



(19) **United States**

(12) **Patent Application Publication**

Atwal et al.

(10) **Pub. No.: US 2003/0061067 A1**

(43) **Pub. Date: Mar. 27, 2003**

(54) **SYSTEM AND METHOD FOR WEB SERVICES PACKAGING**

(22) Filed: Sep. 23, 2002

(75) Inventors: **Amar Atwal**, Ottawa (CA); **Joseph Chiu**, Ottawa (CA); **Frederic LeBel**, Ottawa (CA); **Stephen Mereu**, Ottawa (CA); **Rick Mutzke**, Ottawa (CA); **Raul Rupsingh**, Brampton (CA); **Matt Schnarr**, Ottawa (CA); **Don Slaunwhite**, Ottawa (CA)

**Related U.S. Application Data**

(60) Provisional application No. 60/324,191, filed on Sep. 21, 2001.

**Publication Classification**

(51) **Int. Cl.<sup>7</sup>** ..... **G06F 17/60**  
(52) **U.S. Cl.** ..... **705/1**

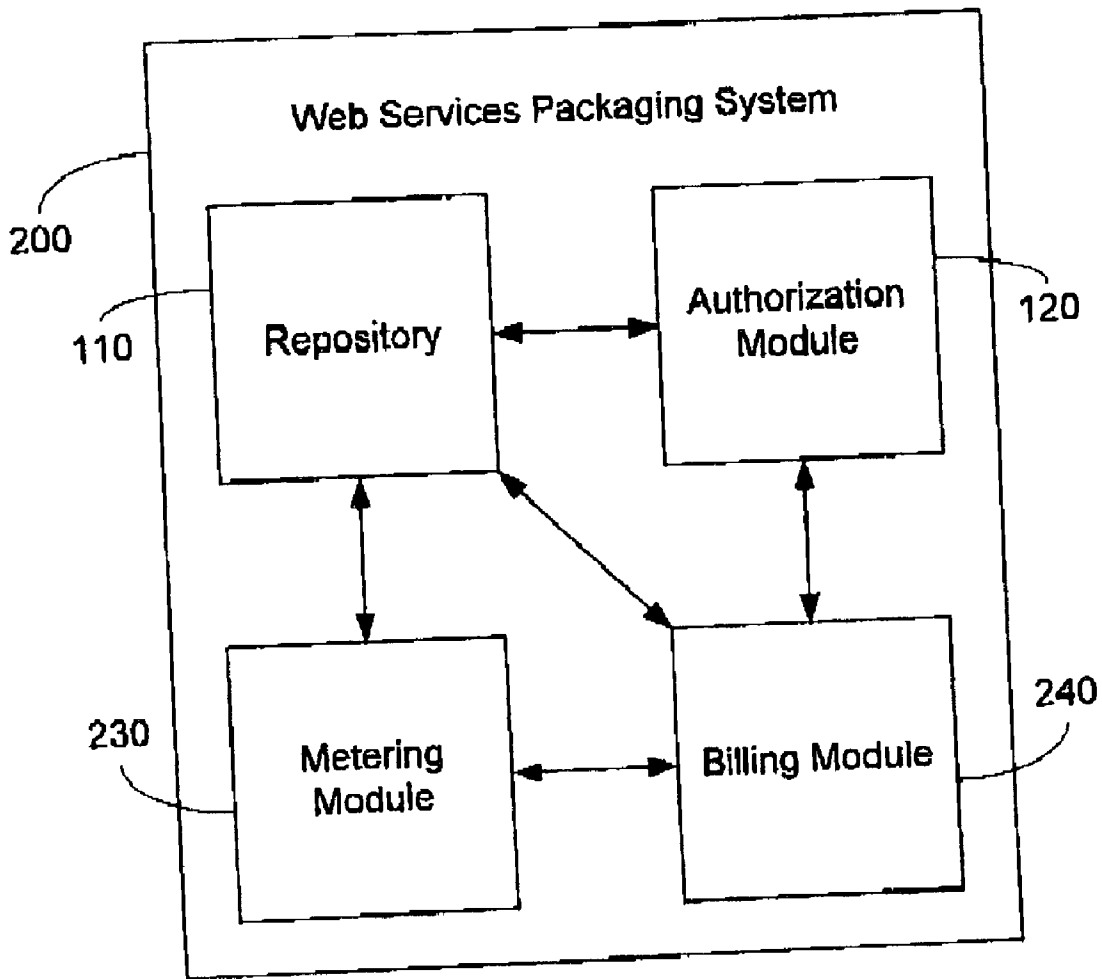
Correspondence Address:  
**Norman P. Soloway**  
**HAYES SOLOWAY P.C.**  
**130 W. Cushing Street**  
**Tucson, AZ 85701 (US)**

(73) Assignee: **Corel Corporation**

(21) Appl. No.: **10/252,897**

(57) **ABSTRACT**

A web services packaging system is provided. The web services packaging system comprises a repository for storing information relating to methods of a plurality of web services organized into bundles, and an authorization module for authorizing a client application to access methods in one of the bundles. Each bundle has one or more methods from one or more web services.



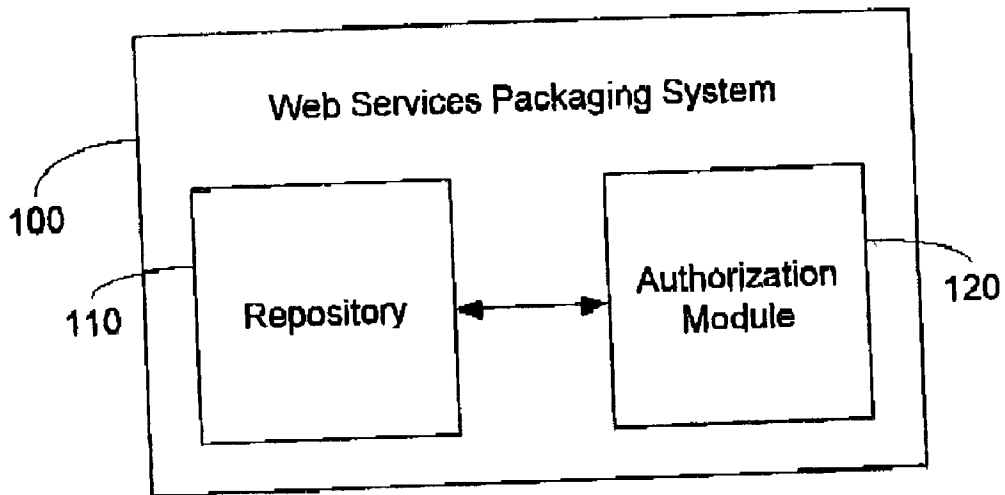


Figure 1

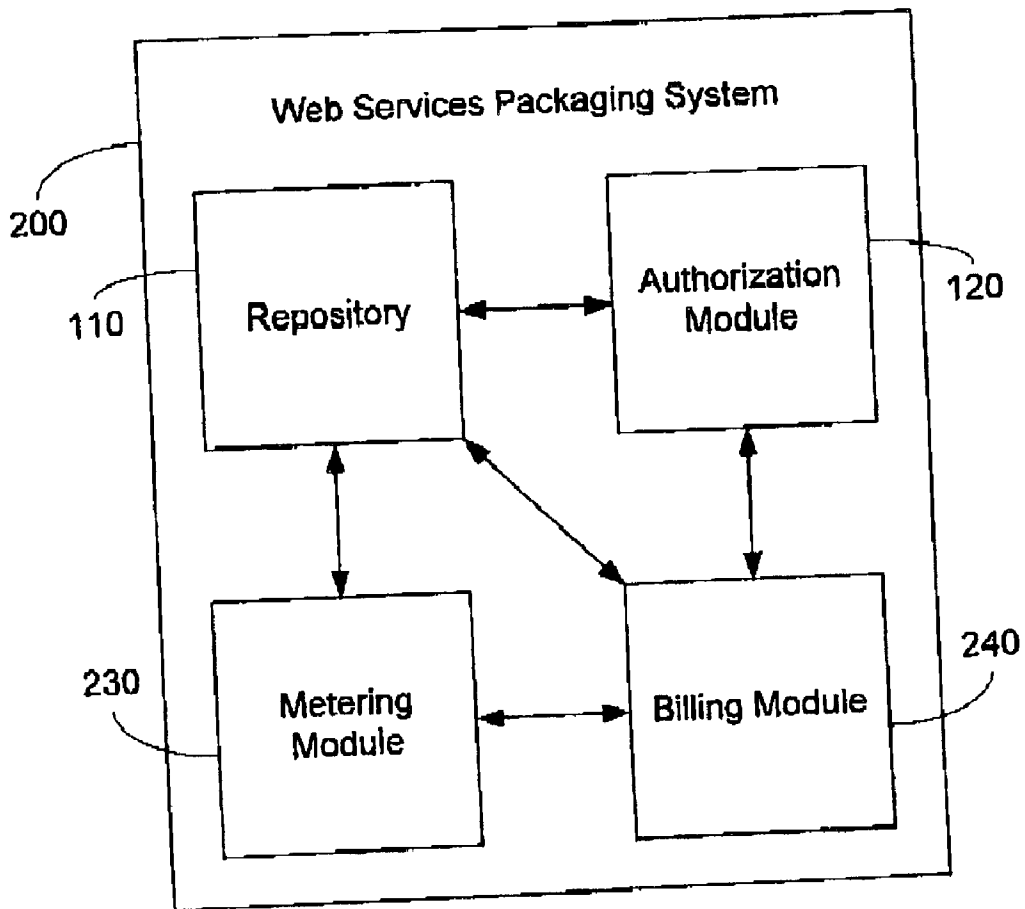


Figure 2

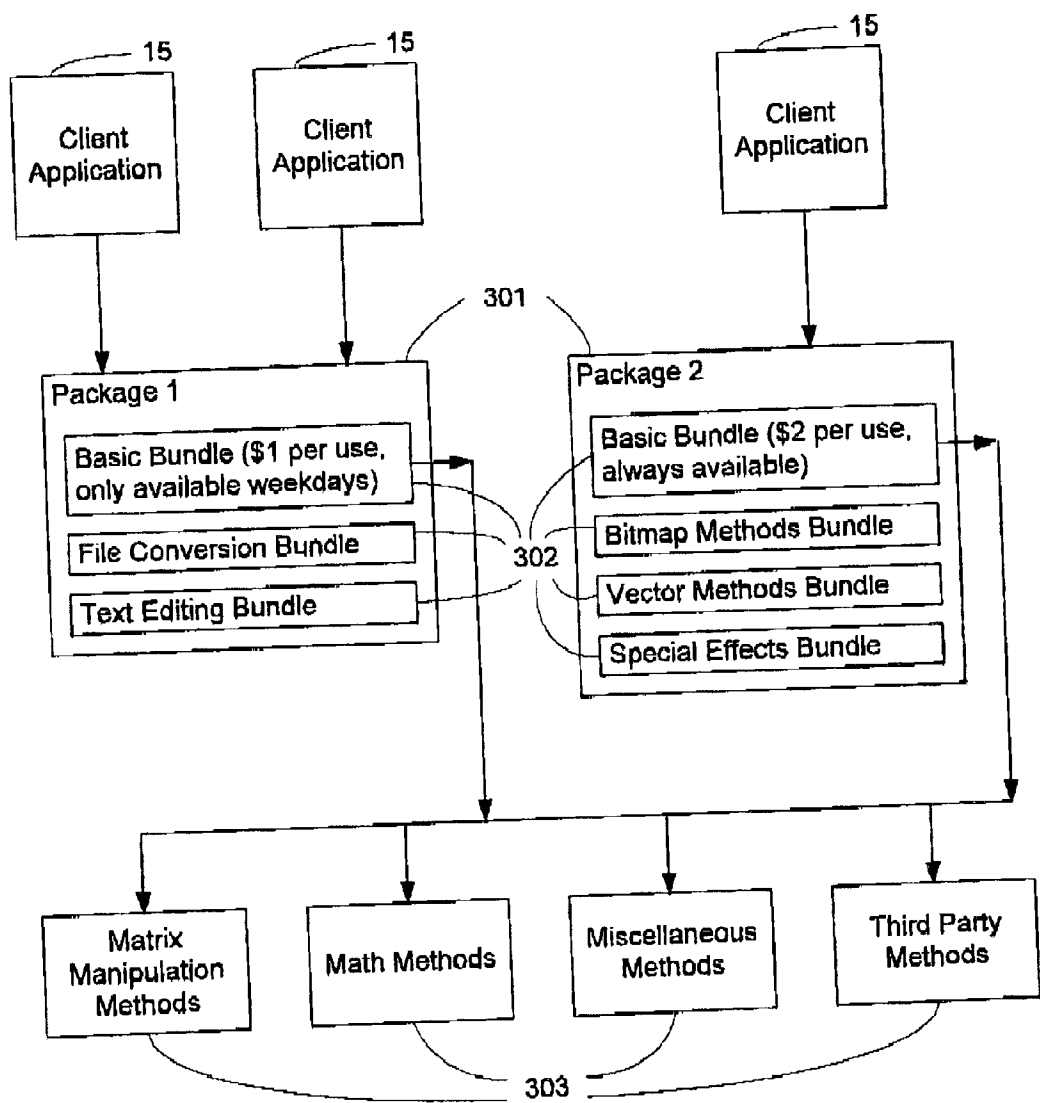


Figure 3

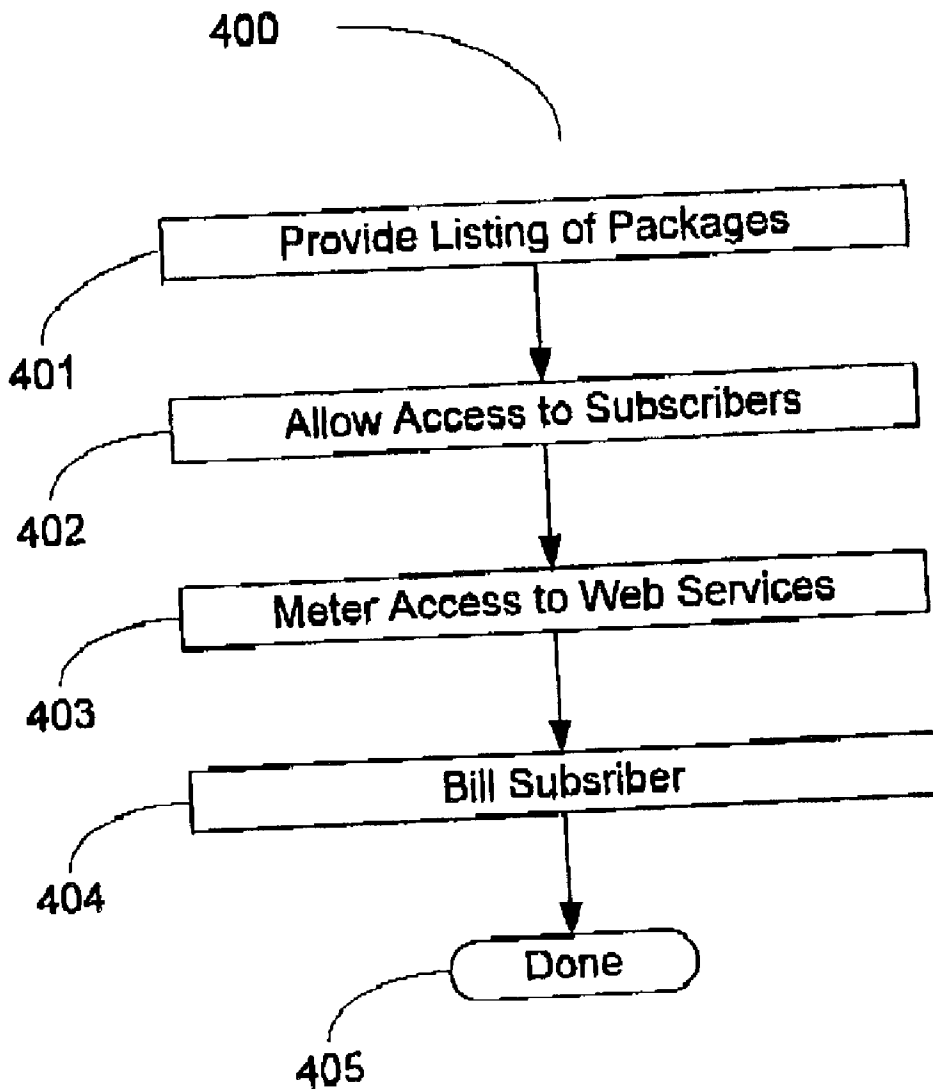


Figure 4

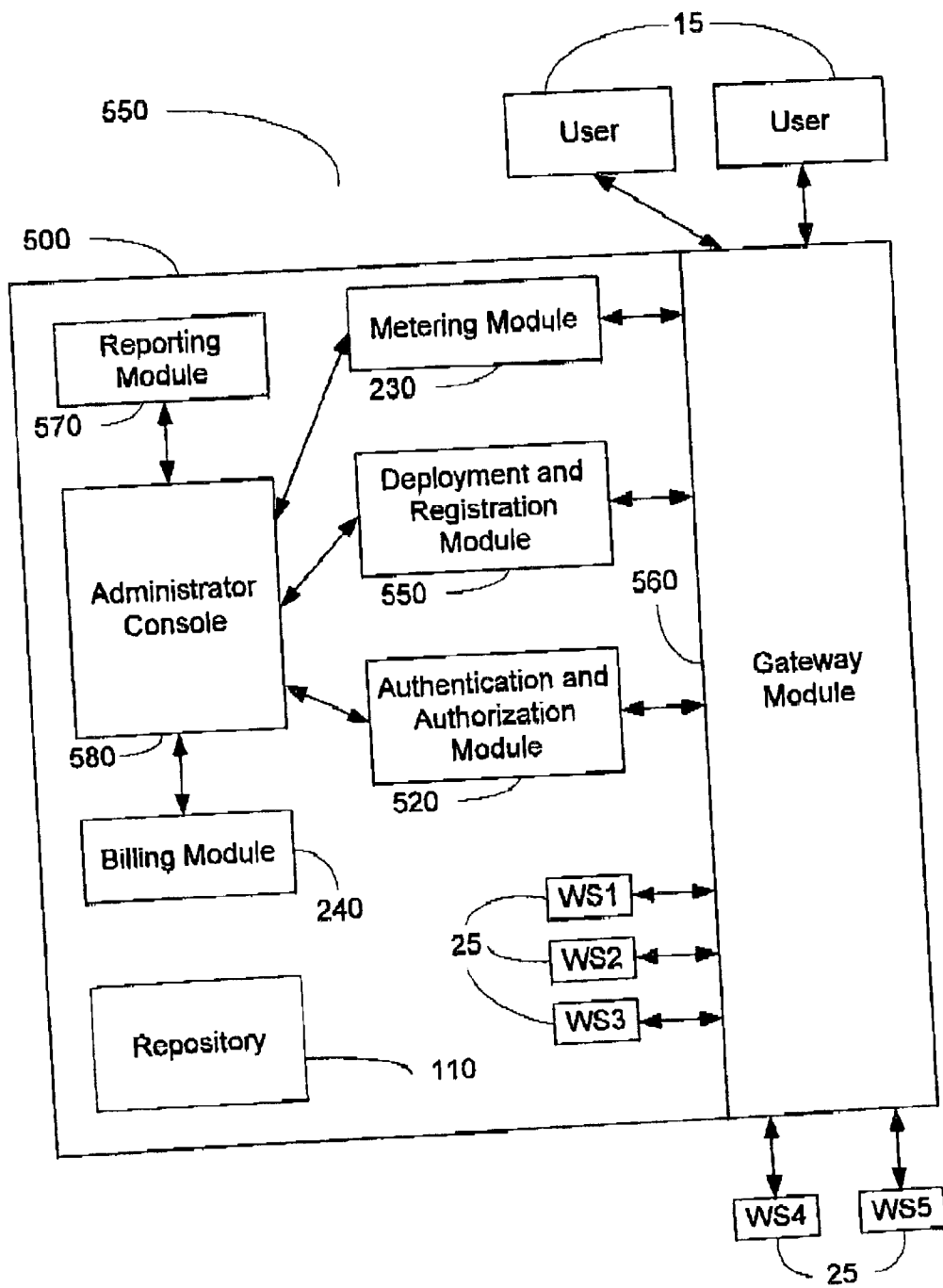


Figure 5

## SYSTEM AND METHOD FOR WEB SERVICES PACKAGING

### PRIORITY

[0001] This patent application claims priority from U.S. Patent Application No. 60/324,191 entitled "Web Services Infrastructure", Atwal et. al., filed Sep. 21, 2001.

### FIELD OF THE INVENTION

[0002] This invention relates to an infrastructure for billing and authorizing web services. In particular, the invention relates to web services packaging.

### BACKGROUND OF THE INVENTION

[0003] Software developers wish to provide programmatic functionality over the Internet through the creation of web services. These web services provide some valuable technology in which the developer has expertise. A web services is often deployed in such a way that the user of the web service has a direct connection with a server hosting the web service. For example, if there are ten servers hosting different web services, then there are ten "connection points" into the different web services.

[0004] One problem that arises from this process of exposing the web services for consumption over the web by client applications is that in order to create revenues, the web services must also incorporate billing of the users, auditing of all transactions that occur, administration of users and billing schemes and any other common functionality that arises from the fact that a web service is being sold to a customer. For example, each web service provider can host any number of web services. Each web service can contain any number of functions (or methods), which can potentially be billed in a different manner. Service providers might wish to tailor billing and authorization schemes for each function to its clients. Therefore, the permutation of user to billing/authorization pairs to be maintained can be large.

[0005] There are many complexities regarding billing and authorization. Each billing and authorization method can take on any number of parameters. For example, a pay per use billing scheme can be tagged with any price. Groups of functions, perhaps across web services, might only make sense to be billed and authorized the same way. For example, there are blur image effect filters published by different companies and distributed across different web services. Billing schemes can vary across time and temporal cycles. For example, a web service provider might want to charge more during busy hours. Authorization may be independent of billing schemes. For example, the administrator might want to temporarily deny a user from accessing certain functions. However, the billing scheme should remain effective once the restriction is lifted. Finally, some billing methods are intrinsically tied to authorization. For example, the client might pay \$100 to use a given web service function one hundred times. Afterwards, the web service will have to become unavailable.

[0006] There is currently no way of providing billing and authorization flexibility in web services. It is therefore desirable to provide means for providing better billing and authorization of web services.

### SUMMARY OF THE INVENTION

[0007] It is an object of the invention to provide a novel system and method for providing better billing and autho-

rization of web services that obviates or mitigates at least one of the problems described above.

[0008] In an aspect of the present invention, there is provided a web services packaging system. The web services packaging system comprises a repository for storing information relating to methods of a plurality of web services organized into bundles, and an authorization module for authorizing a client application to access methods in one of the bundles. Each bundle has one or more methods from one or more web services.

[0009] In another aspect of the present invention, there is provided a method for packaging web services. The method comprising steps of storing information relating to methods of a plurality of web services organized into bundles, and authorizing a client application to access methods in one of the bundles. Each bundle has one or more methods from one or more web services.

[0010] In another aspect of the present invention, there is provided computer readable media storing the instructions and/or statements for use in the execution in a computer of a method for packaging web services. The method comprises steps of storing information relating to methods of a plurality of web services organized into bundles, and authorizing a client application to access methods in one of the bundles. Each bundle has one or more methods from one or more web services.

[0011] In another aspect of the present invention, there is provided electronic signals for use in the execution in a computer of a method for packaging web services. The method comprising steps of storing information relating to methods of a plurality of web services organized into bundles, and authorizing a client application to access methods in one of the bundles. Each bundle has one or more methods from one or more web services.

[0012] In another aspect of the present invention, there is provided a computer program product for use in the execution in a computer of a method for packaging web services. The computer program product comprises a repository for storing information relating to methods of a plurality of web services organized into bundles, and an authorization module for authorizing a client application to access methods in one of the bundles. Each bundle has one or more methods from one or more web services.

[0013] Other aspects and features of the present invention will be readily apparent to those skilled in the art from a review of the following detailed description of preferred embodiments in conjunction with the accompanying drawings.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The invention will be further understood from the following description with reference to the drawings in which:

[0015] **FIG. 1** is a diagram showing a web services packaging system in accordance with an embodiment of the invention;

[0016] **FIG. 2** is a diagram showing bundling on a per-client application basis in accordance with an embodiment of the invention;

**[0017]** FIG. 3 is a flowchart showing a method for billing and authorization of web services in accordance with an embodiment of the present invention; and

**[0018]** FIG. 4 is a diagram showing a web service deployment environment in accordance with an embodiment of the invention.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

**[0019]** A web service comprises one or more methods which users or client applications may invoke. Different web services may have a different amount of methods. In embodiments of the present invention, web services are registered with a central repository. Identifications of the methods of the web services are stored in the repository and grouped into bundles. A bundle of methods contains one or more methods from one or more web services. For example, consider four web services where a first web service contains 10 methods, a second web service contains 20 methods, a third web service contains 100 methods, and a fourth web service contains 50 methods. A bundle may be created to contain one method which belongs to the first web service, two methods which belong to a second web service, and six methods which belong to a fourth web service. Another bundle may be created to contain six methods which belong to the first web services, 11 methods which belong to the second web service, 71 methods which belong to the third web service, and all 50 methods which belong to the fourth web service. Other combinations of methods belonging to a plurality of web services containing a plurality of methods may be created.

**[0020]** FIG. 1 shows an example of a web services packaging system 100 in accordance with an embodiment of the present invention. The web services packaging system 100 comprises a repository 110, and an authorization module 120. The repository 110 contains information relating to methods of a plurality of web services organized into bundles. Each bundle has one or more methods from one or more web services. The repository 110 may be a database with fields and tables for storing the web service addresses, web service methods, web service method names, client application rights, costs, and other information. The authorization module 120 is used for authorizing a client application to access methods in one of the bundles. The authorization module 120 restricts access to the web service methods to authorized or registered client applications. The authorization module 120 communicates with the repository 110 to exchange information for authorization purposes. These components may be software code implemented on a server or code embedded in hardware. Components may be added or removed from the web services billing system 100. For example, a registration unit may be added to the web services billing system 100 for registering web service with the system 100.

**[0021]** FIG. 2 shows another example of a web services packaging system 200 in accordance with an embodiment of the present invention. The packaging system comprises a repository 110, an authorization module 120, a metering module 230, and a billing module 240. The metering module 230 is used to track the usage of web services by client applications. The usage of the web services may be logged in the repository 110. The billing module 240 is used to bill client applications for their use of the web services.

**[0022]** The repository 110 communicates with the other components and exchange information. The authorization module 120 communicates with the repository 110 and the billing module 240, and exchanges information for authorization purposes. The metering module 230 communicates with the repository 110 and the billing module 240, and exchange information. The billing module 240 also communicates with the other components and exchange information for billing purposes. The metering module 230 records events that take place that can later be used for analysis. One client of the metering module 230 is the billing module 240. Billing schemes may be created such that the billing module 240 requires information regarding which web services methods 303 a client application 15 has used. Such information is tracked by the metering module 230 and stored in the repository 110. Alternatively, the metering module 230 may pass the web services methods 303 usage information directly to the billing module 240 to be used or stored for later use.

**[0023]** The following description will refer to web services packaging systems 200. However, the authorization schemes portion of the following description also applies to the web services packaging system 100.

**[0024]** With reference to FIG. 3, the following describes how the web services packaging system 200 is used to package web services. FIG. 3 shows client applications 15 assigned to packages 301 comprising of bundles 302. The bundles 302 comprise web services methods or routines 303 which a client application 15 may use when calling web services methods 303. In this example, the basic bundle 302 contains matrix manipulation, math, miscellaneous, and third party web services methods 303.

**[0025]** Web services methods are registered in the repository 110. The web services methods are then grouped into bundles 302. Client applications 15 may also register with the web services packaging system 200. Client applications 15 usage rights relating to the bundles 302 are stored in the repository 110. Thus, by accessing the repository 110 to determine if a client application 15 has usage rights to a bundle 302, the authorization module 120 determines if a client application 15 is authorized to access a web services method in that bundle 302.

**[0026]** A package 301 is a set of bundles 302. For example, in FIG. 3, Package 1 contains a basic bundle, a file conversion bundle, an a text editing bundle. Package 2 contains a basic bundle, a bitmap methods bundle, a vector methods bundle, and a special effects bundle. As is described above, a bundle is a set (or group) of web services methods 303 which may come from several different web services. Other packages 301 containing other combinations of bundles 302 may be created.

**[0027]** Web services methods 303 of the bundles 302 in a package 301 are billed and authorized in the same manner. Advantageously, a bundle 302 simplifies the authorization to access a set of web services methods 303 in the bundle 302. If the user or client application has authorization to access the bundle 302, then the user or client application has authorization to access each method 303 in the bundle 302.

**[0028]** A package 301 allows for customization of billing for the bundles 302. For example, consider two packages 301, Package 3 and Package 4, where each package contains

the same two bundles **302**, Bundle **1** and Bundle **2**. Package **3** may set Bundle **1** on a per-per-use billing scheme and Bundle **2** on a flat rate billing scheme. Package **4** may set both Bundle **1** and Bundle **2** on a flat rate billing scheme. Thus, packages **301** allow for billing flexibility and billing customization of bundles **302**. Furthermore, by having different combinations of methods **303** in different bundles **302**, packages **301** allow for billing flexibility and billing customization of web services methods **303**.

[**0029**] Advantageously, providing web services functionality through packages **301** eliminates much of the logistics needed to manage/present billing and authorization requirements. Tables or records in the repository **110** may contain identifications of methods **303** belonging to web services registered with the web services packaging system **200**. Methods **303** of different web services may be organized together into a bundle **302**.

[**0030**] Packages **301** organize bundles **302** into default billing/authorization plans to which client applications **15** may subscribe. The same bundle can reside in different packages **301**, with different billing/authorization specifications. In the example shown in **FIG. 3**, the basic bundle is contained in both packages **301**. Thus, subscribers can be offered combinations of methods **303** they desire without having to subscribe to a web service.

[**0031**] Preferably, a client application **15** may only subscribe to one package **301**. This limitation advantageously reduces the logistics since allowing a client application to subscribe to more than one package **301** may create a potential scenario of the two packages **301** both containing the same bundle **302** (or even different bundles **302** with the same methods) but associated with a different billing/authorization plan. For example, the basic bundle in Package **1** is priced at \$1 per use and is limited to use on weekdays. The basic bundle in Package **2** contains the same web services methods the basic bundle in Package **1**. However, there are no limitations to availability for the basic bundle in Package **2** and therefore, a price of \$2 per use is assigned to Package **2**.

[**0032**] In one example of a billing scheme, a web service administrator can override the default billing/authorization plan specified in the package **301** to which a client application **15** subscribes. Advantageously, this allows for further flexibility for billing client applications **15**.

[**0033**] Billing and/or authorization schemes may be created to vary depending on the time of day, day of the week, week of the month, etc. Alternatively, billing/authorization may expire after a set amount of time has passed since the package **301** was assigned to the client application. For example, a client application may sign up for a trial package of web services that allows the client application to use the services for a trial period, like a month, after which access will expire. Thus, the system **200** may be designed to be aware of time and temporal cycles using time stamps as to when the client application signed up, the server clock, and other means of measuring time available to the server upon which the web services packaging system is implemented.

[**0034**] The metering module **230** and authorization module **120** modules may employ a plug-in architecture which allows new components to be easily added without changing the core architecture. There are several ways in which this

plug-in architecture can be accomplished. One example provides that the core architecture searches for the existence of billing/authorization schemes on a particular location of the server in which the web services packaging system **200** is implemented. Since the architecture can be designed to always search the same location, additional schemes can be added. For example, other bundles **302** may be added to a package **301**, or other web services methods **303** may be added to a bundle **302**.

[**0035**] The determination of a client application's permissions to use a given resource may depend on the current standing of the client application's billing data. For example, if a client application's billing type specifies that the client application has paid a certain fee which allows the client application to make X number of calls to a bundle of methods, then on the (X+1)th call that client application will not be authorized to make the call. To achieve this, the authorization module **120** may query the metering module **230**, or the metering module **230** may set some state in the authorization module **120**.

[**0036**] **FIG. 4** shows a method for billing and authorization of web services (**400**) in accordance with an embodiment of the present invention. The method (**400**) begins with providing a listing of web service functionality (i.e., a listing of packages **301**) to which a client application **15** may subscribe (**401**). Subscribed client applications are given access to method calls **303** of web services contained in bundles **302** of a package **301** to which they are a subscriber (**402**). Access to web services method calls **303** contained in the bundles **302** is metered (**403**). The web services usage is billed pursuant to the billing scheme of the package **301** (**404**). The method is done (**405**).

[**0037**] Other steps may be added to the method (**400**), such as registering a web service with the web services packaging system **200**, registering a client application **15** with the web services packaging system **200**, and storing in the repository **110** the web services and the client applications **15** which are authorized to access the web services methods **303**.

[**0038**] **FIG. 5** shows an example of a web service deployment environment **501** in accordance with an embodiment of the present invention. The web service deployment environment **501** comprises client applications **15**, a web services management system **500**, and remote web services **25**. The web services management system **500** comprises a repository **110**, an authentication and authorization module **520**, a metering module **230**, a billing module **240**, a deployment and registration module **550**, a gateway module **560**, a reporting module **570**, an administrator console **580**, and locally hosted web services **25**. All components have access to the repository **110**. Components may be added or removed from the web services management system **500**.

[**0039**] The repository **110**, the metering module **230**, and the billing module **240** are similar to those described above. The administrator console **580** is used to perform administration tasks, including creating bundles **302** and packages **301**, provisioning client applications, setting or modifying access rights, generating reports, and other tasks desired for administration purposes. Another client of the metering module **230** is the administrator console **580** that enables the administrator of the web services packaging system **500** to track how the system **500** has been used. Statistics may also



be generated and stored in the repository **110** in order to help the administrator make business decisions such as offering new services, adjusting prices, and other business decisions relating to the packaging of web services.

[0040] The authentication and authorization module **520** is used when client application attempts to access a web services method. The system **500** calls on the authentication and authorization module **520** to determine if the client application indeed is authentic and authorized to use the web service **25**. The metering module **230** is used to track the usage of web services and packages **301**. The gateway module **560** is used for routing communication between client applications **15** and web services **25**. The reporting module **570** generates reports on the usage of the system **500** by administrators for billing and auditing purposes.

[0041] The deployment and registration module **550** is used to register web services and/or client applications. When a web service registers with the system **500**, its information is stored in the repository to be used for creating bundles **302** and packages **301**. When a client application registers with the system **500**, its credentials are stored in the repository to be used for authentication, authorization, and billing purposes. A remote web service provider may be billed for registration into the system **500** or payed a percentage of the revenues for packages sold to client applications.

[0042] The purpose and practical use of the web services packaging system **500** is that it be deployed on the servers which host a company's web services **25**. The web services **25** are registered with the deployment and registration module **550** and an administrator may set up client application rights, billing schemes and any other infrastructure for these services. Furthermore, a host company's web services **25** may be complimented with remotely located web services **25**. In addition the host company may host virtual web services which are comprised of other web services but connected in such a way to provide new functionality.

[0043] This web services packaging system **500** provides value to the web services provider, in that it allows for the tailoring of functionality on a per-client application basis. The alternative would be to develop different versions of its web services **25** for different customers, thus requiring duplication of code and additional strains on disk space, to maintain multiple copies of similar code. The web services packaging system **500** provides the framework around which the web services provider can track usage of the web services **25** and charge accordingly, on a per-client application basis.

[0044] The web services packaging system **500** provides an elegant, comprehensive and extendable system for handling the most complex of web services **25** billing and authorization scenarios. To web services providers, it means reduced administration, resource and upgrade costs. To the users of client applications, it opens up the possibility of finely grained customized billing and authorization options.

[0045] The web services packaging system **100**, **200** according to the present invention may be implemented by any hardware, software or a combination of hardware and software having the above described functions. The software code, either in its entirety or a part thereof, may be stored in

a computer readable memory. Further, a computer data signal representing the software code which may be embedded in a carrier wave may be transmitted via a communication network. Such a computer readable memory and a computer data signal are also within the scope of the present invention, as well as the hardware, software and the combination thereof.

[0046] While particular embodiments of the present invention have been shown and described, changes and modifications may be made to such embodiments without departing from the true scope of the invention.

What is claimed is:

1. A web services packaging system comprising:

a repository for storing information relating to methods of a plurality of web services organized into bundles, each bundle having one or more methods from one or more web services; and

an authorization module for authorizing a client application to access methods in one of the bundles.

2. The web services packaging system as claimed in claim 1, wherein at least one bundle has one or more methods from at least two web services.

3. The web services packaging system as claimed in claim 1, wherein client application usage rights to the bundles are stored in the repository.

4. The web services packaging system as claimed in claim 1, further comprising:

a metering module for logging the access; and

a billing module for billing the client application.

5. The web services packaging system as claimed in claim 4, wherein the bundles are organized into packages.

6. The web services packaging system as claimed in claim 5, wherein client application usage rights to the packages are stored in the repository.

7. The web services packaging system as claimed in claim 5, wherein prices of the packages are stored in the repository.

8. The web services packaging system as claimed in claim 1, further comprising an administrator console for administering the web services packaging system.

9. The web services packaging system as claimed in claim 1, further comprising a deployment and registration module for registering web services and client applications.

10. The web services packaging system as claimed in claim 1, further comprising a reporting module for reporting usage of the web services packaging system for auditing and billing purposes.

11. The web services packaging system as claimed in claim 1, further comprising a gateway module for routing communication between the client application and a web service.

12. The web services packaging system as claimed in claim 1, wherein the repository is a database with fields and tables for storing the web services addresses and method calls.

13. A method for packaging web services, the method comprising steps of:

storing information relating to methods of a plurality of web services organized into bundles, each bundle having one or more methods from one or more web services; and

authorizing a client application to access methods in one of the bundles.

14. The method as claimed in claim 13, wherein the step of storing comprises the step of storing at least one bundle having one or more methods from at least two web services.

15. The method as claimed in claim 13, further comprising the step of organizing a group of web services methods into a bundle.

16. The method as claimed in claim 13, wherein the storing step comprises the step of storing client application usage rights to the bundles in the repository.

17. The method as claimed in claim 13, further comprising the steps of:

logging the access; and

billing the client application.

18. The method as claimed in claim 17, further comprising the step of organizing the bundles into packages.

19. The method as claimed in claim 18, further comprising the step of storing client application usage rights to the packages in the repository.

20. The method as claimed in claim 18, further comprising the step of storing prices of the packages in the repository.

21. The method as claimed in claim 13, further comprising the step of administrating the web services packaging system.

22. The method as claimed in claim 13, further comprising the step of registering web services and client applications.

23. The method as claimed in claim 13, further comprising the step of reporting usage of the web services packaging system for auditing and billing purposes.

24. The method as claimed in claim 13, further comprising the step of routing communication between the client application and a web service.

25. The method as claimed in claim 13, wherein the step of storing comprises the step of storing web services addresses and method calls in a database with fields and tables.

26. Computer readable media storing the instructions and/or statements for use in the execution in a computer of a method for packaging web services, the method comprising steps of:

storing information relating to methods of a plurality of web services organized into bundles, each bundle having one or more methods from one or more web services; and

authorizing a client application to access methods in one of the bundles.

27. Electronic signals for use in the execution in a computer of a method for packaging web services, the method comprising steps of:

storing information relating to methods of a plurality of web services organized into bundles, each bundle having one or more methods from one or more web services; and

authorizing a client application to access methods in one of the bundles.

28. A computer program product for use in the execution in a computer of a method for packaging web services, the computer program product comprising:

a repository for storing information relating to methods of a plurality of web services organized into bundles, each bundle having one or more methods from one or more web services; and

an authorization module for authorizing a client application to access methods in one of the bundles.

\* \* \* \* \*