

WEB SERVICES GATEWAY

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 managing functionality for web services. In particular, the invention relates to a web services gateway.

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. This makes addition of web server independent common features, such as application-level authentication, authorization and transaction logging tedious, time consuming and prone to errors at the integration layer.

[0004] FIG. 1 shows an example of a standard web service deployment environment. The standard web service deployment environment comprises an end user application 10, web services providers 20, and connections 30. The web services providers 20 have web services WS1 to WS_n, where n is an integer greater than zero. The end user application 10 connects directly with each web services provider 20 to obtain different web services 25 that the end user application 10 request. In fact, each end user application 10 connects directly with each requested web service 25 from each web services provider 20 to obtain each different web service 25 requested. Thus, if the end user wishes to obtain X different web services 25, then the end user application 10 must make a connection 30 to each of the X web services 25.

[0005] One problem that arises from this process of exposing the web services 25 for consumption over the web by client applications is that in order to protect unauthorized access of these web services 25 over the Internet, all the web services 25 must somehow incorporate authentication and authorization of users and other security measures. When a user wishes to use a web service 25 on a server, the server usually needs to ensure that (a) the user is authentic and (b) is authorized to access the web service. This authentication of the user is typically done by sending the user's name and password to the server which then verifies the given data before granting access. Since the authentication data is sensitive, it is desirable to send the authentication data over a secured channel, such as the hypertext transfer protocol over secure socket layer (https), which encrypts the data. Once the user is authenticated, the user's access to the web service is verified. This is typically done by querying an Access Control List (ACL) for the user's access rights.

[0006] Complexity is added and efforts are duplicated if every web service 25 provided by a developer implements or is even aware of all of the above infrastructure. Aside from

the infrastructure common to the services 25 provided, a developer may wish to provide value adding functionality to all or some of the web services 25. There is currently no way of adding these methods without having to reimplement them for each web service 25 or without the web service 25 even being aware that they exist.

[0007] Additionally, a company may wish to combine several web services 25 or parts of several web services 25 into an existing or new web service 25. It is time consuming for a developer to construct new web services 25 that call these other web services 25, and limiting in that the combinations are fixed at design time. There is no way to dynamically aggregate web services 25 based on a user's identity or some other criteria.

[0008] Furthermore, the location or address of a web service 25 must remain fixed so that its client applications will always know where to find it. However, the logistics of deployment may dictate that a web service 25 needs to be moved or exists on multiple servers at the same time. A user may not be able to find a web service 25 that has been moved from one location to another.

[0009] Web services 25 may be created from existing libraries of functionality with an existing application programming interface (API). In order to restrict access and bill by method, identification data must be provided by the client for every invoked method call. Adding these additional parameters, or any parameter that may be required by the particular business logic involved to an existing API is both complex and time consuming.

[0010] Currently, web service 25 capabilities remain uniform, irrespective of the identity of the client accessing them. This implies that there is no means of tailoring those web services 25, based on the presumed or assigned roles of the clients. For developers creating applications which consume a company's web services 25, this software 'contract', i.e., the list of functionality provided by a particular web service 25, is fixed. For companies providing these web services 25, there is no standard means of modifying the contract for different developers. Also, these developers, provided with a particular contract for web services 25, cannot count on the company providing the web services 25 with a means of limiting usage of these web services 25 to consumers of the application being developed by the developer.

[0011] Typically, the problems listed above are addressed by creating a library encapsulating the common or new functionality which is then consumed from each location where it is needed. This is inadequate here, as it does not offer the flexibility of leaving the web services 25 in question entirely intact and unaware that they are part of the infrastructure. Nor does it allow a dynamic data-driven approach to the problem since the behaviors are fixed at design time.

[0012] Current solutions also do not describe any mechanism for a means to customize that contract in any way, either for the developer of the application that consumes the web services 25, or for the user of that application. In order to provide this customization, the publisher of the web service 25 must develop several versions of the web service 25 in parallel, each version with capabilities unique to each developer.