2002

Charging Control and Transaction Accounting Mechanisms using IRTL (Information Resource Transaction Layer) Middleware for P2P Services

Junseok Hwang
Syracuse University

Praveen Aravamudham
Syracuse University

Elizabeth D. Liddy
Syracuse University, liddy@syr.edu

Jeffrey M. Stanton
Syracuse University, jmstanto@syr.edu

Ian MacInnes
Syracuse University, IMacInne@syr.edu

Charging Control and Transaction Accounting Mechanisms using IRTL (Information Resource Transaction Layer) Middleware for P2P Services

Junseok Hwang[1], Praveen Aravamudham[1]
Elizabeth Liddy[1], Jeffrey Stanton[1] and Ian MacInnes[1]

[1] CST 4-291, School of Information Studies, Syracuse University,
Syracuse, NY, 13224, U.S.A
{jshwang, paravamu, liddy, jmstanto, IMacInne}@syr.edu
http://web.syr.edu/~jshwang/IRTL/

Abstract. One of the many challenges associated with p2p computing is providing mechanisms for control over accounting transactions among peers. With the increase in the number and variety of p2p applications, there is a need to distinguish between services that can be charged and services that can operate without charging mechanisms. In this paper we present an Information Resource Transaction Layer (IRTL) middleware architecture that addresses some of the technical challenges associated with heterogeneous resource transactions in the p2p-computing environment. We propose to handle charge control, transaction accounting, reputation management and several other p2p parameters through the IRTL.

1 Introduction

P2P computing has effectively made the Internet into a giant copy machine, allowing users and consumers to communicate and swap resources far more freely than businesses that produce content ever expected. One result is abundantly clear: With the increase in the number and variety of p2p applications, there is a need to distinguish between services that can be charged and services that do not require charging mechanisms. While some p2p services such as Napster allowed free and unrestricted copying of files, such services failed to differentiate between resources that are copyright protected and those that are not. The challenge of pure P2P computing - that is, sharing content without central control, organization or intermediaries - is that no one seems to control it so no one seems to be able to make money from it. This raises a number of interesting questions about p2p services: Could a p2p service successfully implement a payment mechanism similar to those adopted by business-to-business or business-to-consumer companies? But traditionally payment mechanisms for accessing resources through the Internet have not found many ways to generate revenue or, more importantly, to distribute revenue to content creators. So what are some of the workable ways to control charge and maintain accounting for resources accessed by peers? This is the set of questions raised by planners of services such as "Mojo Nation" and this is the set of questions we hope to explore by providing a research platform for experimentation with p2p control and charging management.

In this paper we present an IRTL (Information Resource Transaction Layer) middleware architecture that addresses some of the technical challenges associated with heterogeneous resource transactions in the p2p-computing environment. This IRTL facilitates discovery, valuation, negotiation, coordination, charging and exchange of resources among peer users as a middleware platform. We present the IRTL design and architecture and describe the charging and transaction accounting mechanisms using the IRTL. Finally, we close the paper with a short discussion on future works and implications.

2 Background and Related Works

Over recent years, the application and service nature of the Internet has gradually begun to change from a predominantly client-server orientation to an environment that also allows for peer-to-peer (p2p) services. The continuing evolution and growth of peer-to-peer services has provided distinctive challenges in the optimal use of computing resources, the management of quality-of-service, and the coordination of transactions between peers [1]. Various peer-to-peer service architectures and service mechanisms have been developed in response to these challenges.

Many peer-to-peer systems are based on coordination and allocation systems, including virtual machine and agent environment, that limit or balance access of the system to each peer's computing power, data and other resources. Sun Microsystems's JXTA project [2] provides a set of protocols to discover other peers and peer groups as well as to establish message and information channels between and among peers. As an enabling layer, JXTA aims to provide an architecture that can integrate and thus provide authentication, access control, audit, encryption, secure communication, and non-repudiation [2].

Grid computation tries to integrate and utilize dispersed resources to perform one task. Necessary services for grids include authentication, authorization, security and grid-wide name spaces for resources and resource registration/discovery. These services further facilitate resource accounting, job scheduling, and job monitoring. One notable example in the area of grid computing is the Globus project (http://www.globus.org/), which provides fundamental architecture and services for computational grids. A main focus of Globus is the open source middleware development platform, the Globus Toolkit. This toolkit provides infrastructure software to build grids and to provide solutions for many aspects of grids.

The Open Grid Services Architecture (OGSA) [3] builds on concepts and technologies from the Grid and Web Services [4] communities, the architecture defines a uniform exposed service Semantics (the Grid service); defines standard mechanisms for creating, naming, and discovering Transient Grid service instances; provides location transparency and multiple protocol bindings for service instances; and supports integration with underlying native platform facilities. The Open Grid Services Architecture also defines, in terms of Web Services Description Language (WSDL) [5] interfaces and associated conventions, mechanisms required for creating and composing sophisticated distributed systems, including lifetime management, change management, and notification. Service bindings can support reliable invocation, authentication, authorization, and delegation, if required.

Early peer-to-peer systems typically relied on the generosity of users to open their systems to others. A peer-to-peer network without transactions and accounting is, in socio-economic terms, a common pool resource. Such common resources can experience a "tragedy of the commons" unless appropriate institutions exist to counteract destructive incentives for overuse [6]. Accounting and transaction systems can resolve the free rider problem by providing incentives for users to provide access to their resources [7]. Monetary pricing or other valuation schemes can ration access in a similar fashion to traditional markets. For such a system to be economical for a large volume of electronic transactions, however, a protocol for electronic micro-payments must exist. Proposals for services (e.g., Mojo Nation) that include such systems of micro-payments, attempt to solve the tragedy of the commons problem by placing “financial fences” around scarce resources. Such systems can involve one or more separate currencies developed and maintained for this specific purpose. The protocol would include provisions for assigning units to currencies, for determining whether and how denominations are expressed (e.g., granularity), and for associating user-generated names with denominations.

Although some peer-to-peer systems depend on common or freely shareable resources, sustainable future systems will probably have to center on the exchange of valuable resources whose owners have a stake in receiving compensation for their creation (e.g., [8]). Intellectual property owners will want to ensure that network users pay for information that they access, borrow, rent, or lease. A peer-to-peer information resource transaction system must include mechanisms to facilitate tracking and transfer of ownership rights.

Few middleware platforms supporting charging and accounting control for p2p computing and networking services. Adar et al. [9] conducted an extensive user traffic analysis on Gnutella [10] to show the significant problem of free riding. The authors argued that the free riding problem eventually degrades the system performance and increases the system vulnerability. To over come the free riding problem, many have suggested that a market based architecture that imposes structures for charging and transaction accounting control is essential. To date, however, no such architecture has established itself as a predominant model. Although a number of projects (some of them open source) have attempted to develop a workable model, we believe that considerably more research and experimentation must occur to resolve the major challenges involved. Our proposed Information Resource Transaction Layer provides a test-bed for such experimentation.

3 IRTL (Information Resource Transaction Layer) Middleware

The middleware platform creates a “glue” layer of various service managers that will integrate the services to fulfill different functional requirements [11]. This glue layer, termed the Information Resource Transaction Layer (IRTL), will integrate services from existing resource management platforms. Existing platforms such as Sun’s JXTA can be used for developing a p2p infrastructure with the IRTL. It defines a set of protocols that can be used to build any p2p application with flexibility to create p2p environments for application specific requirements. [12] The IRTL can utilize those protocols to support the middleware for information resource management in the p2p environment.

The proposed IRTL architecture comprises four major overlaid sections called the Service Identification Layer, Manager Layer, XML Generator Layer and the Metadata Information Base Layer. These layers lie on top of an Application Layer and a typical Internet Protocol network as illustrated in Figure 1.

The Application Layer comprises of a number of p2p applications that make a connection to the Middleware layer. Typically, SOAP [13] [Simple Object Access Protocol] is the mechanism used to embed requests for resources.

The Service Identification Layer contains WSDL documents, which offer a description on the type of services managed by the middleware layer. Characteristics such as port types, service types and message types are contained inside a WSDL document.

The Manager Layer facilitates decisions about the use of resources and controls usage of resources on behalf of a peer. The Manager elements allocate and manage resource sharing among different peer systems and networks with different resource definitions and identifiers through metadata information. Resource characteristics embedded inside metadata are subsequently passed on between managers. A Manager also provides a mechanism for managing resources within a resource domain where it has direct control and provides a mechanism to manage resources with other peer domains. A Manager supports resource allocation and provisioning at the system boundary among multiple peers.

The Metadata Information Base layer is a distributed database that manages metadata information for the resources and profiles of the peers. Metadata provides an extensible markup mechanism for documenting the structure and content of heterogeneous information resources: Our architecture facilitates the use of automated application of markup and thus simplifies the related processes of resource discovery and resource inventory maintenance. Information resources for reputation management, trust assessment, charging and accounting control, privacy, and security will be maintained in the Metadata Information Base.

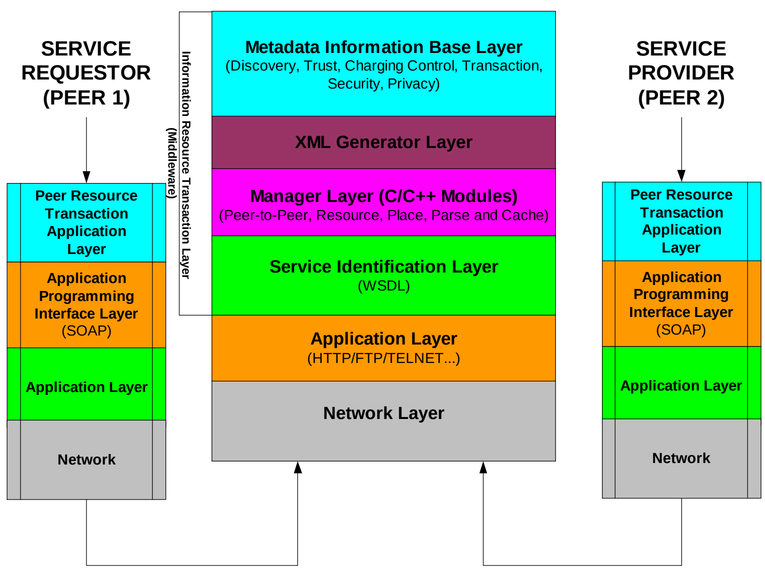

Fig. 1. IRTL Middleware Architecture

The managers will relate to one another through the Metadata Information Base for dynamic management of metadata services among transacting peers. Instantiating a "manager" that controls metadata information, or even transaction accounting as to each user's role in sharing data simplifies communication inside the middleware layer. Further, this manager can establish connections to other components inside the middleware layer through WSDL methods, and XML can be used as the primary markup language to embed data content. Instantiating several managers to perform specific tasks pertaining to the p2p application helps in controlling middleware services in general, and also in determining application specific information to help analyze if applications are being subjected to change or even non-existent within the p2p domain and hence knowing the state of the application. In the p2p example, we call the "Peer-to-Peer Manager", which primarily contains metadata information and transaction accounting information. There can also be a "Resource Manager", which controls user information, and user(s) role in resource sharing. These modules can act interdependently with each other sharing private and public data. Details about various managers can be found from the project website shown in the title of this article.

4 Charging Control and Transaction Accounting through IRTL

In this section, we present how the IRTL handles charge control and transaction accounting mechanisms with a simple example.

4.1 Charging Control

Figure 2 shows the breakdown of the different sub-layers inside the IRTL layer, with emphasis on the Peer-to-Peer Manager and the Resource Manager for charge control and Transaction Accounting. In Figure 2, the IRTL receives requests from Peer 1 and Peer 2 to retrieve a research paper. The internal logic schema for charge control is represented in Figure 3. Figure 3 also outlines the algorithms used in charging control with emphasis given to peer relationships that are used in request servicing.

The request traverses through the Service Type Identifier, detecting "Web Services" type of service application. The request is embedded inside a SOAP document, which is sent to the Service Type Identifier (STI) via HTTP. The STI sends a request to the Universal Description, Discovery and Integration (UDDI) [14] to invoke WS-Inspection Documents, which identify Peers that have published research papers similar to the request. Peer 3 is a possible Service Provider that a Peer identified through the UDDI layer. The assumptions made are as follows:

- Peer 2 is a "buddy" of Peer 3
- Peer 1 is an anonymous peer logged in for a one-time transaction.
- The Network is not BUSY
- There are no copyright or patenting rights on the information requested.

The UDDI layer invokes the Place Manager, which detects the specific server as to where the Information is located. (We assume the Metadata Information Base Layer is comprised of many servers that are used to distribute the vast plethora of information types and their associated structures). The Peer-to-Peer Manager is called next which establishes a connection with the MIB layer. The p2p Manager requests "Buddy List" information on the requestor Peers. "Buddy Lists" is one of many schemas we propose for handling charge control. The mode of operation is similar to the ones present in Messengers (Yahoo! Messenger, MSN, AOL AIM Messenger etc.). A peer can add another peer to his buddy list upon consent from the peer. The Metadata Information Base Layer contains information on "Buddy Lists" on a simple database that adds in entries made from a peer each time a buddy is added or deleted. In our example, Peer 2 is a buddy of Peer 3, and Peer 1 is an anonymous peer. Also, Peer 1 does not offer payment or exchangeable information for access. Hence, Peer 2 is offered service, and then Peer 1 again based on other requestor Peers.

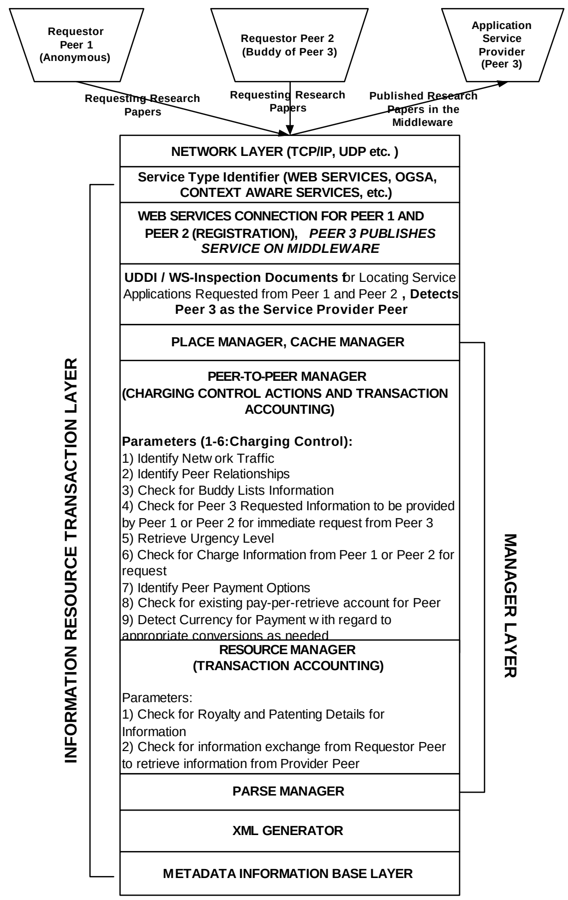

Fig. 2. Charging Control and Transaction Accounting with IRTL

The different parameters through which the Peer-to-Peer Manager emphasizes charge control are:

- Detecting Network Traffic (If Busy or not available, Stall Both requests from the Peers).
- Detect Peer relationships between the Requestor Peers. If both peers are buddies, then handle request based on a First-in-First-out basis.
- Detect Buddy List information between Requestor Peers and the Provider Peer. If a buddy is matched, more preference is given to the buddy peer, and less preference to the anonymous peer.
- Retrieve Urgency Level of Request made from the Requestor Peers.
- Detect information that could be exchanged for information. For example, if the Provider Peer has submitted a request to the IRTL for information that is not present, the IRTL makes a record of the request, and matches them for possible exchangeable information for future requests.

- Detect "Payment" information from peers. For example, if the Provider Peer submits a charge for information that needs to be accessed, the Information becomes "pay-per-view"; the Requestor Peers need to pay in order for information access.

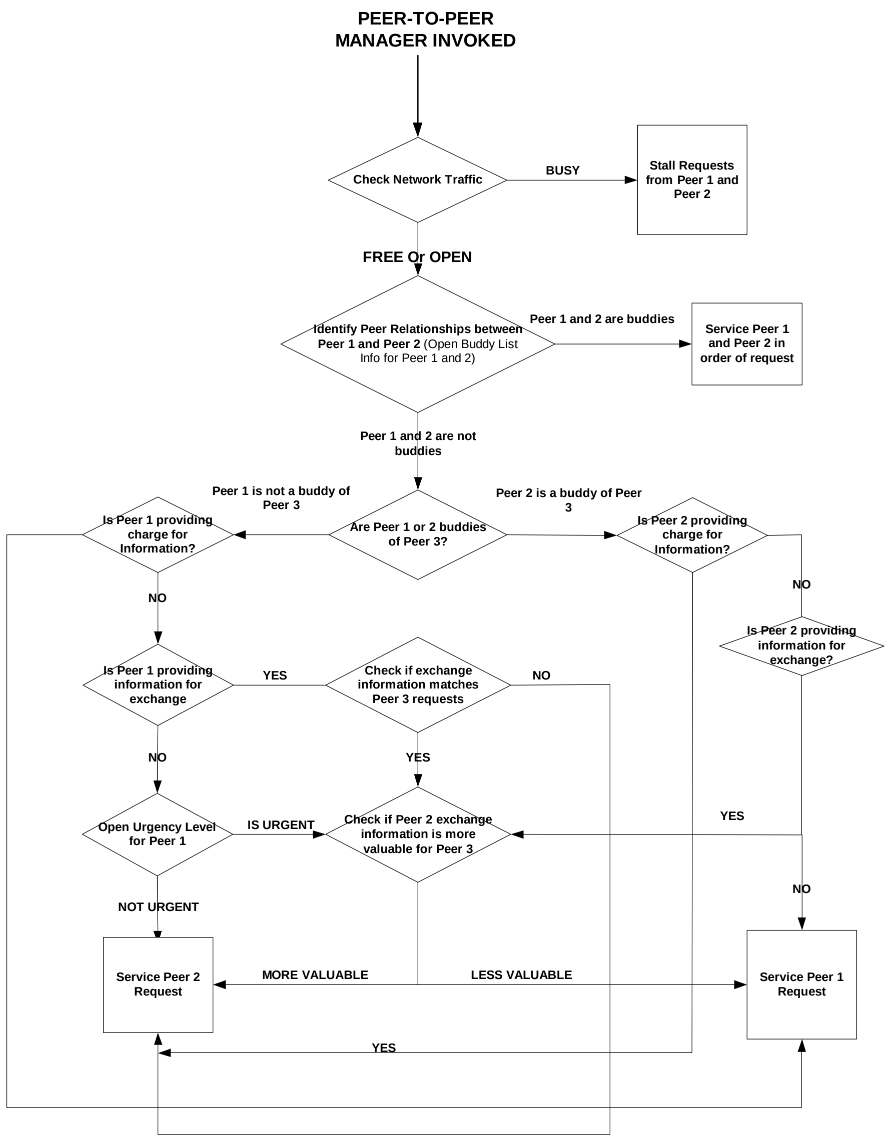

Fig. 3. Peer-to-Peer Manager Logic Representation for Charging Control

4.2 Transaction Accounting

Transaction Accounting is primarily managed inside the Peer-to-Peer Manager and the Resource Manager. The Peer-to-Peer Manager retrieves information about Peer payment options setup. The IRTL offers a system similar to the PayPal system to Peers through which Peers can setup payment methods for information access. Other payment mechanisms like “Split-Payment” are also made available. Split-Payment is a schema by which Peers can pay based on an installment or a split basis. Discounts, and Deal Information on Peers are also available for Peers for bulk information access. The Resource Manager primarily handles copyright and patenting details on information. Patented and Copyrighted Information are disclosed to Requestor Peers, with algorithms in place for detection and enforcement. Since, most p2p models rely on “Information Exchange”, we propose a model capable of detecting specific information for exchange from the Requestor Peers to the Provider Peers.

Figure 4 primarily deals with cases encountered during p2p information transfer noting the different types of transfer algorithms encountered from charging to exchange principles. Charging principles are responsible for detecting the type of payment ensued, while exchange principles follow a more complex routine in determining the nature and validity of information. The different parameters followed by the Peer-to-Peer Manager and Resource Manager for Transaction Accounting is as follows (Figure 4).

- Detection of Payment Type for Transaction by the Resource Manager. (There are 3 payment types considered: Micro Payment, Split Payment and Information Exchange).
- In the case of Micro Payment, a connection to the Service Provider is made to disclose payment amount for acceptance. If the request is accepted, the Application Requestor Peer is serviced. The Peer-to-Peer Manager sends a request back to the Service Type Identifier Layer through XML’ised data embedded inside a SOAP response generated by the STI. If the Service Provider denies the payment amount put forward by the Requestor Peer, the service is terminated for the Requestor Peer.
- In the case of Split Payment also, the Service Provider is notified for a response. Note: the Requestor upon connection sends the Split Payment details via SOAP, which traverses through to the Peer-to-Peer Manager. If accepted, the Application Requestor Peer is serviced. If the Provider denies the transaction, a request is formulated through the Peer-to-Peer Manager to be sent to the Requestor seeking other Payment options. If the Requestor Peer changes Payment type to Micro Payment or Information Exchange, then once again the Provider is notified of the change seeking a response. If approved, the Requestor Peer is serviced, and if denied, the service is terminated. We do consider Micro Payment for approval once Split Payment is denied, since we aim at offering Requestor Peer several opportunities for service.
- Information Exchange is more complicated compared to other options, since the extent of information, which can be termed as “valuable”, can never be appropriately determined. The approach we follow is to collect Service Requests made by the Service Provider at an earlier stage to determine “valuable information”. Since records of information requests are continuously updated in the Metadata Information Base Layer, a simple searching scheme to match the information exchangeable by the Requestor Peer to data already present in the Metadata Information Base Layer yields “valuable” information. If a match does occur, the requestor peer is serviced, and if not, service is denied.
- “Buddy List Information” can also be used for transaction accounting. Suppose the Requestor Peer is found in the buddy list of the Provider Peer, a response is made to the Provider Peer seeking approval for transaction. If approved, the Requestor Peer is serviced, and if denied, the service is terminated.

The Metadata Information Base Layer records every transaction occurrence for information request. This helps in maintaining a “History” of Peer requests for identifying possible exchangeable information and also for Reputation Management. Additionally, copyrighted and patented information need to be considered for access. Since security for information can vary from peer to peer, different levels for security are enforced in the Resource Manager identifying types of information that can be accessed corresponding to peer's.

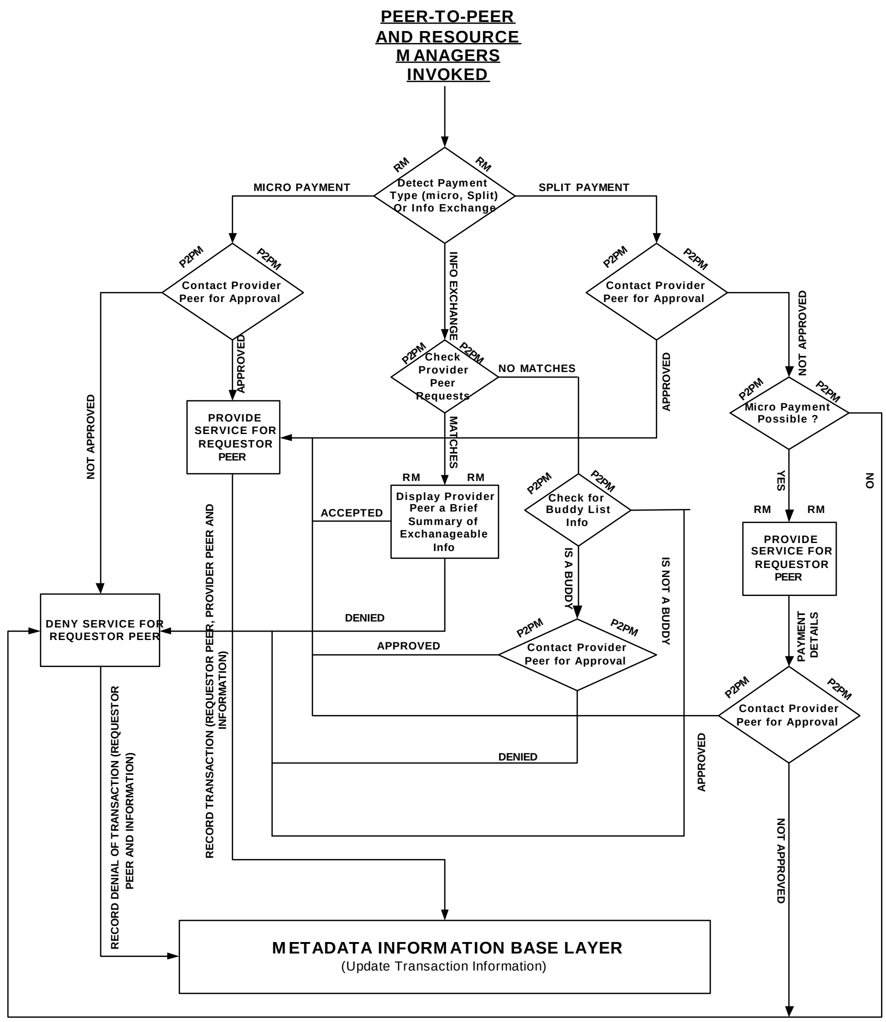

Fig. 4. Transaction Accounting Logic Representation

5 Conclusion

In future p2p computing, the heterogeneous resources available for sale or barter will naturally include the basic building blocks of communications bandwidth, data storage, and computational power but eventually will also include more abstract resources such as databases, directories, catalogs, resource maps, security keys, identities, membership lists, price lists and many other information resources. For information resource transaction services in networked, distributed computing environments, a middleware approach would provide consistent, standardized solutions to problems of transaction security, directory services, quality of service, discovery of resources, and accounting. In this paper, we presented a middleware platform architecture and design, the IRTL (Information Resource Transaction Layer), for information transaction services that provide access and integration across heterogeneous p2p service platforms. In particular, we demonstrated one possible use of IRTL for charging control and transaction accounting of p2p services.

One of the many challenges offered by p2p computing is effective handling of reputation management. A usable reputation management system must include mechanisms that allow for automatic correction of publicized reputations. In a peer-to-peer system there is a further complication that public reputation information must exist in distributed form without a central administrator or repository. In the IRTL, this function can be utilized by instantiating a "Reputation Index" (RI) for Peers. A reputation index is a count of activities made by a peer when accessing the IRTL. We are currently investigating this reputation management issue further to identify generic factors affecting the RI in relation to the transaction accounting of peer-to-peer service environments.

References

1. Oram, A. (Ed.) Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly and Associates. (2001)
2. Gong, L. Project JXTA: A Technology Overview. Sun Microsystems, Inc. Palo Alto, CA, USA, (2001) http://www.jxta.org/
3. Foster I., Kesselman, C., Nick, J. and Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, the Fourth Global Grid Forum, Toronto, Canada, (2002) http://www.globus.org/research/papers/ogsa.pdf
4. Web Services: W3C Architecture Domain http://www.w3.org/2002/ws/
5. Christensen, E., Curbera, F., Meredith, G. and Weerawarana., S. Web Services Description Language (WSDL) 1.1. W3C, Note 15, (2001) www.w3.org/TR/wdsl.
6. Ostrom, E. Governing the Commons: The Evolution of Institutions for Collective Action. Cambridge: Cambridge University Press. (1990)
7. Gupta, A., Stahl, D., Whinston, A. The Internet: A Future Tragedy of the Commons? Paper Presented at the Conference on Interoperability and the Economics of Information Infrastructure (1995) http://cism.bus.utexas.edu/alok/wash_pap/wash_pap.html.
8. Korba, L. Towards an Agent Middleware Framework for E-commerce. Netnomics, 2 (2), (2000) 171-189
9. Eytan Adar, and Bernardo A. Huberman, "Free Riding on Gnutella," FirstMonday, (2000)
10. The Gnutella Homepage, http://gnutella.wego.com/
11. Internet2: Middleware Initiatives http://middleware.internet2.edu/
12. JXTA: The JXTA Solution to p2p technology October 2001 http://www.javaworld.com/javaworld/jw-10-2001/jw-1019-jxta.html
13. SOAP: W3C Simple Object Access Protocol Version 1.1, May 8, (2000) http://www.w3.org/TR/SOAP/
14. UDDI: Universal Description, Discovery and Integration http://www.uddi.org/