INTERNET BILLING SERVICE DESIGN AND PROTOTYPE IMPLEMENTATION

by

Marvin A. Sirbu*

Abstract

A group of students in the M.S. program in Information Networking at Carnegie Mellon University have designed and implemented a prototype of an Internet Billing Service--an electronic credit card service for the Internet environment. The service provides account management, authentication, access control, credit verification, management reporting, billing and collection services to network-based service providers.

1. Introduction

A worldwide data networking infrastructure is gradually falling into place which will allow consumers and service providers to interact in a vast electronic marketplace. In France some 9,000 services are available over the Minitel network. In the U.S., access to electronic databases generates billions of dollars a year in business. As networked computers proliferate at home and in the workplace, more and more consumers are in a position to shop in the electronic marketplace.

---

* This report represents the collective work of 13 students in CMU's Master of Science Program in Information Networking and is derived from their final project report: Richard Batelaan, Richard Butler, Chun Yi Chan, Tie Ju Chen, Michael Evenchick, Paul Hughes, Tao Jen, John Jeng, Jon Millett, Michael Riccio, Ed Skoudis, Chris Starace, Peter Stoddard. The project was directed by William Arms, John Leong and Dennis Smith of CMU. Dhananjay Gode served as Teaching Assistant and prepared the first draft of this paper.

Already the Internet, a loose confederation of independently managed networks, links eight million users and some one million computers on 10,000 separate subnetworks in more than 40 countries.[1] Once used only by universities and research organizations, the Internet is used today by individuals and corporations for a wide range of commercial purposes including disseminating information, searching remote databases, and providing access to specialized computer resources. Major information service providers such as Dialog and BRS can now be reached via the Internet.

While it is relatively simple for an entrepreneur to set up a small computer capable of providing information to the worldwide Internet community, it is much more difficult to arrange a mechanism to charge users for the services rendered and to collect payments. Current billing mechanisms for electronic services are costly and inconvenient for both service providers and end users. In the absence of a centralized billing service, users must initiate a service agreement with each service provider before using its services, and must keep track of its access point, password, and bills. Also, there is no central directory of service providers. It is uneconomical for small service providers to advertise, check credit, authenticate users, control access, bill and collect payments, maintain audit trails, and keep usage statistics.

Both service providers and end users need a reliable, easily accessible, fast and inexpensive intermediary, a billing service. The billing service could be compared to an electronic credit card for services on a network. It would allow small service providers to contract out the functions of billing users and collecting payments and concentrate on providing services.

This document describes the design and implementation of a prototype of a computer-based Internet Billing Server (IBS) developed by a project team from the Information Networking Institute (INI) at Carnegie Mellon University. The project team had two major tasks: (1) specify functional requirements for a full scale billing server, and (2) design and develop a prototype which, while satisfying only a subset of these requirements, demonstrates the feasibility of the concept. Specification of the full set of requirements brings out important issues and ensures that the prototype design has no major flaws which limit its extensibility to a full-scale system. We will summarize the full set of requirements, noting in passing where the actual prototype differs.[2] Our design demonstrates how an Internet Billing Server could facilitate the emergence of a real electronic marketplace.

2. Unique Characteristics of Network-Based Markets

The design of a network billing server is difficult because markets for electronic services are different from markets for physical goods and non-electronic services. A network billing server must be designed to take these differences into account, in order to avoid fraud and disputes. These differences are outlined below.

First, in a network, the users, the service providers, and the billing service are geographically separated. Credit cards are designed for situations where the users physically present their credit cards at the time of purchase so that merchants may validate their signature. Although credit cards are used today for placing orders over the phone, such methods are highly insecure; ordering over a network makes it difficult to verify the identity of the parties. Indeed, to reduce fraudulent charges, many merchants will only ship goods ordered over the phone to the billing address of the credit card holder. Secure network authentication protocols, such as Kerberos,[3] may be part of a solution but the legal liability and responsibility of participants in an electronic market is not well defined.

Second, given the high processing speeds of electronic services, a user can accidentally run up a huge bill within a matter of seconds without having an opportunity to cancel it. In contrast, if there is a mistake in ordering a physical product, it can be corrected before the product is shipped or the product can be returned after delivery. Even though a non-electronic *service* cannot be returned, the user can still cancel it while it is being performed: one can leave a hotel if its service is not satisfactory or is too expensive. Providing a similar capability for halting the provision of network based services in midstream is complex.

Third, in contrast to physical goods, it is difficult to determine the price for an electronic service in advance. For example, if the price of a database query were based on the number of bytes of information it generates, it would be difficult to determine the query's price without searching the database. It is infeasible to let the user have a look at the information to assess its value; it is also infeasible to price information solely upon objective measures such as its size in bytes. This difficulty in judging the quality of information may give rise to disputes which are difficult to resolve. This makes it important to carefully define the legal role of the billing server.

Fourth, electronic information can be easily duplicated and redistributed. This makes product "returns" meaningless because a user can copy an electronic file before returning it. It is also much easier to copy and redistribute an electronic version of a book than copy and redistribute a printed version of the same book.

The INI Internet Billing Server is able to address some, but not all, of these issues. Security is provided using passwords and encryption. The IBS also provides a capability for setting and enforcing spending limits. The billing server provides a flexible mechanism for charging for network services, and for price negotiation, but it does not pretend to resolve the problem of determining a service's value. Nor does it address the issue of illegal copying and redistribution of purchased information.

3. Functions Provided by a Billing Server

A billing server plays the same role vis-à-vis end users and service providers as a credit card company does vis-à-vis cardholders and merchants. Consider a credit card holder going to rent a car. He begins by identifying himself to the rental car company by presenting his credit card and driver's license. He negotiates with the car company for the service he desires, the cost per day and the maximum number of days he expects to keep the car. The merchant then verifies the customer's credit with the credit card issuer and places a hold on the customer's credit for the maximum amount of the rental. When the customer returns the car, the transaction is complete, and the car rental agency sends a final invoice to the credit card company, with a copy to the consumer. At the end of the billing cycle, the credit card company sends a bill for all purchases charged to the card, including the car rental, and the customer sends back his payment. The credit card company pays the merchant after deducting its fees.

Our model of transactions in the network marketplace is similar to the car rental scenario: a customer or *end user*—through his computer— interacts over a network with two other computer systems: the *service provider*, and the *Internet Billing Server*. as illustrated in Figure 1.

[put figure 1 here]

All of the steps described above for renting a car have their counterparts in the network marketplace. However, instead of face-to-face communications as in the car rental scenario, the end user's computer interacts with the service provider's computer over the network. Each step in the interaction forms part of the Internet Billing Protocol (IBP), a proposed standardized method of interaction among an end user's computer, the service provider's computer, and the Internet Billing Server computer.

The Internet Billing Server is more than a computer and a set of standardized protocols, however; it is a model for a business which provides valuable services to network marketplace entrepreneurs. The Internet Billing Service acts as a factor for the service provider, providing prompt payment while taking over all aspects of billing and collections. To be successful as a business, the Internet Billing Service must satisfy two sets of customers. It must attract merchants by giving them easy access to a large group of customers, and providing them a cost-effective way to receive payment for services provided. It must make it as easy as possible for service providers to make use of the Internet Billing Server, working with the providers to modify both client and server software to implement the Internet Billing Protocol. It must attract end users by providing a powerful and flexible capability for managing end-user accounts and by giving end users access to a large number of service providers.

While we have described the Internet Billing Service as an independent business, large organizations often have a need to create an internal equivalent of an Internet Billing Server. For example, the central administration at a university such as CMU could operate a billing server as a single mechanism to bill for online library access, computing services, printing services, and electronic mail. While we recognize this as another potential application for the billing server software developed in this project, the focus has been on the design and implementation of a public, third-party billing server.

As designed by the project team, the INI Internet Billing Server provides the following functions to service providers and end users:

- *Account Management.* The IBS enables end users to establish an account relationship with the Billing Server which will permit them access to any number of service providers. Service providers establish accounts which enable them to use the IBS to bill their clients for services rendered.

- *Authentication*: The IBS provides a service for authenticating both end users and service providers prior to any transaction and for secure communications between the parties.

- *Access Control*: The IBS provides access control for both end users and service providers. Information associated with an end user account can specifically designate a list of services that may be accessed, or a list of services that specifically may not be accessed.

- *Price Negotiation*: Using the Internet Billing Protocol, the end user may determine the services available from the service provider and the posted prices. The Internet Billing Server can record the mutual agreement of the end user and the service provider on a set of prices, and the maximum amount which the end user has authorized for this set of transactions.

- *Credit Verification:* The Internet Billing Service will verify to the service provider that the customer has sufficient credit to pay for the proposed transaction up to the agreed cap.

- *Final Invoice:* At the conclusion of the transaction, the service provider can send a final invoice to the Billing Server using the IBP. The Internet Billing Server will send an authenticated copy of the invoice to the end user.

- *Periodic Billing:* The Internet Billing Server will generate periodic billing statements to customers detailing all of their transactions and the sums owed.

- *Collections:* The Internet Billing Service will collect funds from end users and make payments from these funds to service providers.

- *Directory Services:* The Internet Billing Service provides a "white pages" and "yellow pages" service for identifying service providers

- *Help Service*: The Internet Billing Server provides an online help manual service.

- *Software libraries*. In the client server model, every service provider must make available to end users client software capable of accessing the service provider's service. A successful Internet Billing Service must provide a set of library routines which make it simple to upgrade both client and server software to support the Internet Billing Protocol. These modules are shown logically in Figure 2.

[put figure 2 here]

4. Design Objectives

In developing the Internet Billing Server and the Internet Billing Protocol we were guided by several fundamental considerations.

- The Internet Billing Server will operate in a transaction-oriented environment. All communications between the parties will be based on a remote procedure call communications paradigm. This is in sharp contrast to most current network-based information services. These typically have been provided via large timeshared computers. Users log in over low-speed networks from dumb terminals–or PCs emulating dumb terminals–and are charged by connect time. However, as desktop computers have replaced dumb terminals and networks have increased in speed, a new information access paradigm has emerged: client-server. In this paradigm, powerful desktop computers running user-friendly client software interact with remote servers on a transaction basis. In a few seconds large files of information can be requested and transferred from servers to clients. File Transfer Protocol, Gopher, and Wide Area Information Service are but a few of the client server protocols used by numerous clients and servers on the Internet. In a client-server environment there is no notion of connect time. Accordingly, services must be billed on a per-transaction basis.

- The billing server should have high availability since, in its absence, the service providers will not be able to offer their services to end users.

- The billing server should be scalable. It is difficult to predict the initial number of customers and the growth pattern, even though the number of potential customers is large. These latter two points suggest that the billing server should be designed to run on replicated, distributed computers, thus providing modular scalability and high availability through redundancy.

- Communications between the parties — end user, service provider, and billing server — should be based on widely available telecommunications standards to ensure the largest market for the services.

- Secure authentication and encryption are critical because all three parties will be connected via insecure public networks. Without a secure authentication mechanism there is a substantial potential for fraud.

- Before using a service, the end user must understand and agree to the prices and terms of the exchange. The transaction protocol in the billing system must support an initial price negotiation between the end user and the service provider. The billing server should be informed about the outcome of this negotiation by both the service provider and the end user. To avoid disputes the billing server should make sure that the user and the service provider have the same version of the agreement.

- The users should be able to limit their financial exposure on a transaction by specifying a spending cap. If the cost of an ongoing transaction exceeds this spending cap then the end user should be able to choose whether to abort the transaction, or continue it by raising the spending cap.

- The billing server should not become a bottleneck slowing the speed of interaction between the end user and the service provider. In particular, the billing server should not be a gateway for communication between the end user and the service provider. Interactions with the billing server should be as few and as simple as possible.

- The billing sever software should help users in their account management. It should support hierarchical accounts so that corporate users can get bills aggregated by organizational units such as departments, regions or divisions. Similarly, a provider of multiple services may use an account hierarchy to organize information on the use of each service.

With these general requirements in mind, the project team prepared a detailed requirements document specifying all of the capabilities required of the Internet Billing Server.[4]

5. Design of the Internet Billing Server

Our prototype Internet Billing Service was implemented using widely available technology. The Billing Server prototype is designed to run on a Digital Equipment Corporation workstation class machine running the Ultrix operating system. It was written in C and uses the Ingres Database Management System. It also uses Transarc Corporation's Base Development Environment (BDE) to provide multithreading, which allows the prototype to process concurrent requests efficiently. For communication between the billing server, end users, and service providers, the prototype uses the remote procedure call (RPC) portion of the Distributed Computing Environment (DCE) provided by the Open Software Foundation and the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. Implementations of DCE are available for the OS2/2.x and Microsoft's Windows NT operating systems as well as Unix.

Authentication is implemented using the Kerberos protocol developed at M.I.T. All communications between the parties are encrypted for security using the Data Encryption Standard (DES) encryption method.

Code libraries enabling rapid modification of client and server software to support the Internet Billing Protocol were written in C. As a test of the complete system, we modified versions of File

Transfer Protocol (FTP) client and server software to make use of the Internet Billing Server. Using these software packages, a service provider could distribute information using FTP and bill for it using the Internet Billing Server.

6. Transaction Sequence

Figure 3 illustrates the sequence of steps involved in the use of the Internet Billing Server.

[put figure 3 here]

Step 0. Establishing an Account

Prior to engaging in a network-based transaction, an end user must first establish an account with the Internet Billing Server. In our design, any number of accounts may be organized in a hierarchical fashion by allowing each account to have *sub-accounts*, each of which is also an account. See Figure 4 for an example of a single hierarchical account structure.

[put figure 4 here]

The hierarchical structure represents authority over accounts; the end user of a parent account has authority over the end user of a sub-account. Every hierarchy has an Account Administrator, who is able to view financial information, and modify certain account characteristics for all the accounts in the hierarchy.

Billing and usage information can be aggregated by various branches of the hierarchical structure, or detailed information for each node in the structure can be supplied. An organization should have the ability to give managers privileges to modify some of the information for the accounts of their subordinates. This hierarchical structure allows the account environment within the Internet Billing Server to mirror the environment within organizations.

Account hierarchies may be individually or collectively billable. In the first case each account is fully billable, i.e., it contains all the financial information such as balance due, adjustments, payments and usage information; the hierarchy is used merely to provide aggregate billing information for management and control. This model may be more appropriate for decentralized organizations. In the second type of hierarchy only the parent or root of the hierarchy is fully billable, i.e., only the root account contains the full billing information for the hierarchy whereas for other accounts only the usage information is listed. The prototype only supports hierarchies where each account is a fully billable account.

Service providers which offer multiple services may want to order their accounts in a hierarchy to help maintain valuable marketing and usage information. Since each distinct service requires a unique Kerberos identifier and an account will not provide multiple Kerberos identifiers, each service must be given a separate account within the Internet Billing Server. By allowing these accounts to be placed into a hierarchical structure, the Internet Billing Server can make one aggregated payment to the service provider instead of a separate payment for each service. In addition, hierarchical accounts make it easier to supply the service providers with one statement containing usage information for all of their services.

Step 1 - User Authentication and Access Control

Since a network is a mutually suspicious environment, the service providers, the end users, and the billing server must authenticate each other prior to any transaction. This step may be compared to credit card users showing their driver's license to prove their identity while using a credit card. As noted, we use a Kerberos-based authentication system for secure communication between end users, service providers, and the billing server. Cross-realm Kerberos authentication may be required in the full-scale system if large users authenticate end users within their organization and then ask the billing server to accept their authentication. However, our prototype does not need cross-realm Kerberos authentication because it functions within a small group of end users and service providers. All communication is encrypted using the Data Encryption Standard for security.

After authentication, the end users may directly request access to a specific service provider, or may search through an index of service providers classified by service categories to select the service they want. The prototype does not support the directory service. The billing server checks the access control lists of both the end user and the service provider to ensure that the end user is allowed to access the requested service. In the full set of requirements, the end-user and service-provider accounts may have two types of access control lists: (1) two positive access control lists -- one listing specific service providers/end users and the other listing categories of service providers/end-users; and (2) similarly, two negative access control lists. End users' lists specify which service providers they can (positive lists) or cannot (negative lists) access; similarly, service providers' lists specify which end users are allowed or not allowed access to them.

The negative access control lists override the positive lists, and determine which specific services or service categories cannot be accessed from the account. Corporate users could use positive access lists to allow access only to company-approved service providers. Parents could use negative access lists, analogous to 900 telephone service blocking, to prevent their children from accessing frivolous or high-cost services. We have implemented only negative access lists for end users and only positive access lists for service providers.

If access is allowed, the billing server issues the end user a Kerberos ticket for the service provider; that authenticates the end user to the service provider. As mentioned before, the end user must have the client software specific to the service provider (for example FTP or Internet Gopher interface software) in order to access the service provider.

Steps 2 and 3 - Price Negotiation and Spending Cap

After getting a Kerberos ticket from the billing server, the end user and the service provider negotiate a price for the requested service and a spending cap for the transaction. This is called an agreement. Note that the end user is communicating with the service provider's computer, not with a human representative of the service provider.

The end user sends a copy of the agreement, encrypted with his private key, to the service provider who forwards the end user’s copy to the billing server, along with his own copy of the agreement. This prevents an unscrupulous service provider from changing the agreement before sending it to the billing server. It also reduces the communication load on the billing server, since it receives only one combined message rather than two separate messages from the service provider and the end user.

The full-scale billing server allows renegotiation of spending caps if the initial spending cap proves to be insufficient. However this capability was not implemented in the prototype.

Step 4 - Verifying Spending Cap and Credit

The billing server decrypts the two copies of the agreement and compares the end user’s version with the service provider’s version. If the two copies match, then the billing server checks if the end user has sufficient funds to pay for the transaction and places a hold on the end user's funds in the amount of the spending cap. It then sends an authorization to the service provider.

In the full-scale server, the end users can specify their preferred payment method; this could be historical billing, advance deposit or credit card. With historical billing the user receives a bill for the services that they used at the end of a specified period of time. Advance payment means that the user deposits funds with the billing server before using services, and receives a periodic statement of the services used and the funds remaining. With credit card billing, their credit card is billed when accumulated charges reach a specified limit. In addition to these three options, corporate users can use purchase orders, a form of historical billing, for making payments. The prototype allows deposit in advance as the only payment method.

Step 5 - Performing the Service

Messages are exchanged between the client and the service provider to perform the service—*e.g.* retrieve information, perform calculations, or spool a print file.

Step 6 - Generating an Invoice

After the service provider has rendered the service, it sends the billing server an invoice detailing the services performed and the actual amounts to be charged. The billing server checks whether the price information on the invoice is identical to the price information received earlier during the price negotiation stage. This protects the users from unscrupulous service providers. The billing server then forwards this invoice to the user. Since the identity and credit capacity of the end users were previously checked by the billing server, the service providers have assured payment for their services. The service providers are required to maintain an audit trail to handle customer inquiries which cannot be resolved by the billing server.

7. Account Management

End user accounts can go through various states, as illustrated in Figure 5. To open an account, the end user account administrator sends a request to the billing server. Once all of the information required for the creation of an account is entered, the account enters the “new” state. An account cannot begin to access services until the billing server's account administrator verifies and approves the account characteristics and the billing server's financial administrator verifies and approves the financial information. Once these verifications are complete, the account is activated. An account which has been activated enters the “active” state and is then allowed to accumulate charges for services it accesses through the billing server.

[put figure 5 here]

An account goes into the “deactive” state if it has an overdue balance for an unreasonable period of time. It goes back to the “active” state if the balance is paid. An account can also enter the “closed” state by end user request or if payment is not received while it is in the “deactivated” state.

Accounts may enter the “paid,” “written off,” or “referred to agency” states after being in the “closed” state. An account enters the “paid” state if the final balance due is paid in full. An account enters the “written off” state if the billing server financial administrator determines that payment for the balance due will not be received. An account enters the “referred to agency” state if the billing server's financial administrator refers the account to a collection agency.

Since the prototype handles only debit model accounts where users pay in advance, there is no need for an approval process. The “new” state is not needed. Again because of payment in advance and the credit check performed during transactions, end users' accounts cannot owe money to the billing server. Therefore, the “deactive,” “paid,” “written off,” and “referred to agency” states are also not needed. In the prototype, when an account is “closed” it is removed from the database. Therefore account states are not supported by the prototype.

Users can access their own account information at the billing server through an interface that allows them to view financial information, and to modify certain account characteristics. The full-scale billing server also provides on-line help to its users. The help, which can be accessed through a keyword search, consists of text screens describing how to perform basic operations. Since on-line help is not central to the billing server, it is not implemented in the prototype.

8. Conclusion

What distinguishes the Carnegie Mellon project from other piecemeal or service-specific solutions is its comprehensive analysis of the network services billing problem. The project has made two contributions: (1) it has highlighted the complex and challenging issues involved in the design of the Internet Billing Server; and (2) it has demonstrated the feasibility of its proposed solution through the successful design and implementation of the prototype. Even though the prototype implements only a subset of the full requirements and may have to be significantly modified, it is an important first step. A commercial service based on the concepts in the INI Internet Billing Server could be the key to the rapid growth of entrepreneurial service providers in the Internet environment.

For further information, please contact The Information Networking Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Tel: (412)-268-7195.

References

[1] John Quarterman. "In Depth. (What can businesses get out of the Internet?)" *Computerworld*, February 22, 1993, p.81.

[2] For a complete report of the project including the full Requirements Specification and Prototype Design Documents, contact the Information Networking Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890.

[3] Jennifer Steiner, Clifford Neuman and Jefferey Schiller. "Kerberos: an authentication service for open network systems." USENIX Winter Conference, 9-12 February 1988, Dallas, Texas.

[4] Richard Batelaan, *et.al. An Internet Billing Server: System Requirements.* Carnegie Mellon University Information Networking Institute, August 1992.

[FIGURES]

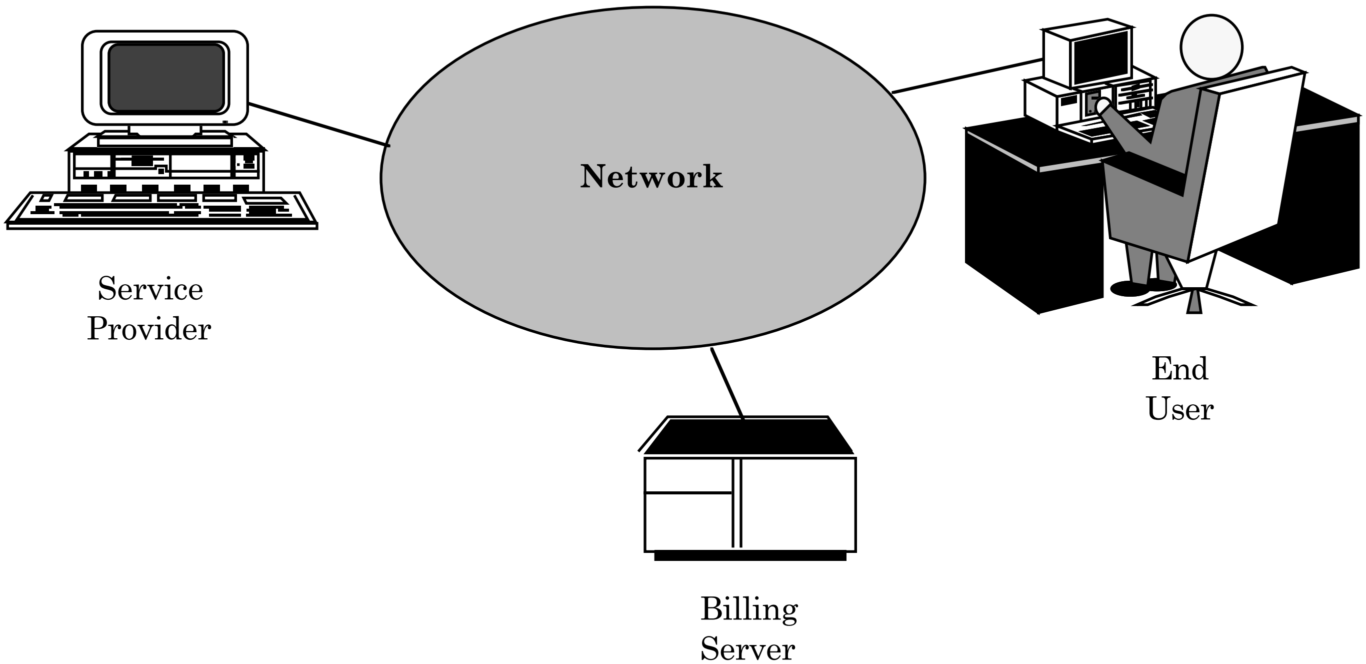

Figure 1. Electronic Marketplace

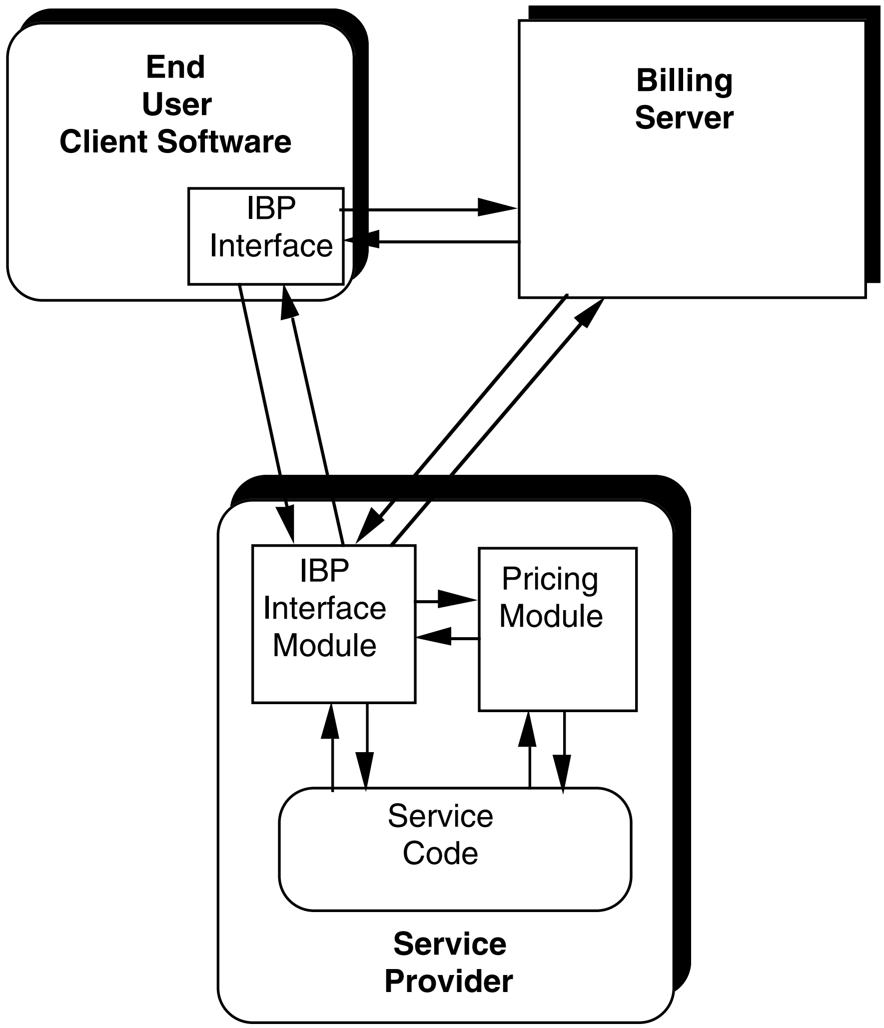

Figure 2.

Interface Libraries Provided by Internet Billing Service

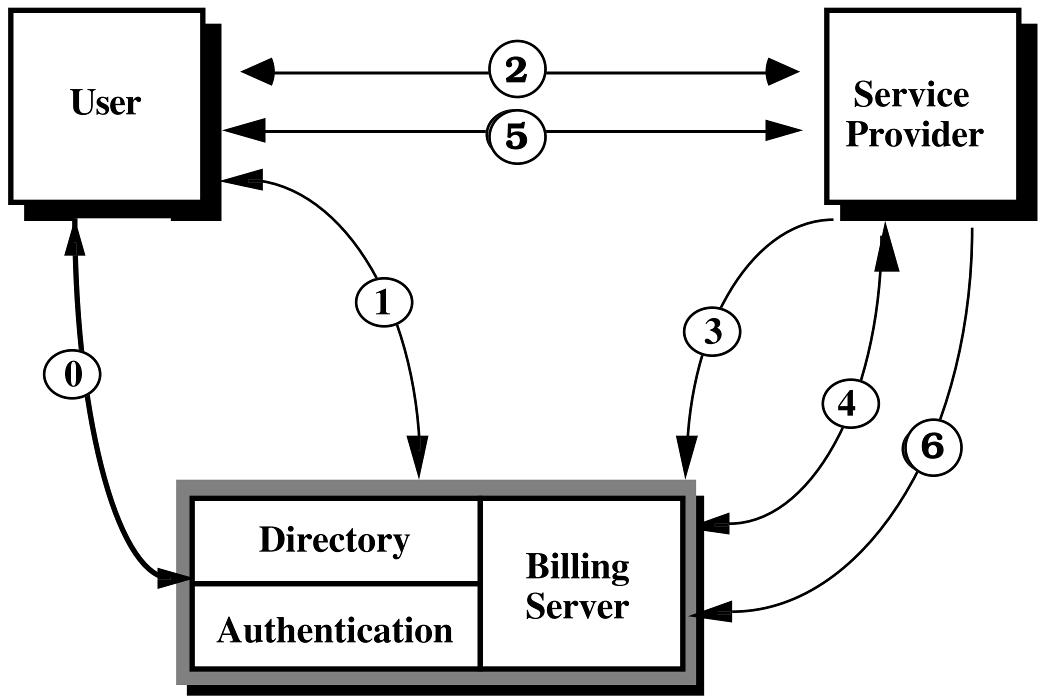

Figure 3: Transaction Sequence for the Billing server

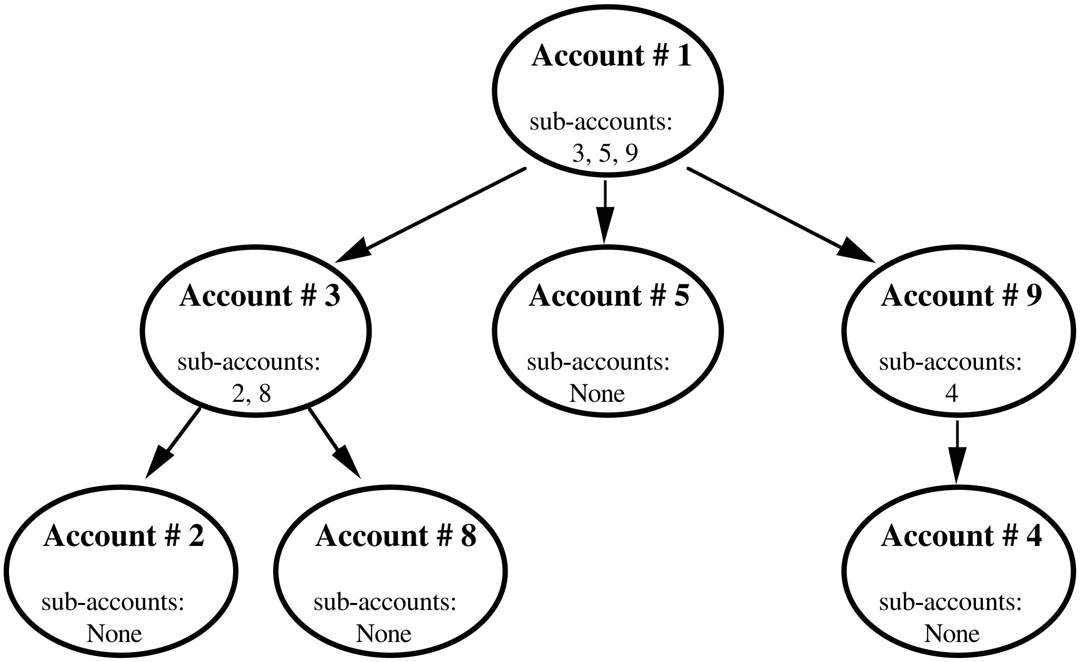

Figure 4: An Example Account Hierarchy

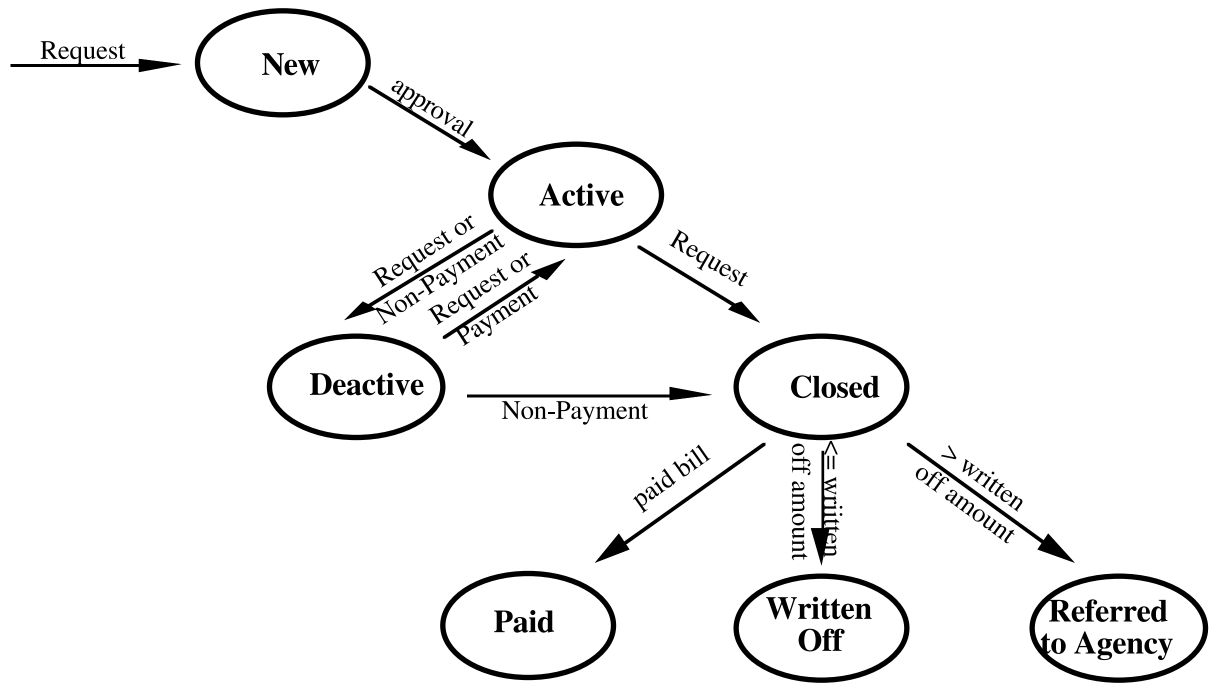

Figure 5: Account State Diagram