.org*.

7.5.5 Trust

This requirement was relevant for customers, merchants, payment system and gateway operators. Intuitively it seems that the HPGs are central components of the hybrid payment system in which all other components should trust. The HPGs are not a single point of trust and not each component needs to trust them, however.

HPGs have payer and payee contracts with the existing micropayment systems in order to perform their interconnection. Hence, the HPGs can be supervised based on such contracts by the operators of these systems. Moreover, due to their role and special functions (e.g., exchanging currencies, providing credit to post-paid customers), the HPGs fall under the supervision of financial authorities. The consequence of such supervision is that we do not see real trust problems. Therefore, the probability that the HPGs misuse their position to commit fraud is low. Probably even lower than the probability that regular users commit fraud. Nevertheless, the impact of the HPGs' fraud would be much higher than of fraudulent users, because it affects the operation of the whole hybrid payment system.

In conclusion, customers and merchants need only to trust their chosen payment system and they can trust the hybrid payment system with equal degree. PSOs and PGOs trust each other since they signed contracts with each other. A solution for enforcing trust between the PSOs, is the introduction of Trusted Third Party (TTP, [1]). In this way, PSOs that have a trust relation with the TTP will also trust each other.

7.5.6 Security

This requirement was present in each requirements category. Section 6.4.2 demonstrated that the security threats on the hybrid payment system are not considerably higher than that on existing payment systems. Additionally, all main security concerns (e.g., non-repudiation, authentication, authorization) were addressed throughout the service and protocol designs (see Sections 5.2.2, 5.5.1, 6.2.2, 6.2.3 and 6.2.6) and the case studies showed that commonly used security mechanisms can be used for the hybrid payment system as well. In conclusion, the security of the hybrid payment systems is comparable with that of existing payment systems and did not degrade considerably with the introduction of Payment Gateways.

7.5.7 Minimal changes

This requirement was relevant for the PSOs. As concluded in Section 5.6, the vast majority of all investigated existing payment systems do not need functional changes to comply with the uniform payment service. Additionally, the first two case studies demonstrated that two existing payment systems can provide the uniform payment service while their functionality remained unchanged (see Section 7.2). Hence, the design of the hybrid payment system satisfies this requirement.

7.5.8 Scalability

This requirement was relevant for PSOs and PGOs. The hybrid payment system scales with respect to the increasing number of users and payment volume because (i) a generic and systematic interconnection method was introduced (see Section 5.3.2), (ii) the usage of multiple gateways was proposed (see Section 6.5.1), (iii) the protocol elements were designed such that scala-

bility problems are avoided (see Sections 6.2.2 and 6.2.3), and (iv) current micropayment systems claim high scalability [2].

7.5.9 Vast majority

This requirement was relevant for the PGOs. This requirement is satisfied since 82% of the investigated existing micropayment systems can provide the uniform payment service and can therefore be interconnected with Payment Gateways (see Section 5.6).

7.5.10 Support for audit

This requirement was part of the legal and regulatory requirement category. This requirement is addressed in the design of the hybrid and uniform payment services (Sections 5.2.2 and 5.5.1) and payment information storage protocol element (Section 6.2.6). These designs and the fact the also current micropayment systems generate and store audit information, demonstrate that this requirement is satisfied.

7.5.11 Conclusion

Based on the previous discussions, we conclude that all hard requirements identified and formulated for this design phase of the hybrid payment system addressed in the design of the hybrid payment system.

Table 7.8 summarizes the requirements for each category and the score (or degree of satisfaction) of the requirements.

Table 7.8 *Summary of evaluation*

Category	Requirement	Score
Customer	Use a single payment system	✓
	Make cross-border payments	high potential (82% of investigated systems are part of our system)
	User-friendly payment system	comparable with the user-friendliness and convenience of existing payment systems
	Anonymity	✓
	Trust	equal with the trust in existing payment systems
	Security	comparable with the security of existing payment systems
Merchant	Use a single payment system	✓
	Receive cross-border payments	potential is high (82% of investigated systems are part of our system)
	User-friendly payment system	comparable with the user-friendliness of existing payment systems
	Trust	equal with the trust in existing payment systems
	Security	comparable with the security of existing payment systems
Payment System Operator	Minimal changes	the vast majority of investigated payment systems do not need functional changes before interconnection takes place
	Scalability	✓
	Trust	is achieved by signing payer and payee contracts and introduce a TTP

Table 7.8 *Summary of evaluation (Continued)*

Category	Requirement	Score
Payment Gateway Operator	Vast majority	the vast majority of investigated payment systems do can be interconnected
	Scalability	✓
	Trust	is achieved by signing payer and payee contracts and introduce a TTP
Legal and regulatory	Support for audit	✓
	Security	comparable with the security of existing payment systems

7.6 References

[1] Sprenkels, R.A.M et al., An Architecture for Reverse Charging in the Internet, In the Proceedings of IEEE Workshop on IP-oriented Operations and Management (IPOM 2000), Cracow, September 2000

[2] Párhonyi, R. et al., Second generation micropayment systems: lessons learned, In the Proceedings of The Fifth IFIP conference on e-Commerce, e-Business, and e-Government (I3E 2005), Poznan, October 2005

Chapter 8

Conclusions

This chapter presents the contributions and conclusions of the research presented in this thesis, and suggests directions for future work.

Section 8.1 describes briefly the background of this thesis. Section 8.2 presents the main and additional contributions. Section 8.3 presents the conclusion on the main research question. Section 8.4 gives answers on the research problems. Section 8.5 indicates some directions for future work. Section 8.6 formulates the epilogue.

8.1 Introduction

In the next years, the market for low value online content, like music and videos, is expected to grow substantially. To allow “pay-per-use” of such content, micropayment systems are expected to play an important role. Since there are already many competing micropayment systems on the market, customers and merchants are forced to use multiple systems.

Currently, to process a payment, both accounts need to be stored within the same micropayment system. This means that customers and merchants need to use multiple payment systems to serve all their needs. As a consequence, customers and merchants run into problems such as install multiple software packages and hardware devices, learn the usage of several systems, manage multiple accounts and e-wallets, remember multiple passwords, trust different payment system operators and so on. Three alternative solutions can be followed to solve the problems of customers and merchants. The first alterna-

tive solution is to select one existing micropayment system and introduce it world-wide. The second alternative solution is to develop of a new micropayment standard and deploy a system based on this standard. The third alternative solution is to introduce *Payment Gateways*, which interconnect the existing micropayment systems. The Payment Gateways and the interconnected systems create together a *hybrid payment system*, which performs so-called *hybrid payments*.

8.2 Contributions

This thesis describes *how* existing payment systems can be interconnected with Payment Gateways into a globally acceptable hybrid payment system, such that customers and merchants can always use their preferred payment system to pay each other.

The main contributions of this thesis are:

1. The development of a generic and systematic interconnection method for existing payment systems and an architecture of the hybrid payment system (Chapter 5).
2. The design of a uniform payment service, which can be provided on top of the majority of existing payment systems. Changes were proposed for the other systems to be able to comply with this service. This uniform payment service is essential in reaching global acceptability, ensures high scalability of the interconnection method and eases the design of the Payment Gateways (Chapter 5).
3. The design of a hybrid payment protocol such that (i) the threats for the normal operation and security of the protocol are not considerably bigger than that of the existing systems, (ii) hardly any money loss situations can occur, (iii) commonly used security techniques can be employed to secure the interactions between the various components of the hybrid payment systems and (iv) that the protocol introduces minimal overhead in case no interconnection is needed (Chapter 6).
4. The designed uniform payment service could guide the design of future micropayment systems such that new systems can be interconnected easily with existing systems. In this way, the uniform payment

service, possibly extended with interactions that have only local significance, could become a de facto standard for micropayment systems. We expect that future payment systems could become part of the hybrid payment system if they implement the uniform payment service or follow the proposed modifications, in case of distinct systems (Chapter 5).

Additional contributions of this thesis are:

1. The definition of precise accounting and payments related terminology (Chapters 2 and 3).
2. The comparison of currently used accounting terminology and the summary of the results of past and ongoing accounting related activities from the perspective of standardization organizations, research projects, commercial products and platforms (Chapter 2).
3. The comparison and classification of electronic payment systems based on their business and main functional characteristics (Chapter 3).
4. The demonstration that the interconnection method proposed for computer networks can be used for interconnecting the vast majority of investigated micropayment systems (Chapter 5).

8.3 Main conclusion

The main research question of this thesis is whether it is possible that customers and merchants use each their preferred payment systems to pay each other regardless of the system used by the other party?

We identified and compared three alternatives that can solve this question. The first two alternatives have several drawbacks such as resistance against abandoning existing micropayment systems, violation of free market rules, high introduction costs, legislative and regulatory differences between countries, long standardization process. The conclusion was that the third alternative, which proposed the introduction of Payment Gateways, is the most promising alternative and seemed to have a high potential for becoming successful. We also demonstrated that the introduction of Payment Gateways is feasible, by (i)

developing an architecture of the hybrid payment system that includes such gateways, (ii) demonstrating that the hybrid payment system can be implemented and (iii) that the design of our system satisfies the hard requirements. The developed architecture specifies how the interconnection is realized and how the payments are processed.

The Payment Gateways offer a non-intrusive solution since existing micropayment systems operate continuously unchanged. Additionally, customers and merchants can pay each other in a transparent manner as they do not have to deal directly with the Payment Gateways. The Payment Gateways receive money from customers, pay in return the merchants and bridge all differences between the existing micropayment systems. So, each payment processed by the hybrid payment system is in fact a chain of payments processed by the existing payment systems.

Despite of the interconnection, the hybrid payment system can be used without restrictions to pay for all kind of products as the existing payment systems are used. Customers will be offered a very large variety of products because they can now buy products from hundreds of merchants that use other payment systems. The products include online music, video, parking and movie tickets.

8.4 Conclusions per research problem

The next sections answer the research problems formulated in Section 1.4.

8.4.1 How do payments fit into the accounting process?

Chapter 2 studied accounting and the relation between the accounting functions. To model this relation, Chapter 2 presented an accounting architecture and defined a precise accounting terminology. The architecture shows that payments are a sub-function of billing, which is one of the five accounting functions.

The study on the state of the art in accounting concluded that the majority of the accounting work focuses on transport accounting, especially on the metering function as it was expected that the DiffServ and IntServ technologies will

be widely deployed. Product accounting received much less attention. We learned that there are no standards for online micropayments, and the payments function is often outsourced to specialized parties such as banks, Internet Service Providers or other business organizations that operate or are involved in the operation of payment systems.

8.4.2 What are the main characteristics of the payment systems? What kind of payment systems exist?

Chapter 3 presented an extensive study on many existing payment systems that were available on the market at the time of writing this thesis. We developed a characterization model, which is used to describe and categorize the payment systems from business and functional viewpoints. From a business viewpoint, we identified which roles need to be played in an (electronic) payment system. From a functional viewpoint, we identified which characteristics are of importance to be able to interconnect the existing payment systems.

The main functional characteristics are the payment initiation and acknowledgement interactions between payment system users (i.e., consumers and providers) and the system itself, the information (e.g., the source and destination account of a payment, the amount of money to be transferred) exchanged during these interactions, and the payment system usage conditions posed to users. Based on these characteristics, the existing payment systems can be grouped as follows:

- according to the way payments are initiated: (1) systems that require consumers to initiate the payments, and (2) systems that require both users to initiate the payments (we call this jointly initiated payments);
- according to the way payments are acknowledged: (1) systems that provide acknowledgement to one or both users, and (2) systems that do not provide explicit acknowledgements, but deliver the paid product(s) instead;
- according to the usage conditions: (1) pre-paid systems are those that require consumers to transfer money to the system before they can initiate payments, and (2) post-paid systems are those that authorize consumers to initiate payments before they transfer money to the system.

A striking observation is that the functionality of existing micropayment systems is not that different! Behind proprietary interfaces the systems look rather alike and a subset of functionality is found back in many systems. This finding played an important role in the design of the hybrid system, because the functional characteristics determine whether the existing payment systems can be interconnected or not, and influence the interconnection method as well. We believe that, this subset of common functionality could be the starting point for developing a micropayment standard.

8.4.3 What are the requirements for the targeted hybrid payment system?

Based on the conclusions of Chapter 3, literature, expected requirements of the stakeholders of our system, legal and regulatory frameworks, Chapter 4 identified and formulated the requirements for the hybrid payment system. These requirements were grouped into five categories: customer, merchant, Payment System Operator, Payment Gateway Operator, legal and regulatory requirements. These categories contain two types of requirements: hard and less important. The hard requirements are meant to guide and then evaluate the design of the hybrid payment system. The hard requirements were addressed in the design phase presented in this thesis, while the less important requirements will be addressed in the following design phases. Chapter 7 concluded that the design of the hybrid payment system satisfies these requirements.

8.4.4 How is the interconnection modelled and realized?

Chapter 5 and 6 presented an architecture of the hybrid payment system designed in three phases. In the first phase, we used the functional characteristics of existing payment systems and translated the hard requirements into functional requirements and design decisions. In the second phase, we used the functional requirements to design the hybrid payment service. In the third phase, we introduced an interconnection method for existing payment systems and developed the hybrid payment protocol.

The introduced interconnection method was inspired by a standardized method used for interconnecting heterogeneous computer networks into one network. Our method allows the systematic interconnection of payment systems. This method requires the (de)enhancement of payment systems towards a uniform

service level before the interconnection takes place. The common subset observed in the functionality of the existing systems was guiding the uniform payment service design.

It is likely that the interconnection will be provided by multiple Payment Gateways to overcome the shortcomings of a single Payment Gateway. This conclusion comes after comparing the cases of a single and multiple Payment Gateways based on business, legal, reliability, and investments arguments. We expect, however, that for performing of a hybrid payment maximal two Payment Gateways will be involved.

Chapter 7 presented a few case studies, which demonstrated that (i) an implementation of the hybrid payment system is achievable by indicating how the implementation of the various components should be realized and (ii) that the design of the system satisfies the hard requirements.

8.4.5 Which classes of payment systems can be interconnected?

Chapter 5 conducted a compliance analysis of existing payment systems to determine which systems could comply with the uniform payment service. The conclusion of this analysis is that systems, which require a specific type of jointly initiated payments or do not provide explicit acknowledgements, have compliance problems. Such systems cannot provide the uniform payment service and require functional modifications. In case the payment initiations are the source of the compliance problems, these systems need to either support payer initiated payments or the other type of jointly initiated payments. In case the payment acknowledgements create compliance problems, these systems need to provide an acknowledgement to one of the users. After performing the suggested changes these systems will be able to provide the uniform payment service and, therefore, can be interconnected.

Payment systems that fall into the other categories can be interconnected. The information exchanged during payment initiations and acknowledgements, and the payment system usage conditions did not cause any compliance problems.

Because only three out of seventeen studied systems have compliance problems, we concluded that the majority of the systems can be interconnected at

any time without modifying their functionality. The three systems that require functional modifications are click&buy, Bitpass and WebCent.

8.5 Future research

We suggest some possibilities to continue the research presented in this thesis.

Electronic payment systems that use other communication channels than the Internet are, for instance, mobile payment systems. Also these systems have a great potential for micropayments. The interconnection of mobile payment systems and their incorporation into the hybrid payment system would expand even more the market for e-payments. In this way, even more customers and merchants from the Internet- and mobile-worlds could become part of a growing marketplace. Hence, the interconnection of the mobile and Internet-based payment systems using Payment Gateways would be of great interest.

Payment infrastructures are expensive and require a rather long return on investment. As explained in Chapter 3, a failure reason of the first generation micropayment systems was the lack of capital and funding until they become profitable. A cost analysis of how much would cost the implementation of the Payment Gateways was not addressed in this thesis. Nevertheless, such an analysis would provide more information about the hybrid payment system's chance of success from an economical point of view.

A study on potential Payment Gateway Operators would be another subject for further research. Potential operators are, for example, ISPs, banks, or other independent business organizations. Such a study should compare these operators based on an evaluation model and indicate which are the most appropriate to provide a Payment Gateway.

8.6 Epilogue

This thesis presented a technical design of a micropayment system, while other aspects of this system were omitted as they were out of scope. At the time of writing, it is difficult to foresee how the micropayment systems will evolve and whether they will really break through as credit card systems did. Currently

there are only a few operational credit card systems, but which have global coverage. Business and economical aspects of micropayment systems need to be studied to be able to determine their evolution. Until then, our studies show that current micropayment systems have a higher chance of success than their predecessors in the 1990s. We believe that, the end effect of competition will be that only a few, globally accepted micropayment systems will survive.

APPENDIX

A. Payment message diagrams

The next sections describe the functionality of several existing micropayment systems in terms of message sequence diagrams and exchanged information. They also describe the way information about these systems was obtained.

The message sequence diagrams illustrate the involved entities (vertical lines) and the messages exchanged between these entities (arrows). Messages with a roman numbering (I, II, III etc.) occur before payments can be made and do not occur for every payment. The other messages occur for every single payment.

The information exchanged in the various messages is presented in terms of parameters of the messages. The tables containing the parameters also specify the type of the message (e.g., http request or http response).

A.1 Minitix payments

The Minitix time sequence diagram was realized based on:

- Minitix demo & FAQ;
- Minitix implementation documents for merchants;
- contacting Minitix support;
- registering and using Minitix;
- testing Minitix at www.droomtuinen.nl and www.oor.nl;
- inspecting the source of web pages provided by content servers and Minitix;
- capturing network traffic data with Ethereal.

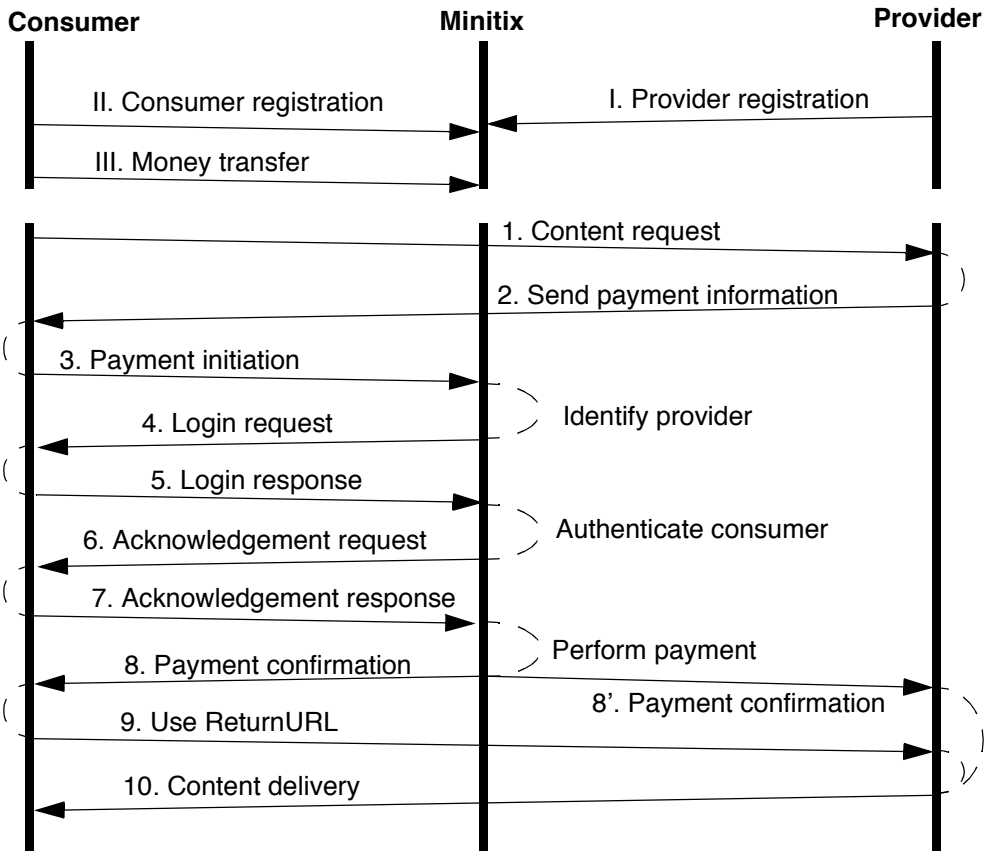


Figure 1.1 *Minitix time sequence diagram*

Table 1.1 *Parameters of Minitix interactions*

Sequence	HTTP message	Parameters (as named by Minitix)
1. Content request	HTTP REQ	Content ID (e.g, URL)
2. Send payment inf.	HTTP RESP	Minitix URL, MType, MVer, MTime, VID, VName, VSite, OrdID, Amount, CurrencyCode, Desc1, Desc2, Desc3, Until, ReturnURL, ErrorURL, CertID, Sig
3. Payment initiation	HTTPS REQ	-same as sequence 2-
4. Login request	HTTPS RESP	VSite, OrdID, CurrencyCode, Amount, Session ID (=SessionLogin)
5. Login response	HTTPS REQ	Session ID, Username, Password
6. Acknowledgement request	HTTPS RESP	Username, CurrentAccBalance, OrdID, CurrencyCode, Amount, VName
7. Acknowledgement response	HTTPS REQ	Session ID, ButtonValue (=submit)
8. Payment confirmation	HTTPS RESP	Currency Code, Amount, ReturnURL
8'. Payment confirmation	HTTPS RESP	VID, OrdID, CurrencyCode, Amount, StackID, TicketID
9. Use ReturnURL	HTTP REQ	Return URL
10. Content delivery	HTTP RESP	Content download page

A.2 Wallie payments

The Wallie time sequence diagram was realized based on:

- Wallie demo and FAQ for merchants and customers;
- Wallie implementation documentation for merchants;
- Using Wallie to buy content from www.computertotaal.nl;
- Inspecting the source of web pages provided by content servers;
- Capturing network traffic data with Ethereal.

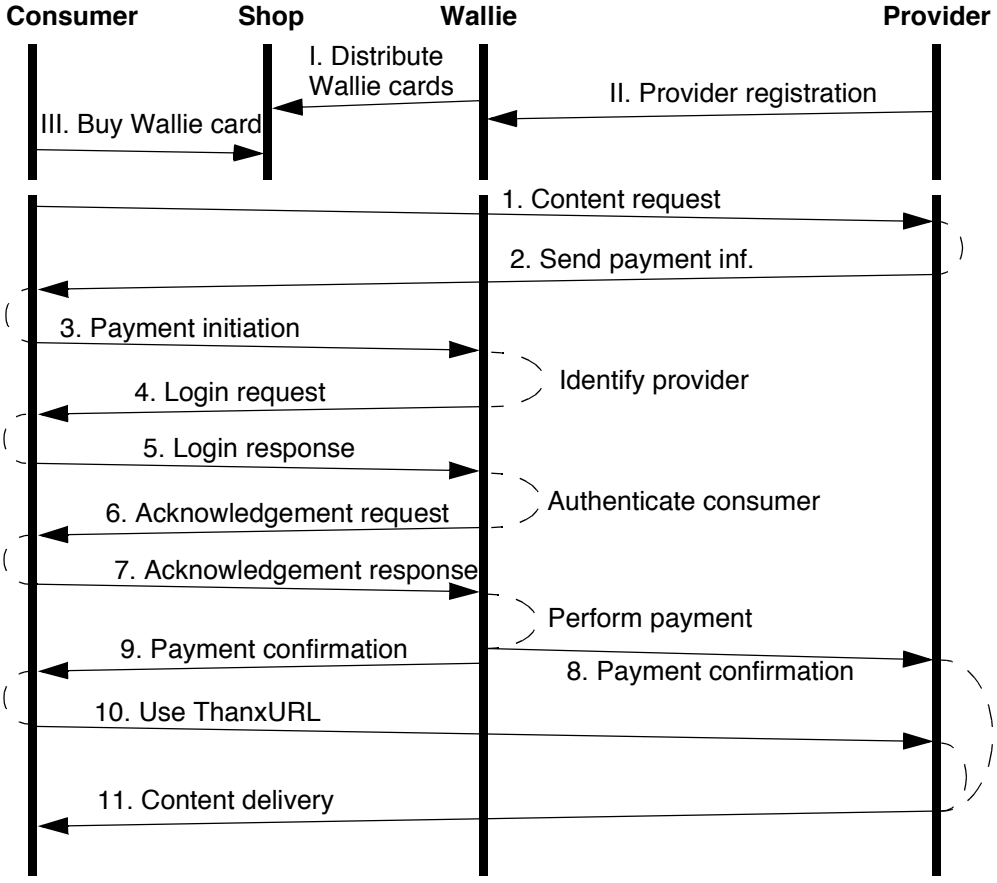


Figure 1.2 *Wallie time sequence diagram*

Table 1.2 *Parameters of Wallie interactions*

Sequence	HTTP message	Parameters (as named by Wallie)
1. Content request	HTTP REQ	Content ID (e.g, URL)
2. Send payment inf.	HTTP RESP	Wallie URL, MerchID, ShoppingCartID, Amount (expressed in Eurocents)
3. Payment initiation	HTTPS REQ	-same as sequence 2-
4. Login request	HTTPS RESP	MerchName, Amount (other params?)
5. Login response	HTTPS REQ	AccountID (other params?)
6. Acknowledgement request	HTTPS RESP	MerchName, Amount, AccountId, CurrentAccBalance, NewAccBalance (other params?)
7. Acknowledgement response	HTTPS REQ	ButtonValue (other?) (or GOTO step 5 and fill in another AccountID)
8. Payment confirmation	HTTP REQ	ResultCode, ShoppingCartID, TransDate, TransTime, Password, TestMode, TransferAmount, TransactionCosts, TotalAmount, Sleutel
9. Payment confirmation	HTTPS RESP	MerchName, Sleutel, Date, Time, Amount, ThanxURL
10. Use ReturnURL	HTTP REQ	ThanxURL
11. Content delivery	HTTP RESP	Content download page

We note that, (i) due to the consumer-side Flash interface and SSL encryption, some of parameters of the login and acknowledgement parameters could not be defined precisely, and (ii) the web server of Wallie supports SSL encryption up to 128 bits depending on the browser capabilities of the consumer.

A.3 Way2Pay payments

The Way2Pay time sequence diagram was realized based on:

- Way2Pay FAQ and implementation documents for merchants;
- registering to Way2Pay and buying content from www.yeahronimo.nl;
- inspecting the source of web pages provided by content servers and Way2Pay;
- capturing network traffic data with Ethereal.

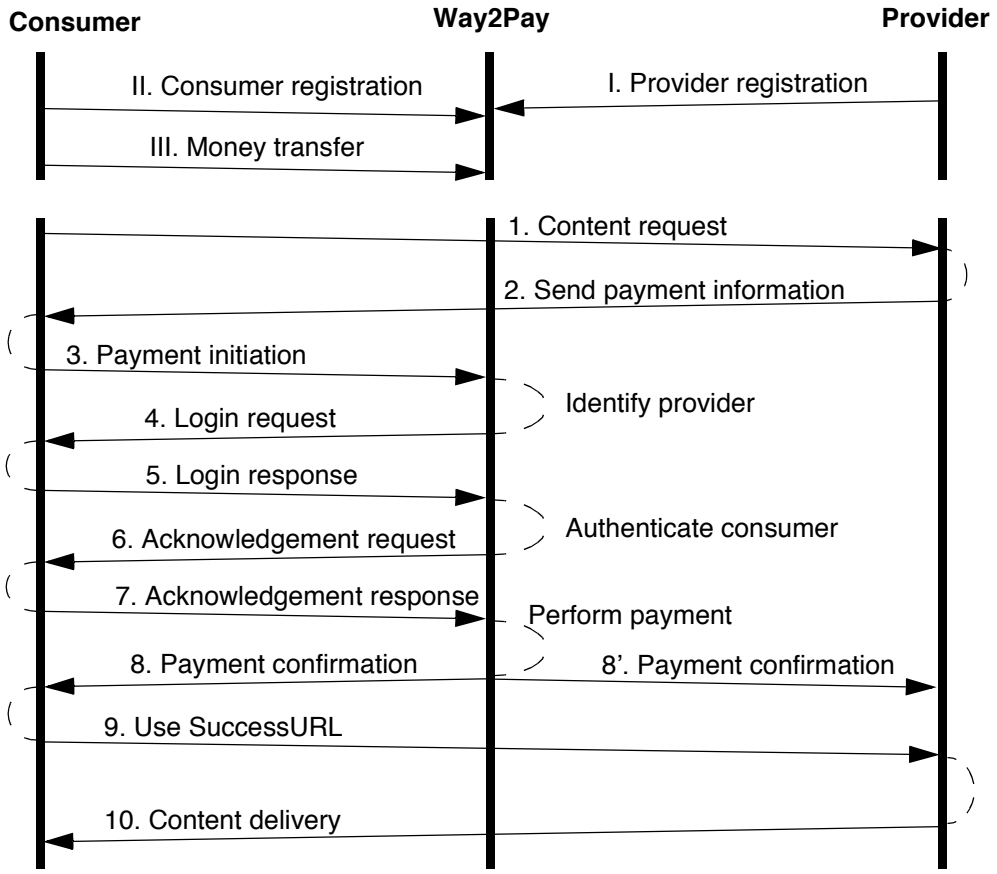


Figure 1.3 *Way2Pay time sequence diagram*

Table 1.3 *Parameters of Way2Pay interactions*

Sequence	HTTP message	Parameters (as named by Way2Pay)
1. Content request	HTTP REQ	Content URL
2. Send payment inf.	HTTP RESP	Way2Pay URL, MerId (merchant id = merchant email address), MName (merchant name), TID (transaction id), ItemName, AMT (amount of money), SuccessURL, FailureURL
3. Payment initiation	HTTPS REQ	-same as sequence 2-
4. Login request	HTTPS RESP	MerchID, MName, TID, ItemName, Amount, SuccessURL, FailureURL
5. Login response	HTTPS REQ	MerchID, MName, TID, ItemName, Amount, SuccessURL, FailureURL, Emailaddress, Password
6. Acknowledgement request	HTTPS RESP	MName, MerID, ItemName, Amount
7. Acknowledgement response	HTTPS REQ	Way2Pay script URL, TKN (=W2P session id), Code (=ack or nack)
8. Payment confirmation	HTTPS RESP	SuccessURL (suppose the payment is successfully processed), MName, MerID, AMT, ItemName, TID, MPurID (W2P reference number), ErrNo (fault code, 0 if successful), ErrDesc (error description, 'thank you' message if successful)
8'. Payment confirmation	HTTPS REQ	MName, MerID, Amount, ItemName, TID, MPurID, ErrNo, ErrDesc
9. Use ReturnURL	HTTP REQ	SuccessURL, TID, MPurID, ErrNo, ErrDesc
10. Content delivery	HTTP RESP	Content download page

A.4 PaySafeCard payments

The PaySafeCard (PSC) time sequence diagram was realized based on:

- PSC demo & FAQ;
- PSC document for merchants;
- testing at www.foilshop.de;
- inspecting the source of web pages provided by providers and PSC;
- capturing network traffic data with Ethereal.

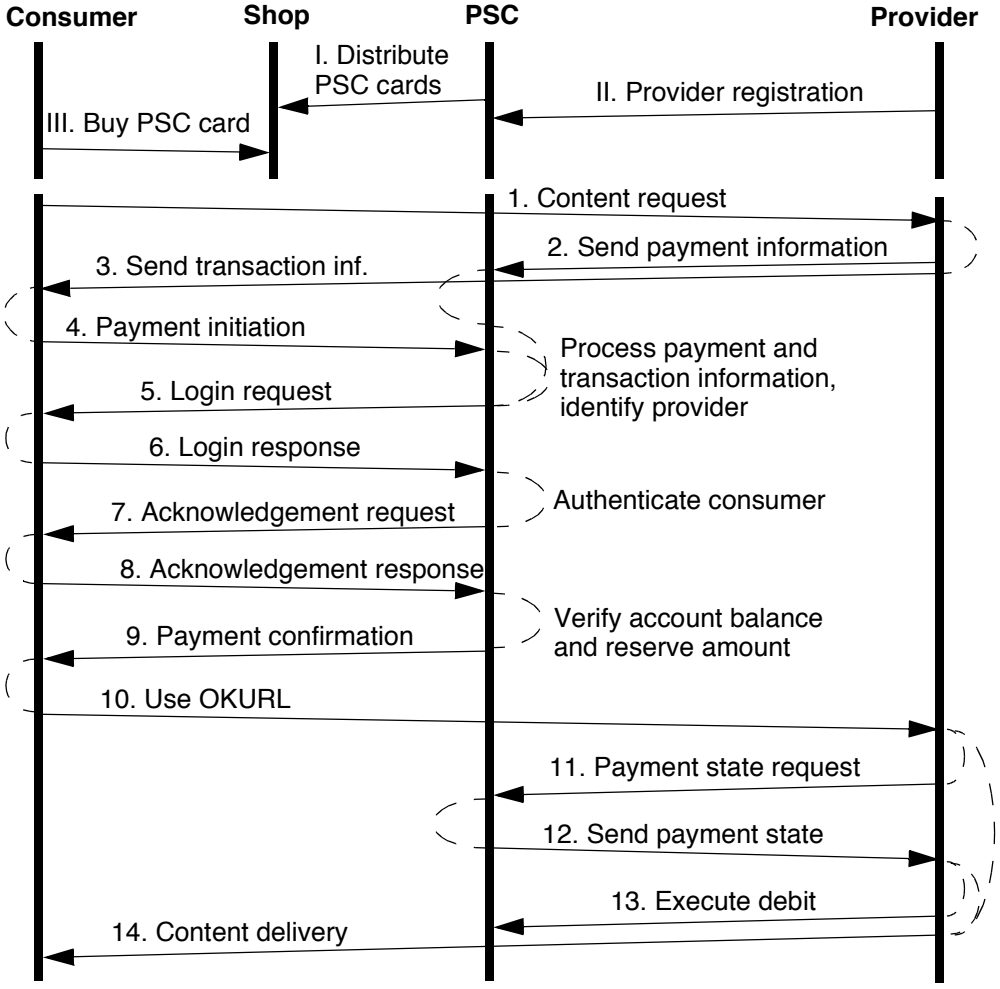


Figure 1.4 *PaySafeCard time sequence diagram*

Table 1.4 *Parameters of PaySafeCard interactions*

Sequence	HTTP message	Parameters (as named by PaySafeCard)
1. Content request	HTTP REQ	Content ID (e.g, URL)
2. Send payment inf.	HTTPS REQ	MerchID, MerchTransID, Amount, Currency, ReportingCriteria, Limitamount, MicropaymentID, OKURL, NOKURL, Config
3. Send transaction inf.	HTTPS RESP	PaySafeCard URL, MerchTransID (and other params)
4. Payment initiation	HTTPS REQ	-same as sequence 3-
5. Login request	HTTPS RESP	MerchID, MerchTransID, Amount, Currency, Language
6. Login response	HTTPS REQ	MerchID, MerchTransID, Amount, Currency, Language, AccountID
7. Acknowledgement request	HTTPS RESP	(params?)
8. Acknowledgement response	HTTPS REQ	ButtonValue (other params?)
9. Payment confirmation	HTTPS RESP	OKURL
10. Use OKURL	HTTP REQ	OKURL
11. Payment state request	HTTPS REQ	MerchID, MerchTransID, Config
12. Send payment state	HTTPS RESP	State
13. Execute debit	HTTPS REQ	MerchID, MerchTransID, Amount, Currency, Close, Config
14. Content delivery	HTTP RESP	Content download page

We note that, this system has not been fully tested and some of the parameters could not be discovered.

A.5 Paynova payments

The Paynova time sequence diagram was realized based on:

- Paynova FAQ;
- testing Paynova at q-park.se;
- inspecting the source of web pages provided by providers and Paynova;
- capturing network traffic data with Ethereal.

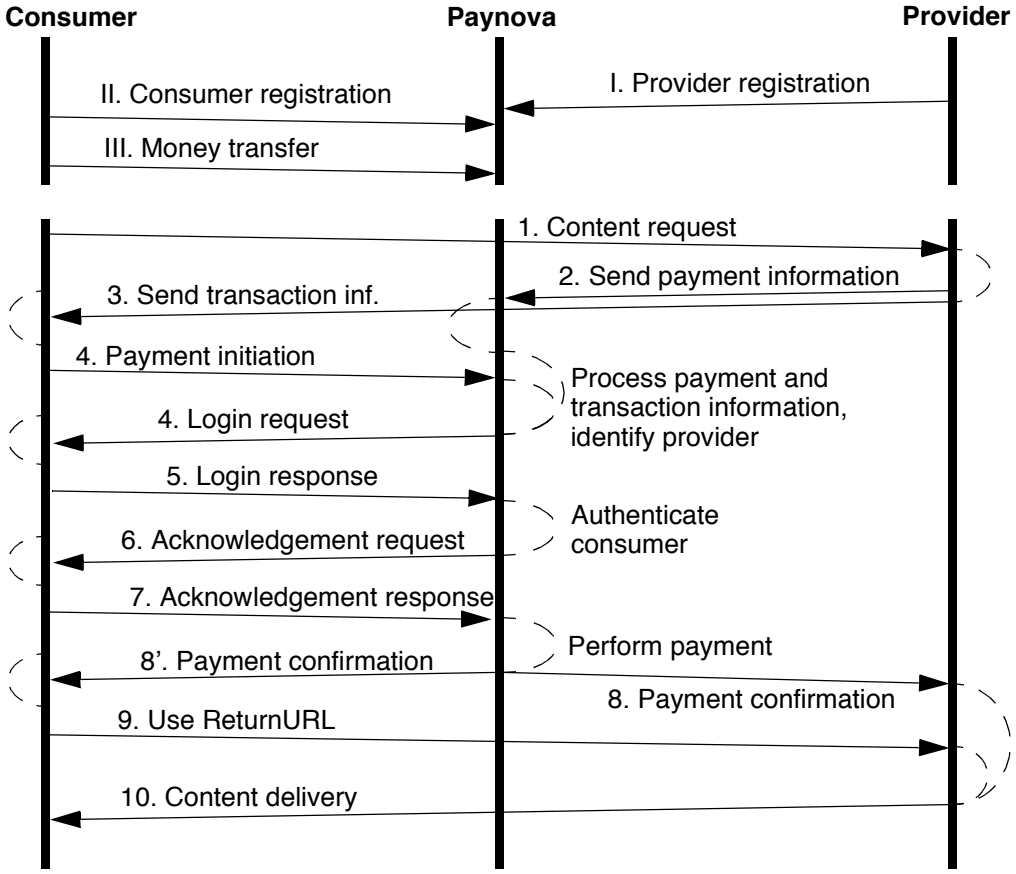


Figure 1.5 *Paynova time sequence diagram*

Table 1.5 *Parameters of Paynova interactions*

Sequence	HTTP message	Parameters (as named by Paynova)
1. Content request	HTTP REQ	Content ID (e.g, URL)
2. Send payment inf.	HTTPS RESP	SessionKey, MerchID, OrderDescr, Amount, Currency, ReturnURL
3. Send transaction inf.	HTTPS RESP	SessionKey
4. Payment initiate	HTTPS REQ	-same as sequence 3-
4. Login request	HTTPS RESP	MerchName, MerchAddress, Amount, Currency, OrderDescr
5. Login response	HTTPS REQ	Username, Password
6. Acknowledgement request	HTTPS RESP	?
7. Acknowledgement response	HTTPS REQ	?
8. Payment confirmation	HTTPS REQ	?
8’ . Payment confirmation	HTTPS RESP	ReturnURL (other params?)
9. Use ReturnURL	HTTP REQ	ReturnURL
10. Content delivery	HTTP RESP	Content download page

We note that, this system has not been fully tested and some of the parameters could not be discovered.

A.6 Bitpass payments

The Bitpass time sequence diagram was realized based on:

- Bitpass FAQ and documents for merchants;
- contacting Bitpass support;
- testing Bitpass at www.digitaldroomedia.com;
- inspecting the source of web pages provided by providers and Bitpass;
- capturing network traffic data with Ethereal.

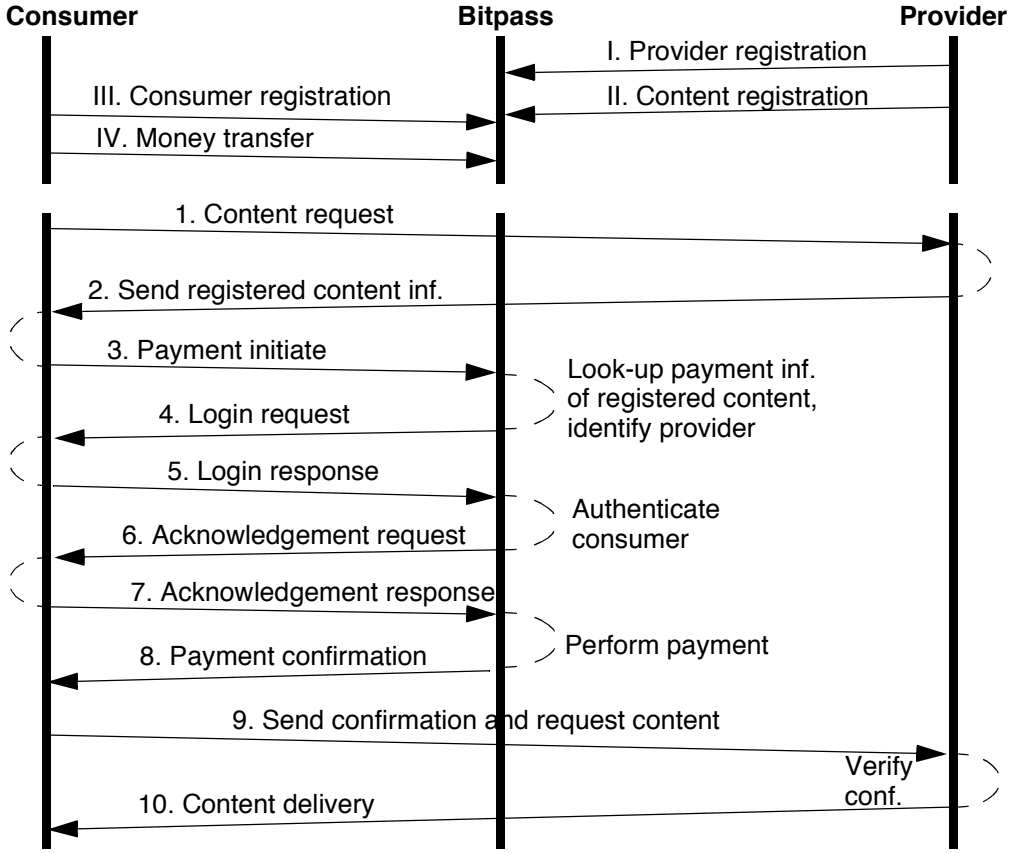


Figure 1.6 *Bitpass time sequence diagram*

Table 1.6 *Parameters of Bitpass interactions*

Sequence	HTTP message	Parameters (as named by Bitpass)
1. Content request	HTTP REQ	Registered Content ID (e.g., URL)
2. Send registered content inf.	HTTP RESP	URL
3. Payment initiate	HTTPS REQ	-same as sequence 2-
4. Login request	HTTPS RESP	ItemName, Amount, Duration, Merch-Site
5. Login response	HTTPS REQ	ItemID, Timestamp, Ticket, Emailaddr, Password
6. Acknowledgement request	HTTPS RESP	?
7. Acknowledgement response	HTTPS REQ	?
8. Payment confirmation	HTTPS REQ	Provider URL, Ticket (other?)
9. Send confirmation and request content	HTTP RESP	Ticket and Content URL
10. Content delivery	HTTP RESP	Content download page

We note that, this system has not been fully tested and some of the parameters could not be discovered.

B. ISDL Notations

Table B.1 *ISDL notations*

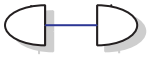
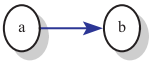
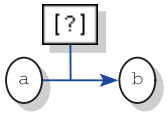
ISDL Notation	Description	Example
	Action: An action models the *successful completion* of some unit of activity that is performed by a single entity.	Example of unit of activity is sending a data packet.
	Interaction: An interaction models the *successful completion* of some unit of activity that is performed by two or more entities in cooperation.	Example of an interaction is the completion of a sales transaction by a seller and a buyer.
	Start **a**: action **a** can always occur, i.e., action **a** can occur from the beginning of the behaviour execution	
	Enabling relation: action **a** must have occurred before action **b**, i.e., action **b** can only occur after action **a** has occurred.	
	Uncertainty attribute: defines whether an action must or may occur when a certain alternative causality condition is satisfied.	In this case: action **b** may occur after **a**. An exclamation mark instead of the question mark would mean that action **b** must occur after **a**.

Table B.1 *ISDL notations (Continued)*

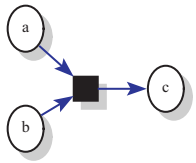
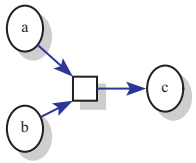
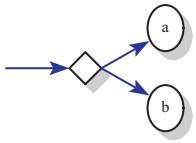
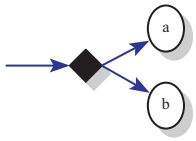
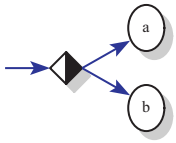
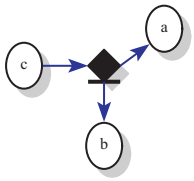
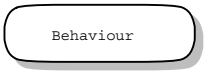
ISDL Notation	Description	Example
	Conjunction: actions **a** and **b** must have occurred before action **c** is allows to occur, i.e., action **c** can only occur after actions **a** and **b** have occurred.	
	Disjunction: the occurrence of **c** either depends on the occurrence of **a** and is then independent of **b**, or the occurrence of **c** depends on the occurrence of **b** and is then independent of **a**.	
	Choice relation: either action **a** or **b** will occur, i.e., both actions cannot occur. Conditions should be defined to determine which action should occur.	
	Concurrency relation: actions **a** and **b** can occur independently from each other.	
	Interleaving relation: action **a** occurs first and in between action **b** also occurs, or action **b** occurs first and in between action **a** also occurs.	

Table B.1 *ISDL notations (Continued)*

ISDL Notation	Description	Example
	Disabling relation, i.e., action **a** disables action **b**: action **a** is allowed to occur after action **c**, provided that action **b** has not occurred before nor simultaneously with action **a**. If action **b** occurs after action **c**, then action **a** cannot occur anymore.	
	Behaviour type: models what a system or a system part does (its functionality).	
	Exit point: an exit point of a behaviour defines a causality condition that can be used to enable actions of other behaviours.	
	Entry point: an entry point of a behaviour allows actions in this behaviour to be enabled by conditions defined in other behaviours.	
	Interaction contribution: an action can be decomposed into an interaction, such that the interaction contributions are assigned to distinct behaviours.	

More information about ISDL is available here: http://isdl.ctit.utwente.nl

Abbreviations

AAA	-	authentication, authorization and accounting
AAAarch	-	authentication, authorization and accounting architecture
ATM (1)	-	asynchronous transfer mode
ATM (2)	-	automatic teller machine
C-UPS	-	consumer-side uniform payment system
CDR	-	call detail record
DCS	-	data collecting and storing
DES	-	data encryption standard
DiffServ	-	differentiated services
DTS	-	data transfer system
EBPP	-	electronic bill presentment and payment
EC	-	European Commission
ECB	-	European Central Bank
ECML	-	electronic commerce modeling language
EFTA	-	Electronic Funds Transfer Act
EPC	-	European Parliament and Council
FIPPA	-	Federal Internet Privacy Protection Act
GSM	-	Global System for Mobile Communications
HCons	-	consumer using the hybrid payment system
HProv	-	provider using the hybrid payment system
HPG	-	hybrid payment gateway
HPS	-	hybrid payment system
HSAP	-	service access point of the hybrid payment system
HTTP	-	hypertext transfer protocol
HTTPS	-	secure hypertext transfer protocol
IETF	-	Internet Engineering Task Force
ID	-	identifier
IFX	-	Interactive Financial eXchange Forum
IntServ	-	integrated services
IOTP	-	Internet open trading protocol

IPDR	-	IP detail record
IPFIX	-	IP flow information export
IPTS	-	Institute of Prospective Technological Studies
IRTF	-	Internet Research Task Force
ISDL	-	interaction system design language
ISP	-	Internet service provider
MD5	-	message digest 5
MPTP	-	micropayment transfer protocol
P-UPS	-	provider-side uniform payment system
P2P	-	person-to-person
PBA	-	provider based accounting
PDU	-	protocol data unit
PE	-	protocol entity
PG	-	payment gateway
PGO	-	payment gateway operator
PGR	-	payment gateway register
PKI	-	public key infrastructure
PSP	-	payment service provider
PSO	-	payment system operator
RADIUS	-	remote authentication dial-in user service
RFC	-	request for comment
RG	-	research group
RSA	-	Rivest Shamir Adelman encryption technology
RTFM	-	real-time traffic flow measurements
SAP	-	service access point
SET	-	secure electronic transactions
SBA	-	server based accounting
SHA	-	secure hash algorithm
SP	-	service primitive
SSL	-	secure socket layer
TLS	-	transport layer security
UPS	-	uniform payment system
URL	-	uniform resource locator
USAP	-	service access point of the uniform payment system
UMSA	-	Uniform Money Service Act
W3C	-	World Wide Web Consortium
WG	-	work group

Samenvatting (Dutch)

Marktonderzoekers verwachten in de komende jaren een groei van de markt voor laag geprijsde online content-gebaseerde dienstverlening zoals muziek en video's. De verwachting is dat de consumenten deze diensten zullen gaan betalen op basis van de mate van gebruik ("pay-per-use"), in plaats van op basis van abonnementen. Dat betekent dat micro betaalsystemen (micro payment systems) voor deze dienstverlening steeds belangrijker zullen worden. Op dit moment is er een groot aantal micro betaalsystemen op de markt. Daardoor zijn online klanten en winkeliers gedwongen om tegelijkertijd verschillende betaalsystemen te gebruiken. Dat betekent voor de klanten dat ze telkens andere systemen moeten vertrouwen en leren gebruiken en even zo vele wachtwoorden of pincodes moeten onthouden. Een winkelier die zo veel mogelijk klanten wil bedienen moet verschillende betaalsystemen aanbieden. De winkelier heeft daardoor te maken met onderhoud en beheer van even zo vele betaalsystemen en contractuele relaties. Voor de online winkelier en online klant leidt dit tot ongemak en extra kosten. Een eenvoudige oplossing voor dit probleem is de introductie van één wereldwijd micro betaalsysteem dat iedereen gebruikt. Dit lijkt echter geen realistisch scenario. Een andere mogelijkheid is het ontwerpen en introduceren van een nieuwe standaard voor micro betalingen. In dit geval zou het ontwerpproces jarenlang duren en het is onzeker of het succesvol wordt. Dit proefschrift doet verslag van onderzoek in een andere richting. We stellen voor gebruik te maken van een zogenaamde hybride betaalsysteem (hybrid payment system).

Het hybride betaalsysteem maakt gebruik van een zogenaamde Payment Gateway, een component die verschillende betaalsystemen met elkaar koppelt. Het idee is dat elke klant of winkelier gebruik maakt van slechts één betaalsysteem. De Payment Gateway koppelt het betaalsysteem van de klant aan het betaalsysteem van de winkelier en zorgt ervoor dat de klant de winkelier via dit hybride betaalsysteem kan betalen. De Payment Gateway kan alleen succesvol zijn als

het op grote schaal wordt geaccepteerd en gebruikt en het geen afbreuk doet aan de veiligheid en betrouwbaarheid waarmee betalingen worden verricht. In dit proefschrift presenteren we de architectuur voor een hybride micro betaalsysteem dat aan deze eisen voldoet.

Het ontwerp van het hybride betaalsysteem heeft in een aantal stappen plaatsgevonden. Allereerst zijn de eisen waaraan de architectuur moet voldoen afgeleid van de gebruikseisen van klanten en winkeliers, en die bepaald worden door bestaande wet- en regelgeving voor het betaalverkeer. Daarna is de betaaldienst (hybrid payment service) van het hybride betaalsysteem gedefinieerd. Vervolgens is onderzoek gedaan naar de interconnectie methode en is de architectuur voor het hybride betaalsysteem ontworpen. De gekozen methode vereist harmonisatie van de functionaliteit van de bestaande betaalsystemen tot een uniforme betaaldienst. Met deze uniforme betaaldienst leggen we vast over welke functionaliteit micro betaalsystemen moeten beschikken om te kunnen worden gekoppeld. Daarna is het hybride betaalprotocol (hybrid payment protocol) gedefinieerd voor de uitwisseling van de gegevens tussen de bestaande betaalsystemen. Aan de hand van praktijkstudies op basis van bestaande betaalsystemen is de praktische waarde en de logische consistentie van de uniforme betaaldienst en het hybride betaalprotocol gevalideerd. Het onderzoek is afgesloten met een evaluatiefase waarin is onderzocht in welke mate de architectuur van het hybride betaalsysteem voldoet aan de opgestelde eisen.

De hybride betaaldienst die we in dit proefschrift presenteren bepaalt hoe klanten en winkeliers het hybride betaalsysteem zullen gebruiken en ervaren.

De uniforme betaaldienst is gebaseerd op een onderzoek van content-gebaseerde diensten en de wijze waarop deze kunnen worden afgerekend. Dit heeft geleid tot het identificeren van een aantal kenmerken aan de hand waarvan micro betaalsystemen voor deze diensten kunnen worden beschouwd. We hebben de structuur en functionaliteit van bestaande betaalsystemen geanalyseerd, en in kaart gebracht welke bedrijfsfuncties de systemen vervullen. We hebben aan de hand van deze inzichten een overzicht van bestaande systemen in kaart gebracht. De uniforme betaaldienst is een harmonisatie van de functionaliteit van de bestaande betaalsystemen aan de hand van de opgestelde kenmerken. Het doel is om de functies van de uniforme betaaldienst te realise-

ren met de functies van een bestaand betaalsysteem zonder de bestaande betaalsystemen zelf aan te passen. We breiden het specifieke bestaande betaalsysteem uit tot een uniform betaalsysteem, waarbij de uniforme betaaldienst zo ontworpen is dat de transformatieregels eenvoudig zijn en de informatieopslag beperkt kan blijven. De meerderheid van de bestaande betaalsystemen kan zo worden gekoppeld. Aan de hand van twee praktijkvoorbeelden laten we zien hoe betaalsystemen zonder aanpassingen met elkaar kunnen worden gekoppeld.

Het hybride betaalprotocol regelt de uitwisseling van informatie tussen de geüniformeerde betaalsystemen. In het ontwerp is nadrukkelijk rekening gehouden met de schaalbaarheid, efficiëntie, veiligheid en betrouwbaarheid van het protocol. Deze criteria zijn belangrijk omdat de Payment Gateway grote transactievolumes moet kunnen verwerken en betalingen veilig en snel moeten kunnen worden verwerkt om content-gebaseerde diensten direct te kunnen verlenen. De veiligheid en betrouwbaarheid van de gebruikte betaalsystemen wordt niet aangetast als ze deel gaan uitmaken van een hybride betaalsysteem. Voor de uitwisseling van gegevens tussen de betaalsystemen kan van gangbare beveiligingstechnieken gebruik gemaakt worden.

Het is op dit moment (augustus 2005) moeilijk te voorspellen hoe micro betaalsystemen zich verder zullen ontwikkelen en of ze een zelfde succes zullen hebben als credit card systemen. We verwachten echter dat, als gevolg van de competitie tussen de verschillende systemen, slechts een beperkt aantal wereldwijd opererende systemen zullen overleven.

Publications by the author

1. Párhonyi, R., Nieuwenhuis, L.J.M., Pras, A.,
The fall and rise of micropayment systems,
In the "E-money, e-payments and m-payments handbook",
Lammer, T. (ed.), Springer, November-December 2005

2. Párhonyi, R., Nieuwenhuis, L.J.M., Pras, A.,
Second generation micropayment systems: lessons learned,
In the Proceedings of The Fifth IFIP Conference on e-Commerce, e-Business and e-Government (I3E 2005), Poznan, Poland, October 2005

3. Párhonyi, R., Quartel, D., Pras, A., Nieuwenhuis L.J.M.
An interconnection architecture for micropayment systems,
In the Proceedings of The Seventh International Conference on Electronic Commerce (ICEC 2005), Xi'an, China, August 2005
ISBN 1-59593-112-0 (abstract), ISBN 1-59593-113-9 (full paper)

4. Párhonyi, R., Pras, A., Quartel D.,
Collaborative micropayment systems,
In the Proceedings of the XIX. World Telecommunications Congress (WTC 2004), Seoul, Korea, September 2004
ISBN 89 950043-1-2 93560 (abstract), ISBN 89-950043-2-0 98560 (full paper)

5. Párhonyi, R., Quartel, D., Pras A.,
A Provider Based Accounting Architecture,
Published in the Proceedings of the IEEE Workshop on IP Operations and Management (IPOM 2002), Dallas, Texas, October 2002
ISBN 0-7803-7658-7

6. Pras, A., van Beijnum, B.J., Sprenkels, R.A.M., Párhonyi, R.,
Internet Accounting,
In the IEEE Communications Magazine, Vol. 39, No. 5, May 2001

7. Párhonyi, R., van Beijnum, B.J.,
Domain based metering,
In the Proceedings of the 6th Open European Summer School (EUNICE 2000), Enschede, The Netherlands, September 2000
ISBN 90-3651-4983

8. Sprenkels, R.A.M., Párhonyi, R., Pras, A., van Beijnum, B.J., de Goede, B.L.,
An Architecture for Reverse Charging in the Internet,
In the Proceedings of IEEE Workshop on IP-oriented Operations and Management (IPOM 2000), Cracow, Poland, September 2000
ISBN 83 8830 900 5

About the author

Róbert Párhonyi holds an M.Sc. degree in Computer Science from the "Babes-Bolyai" University of Cluj-Napoca, Romania.

From August 1998 to October 1999 he worked as a software engineer and database administrator at OTS Services S.R.L. in Cluj-Napoca.

Since November 1999 he is working as a Ph.D. student in the Network Management and later in the Architecture and Services of Networked Applications group within the Electrical Engineering, Mathematics and Computer Science Faculty of the University of Twente, Enschede, The Netherlands. His worked involved among others network management, design of distributed systems, Internet accounting, electronic payment systems and micropayment systems.