Electronic Commerce Research and Applications, (c) Elsevier, vol. 6, no. 1, March 2007, Elsevier, pp. 91-101.

NetPay: An off-line, decentralized micro-payment system for thin-client applications

Xiaoling Dai[1] and John Grundy[2]

Department of Mathematics and Computing Science
The University of the South Pacific, Laucala Campus, Suva, Fiji[1]
dai_s@usp.ac.fj

Department of Computer Science and Department of Electrical and Electronic Engineering
University of Auckland, Private Bag 92019, Auckland, New Zealand[2]
john-g@cs.auckland.ac.nz

Abstract

Micro-payment systems have become popular in recent times as the desire to support low-value, high-volume transactions of text, music, clip-art, video and other media has increased. We describe NetPay, a micro-payment system characterized by de-centralized, off-line processing, customer anonymity and relatively high performance and security using one-way hashing functions for encryption. We describe the motivation for NetPay and its basic protocol, describe a software architecture and two NetPay prototypes we have developed, and report the results of several evaluations of these prototypes.

Keywords: Electronic commerce; Micro-payment; software architecture; electronic wallet

1. Introduction

Current macro-payment systems used by most E-commerce sites are not suitable for high-volume, low-cost transactions, such as charging on a per-page basis for web site browsing. These macro-payment payment technologies suffer from use of heavy-weight encryption technologies and reliance on always on-line and slow-response authorization servers. Micro-payment systems offer an alternative strategy of pay-as-you-go charging, even for very low cost, very high-volume charging. There are a variety of micro-payment systems, such as Payfair [25], Millicent [19], Mpay [11], e-coupons [22] and PayWord [24]. Most existing micro-payment technologies proposed or prototyped to date suffer from limitations with communication, security, lack of anonymity or being vendor-specific.

To address these we issues we have developed a new micro-payment protocol called NetPay to address these problems [4]. NetPay uses “electronic coins” (E-coins) encoded as a “payword chain” of elements encrypted by fast one-way hash functions. The NetPay protocol shifts the communication traffic bottleneck from a broker and distributes it among the vendors by using transferable E-coin Touchstones and Indexes. Customers are prevented from double spending as the index of the payword chain indicates the balance of the customer’s e-wallet, and the touchstone can be used to verify the payword chain has not been tampered with [4].

In this paper, we give an overview of existing micro-payment approaches and briefly discuss the limitations of these models. We present the NetPay micro-payment model and an architecture we have been developing to realize NetPay-based e-commerce systems. We describe a design and prototype implementation of NetPay for deployment with thin-client vendor user interfaces for customers. We describe two example applications which include NetPay micro-payment support, a purpose-built E-newspaper and a pre-existing component-based E-journal web site. Both use NetPay support to sell content on a per-page usage basis. Our protocol is compared with previous micro-payment protocols and we describe three kinds of evaluations we have conducted on our prototypes. We conclude with an outline of our further plans for research in this area.

2. Motivation

Consider a customer browsing an electronic journal site. To access content the customer typically logs in, identifying themselves, searches for articles of interest, and browses and/or downloads the articles to print or read off-line. In a similar manner, customers browsing electronic news provider sites will typically browse headline pages and select articles of interest to view. Many journal and newspaper sites once provided such content for free, or relied on only revenue from on-line, paid advertisements. However, due to the need to recover costs for providing such services, more and more content providers are switching from once free content or services to a paid subscription model or a "pay-per-click" model [19] [23]. Some existing e-newspapers and e-journals provide free content with embedded advertisements for revenue, while others require subscription to typically all of the newspaper content. Some even provide a printed hard copy in addition to the electronic one. Other forms of emerging on-line content provision include purchase of music and video clips, clip art, stock market and other financial data, and so on [12] [21]. For example, on-line music can be downloaded as a single at a time from an on-line music site by paying small amounts of money per single. There is also a multitude of game sites [2] using a small fee-per-play charging model, and various clip-media services where customers can purchase graphics, audio, and video online [17].

Many "free" content sites currently use intrusive embedded advertising for revenue which are often annoying to customers and the revenue for the vendor very difficult to predict. Alternatively where a subscription model is used, the large up front cost can prove a great deterrent to potential customers. It can also be inefficient if the customer actually wants to use a small portion of the information or services for which they have paid. Payment on a per-click basis using traditional macro-payment (e.g. credit card or digital cash) schemes is infeasible due to cost and performance overheads of using slow response authorization servers. An alternative is to use micro-payment systems. In the above scenarios a customer could find and download an article, song or clip-art and only pay a small amount of money e.g. 1c, 2c, 10c or 20c on a pay-per-use basis. Key requirements for such micro-payment systems are generally agreed to be [9] [11] [13] [24]:

- Ease of use for customers, ideally requiring nothing but point-and-click to purchase
- Security of the electronic coins ("e-coins") from both fraud and double-spending by customers
- Ideally anonymous like traditional cash – the vendor has no idea who the customer is
- Vendor-transferable e-coins allowing customers to buy coins from a broker and spend at many different e-commerce sites
- Off-line processing of payments i.e. no on-line bank authorization server needed by vendor or client during payment processing, and highly scalable architecture to support very large numbers of clients concurrently using a vendor site with low-impact on vendor site efficiency

The area of micro-payment on the Internet has attracted much research over the past 10 years. *Millicent,* a micro-payment system by Digital Equipment Corp [19] uses no public-key cryptography and is optimized for repeated micro-payments to the same vendor. Its distributed approach allows a payment to be validated, and double spending prevented without the overhead of contacting the broker on-line during purchase. Key drawbacks are that the broker must be on-line whenever the customer wishes to interact with the new vendor; the customer must nearly always be able to connect to the broker in order to be sure of the ability to make payments; and the vendor scrip is vendor-specific and has no value to another vendor. The *Mpay* micro-payment system was proposed by IBM [11] and is similar to billing mechanisms of third party value-added services of phone networks. Mpay is based on a notational model and has off-line capability in its daily certificate. Mpay only uses one or no public key operation per purchase, so the transaction cost is low. The major shortcoming of the system is that the customer can pay nothing to the issuer who still needs to pay the bank after purchasing goods. Furthermore, the protocol does not support anonymity for customers due to Mpay's after-the-fact policing requirements.

Several micro-payment systems have been developed that are based on a Payword-based micro-payment protocol. These systems can be classified as credit-based and debit-based. Payword [24] is an off-line credit-based system. The customer only needs to contact the broker at the beginning of each certificate lifetime in order to obtain a new-signed certificate. The system aims to minimize the number of public key operations required per payment using hash operations instead whenever possible. It is a credit-based scheme where a user's account is not debited until some time after purchases. This provides more opportunities for fraud since a large number of purchases can be made against an account with insufficient funds. PayFair [26] is a debit-based micro-payment system that employs some parts of the Payword scheme. A payword chain purchased from the broker will be bound to a specific vendor. NMP [15] is a credit-based protocol that improves the fairness for customers from the Payword protocol. The Payword-based micro-payment systems described above share a key disadvantage - they are all vendor specific. The e-coins (paywords) in these systems are only usable at one vendor and have no value for any other vendor. E-coupons [22] is a credit-based, off-line scheme that allows customers to pay for services from a variety of devices, not requiring users to re-register each device. It uses a delegation approach and a SPKI/SDSI multi-seed certificates [3] to ensure security of the payword chain and low-overhead hashing functions rather than public-key encryption. Unfortunately the paywords are vendor-specific and the protocol complex to implement. No real performance benchmarks are available for most of these payword-based schemes to measure their impact on e-commerce systems incorporating them.

Various micro-payment protocols that have been specifically designed for selling information goods on the Internet have been used to support wireless communication device-based payment e.g. paying for items in a shop via a mobile phone [1] [9] [14]. For example, the Atlas Telecom mobile uses an SMS Kambi micro-payment solution, allowing each mobile phone user to pay for both on-line Internet content and face-to-face shop services. SMS Kambi allows users to anonymously and securely pay via their mobile phone by sending text message to a premium number. Users are charged on their mobile phone invoice. This is an on-line micro-payment system requiring an on-line authorization service and slow response time for pay-per-click on-line content. The Small Value Payment (SVP)-based micro-payment scheme [13] [25] that uses tamper-resistant devices was proposed by Park for wireless communications [9]. The scheme aims to avoid customers and vendors executing the three-way challenge-response protocol for every micro-payment. This can be an important issue for mobile communications where the call charges are still large in comparison with Internet-based communications. It also reduces delay and removes the possibility of incomplete payment protocols due to communications failures. Huang and Chen [14] propose a micro-payment system for use on mobile phones using secret-key certificates. This is predominantly an off-line system with customer anonymity. Most of these systems have not been implemented to a degree that meaningful performance analysis of the protocols have been carried out. Many are either credit-based, allowing potential for fraud or double-spending, or require on-line authorization. In addition, for some it is unclear how vendor-neutral the protocols are, potentially requiring separate customer accounts per vendor.

3. The NetPay Protocol

We have developed a new micro-payment protocol called NetPay which provides a secure, cheap, widely available, and debit-based protocol for an off-line micro-payment system [4]. The NetPay protocol is based on the PayWord protocol [24]. PayWord allows a customer to generate a payword chain and spend the paywords at a specific vendor, the payword chain being vendor and customer specific. The payword chain in NetPay protocol is generated by the broker for every customer who spends paywords from one vendor to another without involving the broker, so the payword chain is not vendor-specific. The paywords can be spent with any vendor. NetPay is a basic offline protocol. NetPay protects the customer's anonymity from vendors and prevents customers from double spending and any internal and external adversaries from forging.

NetPay differs from previous payword-based protocols by using touchstones that are signed by the broker and an e-coin index signed by vendors, which are passed from vendor to vendor. The signed touchstone is used by a vendor to verify the electronic currency – paywords, and signed Index is used to prevent double spending from customers and to resolute dispute between vendors. In this section, we describe the transactions and related issues in the NetPay protocol. The system includes a customer (C), vendor (V), and broker (B). We assume that the broker is honest and is trusted by both the customers and the vendors. The customers and the vendors may be dishonest. The vendors and the customers open accounts and deposit funds with the broker. The payment only involves C and V, and B is responsible for the registration of customers and for crediting the vendor's account and debiting the customer's account. Figure 1 shows the NetPay payment model.

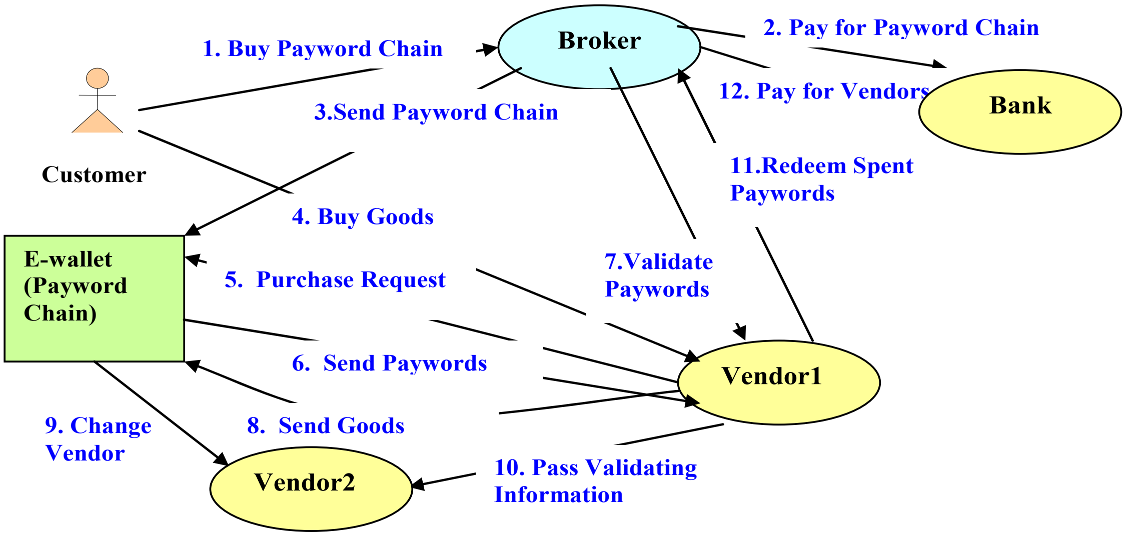


Figure 1: NetPay basic interaction between the parties

Initially a customer accesses the broker's web site to register and buy a number of e-coins from the broker (1) using a single macro-payment (1). The broker sends an "e-wallet" that includes the e-coin chain to the customer (3). When the customer wishes to purchase information goods from Vendor1 site (4), the Vendor1 sends a purchase request to the customer's e-wallet (5) and the e-wallet sends e-coins to the vendor1 (6). Then Vendor1 gets validating information from the broker and verifies the e-coins (7). If the payment is valid, the information goods is sent to the customer (8). The customer may purchase other information goods, their coins being debited. If coins run out, the customer is directed to the broker's site to buy more. When the customer changes to Vendor2 (9), Vendor2 first requests the current e-coin validating information from the Vendor1. Vendor2 contacts Vendor1 to get the e-validating information and then debits e-coins for further information goods (10). At the end of each day, the vendors send all the spent e-coins to the broker redeeming them (11) for real money (12). A variation to this model is to have each customer's e-wallet held on the server-side and passed from vendor to vendor, reducing communication overhead to the customer client PC and allowing the customer to use the e-wallet from different machines. However this approach requires the customer to log into each vendor site initially.

4. The NetPay Architecture

We have developed a software architecture for NetPay-based micro-payment systems for multi-tier thin-client web applications as shown in Figure 2 [5]. The Broker web server provides a point of access for customers to register and buy e-coins. The broker application provides a set of CORBA interfaces vendor applications communicate with it to request touchstones and redeem e-coins. This server also communicates with one or more bank servers to authorize macro-payments. The Broker provides a database holding all customer and vendor account, generated and redeemed coins information.

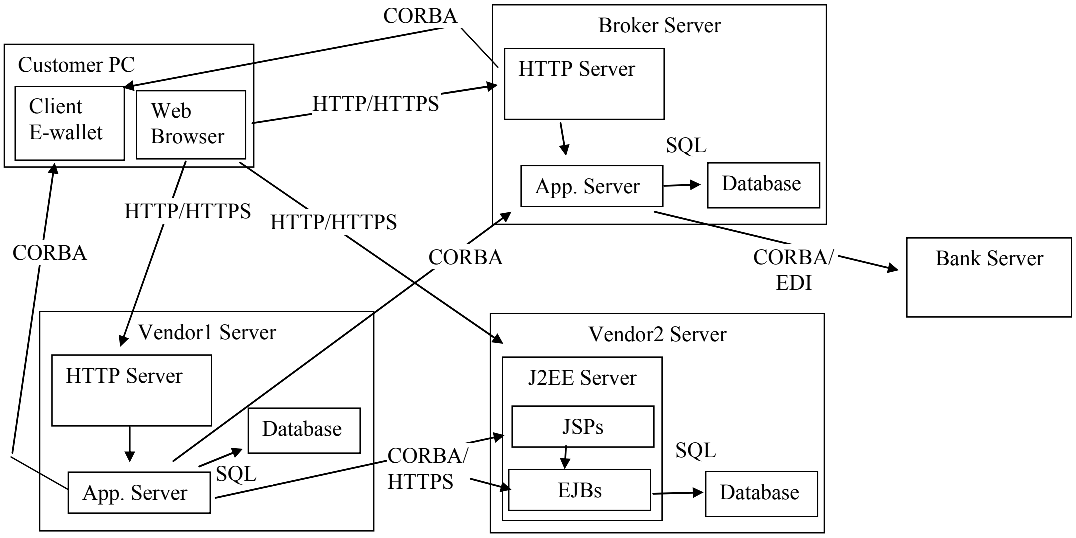


Figure 2: NetPay Architecture.

The customer runs a web browser that accesses the broker and vendor servers. The Vendor web server pages provide content that needs to be paid by customers. Vendors may use quite different architectures. For example, in Figure 2, Vendor #1 uses a web server, custom application server and relational database. Vendor #2 uses a J2EE-based architecture with J2EE server providing JSPs and EJBs, along with a relational database to hold vendor data. The vendor #1 CORBA and the vendor #2 EJB application servers access the broker application server via CORBA to obtain touchstone and to redeem spent e-coins. They communicate with other vendor application servers which may use a CORBA interface like vendor #1 or a J2EE infrastructure like vendor #2 to pass on e-coin indexes and touchstones.

5. NetPay Design and Prototype Implementations

We have developed three kinds of e-wallets for NetPay systems. A server-side e-wallet is held on the vendor server the customer is currently buying content from. Customers can buy articles using the server-side e-wallet and the e-coin debiting time is fast on the server-side e-wallet system. A client-side e-wallet is a Java application to store e-coin information for debit by vendor servers and hosted on the customer PC. Potentially this e-wallet could be hosted on handheld devices like PDAs and mobile phones [14] [22]. Customers can buy article content using the client-side e-wallet at different newspaper sites without the need to log in. However the e-coin debiting time is slower for a client-side e-wallet than the server-side e-wallet due to communication from the vendor to the customer PC's e-wallet application to debit e-coins. A client-side cookie-based e-wallet is stored in a temporary cookie e-wallet for debiting instead of the e-wallet database. This reduces the need for the VAS to communicate with client PC-based e-wallet, caches the e-coins in HTTP request, which holds cookies.

To date we have used a thin-client technology, HTML browsers, to implement our customer clients. We have implemented a broker and several vendor prototypes with "hard-coded" NetPay support for both server-side and client-side NetPay e-wallet systems in order to carry out evaluation of our NetPay protocol. However the major disadvantages for a vendor with hard-coded NetPay support includes the difficulty and time consuming nature of adding NetPay support to existing applications and a lower reusability level for the NetPay implementations.

To overcome these disadvantages, we developed a second set of reusable component-based NetPay vendor facilities using J2EE software components [7]. Figure 3 shows a high-level view of how these various components interact in an E-Journal example system. The E-Journal example system has a number of customer web browser clients used by customers to access the journal site and read article contents. Another web client is used by staff to manage the redemption of spent E-coins with the NetPay broker server. The vendor J2EE server has a number of web pages e.g. JSPs or Servlets and EJBs providing an implementation of the E-journal web system. We add to this a number of NetPay components: EJBs to provide E-wallet management which includes: (1) tracking spending of E-coins by customers; (2) E-coin exchanges with the client-side E-wallet application or server-side E-wallet management; and (3) touchstone exchanges with the NetPay broker or other vendors.

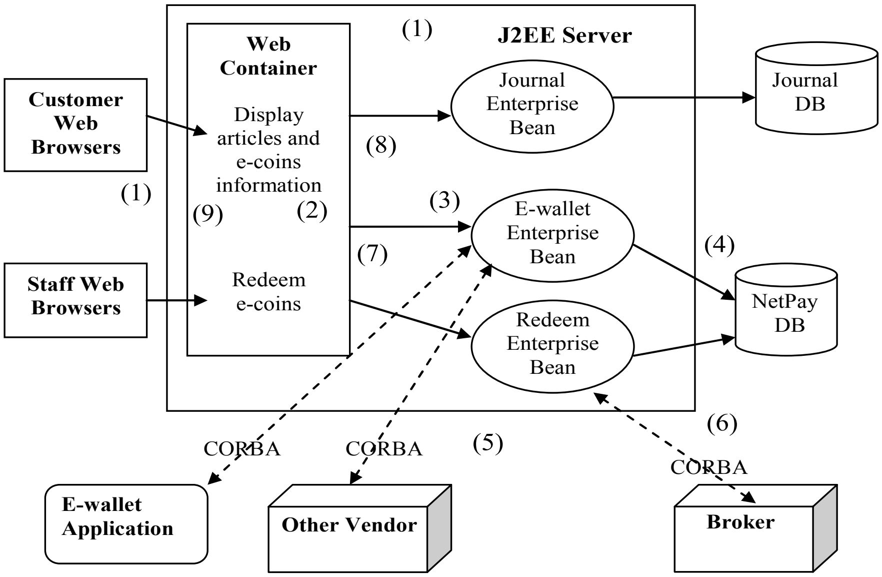


Figure 3: E-journal system with NetPay components

After a customer finds a desired article, he/she clicks the title of the article (1). The web browser requests the article content from the appropriate JSP (2), and this JSP requests payment for the content of the article from the NetPay E-wallet (3). The e-wallet EJB contacts debits the customer's e-coins to pay for the article (4). If there is no e-wallet existing, the e-wallet session bean contacts with either broker or previous vendor to get them (5 or 6). If insufficient coins are available, the customer is directed to the broker site to buy more (7). Otherwise, the journal article content is displayed to the customer (8 and 9).

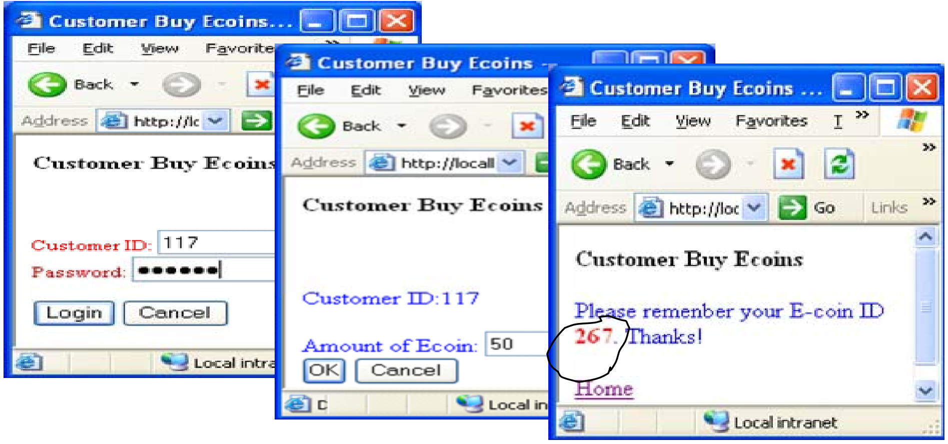
We also provide redemption support for the vendor to communicate with the NetPay broker and redeem customer-spent E-coins for real money. The Ewallet and Redeem EJB components are plugged into the existing E-journal system by deploying them into the E-journal system's J2EE server. The Ewallet component is used to obtain an e-wallet from the broker or another vendor, make payments by using the client-side or server-side E-wallet managed e-coins, and generates payment data. The Redeem component is responsible for selecting payments and sending these to the NetPay broker. EJB deploytool provides an interface to define relationships between enterprise beans. This makes it easier to plug-and-play components. There is no relationship among existing journal component and NetPay components.

6. Prototype Example Usage

In this section we briefly illustrate how a NetPay-enabled micro-payment system works in practice by using two brief example applications, an E-newspaper with hard-coded NetPay support and an E-journal extended to provide micro-payment via plugged-in NetPay software components.

6.1 E-newspaper Example System

We developed a NetPay broker and two E-newspaper vendor systems along with NetPay client-side and server-side E-wallet prototypes. We used a Java application to implement the broker with a CORBA interface for vendors to communicate with. The vendors were implemented using Java Server Pages, which used CORBA to communicate with the broker. The client-side e-wallet was a Java application that communicated with the vendors via TCP/IP sockets. In the server-side e-wallet broker system, when needing to buy some e-coins, the customer logins to the system and enters amount of the e-coins. The HTML interface is used by customer to purchase e-coins as shown in Figure 4. The broker system debits the customer's supplied credit card to pay for the coins by communicating with macro-payment system and then generates e-coins, which are stored in the database. The customer needs to remember the e-coinID e.g. 267 for accessing a vendor site.



Customer Buy Ecoins

Customer ID: 117
Password: ●●●●●●

Login Cancel

Customer Buy Ecoins

Customer ID:117

Amount of Ecoin: 50

OK Cancel

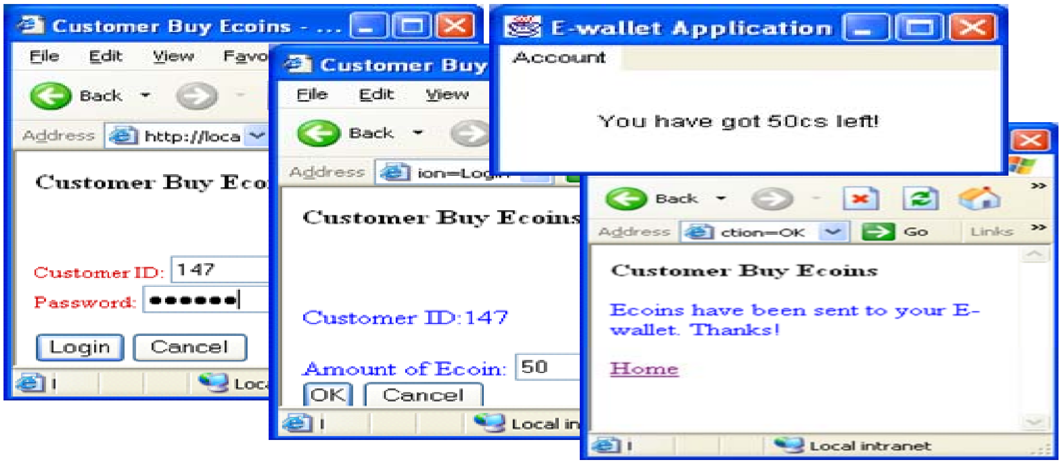
Customer Buy Ecoins

Please remenber your E-coin ID 267. Thanks!

Home

Figure 4: Example of HTML customer buy e-coins with server-side e-wallet broker

The HTML interface for client-side NetPay used by customers to purchase e-coins with the broker is the same as that for server-side NetPay, but the customers do not need to remember e-coin ID. After a customer logins to the system and enters the amount of the e-coins, the e-coins are generated by the system and sent to the customer's e-wallet application on the customer PC as shown in Figure 5. Now the customer can check the e-wallet balance from account menu any time, e.g. there is 50cs in the e-wallet as shown in figure 5a.



Customer Buy Eco

Customer ID: 147

Password: ●●●●●●●

Customer Buy Ecoins

Customer ID:147

Amount of Ecoin: 50

You have got 50cs left!

Customer Buy Ecoins

Ecoins have been sent to your E-wallet. Thanks!

Home

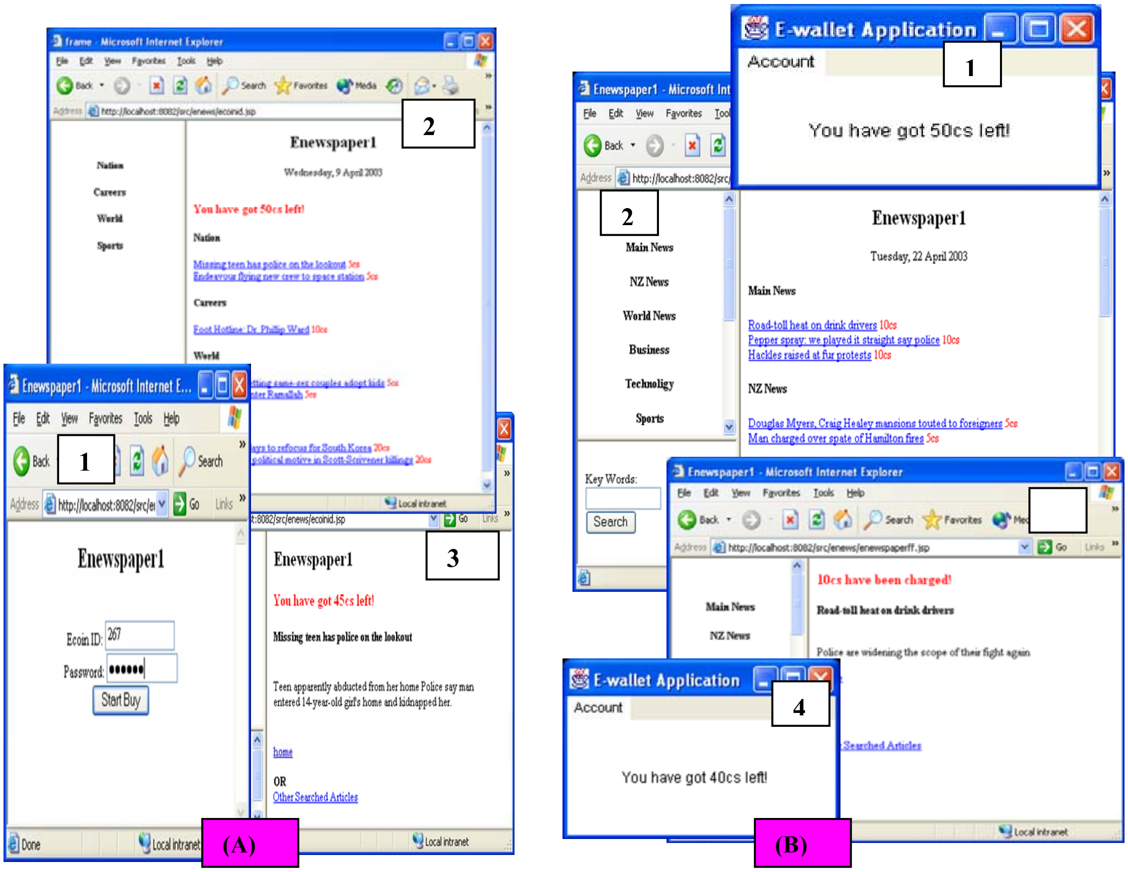
Figure 5: Example of HTML customer buy e-coins with client-side e-wallet broker

Figure 6 shows how a customer buys article contents on an e-newspaper site using the server-side e-wallet and client-side e-wallet NetPay system. In the server-side NetPay vendor, customers are required to login to the newspaper1 site by entering e-coin ID e.g. 267 and password before buy the articles as shown in Figure 6A(1). After the customer logs into the newspaper site, the site obtains the customer's e-wallet from the broker or another vendor in order to debit and verify the e-coins, which are used to pay for the content. The vendor Java Server Pages not only provide searching, browsing and newspaper content for the customer, but also indicate article cost and the amount of e-coins in the e-wallet as shown in Figure 6A(2). After buying an article, the vendor Java Server Pages indicate the amount of e-coins left in the customer's e-wallet as shown in Figure 6A(3). In the client-side NetPay vendor, customers need to run e-wallet application first as shown in Figure 6B(1) and then access Enewspaper1 as shown in Figure 6B(2). When the customer clicks the title of the article, Enewspaper1 system requests e-coins with e-wallet application. If the e-coins valid, the content of the article is displayed on the screen as shown in Figure 6B(3). The customer can check the balance from e-wallet application window as shown in Figure 6B(4). When needing to change to e-newspaper2 site, the customer only needs to go to the site and buy contents. However the customer needs to log in to e-newspaper2 site if buying contents with the server-side e-wallet NetPay system.

6.2 NetPay-enabled E-journal Example System

The main problems of the use a hard-coded approach to adding NetPay facilities to our E-newspaper vendor prototypes is that it is time-consuming, requires modification of existing code structures and possible designs, and the NetPay-implementing facilities are not very reusable. To address these issues we designed and developed several NetPay components using Enterprise Java Beans (EJBs) that can be seamlessly added to existing J2EE-based web applications. NetPay functionality is embodied in Enterprise JavaBean software components and JSP includes or JSP proxy pages, allowing the existing application to be easily micro-payment enabled. Our NetPay EJBs use a

CORBA infrastructure to communicate with customers' client-side e-wallet applications, with the broker server, and with other vendor application servers, whether J2EE-based or not.



frame - Microsoft Internet Explorer

http://localhost:8082/src/enews/ecoinid.jsp

Enewspaper1

Wednesday, 9 April 2003

You have got 50cs left!

Nation

Missing teen has police on the lookout 5cs
Endeavour flying new crew to space station 5cs

Careers

Foot Hotline: Dr. Phillip Ward 10cs

World

Enewspaper1 - Microsoft Internet E...

Enewspaper1

Ecoin ID: 267

Password: ●●●●●●

Start Buy

Enewspaper1

You have got 45cs left!

Missing teen has police on the lookout

Teen apparently abducted from her home Police say man entered 14-year-old girl's home and kidnapped her.

home

OR
Other Searched Articles

E-wallet Application

Account

You have got 50cs left!

Enewspaper1 - Microsoft Int

Enewspaper1

Tuesday, 22 April 2003

Main News

Road-toll heat on drink drivers 10cs
Pepper spray: we played it straight say police 10cs
Hackles raised at fur protests 10cs

NZ News

Douglas Myers, Craig Healey mansions touted to foreigners 5cs
Man charged over spate of Hamilton fires 5cs

Key Words:

Search

Enewspaper1 - Microsoft Internet Explorer

http://localhost:8082/src/enews/enewspaperff.jsp

10cs have been charged!

Road-toll heat on drink drivers

Police are widening the scope of their fight again

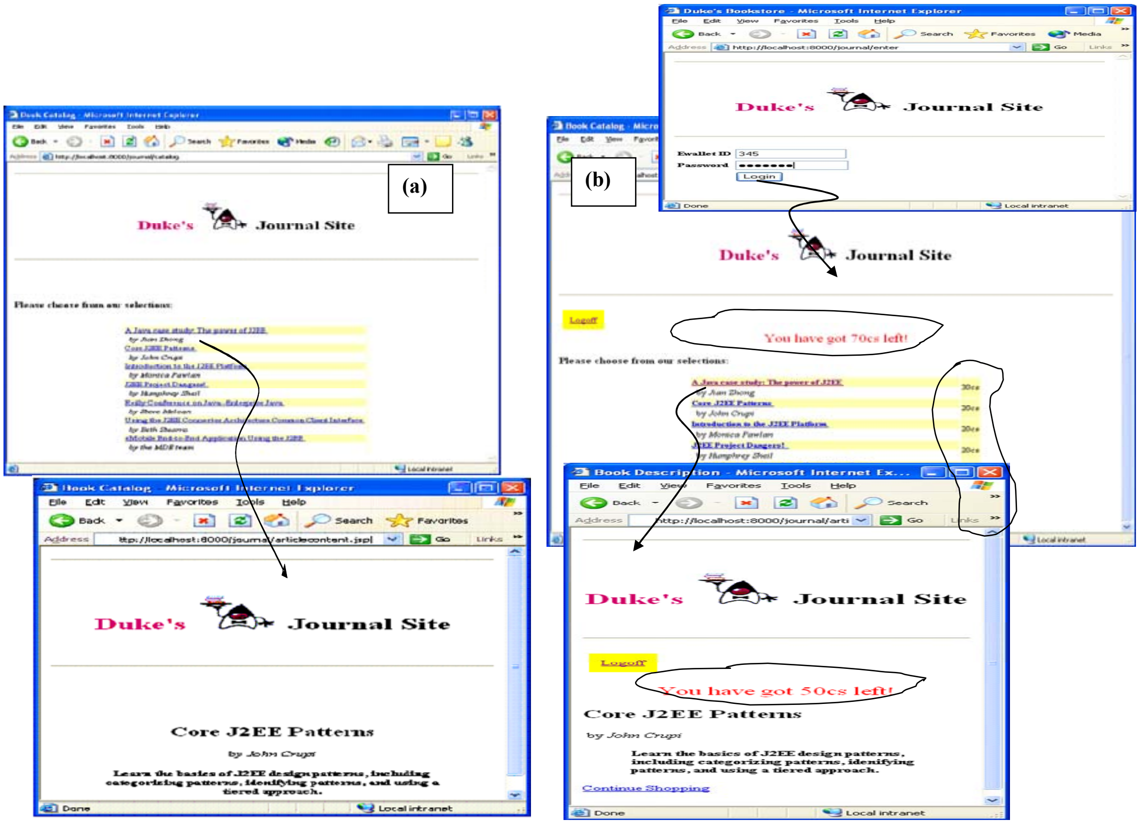
E-wallet Application

Account

You have got 40cs left!

Figure 6: Example of HTML customer buy article content with (A) server-side NetPay enabled and (B) client-side NetPay-enabled E-newspaper1 sites

Figure 7 (a) shows a pre-existing E-journal vendor site implemented using JSPs and EJBs. When using this site the customer browses the site and clicks on an article title to view its contents. Figure 7 (b) shows the click-buy articles interface for the E-journal after extending the system by plugging in our NetPay components. The Login JSP page is added to the system in order to get the e-coin ID from a customer when using server-side e-wallet NetPay. The article and content JSP pages were modified to include our NetPay micro-payment JSP includes so that e-coin credit and article prices are displayed on the screen. Article pricing is stored in the NetPay database and the JSP includes are parameterized with article information (category, title, URL) to look up the appropriate pricing. The scenario of a customer buying article contents with a client-side NetPay-enabled e-journal site is the same as the one above with a customer-buying article with a CORBA-based e-newspaper site.


(a)
(b)
Book Catalog - Microsoft Internet Explorer
http://localhost:8000/journal/catalog
Local intranet
Duke's Bookstore - Microsoft Internet Explorer
http://localhost:8000/journal/enter
Book Description - Microsoft Internet Ex...
http://localhost:8000/journal/arti
ttp://localhost:8000/journal/articlecontent.jsp
Done
Local intranet

Duke's Journal Site

Please choose from our selections:

A Java case study: The power of J2EE
by Jian Zhong
Core J2EE Patterns
by John Crupi
Introduction to the J2EE Platform
by Monica Pawlan
J2EE Project Dangers!
by Humphrey Shal
Kelly Conference on Java, Enterprise Java
by Steve Meloan
Using the J2EE Connector Architecture Common Client Interface
by Beth Stearns
eMobile End-to-End Application Using the J2EE
by the MDE team

Core J2EE Patterns

by John Crupi

Learn the basics of J2EE design patterns, including categorizing patterns, identifying patterns, and using a tiered approach.

Ewallet ID 345
Password ••••••••
Login

Logoff

You have got 70cs left!

Please choose from our selections:

A Java case study: The power of J2EE 20cs
by Jian Zhong
Core J2EE Patterns 20cs
by John Crupi
Introduction to the J2EE Platform 20cs
by Monica Pawlan
J2EE Project Dangers! 20cs
by Humphrey Shal

Logoff

You have got 50cs left!

Core J2EE Patterns

by John Crupi

Learn the basics of J2EE design patterns, including categorizing patterns, idenifying patterns, and using a tiered approach.

Continue Shopping

Figure 7: Example of (a) original E-journal system and (b) NetPay-enabled E-journal system

7. Discussion

In this section, we compare the features of our NetPay protocol with other micro-payment protocols. We also discuss three kinds of evaluations we have carried out on our NetPay prototypes to demonstrate their usability, performance impact on a vendor's e-commerce system, and overall satisfaction of the requirements we outlined in Section 2.

7.1 Micro-payment Systems Comparison

We compare NetPay's characteristics to a number of other well-known micro-payment systems and some more recent micro-payment systems. The comparison criteria we have used below are based on the key requirements identified in Section 2: an easy-to-use micro-payment system; secure electronic coins and no double-spending; anonymity for customers; transferable e-coins between vendors; and robust, low performance impact, off-line micro-payment supported and architecture is scalable for large number of customers. The comparison is for the scenario of a customer (C) reading an on-line newspaper, newspaper vendors (V), and micro-payment brokers (B). Table 1 lists the results of this requirements satisfaction comparison for our NetPay protocol with these other payment systems.

The NetPay system requires no log-in if the client-side e-wallet is used for micro-payment based purchases, even when moving between vendors. SVP [25] allows handset-based "e-wallet" payment similarly without login. Security is via one-way MD5-based hashing functions encrypting the payword chain, as in the PayWord protocol on which it is based [24]. However customers are prevented from double spending as the index of the payword chain indicates the balance of the customer's e-wallet**,** and the touchstone can be used to verify the payword chain. E-coupons [22] uses a more complex multi-seed certificate approach for paywords. Since only the broker knows the mapping between the pseudonyms (IDc) and the true identity of a customer, the protocol protects the customer's privacy. In all protocols, the broker knows who the customers are but in NetPay knows nothing about their transactions with the vendors. NetPay allows customers to move transparently from one vendor site to another, with a single e-coin touchstone and index transfer between vendors. NetPay is an off-line protocol with the number of expensive public-key operations required per payment minimized by using fast hash function operations to get the next payword chain coin, in order to minimize the transaction overhead.

Table 1 Comparison of payment methods with NetPay

System/ property	**Millicent**	**Mpay**	**PayWord**	**NMP**	**SVP**	**e-Coupons**	**NetPay**
Ease of use	**Medium,** C nearly always contact B.	**High,** C only needs to click and see what he pays for.	**Medium,** C generates and manages different e-coins for Vs.	**Medium,** C buys e-coins from broker to use for vendor	**Very High,** C uses device to pay for goods via wireless communications	**Very high,** C can delegate spending to different devices and even different users.	**High,** C clicks and gets the content. No login for client-side e-wallet.
Security	**Medium+,** the system prevents double spending by using V-specific scrip.	**Medium,** but the criminal is free to spend money to buy content for a full day.	**Low,** the system is credit-based scheme to provide more opportunity for fraud.	**Medium,** credit-based but on-line check can be used	**Medium+,** double-spending stopped by on-line check	**High,** double-spend prevented by multi-seed certificates for e-coin paywords	**Medium+,** the system prevents double spending by transferring indexes between Vs.
Anonymity	**Medium+,** C's identity is protected from V.	**Low,** C's anonymity is not supported.	**Low,** Vs know what bought and who by	**Medium**+, C not known by V	**Medium+,** C not known by V	**Medium,** C not known by V but broker knows delegations	**Medium+,** C's anonymity is protected from V.
Vendor-specific E-coins	**Yes,** The scrip has no value to other Vs.	**Account-based**	**Yes,** E-coins have no value to another V.	**Yes**, e-coins for one vendor only	**No,** key idea is to allow any V	**Yes**	**No,** E-coins can be spent at any V.
Low-performance impact and robust	**No,** on-line	**Yes**, off-line	**Yes**, off-line and hashing used to encode e-coins	**Mostly,** off-line but on-line check can be used	**Mostly,** on-line check for double-spend	**Yes,** off-line and hashing	**Yes**, off-line and hashing

7.2 Usability Evaluation

We carried out a usability evaluation which surveyed users of the E-newspaper prototype to assess their impressions of the approach in order to determine if NetPay is usable as far as target users were concerned [6]. We compared three versions of E-newspaper vendors: one using a subscription-based macro-payment scheme, one using a client-side e-wallet and one using a server-side e-wallet. We had a dozen people participate in the experiment, half being experienced on-line shoppers using macro-payment supporting e-commerce sites. We split the participants into groups of three, each group using each version of an E-newspaper system in turn. We had the users carry out a set of registration, browsing, purchasing and viewing tasks. We had the groups use the same system on alternate days to carry out further browsing and purchase activities as well as moving between vendors during these tasks. We used pre- and post-experiment surveys with a set of closed and open questions to gauge users' views on the payment support in each prototype e-commerce system. We used similar criteria in the questionnaires as in Section 7.1 above: ease of use; perceptions of security and anonymity; ability to move between vendors and system response time.

In our survey results, ease of use, efficiency, and satisfaction/preference mainly favored the client-side e-wallet NetPay system. However it was found by users that this incurred an extra delay in page display due to communication from the vendor to the customer PC's e-wallet application, which the other systems don't have. Participants stated that the article contents at different newspaper sites ware easy to access without log in and their E-coin balance can be checked any time. The server-side NetPay system allowed users to read articles on different computers, but customers needed to remember e-coin IDs and had to log into the new newspaper site when changing vendor. This was found to be very inconvenient by users. The article content loading was very fast on the subscription-based system, but the users found that it was not at all convenient to change vendor as re-registration and loss of money resulted. The users generally needed to spend more money in order to subscribe to the whole newspaper provided by each vendor site they even heavy browsing and purchase of content with micro-payment. Open question results revealed that client-side NetPay was found to be significantly preferred over a subscription-based system. In addition, server-side NetPay was somewhat more preferred than the subscription-based system for this E-newspaper application domain.

7.3 Performance Impact Evaluation, Security, Reliability

One potential problem with adopting micro-payment protocols is the processing overhead needed to validate customer purchase requests and debit e-wallet content [7] [14]. To identify the overhead on e-commerce systems incorporating macro-payment and NetPay-based micro-payment approaches we designed and carried out a performance impact evaluation. This evaluation assessed the performance of NetPay-enabled prototype e-commerce web sites to determine the overhead of the micro-payment extensions made to the software, particularly in regard to user response time and database access and update overheads [6]. We again deployed three versions of our E-newspaper vendor e-commerce system: a subscription-based, a server-side e-wallet system and client-side e-wallet system. Ten concurrently running clients deployed on a separate host made 1000 requests to purchase content from the vendor servers. The vendor e-commerce server and SQL Server 2000 database were deployed on dual CPU machine running Windows 2000 Server.

Table 2 shows performance results from the first experiment run. The server-side NetPay takes 64ms for e-coin debiting per article and Client-side takes. 934ms total time due to communication overhead back to the client-side PC hosted E-wallet. The time taken to debit E-coins is taken by the client's e-wallet application, not the vendor's application server. The large overhead in the server for the server-side NetPay prototype is due to the database transactions it carries out to record coin updates and debits to redeem to the broker. Note that multi-threading in the server allows the vendor to serve other clients during NetPay debits but the server-side e-wallet incurs a heavy processing overhead.

Table 2 Original Prototype performances

System	Vendor Server Actions	Original Response Delay Time	Original Server NetPay CPU Usage
Subscription-based	Fetch page data and format	16ms	N/A

Server-side NetPay	Debit E-coins;
	Store E-coin debit info;
	Fetch page data & format;
		80ms	64ms

Client-side NetPay	(client side) Debit e-coins Store E-coin debit info; Fetch page data & format	950ms	60ms

To reduce the e-coin debiting time, we created a transaction temporary file recording the data for redeeming instead of the original approach of directly updating the vendor database. Because of the optimized efficiency of such a temporary file, the e-coin debiting time decreases dramatically for the server-side NetPay system. The performance results of our second tests are shown in Table 3. At the end of each day, the system redeems the coins or updates the database, and then deletes the temporary transaction file. From Table 3, Server-side NetPay takes 14ms for e-coin debiting per article after the application of the temporary file. The impact of the NetPay micro-payments on the vendor application server are greatly reduced, but the client-side e-wallet still incurs considerable response time delay due to the additional overhead of the vendor connecting to a customer PC to debit its e-coins.

Table 3 Prototype performances after using a temporary debited coin cache file

System	Response Delay Time	Server NetPay CPU Time Usage
Server-side NetPay	**30ms**	**14ms**
Client-side NetPay	900ms	12ms

7.4 Qualitative Evaluation of NetPay Micro-payment Systems

From the above evaluations and by analyzing the problem domain we can make several conclusions about using NetPay for micro-payments in e-commerce systems. Firstly, a macro-payment approach is going to be more beneficial in general for the customer who reads a large portion of the on-line newspaper or E-journal articles, or downloads sufficient music or video clips, clip art, or other electronic content from a subscribed site to outweigh any micro-payment pay-per-click advantage. However, using a micro-payment approach wins out when the customer reads a small portion of the articles, articles are low-priced, or if the customer reads articles or downloads small amounts of content from multiple vendors, using their e-coins across any of these vendors. The response time delay when debiting e-coins from a client-side e-wallet can also be mitigated by using cookies to cache e-coins, as in our recent work [8]. Security is always going to be less in off-line protocols than on-line, where on-line debiting of customer account information takes place. However, this is mitigated when using high-security one-way hashing functions and similar technologies to prevent fraud and double-spending of e-coins. Customers prefer micro-payment solutions that don’t require passwords, e-coin IDs or other authentication but rather almost invisible click-and-pay purchase of low-cost content.

A key outstanding challenge with micro-payment systems is being able to spend e-coins at a wide range of vendors. This requires buy-in of a multitude of vendors, or at least several well-used vendors, to enable customers to leverage buy-once, spend-(almost) anywhere behaviour from their micro-payment system. Emerging trends of face-to-face payments with portable devices may help address this need for wide vendor take-up if used very frequently for low-cost purchases. Another issue is minimizing overhead on vendor e-commerce servers. As shown in the NetPay performance evaluations, naïve implementation of server-side e-coin recording and debiting for purchases greatly increases vendor e-commerce system load. Vendors traditionally have enjoyed ‘capture’ of customers via subscription systems or advertising revenue for site visits. Potential benefits for vendors of micro-payment systems include revenue on per-use basis rather than hoping customers go to advertisers sites and simplification of payment for customers through seamless debit of e-coins.

8. Future Research

An extension to our client-side E-wallet model we have recently experimented with is to cache E-coins using cookies after the first click-and-buy at a vendor and debit coins held by the browser and passed to the server as a cookie for each page view. This greatly improves performance as the time consuming vendor server to client e-wallet communication is avoided, and also reduces the customers need to supply e-coin information for server-side e-wallet management. We could also allow the vendor to debit multiple coins for multiple pages from the client-side E-wallet at one time, depending on customer preferences, reducing the number of expensive delays. This approach acts as a form of "pre-paying" a number of pages or on-line content downloads at once in advance.

We are designing a portal infrastructure using web services that will allow a NetPay-enabled vendor to act as a purchasing portal to non-NetPay supporting vendors. The NetPay-enabled vendor will redirect page accesses to these vendors and will manage debiting of a customer's e-coins in the process. This approach will allow for dynamic registration of vendors and move the processing of e-coins from vendor servers. It will also allow vendor servers to ignore the micro-payment management and enhancement of their servers to accommodate it. We are investigating approaches to using NetPay for mobile information content micro-payment applications, both with a server-side e-wallet and client-side e-wallet storage by the mobile device. In addition, we are investigating the use of NetPay-style micro-payment e-coins as a form of efficient document digital signature for B2B e-commerce systems.

References

[1] W. Adi, A. Al-Qayedi, A. Zarooni, and A. Mabrouk, Secured multi-identity mobile infrastructure and offline mobile-assisted micro-payment application, 2004 IEEE Wireless Communications and Networking Conference, vol. 5, no. 1, pp. 879 – 882.

[2] E. Brown, 1999. "Micro-payment schemes promise to make the Web profitable – one penny at a time", NewMedia, June 1997, http://newmedia.com/newmedia/97/08/fea/micropayments_small_change.html

[3] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. Rivest. Certificate Chain Discovery in SPKI/SDSI, Journal of Computer Security, vol.9, no. 4, 2001, pp. 285-322.

[4] X. Dai, and B. Lo, NetPay – An Efficient Protocol for Micro-payments on the WWW. Fifth Australian World Wide Web Conference, Australia, 1999.

[5] X. Dai, and J. Grundy, Architecture of a Micro-Payment System for Thin-Client Web Applications. In Proceedings of the 2002 International Conference on Internet Computing, Las Vegas, CSREA Press, 2002, pp444—450.

[6] X. Dai, and J. Grundy, 2003a. Customer Perceptions of a Thin-Client Micro-payment System: Issues and Experiences. Journal of End User Computing, 15(4), (2003) 62—77.

[7] X. Dai, and J. Grundy, 2003b. Architecture for a Component-based, Plug-in Micro-payment System. In Proceedings of APWEB 2003, Lecture Notes in Computer Science, Springer-Verlag, 2003, pp 251—262.

[8] Dai, X. and Grundy, J.C. Three Kinds of E-wallets for a NetPay Micro-payment System, In Proceedings of the 5th International Conference on Web Information Systems Engineering, Brisbane, Australia, November 22-24 2004, LNCS 3306.

[9] D. Park, C. Boyd, and E. Dawson, Micro-payments for wireless communications, 3rd International Conference On Information Security and Cryptology, Lecture Notes in Computer Science 2015, Springer, 2001, pp. 192—205

[10] A. Furche and G. Wrightson, SubScrip – An efficient protocol for pay-per-view payments on the Internet, The 5th Annual International Conference on Computer Communications and Networks, USA. 1996.

[11] A. Herzberg, and H. Yochai, Mini-pay: Charging per Click on the Web, 6th World Wide Web Conference, Santa Clara, USA, April 1997.

[12] A. Herzberg, Safeguarding Digital Library Contents - Charging for Online Content. D-Lib Magazine, January 1998.

[13] M-S. Hwang, I-C. Lin, and L-H. Li, A simple micro-payment scheme, Journal of Systems & Software, vol. 55, no. 3, March 2001, 221—229.

[14] Z. Huang and K. Chen, Electronic Payment in Mobile Environment, Proceedings of the 13th International Workshop on Database and Expert Systems Applications, Aix-en-Provence, France, 2-6 September 2002, Lecture Notes in Computer Science 2453 Springer..

[15] D-Y. Ji, and Y-M. Wang, A micro-payment protocol based on PayWord. Acta Electronica Sinica, vol. 30, no. 2, 2002, pp. 301—303.

[16] Kim, M., Lee, H., Kim, S., Lee, W., Kang, E. Implementation of anonymity-based e-Payment System for M-commerce, International Conference of Communications, Circuits, and Systems, June 29 - July 1, 2002, CHENGDU, China.

[17] Library ClipArt Site, http://www.libraryclipart.com/news.html

[18] Lowen Color Graphics Site: http://www.lowencg.com/products.html, 2003

[19] M. Manasse, The Millicent Protocols for Electronic Commerce. First USENIX Workshop on Electronic Commerce. July New York, USA, 11-12, 1995.

[20] R. McGarvey, Micropayments enable teensy content purchases. Econtent, 24(1) (2001) 18—21.

[21] MP3 Web Site, 2003. http://www.mp3.com

[22] V. Patil and R.K. Shyamasundar, An Efficient, Secure and Delegable Micro-Payment System, Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, Taipei, Taiwan, 28 - 31 March 2004.

[23] A. Posman, "Would You Pay for Google?", 2002. http://www.clickz.com/media/agency_start/article.php/1013901

[24] R. Rivest, and A. Shamir, PayWord and MicroMint: Two Simple Micropayment Schemes. Proceedings of 1996 International Workshop on Security Protocols, Lecture Notes in Computer Science 1189, Springer, pp. 69—87.

[25] J. Stern, and S. Vaudenay, 1997. "SVP: a Flexible Micro-payment Scheme". Financial Crypto '97, Lecture Notes in Computer Science 1294, Springer-Verlag, 1997 pp161—171.

[26] S-M. Yen, PayFair: a prepaid internet ensuring customer fairness micro-payment scheme. IEE Proceedings-E Computers & Digital Techniques, vol.148, no.6, 2001, pp. 207—213.