CHAPTER 6

CONCLUSION

This chapter presents the contributions and critical analysis of the proposed Integrated Accounting architecture for web services. Suggestions for future work have also been outlined. Before proceeding with the review of the work done, the objectives stated earlier in the introductory chapter are recalled.

The primary objective of the research work is to propose an integrated accounting architecture to automate the accounting functions of web services. The focus of the proposed system is to support different pricing models for service providers. A metering framework has also been designed to measure and record the service usage information of web services. The research work has also been proposed to design, develop and integrate the modules for billing, payment, accounting, and auditing.

6.1 CONTRIBUTIONS OF THE THESIS

The research work has proposed an integrated accounting architecture to automate the various accounting functions for web services including charging and auditing. The accounting functions proposed to be automated are metering, charging, billing, accounting and auditing.

The metering component has been designed to measure and record the web service usage of the service clients. The metered data is used to automate the billing function that raises bills on users for services consumed.
The accounting module is designed to update the payments obtained from the users. The accounting information can be used to generate various management reports that would facilitate decision making. Audit is a statutory function for verification of the corrections of usage and associated payments, and to ensure that proper controls and procedures are in place. The audit module provides a query facility to users, service providers and third party auditors, to track the service usage and related payments and other information.

Prior work has not addressed the need for an overall architecture that integrates and automates the accounting functions. Some previous work has primarily discussed only the metering function, but the implementation issues have not been dealt with. Other accounting functions have been outlined, but the design, implementation and integration requirements have not been dealt with in detail.

An accounting architecture is proposed for automating the accounting functions outlined as above. The AAAAC server component is responsible for the management of the accounting activities. All accounting data, namely, usage data records, bills, payments and all other related information are maintained at the AAAAC server. The AAAAC client is responsible for metering the service usage. These client components are located at the service providers to log the usage statistics. In comparison, AWS architecture provides for only metering and billing functions. The pricing scheme in AWS has been implemented using tags which are applied to AWS resources. Cost allocation report is prepared with usage and cost as per tags.

Different pricing schemes of service providers are proposed to be implemented using policies. Prior work has discussed and analyzed the need for different payment possibilities. AWS pricing scheme includes standard
units and rates, hourly metering and elastic demand. The Priceline model (Priceline.com) is a Name -Your –Own –Price (NYOP) system where a buyer specifies a price for a product or service and asks sellers to match that combination. However the buyers do not know the suppliers until they make a non-refundable purchase. In ‘chicing.com’ NYOP model, buyers were allowed to select the business as well as pricing and the value. However if no dealer accepted the user price, the user gets nothing.

The research work has proposed the use of charging and accounting policies to provide ease and flexibility in specifying the different charging patterns, and also to facilitate frequent updates of the charging schemes to match the changing business requirements. The pricing patterns too can be varied depending on user categorization. Policies also facilitate querying of pricing information by service consumers.

The research work has proposed the use of a WS-Policy standard to define the charging policies. WS-Policy is widely used and supported by many applications. For instance, software packages are available to support the querying facility of policies specified in the WS-Policy standard. A policy generation tool has been developed to automate the creation of a charging policy in the WS- Policy standard format. The tool provides the option to create, view and maintain the charging and accounting policies of service providers.

COPS protocol supports policy control. It is a simple query and response protocol that can be used to exchange policy information between a policy server and its clients. Policy definitions are stored at the servers and the policy enforcements are carried out at the clients. It was primarily defined for use with quality of service protocols such as RSVP. COPS does not support user authentication and protocols such as RADIUS, DIAMETER were proposed to overcome the disadvantages.
The standard RADIUS protocol was primarily developed for authentication and accounting of telecommunication services, and later extended to the Internet. The research work has designed and developed WS-RADIUS protocol to perform the metering function for web services. The AAAAC client component uses the WS-RADIUS protocol to perform metering of the service usage at the service providers. The server component of the protocol is responsible for the initiation and coordination of the metering activities at the service providers.

The research work proposes the use of the IPDR standard to maintain the metered and accounting data, such as, usage records, bills and payments. IPDR is a telecommunication standard for the exchange of accounting related data. Prior work has discussed the measurement issues for service oriented computing, and has suggested the use of logs to store the metered data. However, the log format and content have not been discussed.

The proposed design of the accounting modules has been implemented and tested using a sample E-book service. The web application has been designed with multiple web services with provision for different functions, such as on-line reading, book search, book download etc.

WS-RADIUS protocol has been used to meter the web service usage. A sample application was developed and tested to analyze the performance of the proposed design. The impact on revenue when the user is charged based on the usage has been illustrated using graphs for the sample application.

6.2 CRITICAL ANALYSIS

The research work was implemented and the performance analyzed using a single meter component in the service providers. When the number of
web services offered by a service provider is large and usage instances become high, then the performance may not be satisfactory with a single meter. Multiple meters would be required to meter the usage statistics. The challenge would be to decide and configure the division of work among the multiple meters. It would be also necessary to address the question of how to handle the interactions between the various meters and co-ordinate their functioning.

The research work was implemented and tested, using one AAAAC server to co-ordinate and manage multiple clients. It may be necessary to have multiple such servers to manage heavy usage of web services, to avoid bottlenecks at the server. Research challenges include the design of interactions between the AAAAC servers, and coordinating the functioning of the individual servers.

6.3 FUTURE WORK

The service providers and web services offered by them may be configured into different administration domains. Future work can address the accounting needs related to intra-domain and inter-domain requirements.

Future research could also focus on federated accounting. A mechanism for federated data collection that spans multiple administrative domains can be designed and developed. As future work, the charging policy can be extended to include charging based on Quality of service (QOS) and sharing of resources between multiple users.

Other schemes like location based pricing or distance based pricing could be included to make the charging policy more varied and comprehensive. The protocol can be extended to cover metering for such
complex web services, and to comprehensively cover and test different functionalities of the web services.

Future work could also focus on security. Security is an important aspect of web services accounting. Accounting data have to be confidential and should not be accessed by the other people. Security requirements such as confidentiality, data integrity, and privacy must be addressed by the technology.

6.4 SUMMARY

This thesis addresses the key challenges relating to the accounting of web services. A generic architecture has been proposed to automate the accounting functions of web services, namely, metering, charging, billing, accounting and auditing.

The accounting architecture is designed to support the different pricing schemes that may be offered by multiple service providers. Varying pricing schemes have been supported with the use of Charging and Accounting policies. Policies make it easy to specify and update changes in the pricing structure. The WS-Policy standard has been proposed to define the policies. Different pricing strategies, such as, charging based on web service usage time, the content accessed, and additional functions used, can be easily implemented using policies.

The WS-RADIUS protocol has been developed to perform the metering function for web services. The client component of the protocol is implemented in the AAAAC client to meter the service usage at the service providers. The server component controls the initiation and coordination of the metering activities at the service providers.
The research work is also aimed to design and standardize the format and content of the logs, used for maintaining the service usage, billing and payment. IPDR telecommunication standard has been proposed for the design and maintenance of the service usage and accounting data.

The proposed research work has been implemented and tested, using a sample application, and the results and performance analysis have been presented.