5th International Conference on Web Information Systems Engineering, Brisbane, Australia, November 22-24 2004, LNCS 3306. 

# Three Kinds of E-wallets for a NetPay Micro-payment System

Xiaoling Dai[1] and John Grundy[2, 3]

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[2] and Department of Electrical and Electronic Engineering[3]
University of Auckland, Private Bag 92019, Auckland, New Zealand
john-g@cs.auckland.ac.nz

**Abstract.** We have developed NetPay, a micro-payment protocol characterized by off-line processing, customer anonymity and relatively high performance and security using one-way hashing functions for encryption. In our NetPay prototypes we have identified three kinds of electronic wallets to store e-coins – a server-side wallet, client-side wallet application, and cookie-based wallet cache. We describe the motivation for NetPay and describe the three kinds of e-wallets and their design. We report on prototype implementations of these wallets and end-user perceptions of their use.

## 1 Introduction

Macro-payment systems used by most E-commerce sites are not suitable for charging per-page for web site browsing. Such systems typically use complex encryption technologies and require communications with an authorisation server to request and confirm payment. 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 number of micro-payment systems [8], [11]. Most existing micro-payment technologies proposed or prototyped to date suffer from problems with communication, security, and lack of anonymity or are vendor-specific. In addition, they usually adopt a single strategy for managing the electronic coinage, not always the one customer requires or desire.

We have developed a new micro-payment protocol called NetPay to address these problems. The NetPay protocol allows customers to buy E-coins, worth very small amounts of money, from a broker and spend these E-coins at various vendor sites to pay for large numbers of discrete information or services of small value each. NetPay shifts the communication traffic bottleneck from a broker and distributes it among the vendors by using transferable E-coin Touchstones and Indexes. We have identified three different strategies for managing these E-coins: a server-side E-wallet

that is exchanged among vendor sites as the customer buys information or services; a client-side E-wallet application that resides on the customer's PC and from which vendors debit coins; and a hybrid, cookie-based E-wallet cache.

In this paper, we give an overview of existing micro-payment models and point out the problems with these models. We then briefly describe the NetPay micro-payment protocol. We then describe the three kinds of e-wallets we have designed and prototyped for the NetPay system. We describe the designs for the two NetPay e-wallets and illustrate their usage. We conclude with an outline of our further plans for research and development in this area.

## 2 Motivation

With the development of Internet businesses, more and more content providers are switching once free content or services to a paid subscription model or pay-per-click model, eliminating the revenue relying on only an advertisement market [10]. Today there are already many newspapers and journals in electronic form. Most of newspapers and journals allow their regular subscribers to read the articles on the net for free while they also get a normal paper copy of them. Such a procedure seems to waste resources since the subscribers can print the articles in which they are interested on net and thus there is no need to read the paper copy. Micro-payment systems could be used to make things different for online contents or services. You could read and download an article and only pay a small amount of money e.g. 5c, 10c or 20c. 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 [9]. 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 are a number of micro-payment systems in various stages of development from proposals in the academic literature to systems in commercial use [8], [5], [6], [11]. Though micro-payment protocols have received a lot of attention from researchers and cryptographers, only one micro-payment system, Millicent [12], exists in general public use in Japan. All existing protocols for micro-payments have their strengths and weaknesses in practical applications [2]. In Millicent [8], the third party must be online whenever the user wishes to interact with a new vendor, i.e., the system places a heavy real-time burden on the third party. In Mpay [7], customers can pay nothing to access services for a full day and also the customer's anonymity is not protected. In PayWord [11], the payword chain is customer and vendor specific, i.e., the system locks customers to some sites that they have the payword chains. Most micro-payment approaches provide customers with E-coin scripts or payword chains, or require customers to log onto a vendor web site to access a stored collection of e-coins. Most do not support inter-vendor spending of E-coins.

Most existing micro-payment approaches use a single mechanism for managing the electronic coinage they use. The majority store these e-coins in (usually) encrypted files on the client PCs. They need a specialized application with which this e-coin database is accessed and updated. This approach often requires installation of client-side "e-wallet" applications to manage the e-coins. Some approaches are susceptible to fraudulent alteration of

the e-coins while others require heavyweight, expensive encryption technologies to decode the e-coins each time they are used. Some on-line micro-payment approaches use a server-side approach where an on-line broker manages the e-coins for each customer and decrements the coins available on each spend. This provides a single point of failure or bottleneck for the micro-payment system as a whole and often removes the anonymity of the customer.

## 3 NetPay Protocol

We have developed a new protocol called NetPay that provides a secure, cheap, widely available, and debit-based protocol for an off-line micro-payment system [1]. 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 resolve disputes between vendors. In this section, we describe the key transactions in the NetPay protocol.

Suppose an e-newspaper site wants to use the NetPay micro-payment system to sell articles on a per-page usage basis. The system involves four parties – a NetPay broker site; e-newspaper vendor sites; customer PCs; and a bank macro-payment system. The customers can be classified as registered customers and unregistered customers. Only registered customers can buy e-coins from a broker's site and click-buy an article with a newspaper site. Both types of customers can search and view article titles on line. Initially a customer accesses the broker's web site to register and acquire a number of e-coins from the broker (bought using a single macro-payment). The broker creates an "e-wallet" that includes the e-coin ID, touchstone, and e-coins for the customer. This e-wallet may reside on the client PC (via a special application) or be passed to vendor servers.

The customer browses the home page of the newspaper web site and finds a desired news article to read. Each article will typically have a small cost e.g. 2-10c, and the customer would typically read a number of these. When wishing to read the details of an article, the customer clicks on the article heading and the vendor system debits the customer's e-coins by e.g. 10c (by taking 1, 2 or more e-coins from their payword chain, depending on the monetary value of each, up to 10c in value).

The newspaper system verifies that the e-coin provided by the customer's e-wallet is valid by use of a "touchstone" obtained once only from the broker. If the payment is valid (coin is verified and sufficient credit remains), the article is displayed on the screen. The customer may browse other articles, their coins being debited (the index of spent coins incremented) each time an article is read. If coins run out, the customer is directed to the broker's site to buy more. The vendor keeps copies of the spent e-coins.

When the customer changes to another online newspaper (or other kind of vendor using the same e-coin broker currency), the new vendor site first requests the current e-coin touchstone information from previous vendor's site. The new vendor contacts the previous vendor to get the e-coin touchstone and "spent coin" index and then debits coins for further news articles.

When the previous vendor system is "down", a backup server in the system sends the e-coin ID, the touchstone, and the index to the broker. The new vendor could also contact the broker to get the e-coin touchstone and the "spent e-coin" index. At the end of each day, the vendors all send the spent e-coins to the broker, redeeming them for real money (done by macro-payment bank transfer from the broker to vendor accounts).

## 4 NetPay E-wallets

We have designed three kinds of e-wallets to manage e-coins in the NetPay system. One is hosted by vendor servers and is passed from vendor to vendor as the customer moves from one site to another. The second is a client-side application resident on the client's PC. The third is a hybrid that caches E-coins in a web browser cookie for debiting as the customer spends at a site.

### 4.1 Server-side e-wallet

Some people prefer to access the Internet from multiple computers (e.g. a business person who often travels around). A Server-side hosted e-wallet is suitable for these people. The server-side e-wallet is stored on the vendor server and is transferred from the broker to each vendor when required.

Fig. 1 shows how a vendor application server debits e-coins from the server-side e-wallet. When a customer clicks title of an article on his/her browser (1), the web server sends the request to the vendor application server (2), which then debits e-coins from the customer's e-wallet (3) paying for the content. Customers can buy articles using the server-side e-wallet anywhere in the world and the e-coin debiting time is very fast on the server-side e-wallet system. However customers are required to remember e-coin IDs and password in order to log into a newspaper site when changing vendor. When a customer moves from one vendor to another, their e-wallet contents must be passed from the previous vendor site to the new one. If the first vendor site becomes unavailable, the customer temporarily does not have access to their e-wallet.

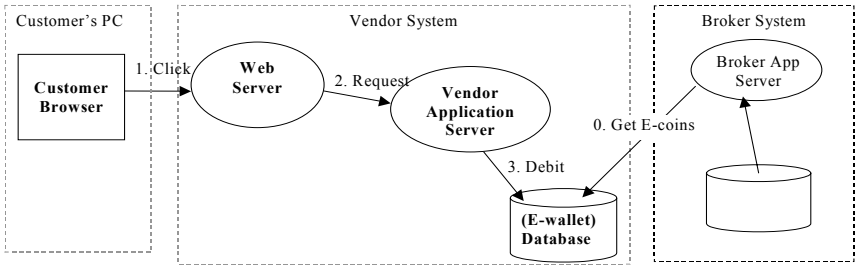


**Fig. 1.** Server-side e-wallet conceptual model.

## 4.2 Client-side e-wallet

Some people prefer to access the Internet using one machine (e.g. those who stay home most of the time or access sites from their work PC only). A Client-side e-wallet is more suitable for these kinds of people. The client-side e-wallet is an application running on the client PC that holds e-coin information.

Fig. 2 shows how a vendor application server debits e-coins from the client-side e-wallet. When buying an article content a customer clicks the title of the article on the web browser (1) and then the web server sends the request to the vendor application server (2). The vendor application server sends the price of the article to the e-wallet application (3) and then the e-wallet application returns the e-coins, paying for the content to the vendor application server (4-5).

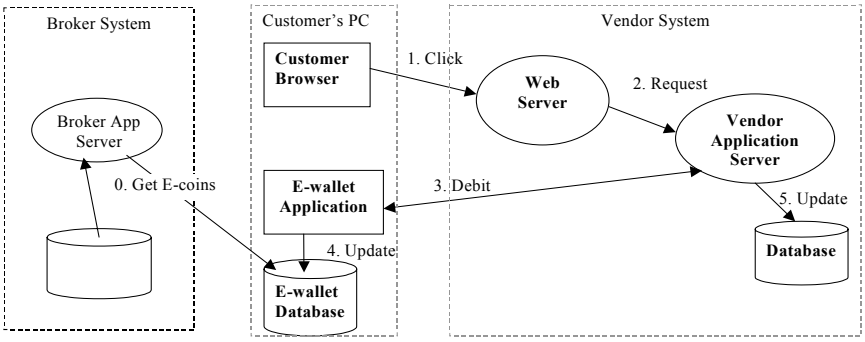


**Fig. 2.** Client-side e-wallet conceptual model.

Customers can buy article content using the client-side e-wallet at different newspaper sites without the need to log in after the e-wallet application is downloaded to their PC. Their e-coins are resident on their own PC and so access to them is never lost due to network outages to one vendor. The e-coin debiting time is slower for a

client-side e-wallet than the server-side e-wallet due to the extra communication between vendor application server and customer PC's e-wallet application.

### 4.3 Client-side cookie-based e-wallet

To reduce the e-coin debiting time with the client-side e-wallet application we can create a temporary cookie-based e-wallet that caches the e-wallet data for debiting instead of the e-wallet database. Fig. 3 shows how a vendor application server debits e-coins from such a client-side cookie-based e-wallet. When a customer finds a desired article, he/she clicks the article heading on the web browser (1). The web server sends the request to the vendor application server (2). Only for the first time when the customer buys content from the vendor web site does the vendor application server need to get the e-coins from the e-wallet application (3). It then creates a "cookie" to cache the remaining customer e-coins, stored in a cookie file on the customer PC (4). Once the cookie is created, the vendor application server debits e-coins from the cookie directly after each HTTP request to buy content (5). The e-wallet application can read the cookie file information to know how many e-coins are left when the customer wants to check the balance of the e-wallet or after the customer has moved to access another vendor site (6). This reduces the need for the vendor application server to communicate with client PC-based e-wallet, caches the e-coins in HTTP request that holds cookies.

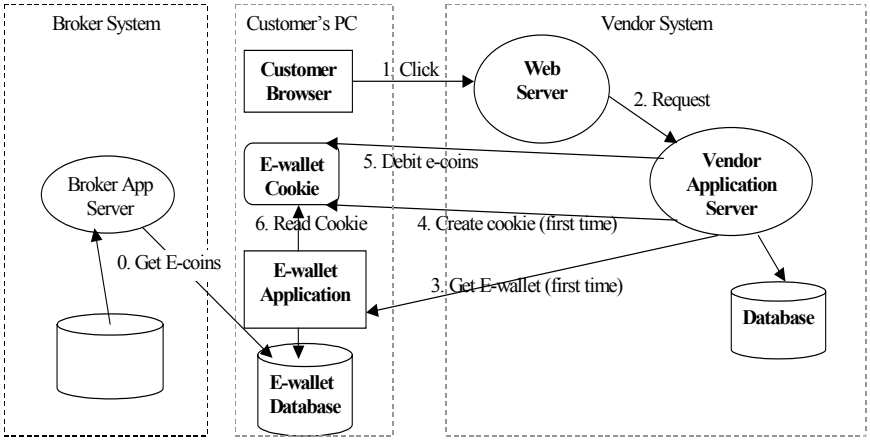


**Fig. 3.** Client-side cookie-based e-wallet

When the customer changes to another vendor, the new vendor contacts the previous vendor to request the touchstone and the index of the e-wallet, and the previous vendor application server gets the remaining e-coins from the cookie file, storing them back into the e-wallet database. It then deletes the cookie. This approach is suitable for a customer performing many purchases from a single vendor, and then changing to another vendor.

## 5 NetPay E-wallet Design

In our current NetPay prototype we have implemented two kinds of e-wallet, a server-side e-wallet and a client-side e-wallet. The broker application sets the e-wallet that stores the e-coins in the server-side or client-side.

### 5.1 Server-side E-wallet NetPay Design

The server-side e-wallet should be transferred from the broker to each vendor in turn that the customer is buying content from. Vendor systems need to know the location of the customer's e-wallet and to get the e-wallet contents. To do this we designed the broker application server so that it provides a set of CORBA interfaces with which the vendor application servers communicate to request an e-wallet location or to get an e-wallet. The vendor application servers also provide a CORBA interface in order for another vendor application server to get the e-wallet if it has been passed to one of them. The e-wallet is thus passed from vendor to vendor as needed. The major problem with this approach is that the new vendor cannot get the e-wallet when the previous vendor crashes or becomes unavailable.

When a customer first clicks the Login&Buy button to purchase e-coins on the browser, the HTTP server runs JSPs handling the request. The Broker application server communicates with a macro-payment system to debit money from the customer bank account and stores the e-coins information in the database.

When the customer goes to a vendor site, he/she needs to login by entering the e-coin ID and the password. A JSP page handles the login request. If the e-wallet does not exist, the vendor's application server communicates with broker application server via CORBA to get the e-wallet location, including host and port of the broker or previous vendor. Then it communicates with the broker/previous vendor via CORBA to get the customer's refreshed e-wallet. This includes ecoinID, touchstone, index, paywords, and amount. After the customer clicks the article handing, a JSP page deals with a display content request. The vendor application server debits e-coins from the server-side e-wallet paying for the content. The key components of the NetPay server-side e-wallet design as illustrated in Fig. 4.

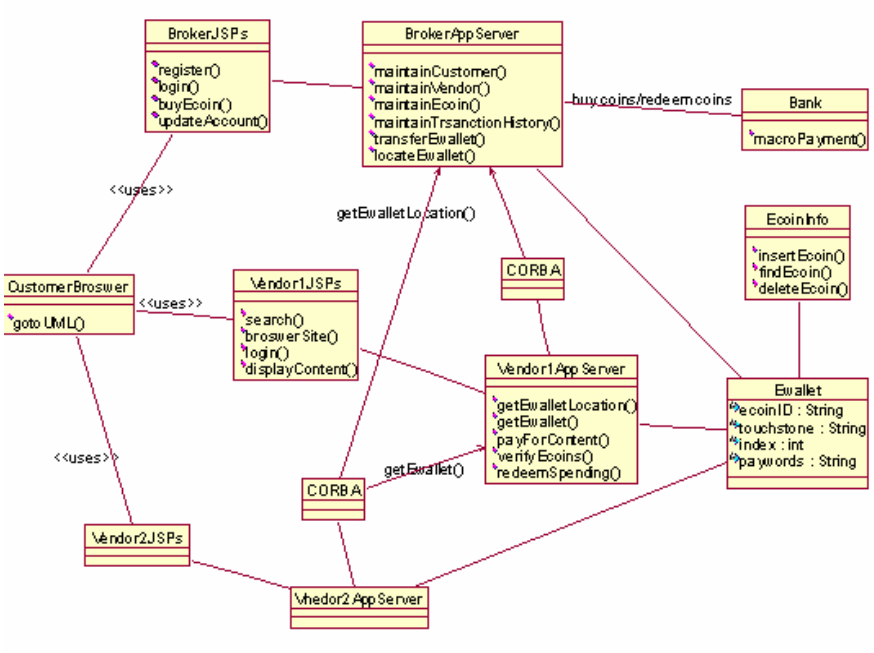


**Fig. 4.** An overview of the NetPay server-side e-wallet design.

## 5.2 Client-side E-wallet NetPay

The client-side e-wallet is implemented as a Java application runs on the client PC. According to our protocol, a touchstone and an index (T&I) of a customer's e-coin should be passed from the broker to each vendor. To do this we have the broker application server provide a CORBA interface for vendor application servers to communicate with to get the T&I to verify e-coins. The vendor application servers also provide a CORBA interface in order for another vendor application server to communication with it to pass the T&I, avoiding use of the broker where possible.

When a customer first clicks the Login&Buy button to purchase e-coins on the browser, JSPs running on the web server handle the request. The Broker application server communicates with macro-payment system to debit money from the customer bank account and then sends the e-coins to the customer's e-wallet on the customer machine.

A JSP page deals with displaying content when the customer clicks on the title of an article. The vendor application server connects with the e-wallet application and sends the price of the article. The customer's e-wallet returns with the e-coins and the location of the T&I to the vendor application server. The vendor application server communicates with the broker or previous vendor via CORBA to obtain the T&I,

which are used to verify the e-coins. The main client-side NetPay e-wallet design features are illustrated in Fig. 5.

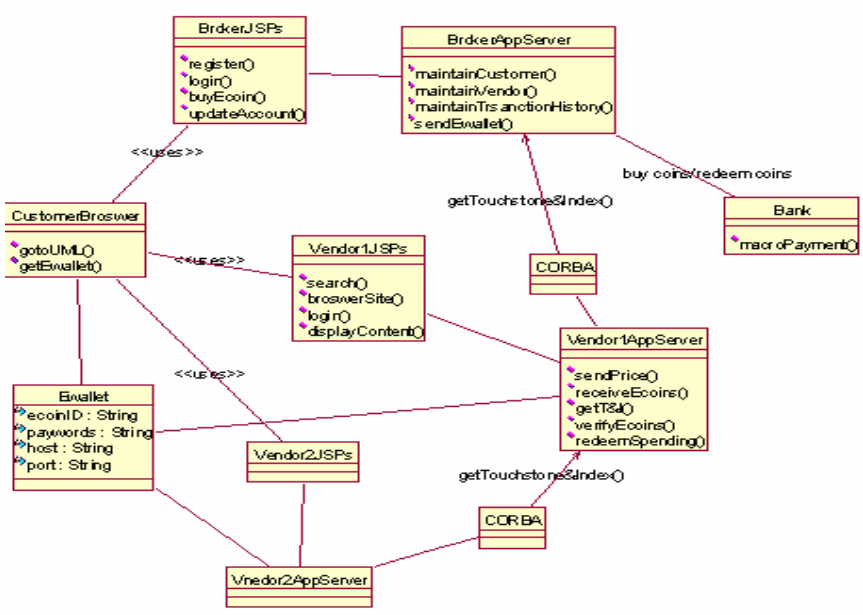


**Fig. 5.** An overview of the NetPay client-side e-wallet design.

The cookie-based E-wallet design is an extension to the client-side e-wallet design that uses browser-based cookies as a temporary cache. When the Vendor application server first communicates with the client PC-hosted e-wallet application, it reads the e-coins from the e-wallet and stores them in a browser cookie. The e-wallet updates its database to indicate the e-coins are now cached by the local browser in a cookie file for the vendor. Each subsequent pay-per-click from the same vendor has one or more e-coins from the cookie removed and stored in the vendor's redemption database. If the customer moves to another vendor, the first new vendor access to the e-wallet application causes an update of the e-wallet e-coins from the cached cookie file from the previous vendor. This cookie file is then deleted by the client PC-hosted e-wallet.

## 6 Example Usage

We briefly illustrate the usage of our E-wallet prototypes for a simple E-newspaper system enhanced with NetPay. Fig. 6 shows an example of NetPay in use for this prototype E-newspaper application. The customer first buys e-coins from a broker (1-2), and these are stored in a client-side E-wallet (3) or requested by a vendor on customer login.

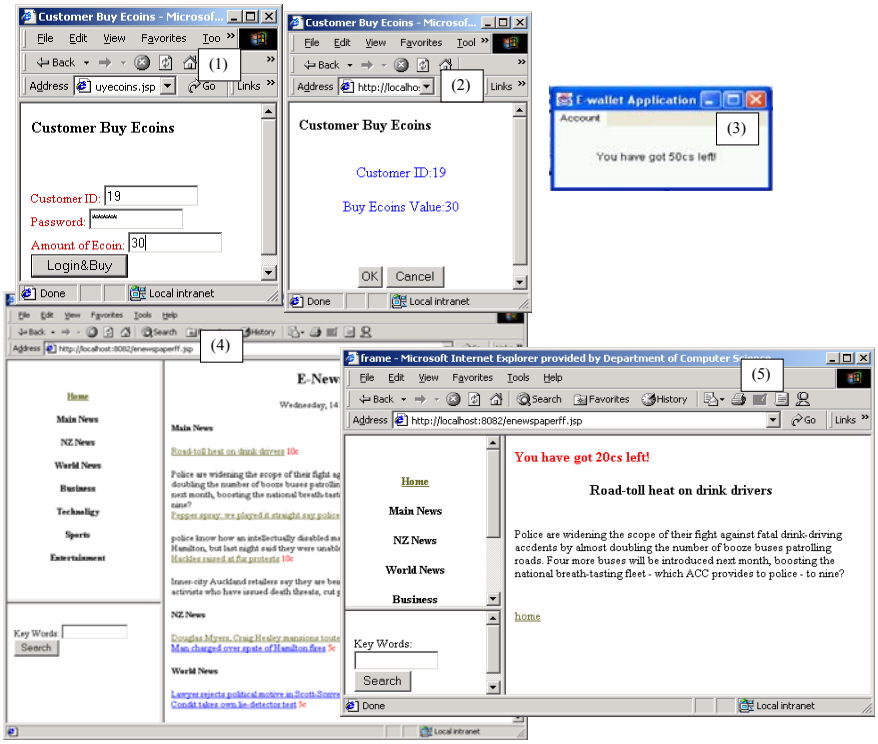


**Fig. 6.** Example of NetPay and e-wallets in use

The customer visits a NetPay-enhanced vendor site (4), and if using a server-side E-wallet logs in providing an E-wallet ID and password generated by the broker. The customer accesses articles on the E-newspaper site, each time their E-wallet being debited (5). The server-side E-wallet held by the vendor site server is debited and the remaining credit displayed to the user after each article is presented, as shown in (5). The client-side E-wallet is held by an application resident on the customer PC, and its balance can be accessed when desired or displayed periodically, as shown in (3). The cookie-based E-wallet provides the same look-and-feel as the client-side E-wallet, with caching of the remaining E-coins in a browser cookie done by the vendor's Net-Pay-enhanced web server.

## 7 Discussion

We focused on designing three kinds of "E-wallets" for NetPay broker and vendor prototypes using a CORBA-based approach. Each approach has advantages and disadvantages. The first requires uses of e-coin ID and password to login to a vendor

system. The later two require that the customers download an e-wallet application and install it on their PC. The e-coin debiting time is slower for a client-side e-wallet than for a server-side e-wallet system due to extra overhead communicating to the client PC. A firewall installed on the customer PC may also interfere with this communication from the vendor server. We implemented two kinds of the NetPay e-wallet and "hard-coded" this support into a prototype vendor application to enhance it with NetPay mcro-payment support. We used these prototype client-side and server-side e-wallets to carry out an evaluation of NetPay usability and performance.

We used three prototypes, one providing macro-payment style subscription-based payment, a second server-side NetPay micro-payment and the third client-side NetPay micro-payment. We carried out some basic usability testing via a survey-based approach with representative target users of NetPay [4]. This survey found that the article content at different newspaper sites was found by users to be easy to access and read without logging into client-side NetPay system. However, users found that this approach could incur a distracting extra delay in page display over the other systems. The server-side NetPay system allowed users to read articles anywhere in the world, but customers needed to remember e-coin Ids in order to use the system. The conventional macro-payment based system was found to be less appealing to our users than the NetPay-enhanced micro-payment implementing e-newspaper vendors.

Our three e-newspaper prototypes have also been tested for application server performance and client response time under heavy loading [4]. The key aim was to test how long a newspaper site takes to serve client requests when extended to use each of the three payment systems, from the time the customer clicks the title of an article to the time the article is fully displayed on screen.

The results of the set of performance impact tests are shown in Table 1. The response time measures how long it takes for a page to be returned from the vendor site.

**Table 1.** Initial prototype performance

| System | Response Delay Time (average) |
|---|---|
| Subscription-based | 16ms |
| Server-side NetPay | 80ms |
| Client-side NetPay | 950ms |

From Table 1, the server-side NetPay takes 64ms for e-coin debiting per article and Client-side takes 934ms total time, though the time to debit coins is taken by the client's e-wallet application, not the vendor's application server. The large response time 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 heavy update overhead. We enhanced the NetPay vendor server components to use a redemption transaction log file with over night update of the vendor redemption database from the log file. This markedly improved server-side performance and reduced server CPU and database overhead by nearly 40%. Further enhancements, such as the use of a server-side

memory database for managing e-coins for redemption and server-side e-wallets could further reduce the impact of NetPay components on vendor server performance.

## 8 Summary

We have described the development of a new micro-payment system, NetPay, featuring different ways of managing electronic money, or e-coins. NetPay provides an off-line, anonymous protocol that supports high-volume, low-cost electronic transactions over the Internet. We developed three kinds of e-wallets to manage coins in a NetPay-based system: a sever-side e-wallet allowing multiple computer access to e-coins; a client-side e-wallet allowing customer PC management of the e-coins, and a cookie-based e-wallet cache to improve performance of the client-side e-wallet communication overhead. Experiences to date with NetPay prototypes have demonstrated it provides an effective micro-payment strategy and customers welcome the ability to manage their electronic coins in different ways.

## References


1. Dai, X. and Lo, B.: NetPay – An Efficient Protocol for Micropayments on the WWW. Fifth Australian World Wide Web Conference, Australia (1999)
2. Dai, X., Grundy, J. and Lo, B.: Comparing and contrasting micro-payment models for E-commerce systems, International Conferences of Info-tech and Info-net (ICII), China (2001)
3. Dai, X., Grundy, J.: 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, June 24-27, 444--450
4. Dai, X. and Grundy J.: "Customer Perception of a Thin-client Micro-payment System Issues and Experiences", *Journal of End User Computing*, 15(4), pp 62-77, (2003).
5. Gabber, E. and Silberschatz, A.: "Agora: A Minimal Distributed Protocol for Electronic Commerce", *Proceedings of the Second USENIX Workshop on Electronic Commerce, O*akland, California, November 18-21, 1996, pp. 223-232
6. Gabber, E. and Silberschatz, A.: "Micro Payment Transfer Protocol (MPTP) Version 0.1". *W3C Working Draf*t, 1995. http://www.w3.org/pub/WWW/TR/WD-mptp
7. Herzberg, A. and Yochai, H. : Mini-pay: Charging per Click on the Web, 1996 http://www.ibm.net.il/ibm_il/int-lab/mpay
8. Manasse, M.: The Millicent Protocols for Electronic Commerce. First USENIX Workshop on Electronic Commerce. New York (1995)
9. MP3 Web Site: http://www.mp3.com
10. Posman, "Would You Pay for Google?", 2002. http://www.clickz.com/media/agency_start/article.php/1013901
11. Rivest, R. and Shamir, A.: PayWord and MicroMint: Two Simple Micropayment Schemes. Proceedings of 1996 International Workshop on Security Protocols, Lecture Notes in Computer Science, Vol. 1189. Springer (1997) 69—87
12. Welcome to MilliCent WORLD Homepage, 2001. http://www.mllicent.gr.jp