

Article

# M&E-NetPay: A Micropayment System for Mobile and Electronic Commerce

Xiaodi Huang <sup>1,\*</sup>, Jinsong Bao <sup>2,\*</sup>, Xiaoling Dai <sup>1</sup>, Edwin Singh <sup>3</sup>, Weidong Huang <sup>4</sup>  
and Changqin Huang <sup>5</sup>

<sup>1</sup> School of Computing and Mathematics, Charles Sturt University, Albury 2640, Australia; sdai@csu.edu.au

<sup>2</sup> College of Mechanical Engineering, Donghua University, Shanghai 201620, China

<sup>3</sup> The University of the South Pacific, Laucala Campus, Suva, Fiji; singh\_edwin@gmail.com

<sup>4</sup> School of Engineering and ICT, University of Tasmania, Hobart 7005, Australia; tony.huang@utas.edu.au

<sup>5</sup> School of Information Technology in Education, South China Normal University, Guangzhou 510660, China; cqhuang@zju.edu.cn

\* Correspondence: xhuang@csu.edu.au (X.H.); bao@dhu.edu.cn (J.B.);  
Tel.: +61-2-6051-9652 (X.H.); +86-21-6779-2583 (ext. 33) (J.B.)

Academic Editor: Angel Garrido

Received: 17 June 2016; Accepted: 27 July 2016; Published: 3 August 2016

**Abstract:** As an increasing number of people purchase goods and services online, micropayment systems are becoming particularly important for mobile and electronic commerce. We have designed and developed such a system called M&E-NetPay (Mobile and Electronic NetPay). With open interoperability and mobility, M&E-NetPay uses web services to connect brokers and vendors, providing secure, flexible and reliable credit services over the Internet. In particular, M&E-NetPay makes use of a secure, inexpensive and debit-based off-line protocol that allows vendors to interact only with customers, after validating coins. The design of the architecture and protocol of M&E-NetPay are presented, together with the implementation of its prototype in ringtone and wallpaper sites. To validate our system, we have conducted its evaluations on performance, usability and heuristics. Furthermore, we compare our system to the CORBA-based (Common Object Request Broker Architecture) off-line micro-payment systems. The results have demonstrated that M&E-NetPay outperforms the .NET-based M&E-NetPay system in terms of performance and user satisfaction.

**Keywords:** mobile and electronic commerce; micro-payment; web services; electronic wallet; mobile networks

## 1. Introduction

Mobile commerce is concerned with conducting business transactions and providing services on portable, wireless devices over the Internet [1]. Due to the exponential growth of the number of the Internet users and the maturation of wireless communication technologies, mobile commerce has rapidly attained the interest of the business vanguard [2].

M-commerce benefits not only consumers, but also business. It is convenient for consumers to purchase goods and services by using their mobiles. M-commerce enables transactions to be conducted in a high-volume, low-cost per-item way. It is obvious that m-commerce has enormous potentials. However, the current micro-payment systems for m-commerce have the following three main problems.

A desirable protocol of micro-payment should support high-volume, low-cost per-item transactions from vendors [3–10]. Several micro-payment protocols have been proposed for electronic payment in m-commerce recently. The examples of such protocols include MPS (multiparty payment scheme) [11], CMP (chaotic micro-payment protocol) [12], NetPay [13] and the recent ones for wearable devices and clouds [14–16]. Many of the proposed protocols, however, suffer the problems of the

dependence on online brokers and a lack of scalability and coin transferability. The interaction between a client and a server in a CORBA-based NetPay system, for example, is mediated by object request brokers (ORBs) on both sides. A problem of this technique is that each node of CORBA has to run ORBs from the same product. In reality, it is difficult for ORBs provided by different vendors to interoperate. In addition, the interoperability does not extend into higher-level services, such as security and transaction management. Furthermore, specific advantages of particular vendors would be lost in this situation. Because this protocol depends on a closely-administered environment, it is unlikely that two random computers can successfully make Distributed Component Object Model (DCOM) or Internet Inter-ORB Protocol (IIOP) calls [17]. As a reasonable protocol for server-to-server communications, CORBA, however, has severe weaknesses in client-to-server communications, especially when client machines are scattered across the Internet.

**Middleware interfaces:** The recently-developed NetPay makes use of an off-line micro-payment model with a CORBA interface as a middleware that interconnects broker and vendor sites [18]. This prototype is suitable for ecommerce applications. In mobile environments where clients (and possibly servers) keep moving, this requires, however, dealing with the changing network addresses and unreliable connections. As a result, this mobility requirement adds additional constraints to the system. Due to its tight coupling between clients and servers, it is obvious that CORBA is not well suited for this environment. In order to overcome this barrier, in this paper, we present an off-line micropayment model that uses web services rather than CORBA as the middleware. By using the Simple Object Access Protocol (SOAP Web service protocol), the mobility requirement is dealt with by proxies that route messages accordingly. Moreover, the sender of a message and the final recipient do not have to be aware of the proxies. Web services offer greater advantages over CORBA, particularly for developing mobile applications. They cater to a large number of users who use either browsers or mobile devices. Web services add in a new functionality of interoperability, which is independent of the development platforms and programming languages used. In particular, Web Services on the .NET framework are widely available in object-oriented and distributed systems. As such, small and enterprise applications enable connecting to each other over the Internet.

**Evaluations:** Rowley [19] and Sumak et al. [20] present comprehensive reviews on e-service evaluation frameworks. The evaluations on specific and particular types of e-services, e-shops and e-business include Barnes and Vidgen [21], Behkamal et al. [22], Parasuraman et al. [23], Schubert et al. [24], and Janda et al. [25]. In this work, we evaluate our system by using three types of evaluations, which include not only user perceptions, but also system performance.

One of the big challenges for micro-payment systems is that e-coins should be allowed to be spent at a wide range of vendors. Micro-payment systems should enable mobile users to leverage buy-once-spend (almost)-anywhere behaviour. In this work, we extend NetPay into M&E-NetPay. M&E-NetPay uses Web Service interfaces as a middleware for interconnecting the sites of brokers and vendors. Web Service interfaces make it simple to transfer e-coins among vendors. E-coins in M&E-NetPay are easily transferred between multiple vendors, so that M&E users can make multiple purchases. Another challenge in the design of micro-payment systems is the minimization of overheads on the servers of the sites of brokers and vendors. As a fully-distributed multi-tier system deployed over several servers, M&E-NetPay is able to achieve the minimal downtime and maximal competence. As reported in performance evaluations, the .NET framework architecture 4.0 [26] with Web Services in M&E-NetPay improves client-to-server communications. This leads to greatly improving system performance. The architecture with Web Services provides fast, secure and inexpensive communications amongst mobile users and vendor systems. In addition, the M&E-NetPay architecture also supports servers running on different platforms and vendor applications developed by using different programming languages. This allows an M&E-NetPay-enabled vendor to act as a purchasing portal for existing non-M&E-NetPay supporting vendors. In particular, an M&E-NetPay-enabled vendor redirects page accesses to these vendors and manages the debit of user

e-coins. As such, existing vendors are encouraged to temporarily use M&E-NetPay micropayment services for dynamic registration.

In summary, we design and develop M&E-NetPay in a way that attempts to address the three above-mentioned problems.

The major contributions of this paper are as follows:

- We present a novel micro-payment model of M&E-NetPay and its architecture.
- A new way for the deployment model with a thin-client architecture and Web service interfaces is proposed, i.e., HTML and Wireless Markup Language (WML)-based interfaces for customers.
- We have implemented the prototype of M&E NetPay including one broker and two vendor sites, which are based on the .NET framework using C# and Active Server Pages (ASP.NET). In particular, two vendor sites of ringtones and wallpaper are implemented.
- The three types of evaluations have been performed on the M&E-NetPay prototype. We compare micro-payment with non-micro-payment in terms of usability, performance and heuristic assessment.

The rest of this paper is organized as follows. Section 2 describes the architecture of M&E-NetPay. The protocol and interactions of M&E-NetPay are given in Section 3. Section 4 presents the implementation of the M&E-NetPay prototype. Section 5 reports the evaluations on the system, followed by related work and comparisons in Section 6. We conclude this paper in Section 7.

## 2. M&E-NetPay Architecture

In this section, we outline the architecture of M&E-NetPay, including the hardware and software architectures.

### 2.1. M&E-NetPay Software Deployment Architecture

Taking into account the general requirements on performance, security, availability and serviceability, we designed the deployment architecture of M&E-NetPay as shown in Figure 1.

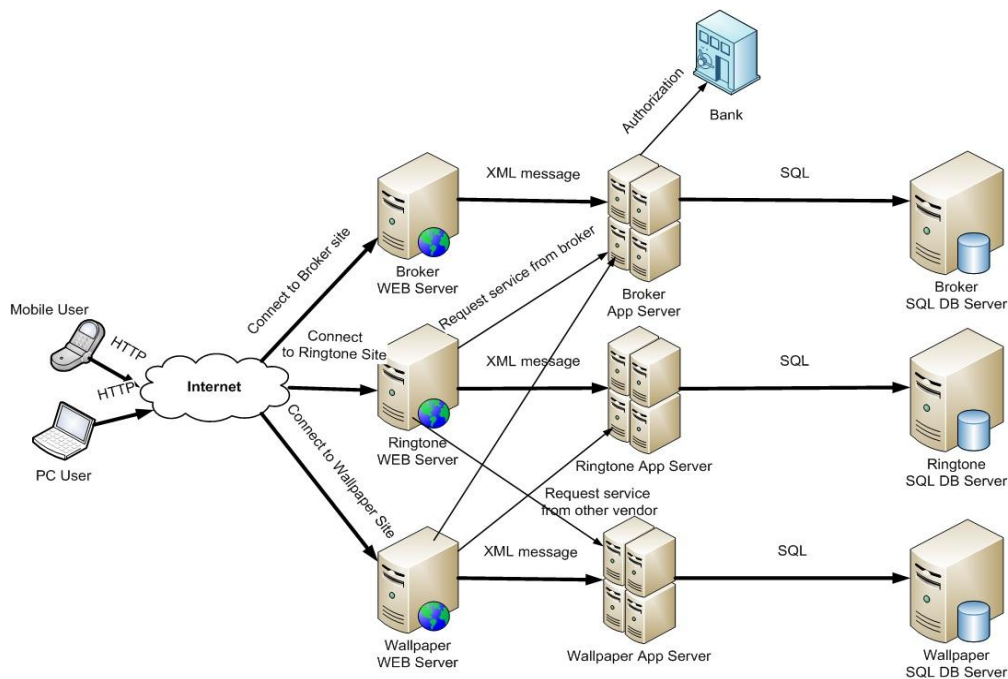


Figure 1. The basic deployment architecture of M&E-NetPay.

As a thin client n-tier application, M&E-NetPay is deployed over three servers: web servers, application servers and database servers. Web servers deploy broker and vendor web/mobile applications. Application servers publish Web services of the broker and vendor. Database servers store required information.

M&E-NetPay is maintainable and serviceable in that any changes result in re-configuring of only part of the application. If the ringtone vendor wants to update its site, for example, then only the web/mobile application on its Web server is re-configured.

### 2.2. M&E-NetPay Software Architecture

The software architecture of the M&E NetPay micro-payment system is shown in Figure 2. The architecture is designed for Microsoft.NET applications. It consists of the following components.

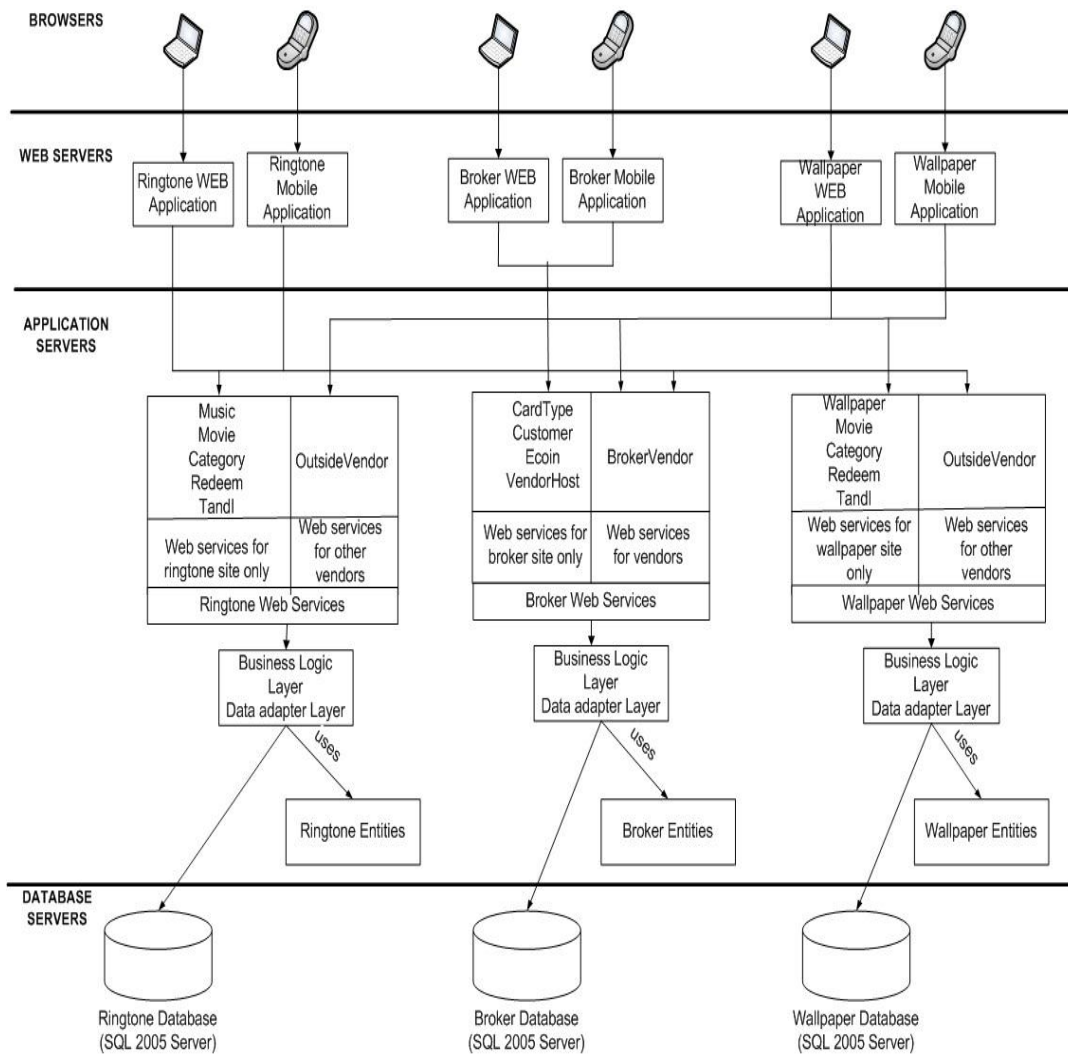


Figure 2. M&E-NetPay basic software application architecture.

Browser: two types of users can access a broker site using their mobile phones or PCs with Internet access. By using a Wireless Markup Language (WML)-based Web browser in their mobile phones, mobile users run the Broker Mobile application with its interface for the small screen of a mobile phone. Internet users can access the Broker Web application through a popular web browser.

Web services: These host the presentation layer. It is much easier to connect remote sites by using web services.

The web service is only available to vendors for accessing certain information from the broker's database. User queries are issued to broker data entities from the client end, and the results are retrieved by data layers. Mobile and Web applications invoke the same Web services hosted on the broker's application server. The broker Web services pass information in an XML-based message to the business logic layer. In our applications, this means that data are retrieved from a database into an entity or entity collection and then updated data are written from an entity back to the database.

**Application servers:** These mainly accommodate Web services, the business logic layer and the data adapter layer. The business logic layer implements all business rules for the application. The business logic layer passes information to the data adapter layer, the broker database, and executes necessary queries. The data adapters exchange data between a data source and a dataset.

**Database servers:** These host relational databases, including the ringtone database, the broker database and the wallpaper database. The database in the broker server records account information and transaction histories of all registered users.

The e-wallet of a user resides on the broker's database until she or he logs on to a vendor site using a given e-coin id [18]. Upon login, her or his e-wallet is transferred to the visiting vendor. The broker helps the vendor to verify e-coins, when she or he purchases items from its site. The broker also allows the vendor to redeem e-coins spent on its site and to request touchstones. These functionalities are provided by the "BrokerVendor" Web service of the broker, as shown in Figure 2.

Similarly, vendor sites also provide their interfaces to both mobile and Internet users. The vendor sites allow users to browse their websites and purchase items. When a user logs in to the ringtone site in our system, the ringtone vendor requests her or his e-wallet from the broker. This function is provided by the Web service of "BrokerVendor" of the broker. If the ringtone vendor finds that the e-wallet of this particular user resides on another vendor site, it then requests her or his e-wallet from the vendor that contains e-coin indexes and touchstones. Each vendor has a Web service called "OutsideVendor", which allows other vendors to retrieve e-wallets of their own users. The e-wallet is then stored on the current vendor's site. Once the user purchases an item, her or his e-wallet is debited.

### 3. Protocol and Interactions of M&E-NetPay

In this section, we describe the protocol and interactions of M&E-NetPay.

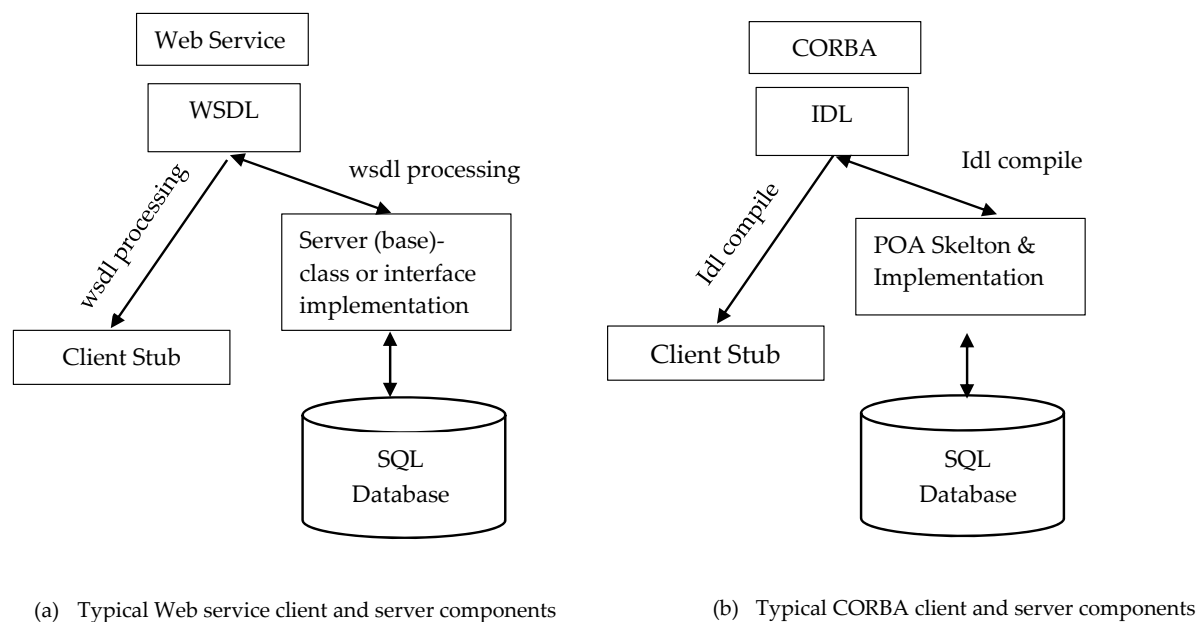
#### 3.1. NetPay and M&E-NetPay Micropayment Protocol

M&E-NetPay is evolved from NetPay. Therefore, we start with describing NetPay. It is an off-line micro-payment system by using a secure, inexpensive and debit-based protocol [13]. The NetPay micropayment system has three models of "e-wallets" that manage e-coins. Like other models, e-wallets in the first model are hosted by vendor servers. An e-wallet is passed from one vendor to another, as a customer visits different sites for e-commerce transactions. The second model is a stand-alone client-side application on a client's PC. The third model is a hybrid one that caches E-coins in a Web browser cookie for debiting, if a customer makes a purchase. The NetPay-based system is developed for the CORBA-based broker, vendor and customer networks. By using a set of CORBA interfaces, the broker application server communicates with the vendor application servers for requesting touchstones and redeeming e-coins [18]. CORBA enables clients to invoke methods on remote objects at a server, regardless of by which language objects are programed and where they are located.

M&E-NetPay replaces the CORBA middleware with Web Services, which provide the interoperability (i.e., platform-independent and language-independent). Using a simple XML-based protocol and SOAP, a Web service is an emerging distributed middleware technique that allows applications to exchange data over the Web. It is a new programming model for building distributed applications by open Internet standards. This new technique manoeuvres the openness of specific Internet technologies to address many interoperability issues of CORBA. Web services use Hyper Text Transfer Protocol (HTTP) to transmit messages. This is a major advantage for building an Internet-scale



application like the M&E-NetPay system, since most of the Internet proxies and firewalls do not have trouble with HTTP traffic. In contrast, CORBA usually has difficulties with firewalls. Moreover, Web Services are platform-independent and language-independent (i.e., a client program can be programmed in C# and running under Windows, while the Web Service is programmed in Java and running under Linux.). Web Services support different interfaces of client-side application programs. Client code may work by constructing “call” objects that are dispatched to a server or may use a higher level interface that hides the communication level entirely through the use of client-side stub objects with an operational interface that imitates the server [27]. The mechanisms for generating client and server components for Web Services and CORBA are illustrated in Figure 3.



**Figure 3.** Basic client and server components from the interface for Web Services and CORBA.

M&E-NetPay uses a secure, inexpensive widely-available and debit-based protocol. The M&E-NetPay protocol differs from the previous protocols in that running on the .NET platform, it uses Web service interfaces as its middleware.

### 3.2. M&E-NetPay Micropayment Interaction

Based on the NetPay protocol of the server-side e-wallet [13,28,29], we extend it into the M&E-NetPay protocol in a way that is suitable for mobile and Internet environments. The M&E-NetPay protocol uses touchstones signed by a broker, as well as e-coin indexes signed by requesting peers. The signed touchstone is used by a vendor to verify the electronic currency: paywords. A signed index prevents customers from double spending and resolves disputes between customers and vendors. We assume that an honest broker is trusted by both customers and vendors. The broker manages the bank accounts of all mobile and Internet users. A bank transfers money to a broker on an online request. The mobile/Internet users access the mobile/web application through Web browsers on mobile phones or PCs. In order to purchase items from vendor sites, a mobile or Internet user needs to register with a broker. Upon successful registration, the user’s account is created. She or he then needs to buy e-coins from the broker by using her or his credit card. She or he is issued a unique e-coin id each time once having bought e-coins from the broker. She or he can log onto a vendor site using the e-coin id and password. In our system, two vendor sites of ringtones and wallpapers are implemented. A user browses the site and selects the ringtone or wallpaper. A small cost is assigned to each ringtone and wallpaper, depending on their demand and ratings. After the user clicks on the download button

in the ringtone site, the broker debits the user account for the cost of the ringtone, e.g., if the user is downloading a ring tone costing “10 c”, then the user’s account is debited by “10 c”.

A user e-wallet is saved on the vendor site last visited. Once the user logs on to the other vendor site for browsing other ringtones, her or his e-wallet is transferred from her or his last visited vendor to the current one. If her or his e-coins are run out, she or he is directed to the broker site to buy more e-coins. At the end of each day, all of the vendors collect the money from the broker in return for e-coins.

As an example, we describe the procedure of macro-payment in M&E-NetPay in the following. Figure 4 also illustrates some key interactions.

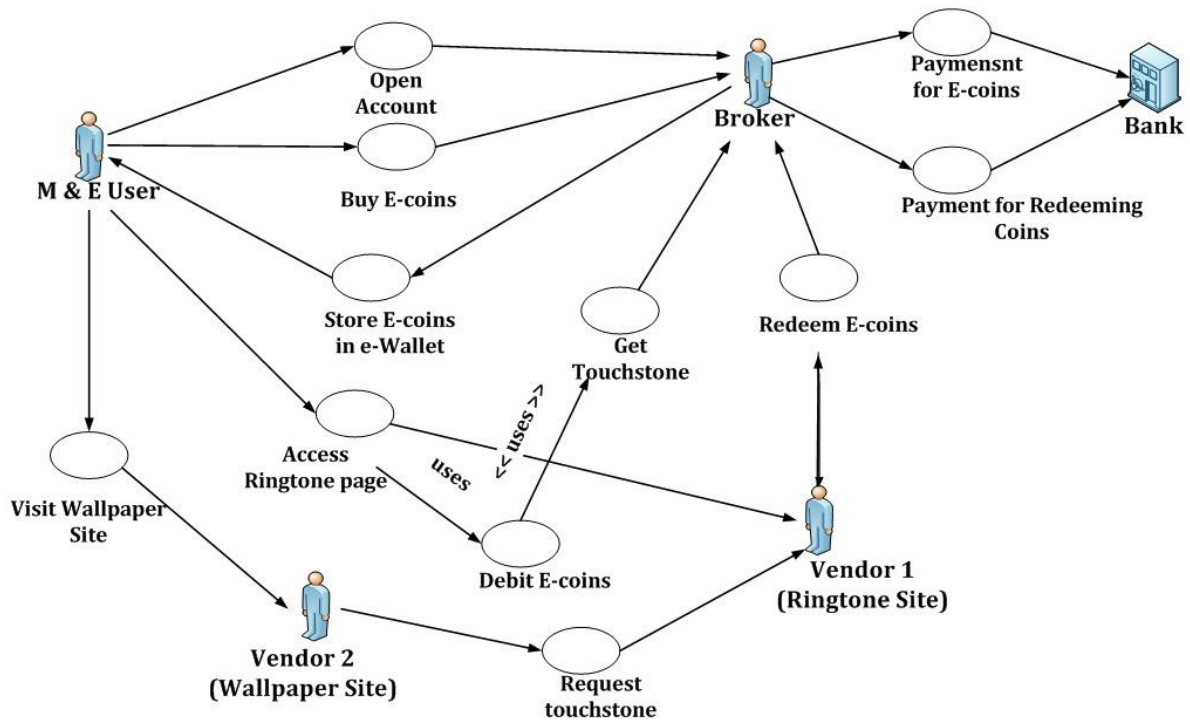


Figure 4. M&E-NetPay component interactions.

Initially, a mobile user registers on a broker’s Web site and buys a number of e-coins.

1. The broker may provide credits as “virtual money”, which is specific to the network only. The P2P network may require peers to use real money to subscribe and/or to use services. In this case, the broker uses a macro-payment, e.g., credit card transactions with a conventional payment party to buy credits.
2. The broker generates an e-coin chain and stores it in an “e-wallet”.
3. When the mobile user selects ringtones to be downloaded from Vendor 1 site, Vendor 1 obtains e-coins from the e-wallet and verifies the e-coins. The mobile user then can download the ringtones.
4. The mobile user may download other ringtones, and her or his coins are debited. If her or his coins run out, she or he is directed to the broker site for buying more. When the mobile user browses Vendor 2, Vendor 2 contacts Vendor 1 in order to obtain the touchstone and index (T & I) and then debits e-coins for this user to download more wallpapers.
5. At the end of each day, the vendors send all of their received e-coins to the broker for redeeming them.
6. For their own credits, vendors may be able to cash them in for real money, again via a conventional macro-payment approach.

As a summary, a mobile user downloads ringtones from Vendor 1. Vendor 1 requests the touchstone, index and e-coins from the broker. After verification, the mobile user is allowed to download ringtones. Vendor 1 sends T & I to the broker. After browsing other vendors, the mobile user wants to download wallpapers from Vendor 2, which contacts Vendor 1 for T & I. If Vendor 1 is off-line, then Vendor 2 requests T & I from the broker.

#### 4. Implementation of M&E-NetPay Prototype

In this section, we present the implementation of the M&E-NetPay prototype.

Our system has implemented one broker and two vendor sites. All applications are developed on the Microsoft.NET platform framework 4.0 [26]. We choose Microsoft Visual Studio 2010 ASP.NET and the C# programming language for frontend implementations and Microsoft SQL Server 2005 for database storage. We use HTML with ASP controls for Web pages and the C# programming language for the back end of the application. The broker and vendors provide access to both mobile and Internet users published on the web servers' IIS. Web service interfaces are implemented on the application servers' IIS, which provides access to the Internet, as well as to remote sites. Vendors and the broker can choose programming languages and operating systems for implementing their systems. A vendor application implemented by the C# programming language on the Windows operating system, for example, can easily communicate with another implemented by the C++ programming language on the UNIX operating system.

To make it more effective and efficient, M&E-NetPay consists of three components: the presentation logic, which presents information to the M&E users; business components, which controls the relationship between inputs and determines business rules; and the data adapter layer, which connects to the database, executes relevant queries and returns the results back to the upper layers. The presentation and business components are communicated only via Web Services, no matter whether they are within a system or between the systems.

Web Services are used as the middleware for M&E-NetPay. Figure 5 shows Web service references on the broker site referenced from the broker Web Service.

```
<applicationSettings>
<BrokerWebApp.Properties.Settings>
<setting name="BrokerWebApp_Customer_CustomerChannel" serializeAs="String">
<value>http://localhost/BrokerWebService/CustomerChannel.asmx</value></setting>
<setting name="BrokerWebApp_CardType_CardTypeChannel" serializeAs="String">
<value>http://localhost/BrokerWebService/CardTypeChannel.asmx</value></setting>
<setting name="BrokerWebApp_Ecoin_EcoinChannel" serializeAs="String">
<value>http://localhost/BrokerWebService/EcoinChannel.asmx</value></setting>
<setting name="BrokerWebApp_VendorHost_VendorHostChannel" serializeAs="String">
<value>http://localhost/BrokerWebService/VendorHostChannel.asmx</value></setting>
</BrokerWebApp.Properties.Settings>
</applicationSettings>
```

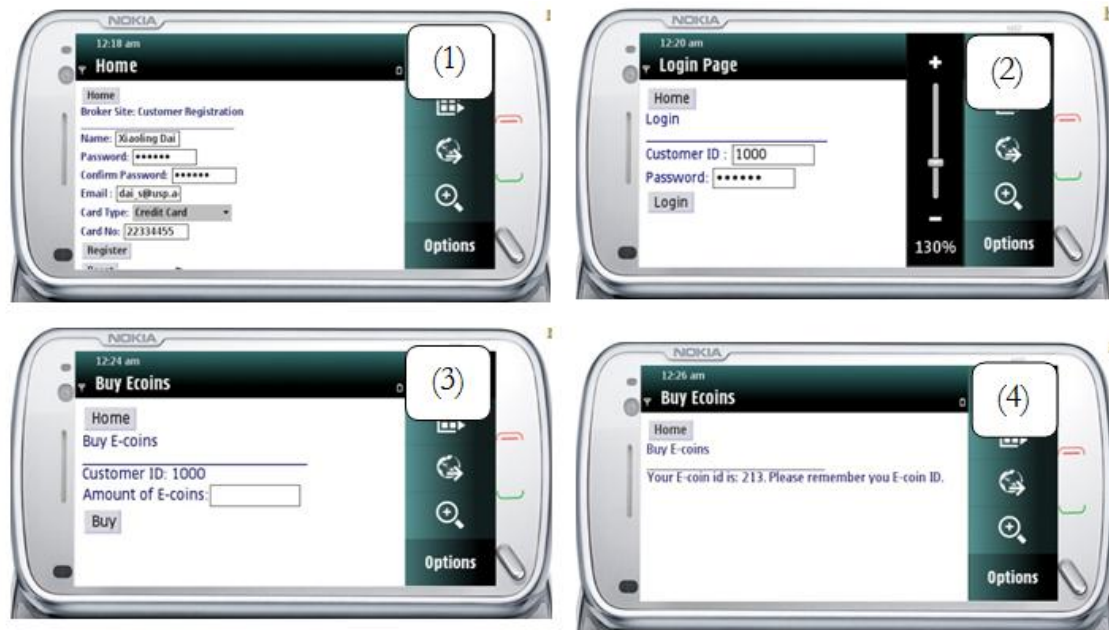
**Figure 5.** Code for Web Service references on the broker Web application.

##### 4.1. Broker

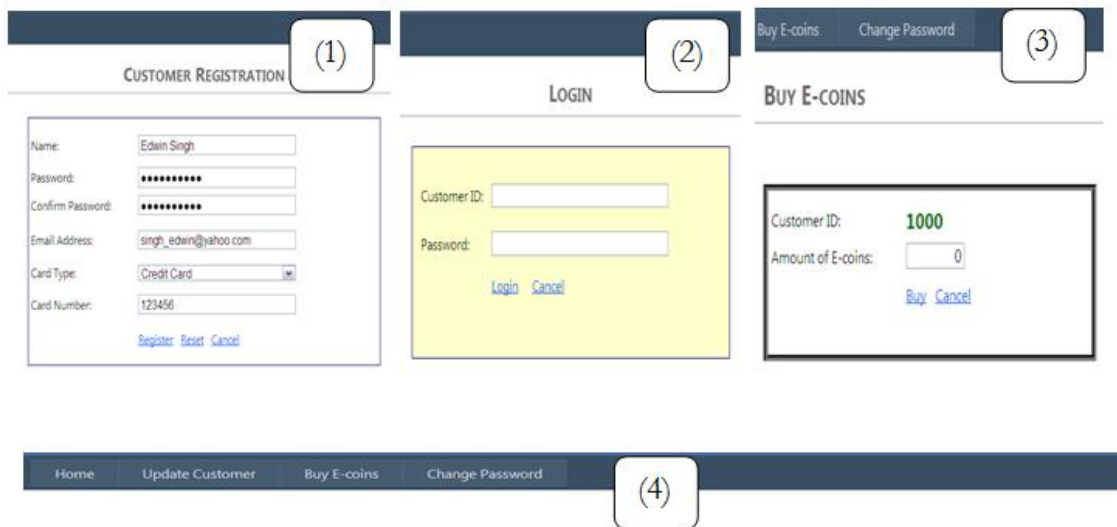
A broker manages customer and vendor accounts, e-coin creation, e-coin redemption, touchstone supply for e-coin verification and macro-payment handling for e-coin purchase and payment to vendors for spent e-coins [13]. The broker database holds user and vendor information. The application server provides business functions. Web service interfaces are for application servers of the broker and vendor. WML interfaces, implemented by using Active Server Pages (ASP.NET) with the ASPX extension, are for mobile users, while HTML interfaces are for Internet users. The Web service interface allows vendors to request e-coin touchstone information, verify e-coins and redeem spent e-coins by other vendors.



Figure 6 shows the screenshots of a customer purchasing e-coins from a broker: (1) registering with the broker to create her or his account; (2) logging in by using the provided customer id and password; (3) authorizing macro-payment by the broker in order to buy e-coins; and (4) debiting the M&E user account for paying e-coins by the bank.



(a)



**E-COIN BUY SUCCESSFUL**

Your E-coin ID is: 211 . Please Remember your E-coin ID.You will need your E-coin ID when purchasing item (s) from any vendor's site.

(b)

**Figure 6.** M&E users purchasing e-coins from a broker. (a) Wireless Markup Language (WML) interfaces for mobile phone users; (b) HTML interfaces for Internet users.

#### 4.2. Customer

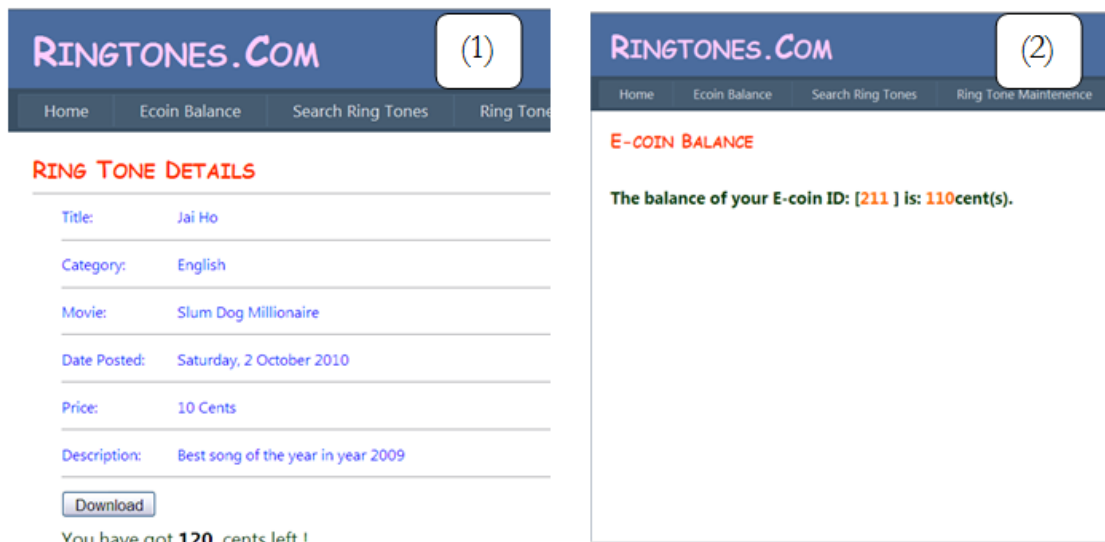
The WML/HTML interfaces of our system are provided for both mobile and Internet users, so that a wide range of customers is allowed to access broker and vendor sites by using a standard Web browser. The use of the thin client technology omits the need to install separate browser software on the client site. The customers use WML/HTML-based ASP.NET pages to browse broker and vendor sites. Being hosted on the server side, the e-wallet of a customer can be transferred from one vendor to another, as the customer makes purchases from those vendors. The e-wallet is held on the vendor server from which the customer is currently buying items.

#### 4.3. Vendor

The site of a vendor displays ASP.NET pages for M&E users to browse. Search functions in sites of the ringtone and wallpaper are provided for users to search for ringtones or wallpapers. The search results are listed as a brief summary of the ringtone or wallpaper with its download cost, as shown in Figure 7a. After downloading an item, the refreshed ASP.NET pages indicate that the amount of e-coins is left with the current vendor in the e-wallet of the user, as shown in Figure 7b.



(a)



(b)

**Figure 7.** M&E user spending e-coins at the ringtone site. (a) WML interfaces for mobile phone users; (b) HTML interfaces for Internet users.

## 5. Evaluation

In this section, we compare M&E-NetPay to non-micro-payment systems. From different perspectives of end users, we evaluate the M&E-NetPay based micro-payment system by collecting and analysing customer views.

### 5.1. Experimental Design

Three types of evaluations on the M&E-NetPay micro-payment are carried out:

- Performance evaluation [30], which compares the performance of the M&E-NetPay prototype with that of the CORBA-based NetPay system in terms of response time. This evaluation aims to assess the potential scalability of the system under heavy loading conditions.
- Usability evaluation, which assesses whether M&E-NetPay is useful as far as end users are concerned. Their opinions about our prototype are surveyed, after potential end users purchase items by using the micro-payment, M&E-NetPay and the alternative CORBA-based NetPay system, respectively.
- Heuristic evaluation, which assesses the overall quality of the user interface. Potential design problems of the user interface of the M&E-NetPay prototype are identified by using a range of common HCI design heuristics.

Experiment prototypes and materials: The evaluations are conducted on two prototypes of M&E-NetPay and CORBA-based NetPay. M&E-NetPay is deployed over three servers:

- Web server, which hosts the presentation layer
- Application server, which hosts Web Services, business logic components and data adapter layer
- Database server, which hosts the relational database

The CORBA-based NetPay system is deployed over three servers:

- Web server, which hosts JSP pages as the presentation layer
- CORBA application server, which hosts business logic components
- Database server, which hosts the relational database

A number of PCs connected to the network is used by the participants. Both prototypes are deployed over multiple machines connected via a high speed LAN.

### 5.2. Performance Evaluation

We carry out experiments on measuring client response time with ten tests. This evaluation aims to compare how long it takes to download wallpapers in the two different payment systems.

Subject: Ten users are a mixture of non-IT specialists, graduate students and college students who volunteer to conduct the evaluation.

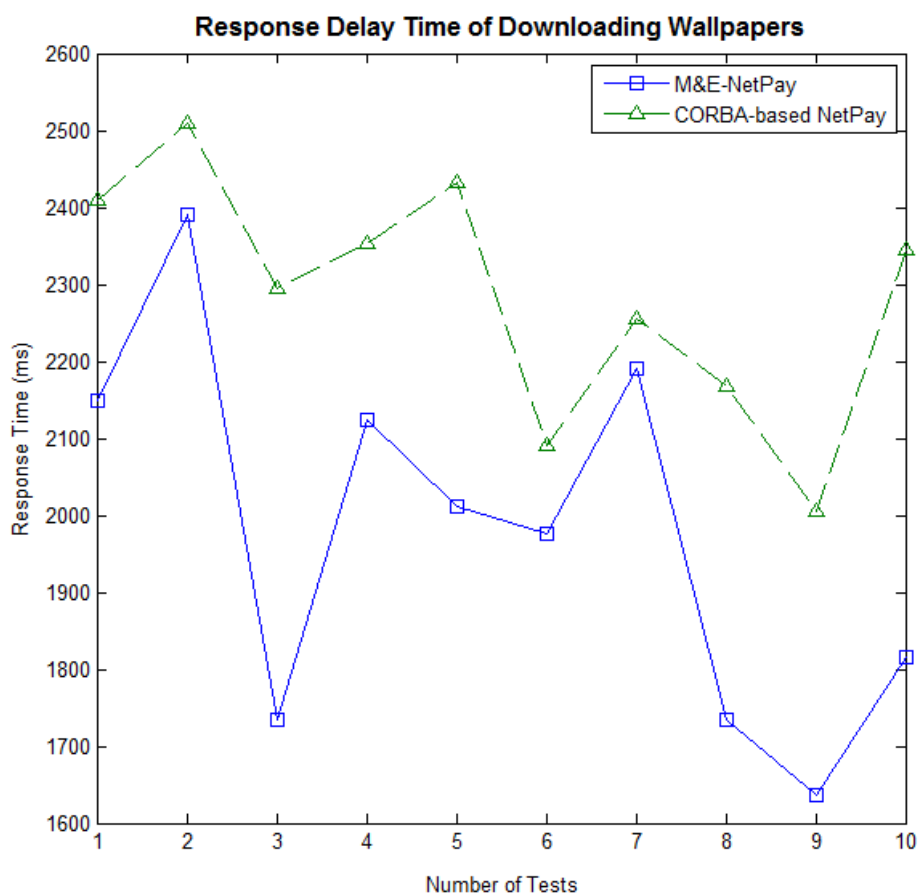
Experimental tasks: The users are required to download the same file from both M&E-NetPay and CORBA-based NetPay systems.

The response times of searching for wallpapers, buying e-coins and redeeming e-coins are recorded. They give an indication of the likely scalability of the prototype systems under heavy loading conditions.

Results: As reported in Table 1 and illustrated in Figure 8, we compare M&E-NetPay to CORBA-based NetPay against the response time of downloading wallpapers. The response time delay is the time for downloading a wallpaper. All ten users download the same wallpaper with the size of 38.4 KB.

**Table 1.** Times for downloading wallpapers.

Test	Response Delay Time with M&E-NetPay (ms)	Response Delay Time with CORBA-Based NetPay (ms)
1	2149	2410
2	2390	2509
3	1734	2294
4	3065	2354
5	2012	2432
6	1976	2091
7	2190	2256
8	1734	2168
9	1637	2005
10	1815	2344
<b>Average</b>	<b>1976</b>	<b>2286</b>



**Figure 8.** Response delay time of downloading wallpaper.

The result of the *t*-test on the data in the two columns of Table 1 rejects the null hypothesis at the default 5% significance level. That is, the two response delay times of downloading the wallpaper from the two systems have a statistically-significant difference. The test parameters are given below: the *p*-value: 0.0033; confidence interval for the difference in population means of the response time in M&E-NetPay and CORBA-based NetPay:  $-502.5709$  and  $-117.8291$ ; the test statistic:  $-3.3878$ ; degrees of freedom (df): 18; and the estimate of the population standard deviation: 204.7455.

It is obvious that the above statistical test result is limited by the size of sample tests. Despite this, the average response delay time for downloading a wallpaper from CORBA-based NetPay is slightly

higher than that from M&E-NetPay. On average, the clients take 1976 milliseconds from M&E-NetPay and 2286 milliseconds from CORBA-based NetPay to download the same wallpaper. The time difference is 310 milliseconds. Except for downloading time, we also compare the two systems against the response times of the respective operations of searching for wallpapers, buying and redeeming e-coins in Table 2.

**Table 2.** Results of searching wallpapers, buying and redeeming e-coins.

Tasks	Average Response Delay Time by M&E-NetPay (ms)	Average Response Delay Time by CORBA-Based NetPay (ms)
Search wallpapers	1501	1703
Buy e-coins	895	920
Redeem e-coins	1990	2110

As listed in Table 2, the searching for wallpapers in M&E-NetPay is 202 milliseconds faster than that in CORBA-based NetPay. Buying and redeeming e-coins also take less time in M&E-NetPay.

There may be other factors that affect the response time of the systems. However, the experiment results still indicate that M&E-NetPay may respond to user interactions faster than NetPay. This observation results from CORBA's limitation in client-to-server communications. In contrast, .NET framework architecture 4.0 with Web Services in M&E-NetPay improves client-to-server communications. It provides relatively fast communications amongst the vendor and broker. In addition, M&E-NetPay, built on a stable, secure and simple architecture, is deployed over multiple servers to share the workload among them.

### 5.3. Usability Evaluation

We survey the satisfaction levels of the participant users, after they download and purchase items. Furthermore, we ask their preferences for the two systems in general: a CORBA-based NetPay system or M&E-NetPay. As we know, usability evaluation involves testing of the usability of an interface by having a group of individuals performing tasks specific to a system, under the general guidance from a facilitator. It is important to realize that it has multiple components with five attributes associated with an interface [29,31,32]. Specifically, efficiency in our evaluation is measured in terms of how easily one can buy items and the speed of downloading them. Errors are regarded as any actions that prevent the successful occurrence of the expected results. Since some errors escalate the users' transaction time, their effect is measured by the efficiency of use. Learnability and satisfaction are a subjective measure provided by each participant in the experiment. Interface memorability is rarely tested as thoroughly as other attributes. However, it is feasible, to some extent, to conduct comparisons and post-test questionnaires of both systems.

The experiments use pre-test and post-test questionnaires. The questions of the pre-test questionnaire are about participants' experience in using mobiles or PCs to download files from the Internet. The post-test questionnaire has the number of questions with scale ratings ranging from one to five, where one is "least favourable" and five "most favourable". The post-test questionnaire also contains open questions for collecting user comments.

Subjects: Fourteen participants are randomly selected with a mixture of non-IT specialists, graduate students and college students. The participants are four non-IT adults, five non-IT graduate students, and the rest are college students. It should be noted that although it is tempting to recruit more participants, it is the general practice to have around 15 participants for usability testing [32].

Experimental tasks: Participants are required to complete the following tasks on M&E-NetPay and CORBA-based NetPay systems, respectively:

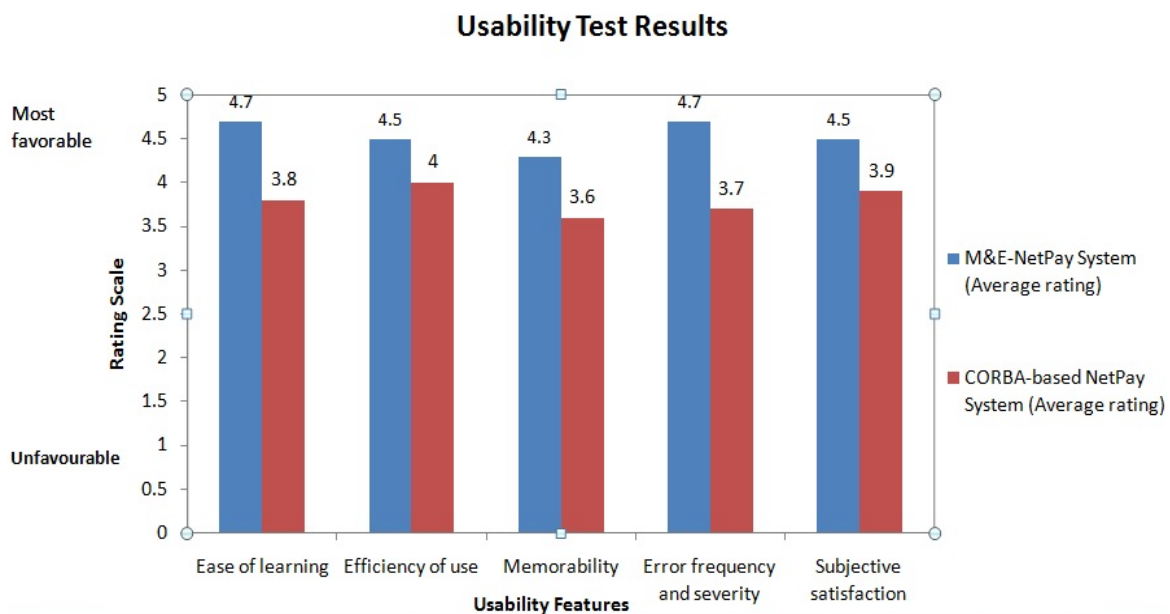


- Create an account with the broker;
- Search for a wallpaper on the wallpaper vendor site;
- Download the wallpaper from the wallpaper vendor site;
- Download a ringtone from the ringtone vendor;
- Buy e-coins from the broker; and
- Redeem e-coins with the broker.

Procedure: Before starting the test, participants need to fill out a pre-test questionnaire. Participants are required to carry out all of the tasks listed in a given sheet for the two systems. After finishing the tasks, they then fill out the post-test questionnaire to answer the questions by ticking one level of the rating. One of the questions asks the participants to rank the overall performance of the systems in order of their preference.

Results: From the answers to pre-test questions, we know that all of that participants have used mobiles or PCs to download files from the web weekly or monthly for free. Only two of them use the online credit-card payment systems to purchase goods online. Fortunately, all participants have had such experiences before. This implies that participants' prior knowledge has the least effect on the experiment results.

We survey the participants' satisfaction with buying e-coins, downloading wallpapers and their preference for the two systems. We analyse the post-test questionnaire outcomes and plot the results in Figure 9.



**Figure 9.** Usability test results with respect to usability features.

Figure 9 shows that the participants significantly favour all of the usability features of M&E-NetPay. With the user friendly interface, M&E-Pay is easy to learn, providing clear instructions on how to accomplish tasks. M&E-NetPay also receives high ratings on its efficiency. The participants comment that the speeds of downloading files (i.e., wallpaper and ringtone) are quite fast with M&E-NetPay in that with a few clicks, they are able to download the file. They also comment that appropriate pop-up error messages prevent them from going off of the right track. The overall average ratings of M&E-NetPay and CORBA-based NetPay are 4.5 and 3.8, respectively. They indicate that the participants prefer M&E-NetPay to CORBA-based NetPay. This fact results from employing new and emerging distributed middleware technique (i.e., Web Services) in M&E-NetPay.

For the open question, some participants write that since M&E-NetPay is available via both mobiles and the web, they will be able to access the system from anywhere at any time with barely any downtime. Twelve participants favour M&E-NetPay.

#### 5.4. Heuristic Evaluation

As the most widely-used inspection method, the heuristic evaluation technique is about identifying usability issues in a user interface by a small number of evaluators (usually one to five) who examine the interface and judge its compliance with usability principles (heuristics) [33–35]. While heuristic reviews are inexpensive and less time consuming, good ideas for improving a user interface may be produced.

**Subjects:** The evaluators include two IT specialists, one accountant and two graduates. They are experts in either software engineering or applied software fields.

**Experimental tasks and procedure:** the evaluators are requested to judge the compliance of the M&E-NetPay interface with usability principles (“the heuristic”). Each individual evaluator examines the interface independently. To aid the evaluators in discovering usability problems, a list of heuristics, as shown in Table 3 [35], is provided, which could facilitate the generation of ideas on how to improve the system.

**Table 3.** Details of the heuristics employed.

Number	Heuristic
1	Visibility
2	Functionality
3	User control and freedom
4	Consistency
5	Help recover from errors
6	Error prevention
7	Memorability
8	Flexibility
9	Aesthetic
10	Help and documentation

With a system checklist provided as a guide, the evaluators are required to first identify the heuristic problems of the interface and then to determine the levels of their seriousness by using the severity ratings as defined in Table 4 [35,36]. The evaluators are also requested to provide recommendations based on their assigned severity ratings.

**Table 4.** Severity of the heuristic evaluation.

Rank	Interpretation
1	Cosmetic problem only: need not be fixed unless extra time is available on a project
2	Minor usability problem: fixing this should be given a low priority
3	Major usability problem: important to fix, so should be given a high priority
4	Usability catastrophe: imperative to fix this before a product can be released

**Results:** The five evaluators evaluate M&E-NetPay by relying on the ten heuristics. The results of the heuristic evaluation are given in Table 5.

**Table 5.** Summary of the findings.

Number	Problem	Heuristic Number	No. of Evaluators	Severity Ranking
1	No sharp colour contrast between product information and its background.	1	2	2
2	No error message is displayed for invalid entries.	5	3	3
3	Multiple options cannot be selected in a menu or dialog box.	2, 8	1	2
4	Insufficient keyboard shortcuts for navigating the activity, function or action.	2	2	2
5	Exit button not provided to exit application from any screen.	2, 3	3	3
6	Not all integers and decimals right-justified.	4	2	1
7	The price associated with the product does not show the currency.	4	4	2
8	No sound used to signal an error.	5	2	1
9	No help topics provided.	10	2	3
10	Borders not used to identify meaningful groups.	1, 7	2	2
11	Titles are not provided on every page.	1, 4	3	2
12	On the login screen, the cursor is not active in the customer id field.	4	2	2

A rating has four levels of severity. The levels of one and two are regarded as minor, which is easily fixed. The levels of three and four should be given high priorities, which have to be fixed. After the evaluation, three major problems have been identified, with each having a severity rating of three. The identified problems, together with their fixing recommendations, are listed in Table 6. We have implemented all recommendations listed in the table.

**Table 6.** Summary of the findings.

Number	Problem	Recommendation	Severity Ranking	Heuristic Number
1	No error message is displayed for invalid entries	Appropriate error messages should be displayed for invalid entries	3	5
2	An exit button not provided to exit the application from any screen.	An exit button should be implemented on every screen	3	2, 3
3	No help topics provided	Implement help topics, as users may not be aware of the function of the menu or command button	3	10

We have described three kinds of evaluations on the M&E-NetPay prototype to assess performance impact, usability and heuristic evaluations. Usability and performance evaluation have been done on two prototypes of CORBA-based NetPay and M&E-NetPay. Even though heuristic evaluation identifies a few errors, M&E-NetPay is successfully implemented in general. The overall result of the evaluations demonstrates that most participants prefer M&E-NetPay. Participants and evaluators are satisfied very much with M&E-NetPay, recommending the system for wide use.

## 6. Related Work and Comparisons

In this section, we review related work and compare relevant systems to our system.

As a micro-payment for an ad hoc network, MPS [11] enables a node to join an existing ad hoc network and allows it to pay each node that relays packets on its behalf in real time. Being a lightweight payment scheme based on hash chains, MPS is flexible in route changes without involving a third party (a bank or a broker), in order to pay the nodes in a new path. MPS supports multiple brokers. Off-line verification makes the protocol more efficient and scalable.

Using a micro-payment protocol, CMP [12] is built on symmetry encryption techniques and chaotic double hash chains. The protocol constructs two PayWord chains: one for the merchant and another for users by using the iteration process of the Henon-like chaotic system. The chaotic hash function generates a payment chain. The use of the symmetric algorithm that encrypts transaction information improves the security and efficiency of CMP. CMP is an off-line system with three stakeholders, users, vendors and a broker.

As an off-line micro-payment system, NetPay [13] is a new micro-payment model in e-commerce. It uses CORBA interfaces to support communication between broker and vendor applications. NetPay improves its performance and security by using fast hashing functions. This prototype is quite suitable for e-commerce applications. In a mobile environment where a client (and possibly the server) keeps moving, which results in changing network addresses and unreliable connections, CORBA, however, is not well suited for this scenario. This is because of CORBA's tight coupling between clients and servers.

We compare our M&E-NetPay protocol to other micro-payment protocols. We have analysed the results from the three types of evaluations of M&E-NetPay prototypes to demonstrate their usability, performance and overall satisfaction of the requirements.

It is generally agreed that the key requirements for a mobile micro-payment system are as follows [3,8,11–13,29,30,37–39]:

- Security: The e-coins must be well encrypted to prevent peers from double spending and fraud.
- Anonymity: Peer users and peer vendors should not reveal their identities to each other or to any other third party.
- Ease of use: This is the ability of M&E users who are able to use the system easily without familiarizing themselves with the M&E user interfaces or being involved in any type of authentication at all times.
- Scalability: The load of communication and transaction of any entity must not grow to an unmanageable size. The load should be distributed among the vendors rather than the broker. Payment systems should be able to cater to the rapidly growing number of users without showing a negative impact on the performance.
- Transferability: The e-coins used for payments should be transferable between multiple vendors. This allows the users to use the same e-coins to make payments across multiple vendors.
- Interoperability: This is the ability of a system that operates in conjunction with other supporting protocols, hardware, software, applications and data layers. Interoperability minimizes the complexity of software development by reusing components and performing inter-component communication. Interoperable systems are language and platform independent.

In the following, we compare M&E-NetPay to several well-known micro-payment systems and also to some more recent micro-payment systems in M&E networks. The comparison criteria are the set of the key requirements: the need for an easy-to-use micro-payment system; the need for secure electronic coins and no double spending; ensuring anonymity for customers; supporting transferable e-coins between vendors; a robust, low performance impact, off-line micro-payment-supported, scalable architecture for a very large number of end users; and the ability of the system to be language and platform independent. Table 7 summarises the comparisons of the M&E-NetPay protocol with other systems.

**Table 7.** Comparison of M&E-NetPay with other micro-payment models. SOAP, Simple Object Access Protocol.

System/Property	MPS	CMP	NetPay	M&E NetPay
Security	Medium (Smart card devices cannot grantee multiple payment protection)	Very High (Uses a chaotic hash function to encrypt messages and services provided)	Very high (prevents users from over spending, prevents vendors from over charging and prevents a third party from forging e-coins)	Very High (Prevents M&E users from over spending, prevents vendors from over charging and prevents a third party from forging e-coins)
Privacy/anonymity	High (The nodes have no information about the user identity)	Very High (The vendor has no information about the user identity; all information services between vendors and users are encrypted)	High (The vendor has no information about the user identity)	High (The vendor has no information about the M&E user identity)
Ease of use	Medium (A lightweight payment scheme; however, too many private, public and sharing secret keys can slow the transaction)	Medium (The use of double hash chains and private and secret keys can slow the transaction)	High (Users need to spend little time in order to buy items from the vendor sites)	High (Provides simple interfaces with easy use; the Web Service that uses the SOAP and simple XML-based protocol makes the payment process faster)
Transferability	Low (Payment chains can only be transferred between the nodes in an ad hoc network; on a new network, a user needs to buy another payment chain from the new broker)	High (The same hash chain can be used on multiple vendor sites)	High (The e-coins can freely be transferred across multiple vendors for users to make multiple purchase)	Very High (The e-coins can freely be transferred across multiple vendors for M&E users to make multiple purchases; Web Service interfaces provide extra simplicity on transferring e-coins between vendors)
Scalability	Low (A change in route does not require contacting the broker; however, the model can only support the nodes in the ad hoc network)	Medium (CMP also requires the vendors to register with the broker if they wish to participate in the M&E commerce, and the use of too many keys can be costly in the future)	Medium (No or less communication with the broker and low volume transactions; however, the CORBA tight coupling between clients and servers in mobile environments makes it less scalable)	High (No or less communication with the broker and low volume information transferred; .NET applications such as a web service are fully accessible at any time on any platform and can support any number of M&E users)
Interoperability	Low (This model is only designed for a fixed network)	Low (Does not support multiple platforms and languages, as no appropriate middleware is specified)	Medium (Supports only a few platforms and languages)	Very High (Supports all platforms and languages)



The above comparisons show that the M&E-NetPay system has advantages over other micro-payment systems.

The security of M&E-NetPay is achieved by using existing security technologies. First, it uses a thin client n-tier architecture. With this deployment architecture, users logging on broker or vendor systems can access only Web servers. From there, transaction information is transferred through a secure channel in an XML message, which cannot be intercepted by a third party. Moreover, Web services on application servers are only available upon the request of mobile/web applications on Web servers. The vendors and broker in M&E-NetPay rely on Web service interfaces of the other party to exchange M&E user information. It is impossible for third parties to log directly or indirectly on to application servers. In addition, application servers are inaccessible from outside the network.

Second, M&E-NetPay relies on the security of Web services. As we know, Web services' security includes three aspects: authentication, which verifies that M&E users and vendors are who they claim to be; confidentiality and privacy, which keep information secret by encrypting the content of a message and obfuscating the sending and receiving parties' identities; and integrity and non-repudiation, which make sure that a message remains unaltered during transit by having the sender digitally sign the message. A digital signature is used to validate the signature and provides non-repudiation.

Finally, M&E-NetPay uses one-way hash functions to generate paywords and prevents M&E users and vendors from over spending and forging paywords from a payword chain. It employs 128-bit encryption of the messages. Since only the broker knows the mapping between the pseudonyms (IDc) and the true identity of an M&E user, M&E user privacy is protected.

In a word, M&E-NetPay has high security features. As an off-line fully-distributed system, the M&E-NetPay is mostly suitable for micro-payments over the WWW. In terms of transferability, e-coins are able to be transferred freely between vendors for multiple purchases. CMP is primarily designed for low value mobile commerce items. The protocol has greater security and faster operation efficiency, but CMP does not support multiple platforms and languages. MPS's design supports multiple brokers. Off-line verification has made the protocol more efficient and scalable. The system, however, still cannot avoid a limited amount of fraud. There may be a wastage of the broker endorsement, which is distributed to the previous path, if the topology of the ad hoc network changes.

M&E-NetPay has greater scalability and performance features, as it supports the rapidly growing number of M&E users. NetPay uses CORBA middleware interfaces to support several programming languages (e.g., Java<sup>®</sup> and C++<sup>®</sup>) and platforms (e.g., Windows<sup>®</sup>, Linux<sup>®</sup>). Vendor systems have to be "hard-coded" with CORBA by communicating with the NetPay broker to exchange messages. In comparison with M&E-NetPay, NetPay has a lesser ability due to the tight coupling between clients and servers as a result of its use of CORBA. M&E-NetPay is the solution to the above problem, as it can support any languages and platforms. Hence, it has a very high rating of interoperability.

## 7. Conclusions

In this paper, we have presented the design and implementation of the M&E-NetPay micro-payment system on the .NET platform and its comprehensive evaluations. Interconnecting broker and vendor sites through XML-based interactions, Web services are used to provide greater interoperability than the CORBA middleware. Apart from generating, redeeming and verifying e-coins, the broker in M&E-NetPay provides e-wallets to customers. Through their interfaces, vendor applications allow users to browse their sites, download items and obtain the valid touchstone and index from a broker or the previous vendor. M&E-NetPay uses a secure, inexpensive and debit-based off-line protocol that allows vendors to interact only with customers after an initial validation of coins. M&E-NetPay achieves a secure and high transaction volume per item by using fast hashing functions that validate e-coin unspent indexes. The results of two types of comparison evaluations on the usability and performance of two systems have demonstrated that the users, as their preference, would adopt M&E-NetPay for widespread use. In the future, we will take further advantage of Web services [40] to generalize the proposed architecture as components of a wider range of M&E-commerce applications.

**Acknowledgments:** This work was partially supported by the National Natural Science Foundation of China under Grant No. 61370229 and the National Key Technology R&D Program of China under Grant No. 2014BAH28F02, the S & T Projects of Guangdong Province under Grant No. 2014B010103004, 2014B010117007, 2015A030401087, and 2016B010109008, and GDUPS (2015).

**Author Contributions:** X.H., X.D. and J.B. proposed the idea of this work and designed the system. E.S. implemented it. All the authors contributed to evaluating this system and writing this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Adi, W.; Al-Qayedi, A.; Zarooni, A.A.; Mabrouk, A. Secured multi-identity mobile infrastructure and offline mobile-assisted micro-payment application. In Proceedings of the IEEE International Conference of Wireless Communications and Networking, 21–25 March 2004; pp. 879–882.
2. Bokai, M.; Mohammadi, S. Exploring adoption of NetPay micro-payment: A simulation approach. In Proceedings of the 2010 IEEE International Conference on Educational and Information Technology (ICEIT), Chongqing, China, 17–19 September 2010; pp. 245–250.
3. Furche, A.; Wrightson, G. SubScrip—An efficient protocol for pay-per-view payments on the Internet. In Proceedings of the 5th Annual International Conference on Computer Communications and Networks, Rockville, MD, USA, 16 October 1996.
4. Hwang, M.; Lin, I.; Li, L. A simple micro-payment scheme. *J. Syst. Softw.* **2001**, *55*, 221–229.
5. Isern-Deyà, A.; Payeras-Capellà, M.; Macià Mut-Puigserver, M.; Ferrer-Gomila, J. Untraceable, anonymous and fair micropayment scheme. In Proceedings of the 9th ACM International Conference on Advances in Mobile Computing and Multimedia, Ho Chi Minh City, Vietnam, 5–7 December 2011; pp. 42–48.
6. Park, D.; Boyd, C.; Dawson, E. Micro-Payments for Wireless Communications. In *Security and Cryptology*; Lecture Notes in Computer Science 2015; Springer: Berlin/Heidelberg, Germany, 2001; pp. 192–205.
7. Pedersen, T. Electronic payments of small amounts. In *Security Protocols*; Lomas, M., Ed.; Lecture Notes in Computer Science No. 1189; Springer Verlag: Berlin, Germany, 1997; pp. 59–68.
8. Reddy, S.; Estrin, D.; Hansen, M.; Srivastava, M. Examining micro-payments for participatory sensing data collections. In Proceedings of the 12th ACM International Conference on Ubiquitous Computing, Copenhagen, Denmark, 26–29 September 2010; pp. 33–36.
9. Rivest, R.; Shamir, A. PayWord and MicroMint Two Simple Micropayment Schemes. In *Security Protocols*; Lecture Notes in Computer Science v. 1189; Springer: Berlin/Heidelberg, Germany, 1997; pp. 69–87.
10. Shin, D.; Jun, M. Micro Payment System Using OTP for Customer’s Anonymous. In Proceedings of the IEEE 2011 International Conference on Information Science and Applications (ICISA), Jeju Island, Korea, 26–29 April 2011; pp. 1–5.
11. Tewari, H.; O’Mahony, D. Multiparty Micropayment for Ad Hoc Network. In Proceedings of the Wireless Communications and Networking, 2003 (WCNC 2003), New Orleans, LA, USA, 20 March 2003; Volume 3, pp. 2033–2044.
12. Ziang, N.; Liu, X.; Zhao, J.; Yang, D. A Mobile Micropayment Protocol Based on Chaos. In Proceedings of the 2009 Eighth International Conference on Mobile Business, Dalian, China, 27–28 June 2009; pp. 284–289.
13. Dai, X.; Grundy, J. Net Pay: An off-line, decentralized micro-payment system for thin-client applications. *Electron. Commer. Res. Appl.* **2007**, *6*, 91–101. [[CrossRef](#)]
14. Chen, L.; Li, X.; Shi, M. A Novel Micro-payment Scheme for M-commerce based on Self-Renewal Hash Chains. In Proceedings of the IEEE International Conference on Communications, Circuits and Systems, Kokura, Japan, 11–13 July 2007; pp. 1343–1346.
15. Yohan, A.; Lo, N.W.; Randy, V.; Chen, S.J.; Hsu, M.Y. A Novel Authentication Protocol for Micropayment with Wearable Devices. In Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication, Danang, Vietnam, 4–6 January 2016; pp. 1–7.
16. Veeraraghavan, P.; Almuairfi, S.; Chilamkurti, N. Anonymous paperless secure payment system using clouds. *J. Supercomput.* **2016**, *72*, 1813–1824. [[CrossRef](#)]
17. Gisolfi, D. Web Services Architect, Part 3: Is Web Services the Reincarnation of CORBA? 2010. Available online: <http://www.ibm.com/developerworks/webservices/library/ws-arc3> (accessed on 20 May 2016).

18. 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, NV, USA, 24–27 June 2002; pp. 444–450.
19. Rowley, J. An analysis of the e-service literature: Towards a research agenda. *Int. Res.* **2006**, *16*, 339–359. [[CrossRef](#)]
20. Sumak, B.; Polancic, G.; Hericko, M. Towards an e-service knowledge system for improving the quality and adoption of e-services. In Proceedings of the 22nd Bled eEnablement: Facilitating an Open, Effective and Representative Society, Bled, Slovenia, 14–17 June 2009.
21. Barnes, S.J.; Vidgen, R.T. An integrative approach to the assessment of ecommerce quality. *J. Electron. Commer. Res.* **2002**, *3*, 114–127.
22. Behkamal, B.; Kahani, M.; Akbari, M.K. Customizing ISO 9126 quality model for evaluation of B2B applications. *Inf. Softw. Technol.* **2009**, *51*, 599–609. [[CrossRef](#)]
23. Parasuraman, A.; Zeithaml, V.A.; Malhotra, A. E-S-QUAL: A multiple-item scale for assessing electronic service quality. *J. Serv. Res.* **2005**, *7*, 213–233. [[CrossRef](#)]
24. Schubert, P. Extended web assessment method (EWAM)—Evaluation of electronic commerce applications from the customer’s viewpoint. *Int. J. Electron. Commer.* **2003**, *7*, 51–80.
25. Janda, S.; Trocchia, P.J.; Gwinner, K.P. Consumer perceptions of Internet retail service quality. *Int. J. Serv. Ind. Manag.* **2002**, *13*, 412–431. [[CrossRef](#)]
26. CodeGuru the .NET Architecture, Published by Wiley, September 21, 2004. Available online: [http://www.codeguru.com/csharp/sample\\_chapter/article.php/c8245\\_\\_1/The-NET-Architecture.htm](http://www.codeguru.com/csharp/sample_chapter/article.php/c8245__1/The-NET-Architecture.htm) (accessed on 20 May 2016).
27. Gray, N. Comparison of Web Services, Java-RMI, and CORBA service implementations, Fifth Australasian Workshop on Software and System Architectures. In Proceedings of the Conjunction with ASWEC 2004, Melbourne, Australia, 13 April 2004.
28. Dai, X.; Grundy, J. Three Kinds of E-wallets for a NetPay Micro-payment System. In Proceedings of the Fifth International Conference on Web Information Systems Engineering, Brisbane, Australia, 22–24 November 2004; pp. 66–77.
29. Huang, X.; Dai, X.; Liang, W. BulaPay: A Novel Web Service based Third-Party Payment System for e-Commerce. *Electron. Commer. Res.* **2014**, *14*, 611–633. [[CrossRef](#)]
30. Yang, C.; Qi, M. Scheme and applications of mobile payment based on 3-D security protocol. In Proceedings of the 3rd ACM International Conference on Mobile Technology, Applications & Systems, Bangkok, Thailand, 25–27 October 2006; pp. 1–4.
31. Dumas, S.J.; Redish, J.C. *A Practical Guide to Usability Testing*; Ablex Publishing Corporation: Norwood, NJ, USA, 1993.
32. Hwang, W.; Salvendy, G. Number of people required for usability evaluation: The  $10 \pm 2$  rule. *Commun. ACM* **2010**, *53*, 130–133. [[CrossRef](#)]
33. Nielsen, J. Heuristic evaluation. In *Usability Inspection Methods*; Nielsen, J., Mack, R.L., Eds.; John Wiley & Sons: New York, NY, USA, 1994.
34. Cao, H.; Wang, L.; Zhu, J. A Trust-Aware Mobile Multiple Micro-Payment Mechanism Based on Smart Agent in Distributed Environment. In Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Applications, ICPCA 2007, Birmingham, UK, 26–27 July 2007; pp. 314–318.
35. Pilla, P. Heuristic Evaluation Report, Heuristic Evaluation of HOMIE, 2009. Available online: <http://www.ccs.neu.edu/home/prasu14/hci/i8/HomieReport.pdf> (accessed on 12 May 2015).
36. Nielsen, J. Severity Ratings for Usability Problems, Papers and Essays, 1995. Available online: <http://www.useit.com/papers/heuristic/severityrating.html> (accessed on 21 April 2016).
37. Mallon, D. *mCommerce Security White Paper Key Security Techniques—mCommerce Security White Paper Version 1.1*; Sybase: Dublin, CA, USA, 2010.
38. Williams, L.; Smith, C. Performance Evaluation of Software Architectures. In Proceedings of the Performance Engineering Services and Software Engineering Research, Buenos Aires, Argentina, 5–8 February 2007.

39. Yang, C.; Wu, C.; Chiu, C.; Chiou, S.; Liao, W. Micropayment schemes with ability to return changes. In Proceedings of the 11th ACM International Conference on Information Integration and Web-based Applications & Services, Kuala Lumpur, Malaysia, 14–16 December 2009; pp. 356–363.
40. Huang, X.D. UsageQoS: Estimating the QoS of Web Services through Online User Communities. *ACM Trans. Web* **2013**, *8*, 1–31. [[CrossRef](#)]



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).