Architecture of CORBA based P2P-Netpay Micro-payment System in Peer to Peer Networks

Kaylash Chaudhary and Xiaoling Dai

School of Computing, Information and Mathematics Science
The University of the South Pacific, Laucala Campus, Suva, Fiji
chaudhary_k@usp.ac.fj [1], dai_s@usp.ac.fj [2]

Abstract: Micro-payment systems have the potential to provide non-intrusive, high-volume and low-cost pay-as-you-use services for a wide variety of web-based applications. We proposed a new model, P2P-NetPay, a micro-payment protocol characterized by off-line processing, suitable for peer-to-peer network service charging. P2P micro-payment systems must provide a secure, highly efficient, flexible, usable and reliable environment, the key issues in P2P micro-payment systems development. Therefore, in order to assist in the design and implementation of an efficient micro-payment system suitable for P2P networks, we describe prototype architecture for a new CORBA based micro-payment model. We present an object-oriented design and describe a prototype implementation of P2P-NetPay for file-sharing P2P system. We compare socket, CORBA, RMI and web-service-based P2P-NetPay prototypes and outline directions for future research in P2P micro-payment implementations.

***Keywords*:** Micro-payment system, P2P-NetPay, CORBA, Web services

1. Introduction

File sharing in peer to peer networks has been an important issue in past years. Many file sharing application were developed for peers to exchange content but this introduced the problem of “free-rider” within the network where peers only consume services rather than provide [1]. Micro-payment system was introduced to liberate “free-rider” problems and escalate efficient use of the peer to peer network. Various micropayment system were established which suffered from the problems of lack of scalability, dependence on online brokers, communication overhead and security [9, 10, 11, 12]. We proposed a new micropayment system, P2P-Netpay [6], which addressed afore mentioned distress for payment system.

Software architecture is the structure of the system, which comprises of software components, the externally visible properties of those components, and the relationship between those components. It is a path for communication among the elements which captures early design decisions for systems. Software architecture plays an important role in the system development. P2P-Netpay software architecture design should be scalable, reliable, secure and flexible.

This paper focuses on architecture of CORBA based P2P-Netpay micropayment system implementing Broker and peers using CORBA. Earlier implementation of P2P-Netpay used java sockets for peer communication [5]. The design and implementation of CORBA system will be discussed.

Web Service is an emerging standard for business to business (B2B) communication with different platforms to handle wide variety of clients [2]. We have also developed our Broker using web service which provides payment service to peers. The peers are implemented using either CORBA or socket for communication between them but they consume services offered by Broker using web service.

The four architectural styles (CORBA, socket, Remote Method Invocation (RMI) and web service) for P2P-Netpay are discussed in this paper with the comparison amongst them. We outline our plans for further research and development in P2P micro-payment system.

2. Motivation

In any software development, architecture plays an important role in achieving design and business goals, quality solution and reusable or extendable solution [3]. One such situation is based on P2P-Netpay protocol implemented via CORBA for peer to broker communication and sockets for peer to peer communication [4, 5].

The Application Programming Interface (API) for socket programming is low-level since it creates communication overheads while two peer applications communicate for transferring of files. It was noted during design/implementation of P2P-Netpay that the programmer was responsible for method of communication. For example, in P2P-Netpay if a peer user wants to send ecoins to peer vendor to buy files, a peer user must establish a socket connection with peer vendor and write to socket while peer vendor reads from a socket. Thus, sockets require more instructions to be executed each time a message is sent or received.

When a file is transferred from peer vendor to peer user, it is read in portions and those small portions are sent to peer user while others are being read. Therefore, a series of messages are sent and received during file transfer. Series of messages sometimes leads to download being aborted. This is due to connection timeout. Each time a peer needs to communicate with other peer, a connection must be established and it must be terminated when finished.

Socket has certain advantages in network programming. Due to P2P-Netpay [6] protocol requirements, every file download deals with ecoins therefore a proper architecture is needed for providing service to peer user.

To unfetter, above concerns using socket, three other methods were considered for development which are RMI, CORBA and Web Services.

3. Overview of P2P-Netpay

P2P-Netpay, which is a basic offline protocol suitable for micro-payments in peer to peer networks is discussed in [4, 5, 6]. Below are some of the briefly discussed micropayment terminologies:

- ***Payword Chain*** – A payword chain is represents a set of E-coins in the P2P-Netpay system.
- ***E-coin*** – An "e-coin" is a payword element and the value of a payword e-coin might be one-cent but could be some other value.
- ***E-wallet*** – An "e-wallet" is used to store e-coins and send e-coins to a vendor paying for information goods.
- ***Touchstone*** – A "touchstone" is used to verify the e-coins whether the ecoins are valid or not.
- ***Index*** – An "index" is used to indicate the current spent amount of each e-coin (payword) chain. For example if you have spent 2cs to buy an information goods, the current index value is 3.

The customer registers and purchases some e-coins (using macro-payment such as credit card) with the Broker/CIS site. Peer-users buy e-coins from Broker/CIS which is sent to peer-user's e-wallet. When buying items from peer-vendor, the peer-user sends e-coins from e-wallet. The peer-vendor verifies the e-coin and allows peer-user to download file. At the end of the day, peer-vendor can redeem e-coins with Broker/CIS for real money. When a peer-user first tries to spend an e-coin the peer-vendor communicates with the Broker/CIS to obtain a validating touchstone for the coin.

Each e-coin encodes a "payword chain" which utilizes a fast hashing function to provide the next valid coin in the chain each time a coin is spent. When a peer-user downloads a file from another peer-vendor, the new peer-vendor obtains the touchstone and index from previous peer-vendor. If the previous peer-vendor is offline then the new peer-vendor contacts Broker/CIS for touchstone and index.

The transfer of e-coins from Broker/CIS to peers is secured by public key encryption. An index is used to indicate the amount of e-coin spent so far which prevents peer-users from double spending and peer-vendors from over debiting [7]. The peer-user and peer-vendor does not reveal identities to any third party or each other. Only the secure Broker/CIS can identify the participants in a particular transaction. In P2P-Netpay, the peer-user needs to contact the Broker/CIS to buy e-coins when e-coins run out.

4. P2P-Netpay Architecture

We have developed a revised software architecture for implementing CORBA based P2P-Netpay micro-payment systems for content sharing in peer-to-peer networks. The interaction between Broker/Central Index Server (CIS), Peer User and Peer Vendor is illustrated in Figure 1.

CORBA is a standard developed by the Object Management Group (OMG) to support distributed object computing [13] which uses Broker/CIS as an intermediary to handle request in a system and separates components interface from its implementation. CORBA is not a language, it is middleware platform. It provides infrastructure for the programming of distributed systems using C, C++, Smalltalk, Ada and Java. Object Request Broker (ORB) forwards operations on objects to the desired object and returns results to the client. These inter-ORB interactions authorize a peer user to communicate with peer vendor or peer user/vendor to communicate with Broker or vice versa.

The CORBA Interface Definition Language (IDL) describes the objects together with their methods and attributes which is independent from the languages in which these interfaces will actually be implemented. There are two IDL files in P2P-Netpay, one for implementation of Broker/CIS operations and other for peer operation. Peers use the Broker/CIS operation to buy ecoins, request Touchstone and redeem ecoins for real money whereas peer operation IDL is used for downloading files from peers. Broker/CIS operation IDL is compiled to generate code for both the peer and the Broker/CIS. The generated code for peers is in the form of object stubs. From the peer's perspective these stubs are function calls directly into the object. In actuality the stubs forward the request to the remote object via the ORB. The compiler also generates skeleton code for the Broker/CIS (typically referred to as a servant in CORBA), which needs to be fleshed-out with the implementation of the requested operations. The same happens for peer operation IDL file.

The P2P-Netpay Broker/CIS system is built on top of the multi-tier web-based architecture presented as follows:

Client tier (HTML Browser): The browser communicates with the web server which runs the JSPs to register peers.

- Web tier (Broker/CIS Web Server and JSPs): In the web tier in the systems, Java Server Pages (JSPs) and JavaBeans are used to service the web browser clients, process request from the clients and generate dynamic content from them. After receiving the client request, the JSPs request information from a JavaBean. The JavaBean can in turn request information from an application server (CORBA). Once the JavaBean generates content the JSPs can query and display the Bean's content. The Broker/CIS keeps track of online peers. It also stores the file names with the host and port of peers. Broker/CIS is designed using CORBA in Java so that it can serve multiple clients at one time.

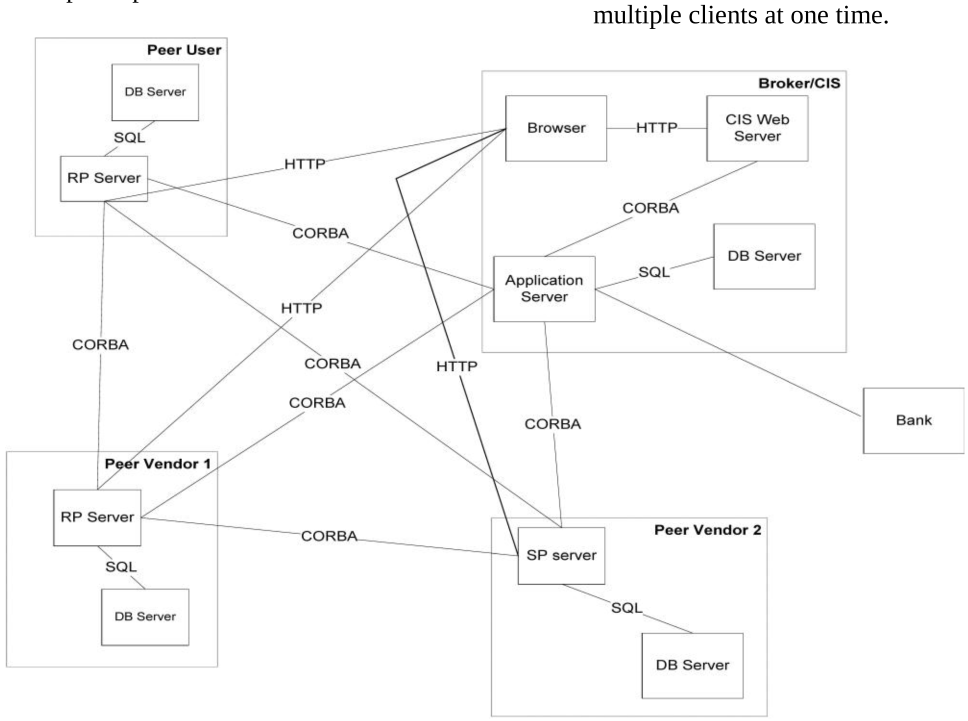

Figure 1: P2P-Netpay interactions

The peer user registers with Broker/CIS site using Java Server Pages (JSP) in web server which communicates through application server using CORBA to register and allow software download. When buying a file, ecoins are transferred using CORBA from peer user to Peer Vendor 1 which in turn requests Touchstone from either Peer Vendor 2 or Broker/CIS depending on where peer user has spent that ecoin before this transaction.

- Application Server tier (CORBA): In the P2P-Netpay Broker/CIS, CORBA is used as the middleware for the application server, which is implemented in the Java language that has a CORBA IDL mapping.
- Database Server tier: On the back-end of the system we use Ms Access to implement the databases accessed via a Java Database Connectivity (JDBC) interface. JDBC, which is a multi-database application programming

interface, provides Java applications with a way to connect to and use relational databases. When a Java application interacts with a database, JDBC can be used to open a connection to the database and SQL code is sent to the database.

The peer system which is either user or vendor is a three tier architecture which includes:

- Client tier (Client Application): This client application communicates with peer server to get requests from other connected peers or CIS/Broker.
- Peer Server tier (Requesting peer server/ Supplying peer server): It is implemented in Java to handle the functionalities such as sharing files, communicating with server, checking balance, redeeming, searching Broker/CIS and browsing peers. This server also uses CORBA IDL mapping for communication between peers.
- Database Server tier: We use Ms Access to implement the database accessed via JDBC. Only the Java application can interact with the database. Peers cannot open the database manually and edit any data since this database is password protected.

5. P2P-Netpay Implementation

The three main functionality of P2P-Netpay are buying ecoins, downloading file and redeem spending.

- ***Buy Ecoins***

Figure 2 shows how a peer buys e-coin in the P2P-Netpay system. The peer checks the amount left in the e-wallet and if wishing to buy e-coin, peer enters the amount and clicks buy button. The client application requests the e-coins through CORBA to Broker/CIS application server. The Broker/CIS application server debits from the peer's credit card, stores e-coins in the database and sends an e-wallet to java application (client application) in peer's computer.

- ***Downloading File***

Figure 3 shows how a peer-user downloads file using P2P-Netpay micro-payment system. After browsing peers or searching the Broker/CIS, peer-user clicks on download popup menu on the title of the file name. The client application sends the request including file name and e-coins through CORBA interface to peer-vendor server (PVS). If the touchstone and index does not exist in its database, the server contacts other Peer Vendor Server (PVS) or Broker/CIS in order to obtain touchstone and index. If the e-coins are valid, the PVS stores in redeem database and sends the file to peer-user. Peer-user server (PUS) than debits the e-coin.

- ***Redeem Spending***

Figure 4 illustrates how a peer-vendor redeems spent e-coins with Broker/CIS. When redeeming spent e-coins, peer-user clicks on redeem button on the client application where the PVS aggregates all payments and sends to Broker/CIS application server using CORBA interface where it verifies spent e-coins and sends the balance to peer-user.

The basic user interfaces are presented in Figure 5. A peer can be a user or a vendor. To use the services of P2P-Netpay, a peer must be registered with the broker and download the application. A peer must remember the Peer ID generated by the system to login as described in Figure 5 (1). For the initial login the peer server connects to Broker/CIS server to get the Peer ID and password using CORBA after which it is stored in the local database. If the login is successful, the main interface of the application appears as in Figure 5 (2). The IP address and port is listed for the peers who are currently online. These IP and port is not to which peer is listening to as in sockets but its the *RemotePeerManagerServer* which provides peers to invoke methods remotely. CORBA has its own servers running to which peers can connect and use the services. There are much functionality provided such as checking balance, searching file, uploading file and redeeming.

Suppose a peer user would like to search for file named "ewallet", user enters the name in (3) and the results are displayed in (4). The results are obtained from Broker/CIS application server. It will only display results of the peer who is currently online. To download a file, right click on the file name and select download from popup menu. The application will connect to peer hosting that particular file using CORBA and download starts after ecoin verification.

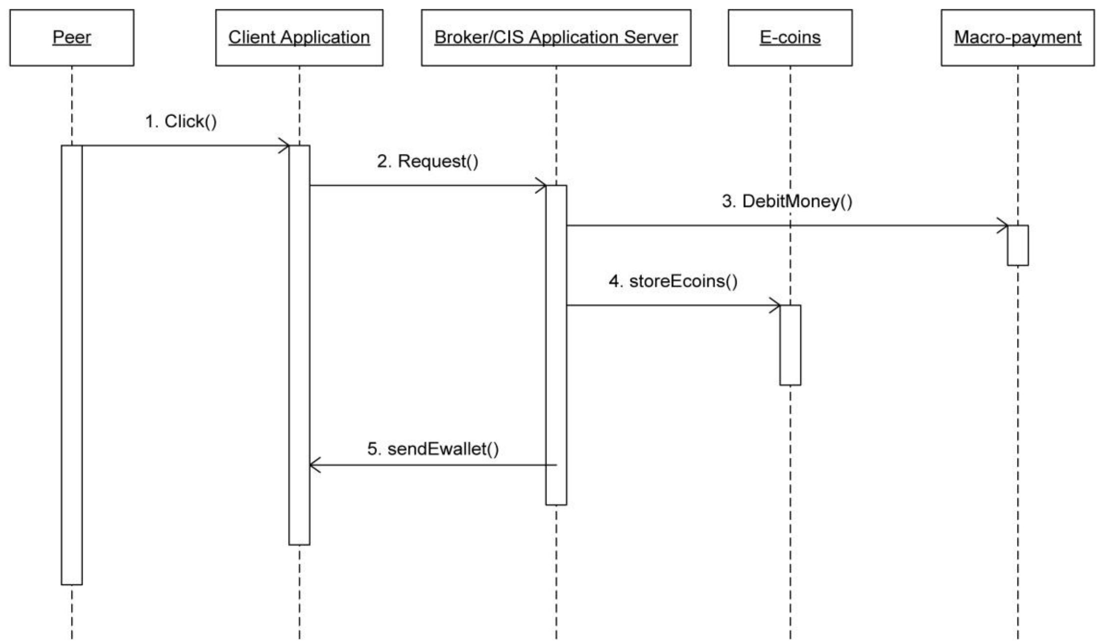

Figure 2: Buy ecoin sequence diagram

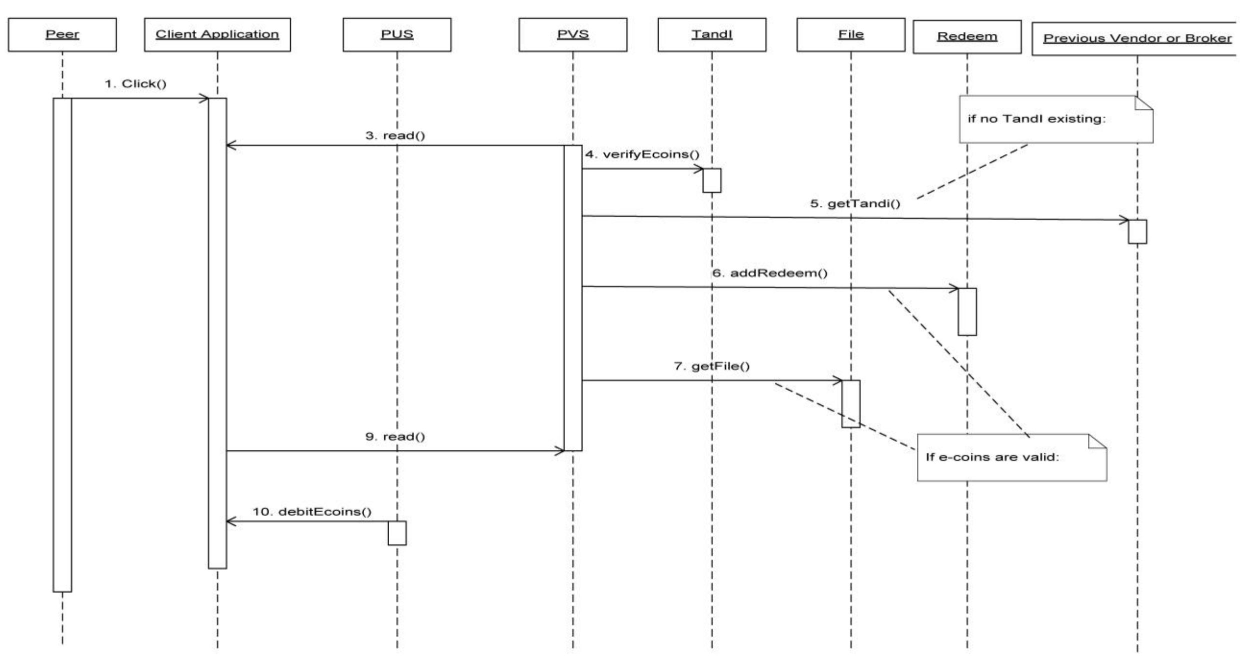

Figure 3: Download file sequence diagram

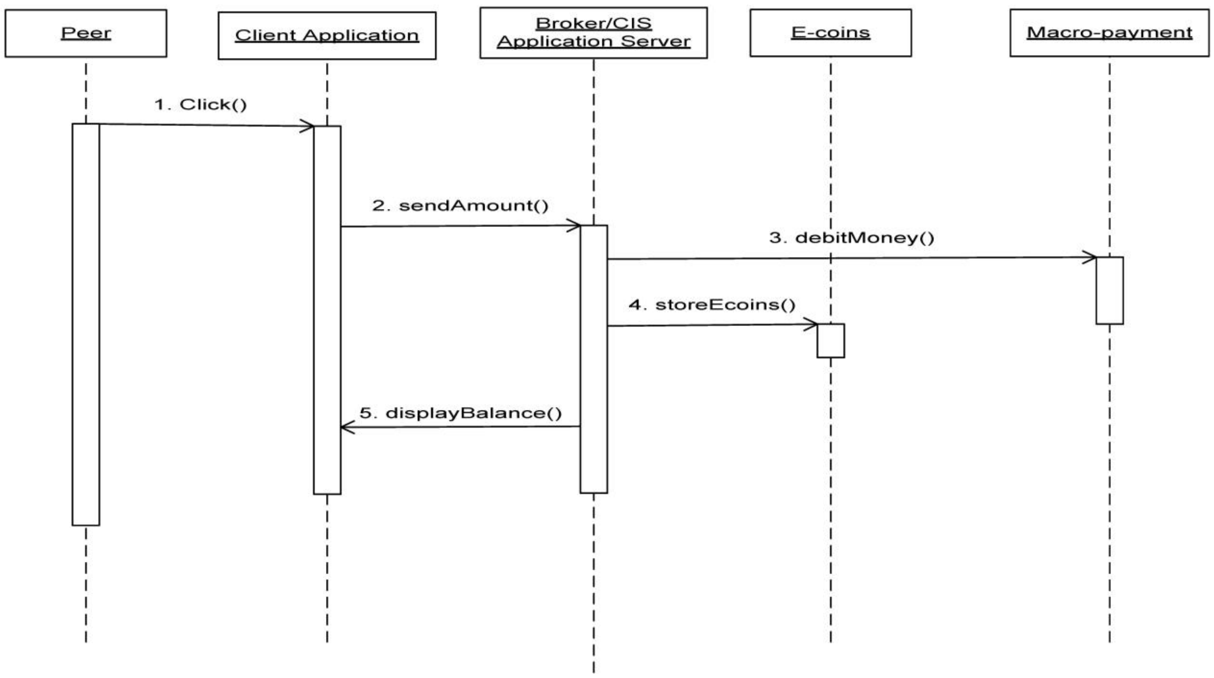

Figure 4: Redeem spending sequence diagram

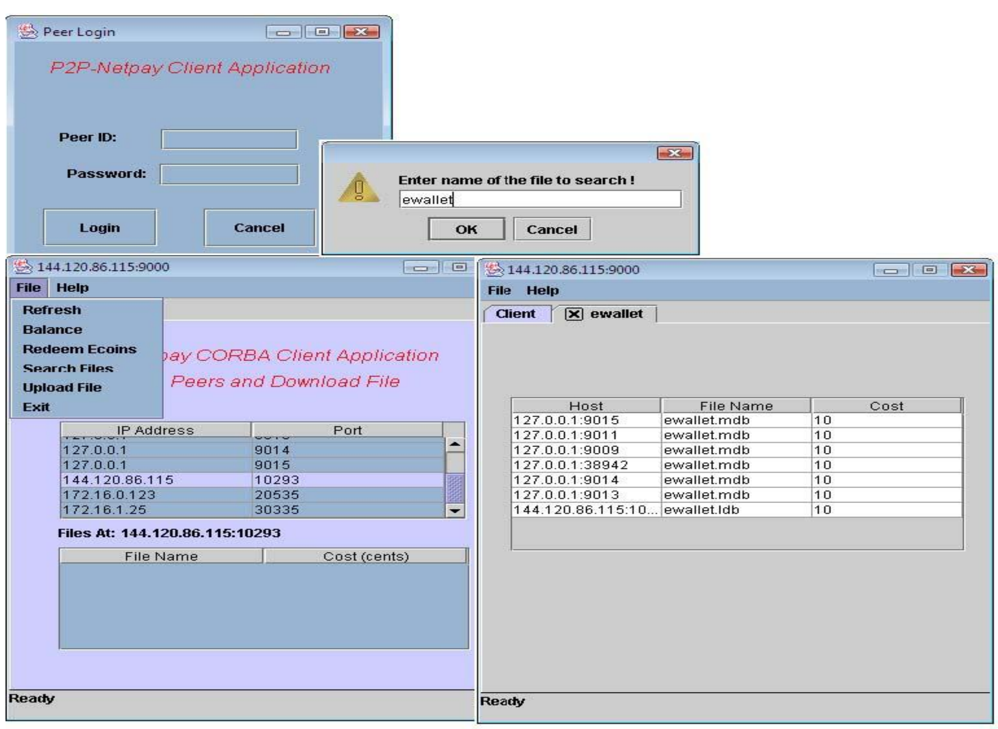

Figure 5: P2P-Netpay user interfaces

6. Comparison of P2P-Netpay Architectures

Different architectures have been investigated for P2P-Netpay peers. Those are CORBA, socket, Web Services and RMI. Architectures have certain advantages and disadvantages. The advantages/disadvantages are in context of P2P-Netpay and not in general. Currently, Web Service is another option for distributed computing infrastructure [3] but the benefits of CORBA outweigh web services such as it supports multiple programming languages, it is a platform middleware, it interoperates with other middleware, it is highly flexible and it supports Remote Procedure Call and message-passing paradigms [2].

Web Services when compared to CORBA also supports multiple programming languages, is designed for web only, supports operation in heterogeneous connection at ends and uses XML to define interfaces and format messages [2].

Table 1 summarizes architecture comparison using five criteria's described below:

- ***Easy to use***: less coding required in achieving goals.
- ***Programming language*:** Some architectures support multiple programming languages while others not. The benefit is different systems implemented via different programming languages can communicate with each other. Why implementation using different languages and not the same? This is to suite other users comfort ability in coding or using a particular language.
- ***Platform*:** Much architecture doesn't support multiple platforms. There may be a situation that a server may be running on different machines and clients on another. To make these clients and servers to communicate, architecture must be platform independent.
- ***File downloading*:** files broken into parts and then sent to peers for reassembling using previous architecture.
- ***Processing time*:** time taken to process requests. The processing times are summarized in [8] and the results presented in the following table for time is based on [8].

Table 1: P2P-Netpay architecture comparison

Property	***Socket***	***Web Services***	***RMI***	***CORBA***
Easy to use	**Low**, more codes required to handle series of messages for downloading file	**Medium**, messages are sent as SOAP; codes are required to build and read SOAP message	**High,** less to code	**High**, functionality is achieved with less coding
Programming language	**No** support for multiple programming language	**High**, supports multiple programming language;	**No** support for multiple programming language	**High,** implementation for either Broker/CIS or peer may differ. Broker/CIS may be implemented in Java whereas client can be implemented as C++
Platform	**No** support for multiple platform	**No** support for multiple platform	**No** support for multiple platform	**High**, currently P2P-Netpay has only windows based application; there is a possibility for different platform applications.
File downloading	**Low**, breaking down of file into parts and sent as series of messages which results in download being aborted	- Very difficult impossible to implement peers as Web services - uses HTTP for communication - Each peer cannot host files on web servers and keep on updating since peers needs to have knowledge about HTML, JSP etc.	**High**, a remote method is invoked – no breaking down of file into parts	**High**, a remote method is invoked – no breaking down of file into parts
Processing time	**Less** processing time	**Medium**	**Less** processing time	**Medium**, since it supports distributed system.

The above comparison shows that CORBA architecture is recommended for P2P-Netpay. Socket is very simple to understand and program. Peer/server must listen to a port in order to communicate and have to cater for each message sent and received. Sometimes the connection times out which aborts downloading requiring establishing connection again. Web services, on the other hand, is a new technology which uses XML to define messages sent to peers by Broker/CIS. Only the Broker/CIS can be implemented as Web service which has three-tier architecture when compared to multi-tier architecture for CORBA.

RMI is very similar to CORBA but it doesn't support multiple programming languages and platforms. The processing time for CORBA is higher when compared to RMI and sockets as in [8]. This is because the number of servers increase the processing time increases. In P2P-Netpay, only two servers will communicate so the processing time will be equivalent to socket and RMI.

7. Summary

We have developed a revised prototype architecture to support an efficient, secure and anonymous micro-payment system for high-volume, low-cost file sharing system. This incorporates a Broker/CIS which is used to generate, verify and redeem e-coins, a peer e-wallet stored on peer machine and peer application server components. All communication between peers and Broker/CIS – peer is through CORBA interface. Our initial prototype used CORBA architecture for Broker/CIS and socket for peers. Due to disadvantages of socket and in turn advantages of CORBA resulted in a new architecture for P2P-Netpay. We also compared four architectures for P2P-Netpay and decided that CORBA is more suitable architecture since P2P-Netpay is more likely to expand in future. Currently, we are investigating on web based file sharing application using P2P-Netpay payment service.

8. References

[1] Adar, E. and Huberman, B.: Free Riding on Gnutella, First Monday, 5(10), (2000)

[2] Baker, S., "Web Services and CORBA" *Lecture Notes in Computing Science*, vol. 2519, 2010, pp. 618-632. Publisher: Springer Berlin / Heidelberg

[3] CA, B., Barai, M. and Caselli, V., "Service Oriented Architecture with Java", Packt Publishing Ltd, 2008

[4] Chaudhary, K., Dai, X. & Grundy, J., "Experiences in Developing a Micro-payment System for Peer-to-Peer Networks", *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 5, no.1, March 2010, pp. 23 – 42. Publisher: IGI Global

[5] Chaudhary, K. & Dai, X., "P2P-NetPay: An Off-line Micro-payment System for Content Sharing in P2P-Networks", *Journal of Emerging Technologies in Web Intelligence (JETWI)*, vol.1, no.1, August 2009, pp. 46 - 54. Publisher: Academy Publisher.

[6] Dai, X., Chaudhary, K. and Grundy, J.: "Comparing and Contrasting Micro-payment Models for Content Sharing in P2P Networks", *Third International IEEE Conference on Signal-Image technologies and Internet-Based System (SITIS'07)*, 16 - 19 December 2007, Published by IEEE Computer Society, pp. 347-354

[7] Dai, X. and Grundy, J.: "Off-line Micro-payment System for Content Sharing in P2P Networks", *2nd International Conference on Distributed Computing & Internet Technology (ICDCIT 2005)* ,December 22-24, 2005, Lecture Notes in Computer Science Vol. 3816, pp 297 –307

[8] Eggen, R. and Eggen, M., "Effciency of Distributed Parallel Proceeing using Java RMI, Sockets, and CORBA", www.imamu.edu.sa/dcontent/IT_Topics/java/paper3.pdf

[9] Wei, K., Smith, A. J., Chen, Y. R. and Vo, B.(2006), "WhoPay: A scalable and anonymous payment system for peer-to-peer environments", in *Proc. 26th IEEE Intl. Conf. on Distributed Computing Systems, Los Alamitos, CA*, Computer Society Press, 2006, pp. 13-23.

[10] Yang, B. and Garcia-Molina, H.(2003), "PPay: micropayments for peer-to-peer systems", in *proc. Of the 10th ACM conference on computer and communication security*, ACM press, 2003, pp. 300- 310.

[11] Zghaibeh, M. and Harmantzis, F.C.(2006), "Lottery-based Pricing Scheme for Peer to Peer Networks"**,** *ICC apos;06. IEEE International Conference on Communications, 2006,* Volume 2, June 2006, pp. 903 – 908.

[12] Zou, E. J., Si, T. , Huang, L. and Dai, Y. (2005), "A New Micro-payment Protocol Based on P2P Networks", *Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05),*IEEE Computer Society Press, 2005, pp. 449 – 455.

[13] OMG's CORBA : http://www.corba.org/

Author Biographies

Kaylash Chaudhary received his MSc degree in Computing Science from University of the South Pacific in Fiji in 2009. He is now an assistant lecturer in the School of Computing, Information & Mathematical Sciences, University of the South Pacific. His research interests include software engineering, distributed system design and implementation, software architecture, electronic micro-payment systems for file-sharing, in peer-to-peer networks, mash-ups, and service-orientated architecture.

Xiaoling Dai received her B. S. degree in Mathematics with first class honors from Hebei University in China in 1984. In 2004, she received the Ph. D. degree in Computing Science from University of Auckland in New Zealand. She is now a Senior Lecturer in the School of Computing, Information & Mathematical Sciences, University of the South Pacific from 2005. Her research interests include component-based software engineering, distributed system design and implementation, software architecture, electronic micro-payment systems for e-commerce, file-sharing, or m-commerce in client-server, peer-to-peer, and mobile networks, web service security and service-oriented software engineering.