

[0049] FIG. 5 shows another example of a web services infrastructure 501 in accordance with an embodiment of the present invention. The web services infrastructure 501 comprises client applications 15, client application connections 31, a gateway module (or gateway module) 500, web services WS1, WS2 and WS3, and web service connections 32. More or fewer client applications 15, web services 25, and their respective connections 31 and 32 may exist in the web services infrastructure 501. The gateway module 500 comprises a client application interface unit 310, a communication processor 311, a web services interface unit 312, an authentication module 520, an authorization module 525, and a web service registry repository 530. The gateway module 500 is a centralized access point for client applications 15 to connect to web services 25. The authentication module 520 and the authorization module 525 may alternatively be contained in a combined authentication and authorization module. Other components may be added to the gateway module 500, as described below.

[0050] The client application interface module 310 may operate on hypertext transfer protocol (http) and SOAP. The communication processor 311 and the web services interface unit 312 may be similar to those described above referring to FIG. 3. The authentication module 520 and the authentication module 525 may comprise code for authenticating and authorizing a client application 15. The web service repository 530 is a centralized registry of web services being exposed. It may store a unique identifier (ID) of the web service 25, its location, an API contract request string, and a brief description, all of which is mapped to the web service's unique name or uniform resource identifier (URI).

[0051] The gateway module 500 is an application that sits between client applications 15 and the web services 25 being consumed, intercepting communication between them. Some communication between client application 15 and web service 25 occurs over the SOAP protocol, while some communication includes the exchange of an API contract description, such as a web service description language (WSDL) contract document. The gateway module 500 acts as a SOAP processor with respect to communication between a web service 25 and a client application 15 used by a client application user. Furthermore, the gateway module 500 acts as an API contract (for example, WSDL) processor with respect to communication between a web service 25 and a client application 15 used by a client application developer. Therefore, the gateway module 500 transparently alters both the way the client application 15 calls the web service 25, and how the web service 25 appears to the client application 15 without either party being aware of the gateway module 500.

[0052] FIG. 6A shows another example of a method for managing functionality for one or more web services 25 (600) in accordance with an embodiment of the present invention. This example relates to a client application user. To request the use of a web service 25, a client application 15 sends a SOAP based method call through the client application connection 31. The client application interface unit 310 receives the method call (601) and passes the method call to the communication processor 311. The communication processor 311 determines which web service 25 is associated with the method call. The method call is then passed to the authentication module 520 to authenticate the method call as coming from the client application 15 (602).

The method call is passed back to the communication processor 311. If the client application is authentic (602), then the method call is passed to the authorization module 525 to determine if the client application 15 has authorization to use the requested web service 25 (603). The method call is passed back to the communication processor 311. If the client application is not authentic (602), then the method call is rejected (609). If the client application 15 is not authorized to use the method in the web service 25 (603), then the method call is rejected (610). A rejected method call may be returned back to the client application 15 through the corresponding client application connection 31. The rejection may include an error message explaining why the method call is rejected. Alternatively, a rejected method call may be ignored and the corresponding client application connection 31 closed. If a method call is rejected (609) or (610), the method is done (611).

[0053] If the client application 15 does have authorization (603), then the method call is passed to the web services interface unit 312. The web services interface unit 312 searches the web service registry repository 530 to determine the location of the requested web service 25 (604). The web service registry repository 530 provides a mapping from the identity of the web service (a URI) to the physical location of the web service 25 and any other attributes of the web service 25 that are desirable to assist the gateway module 500 to interpret, process, and make actual requests or method calls of said web service 25. Once the location of the web service 25 is determined (604), the method call is delegated to the web service 25 via the corresponding web service connection 32 (605).

[0054] If the method call does not have a corresponding response (606), then the method is done (610). Otherwise, the web services interface unit 312 receives a corresponding response from the web service 25 (607), in response to the method call. The web services interface unit 312 passes the response to the communication processor 311 to be passed back to the client application interface unit 310. The client application interface unit 310 sends the response through the appropriate client application connection 31 to the client application 15 from which the method call originated (608). The method is done (611).

[0055] FIG. 6B shows another example of a method for managing functionality for one or more web services 25 (620) in accordance with an embodiment of the present invention. This example also relates to a client application user. To request the use of a web service 25, a client application 15 calls an authentication method through the client application connection 31. The authentication method call contains the client application credentials such as user name and password. The client application interface unit 310 receives the authentication method call (621) and passes the authentication method call to the authentication module 520 to authenticate the client application credentials (622). If the credentials are authentic (622), i.e., the client application is registered with the gateway module 500, then the authentication module 520 issues an authentication identifier (ID) (523) and passes the authentication ID to the client application interface unit 310 to pass to the client application 15. If the credentials are not authentic (622), then the authentication call is rejected (631) and the method is done (633). Alternatively, the authentication module 520 issue an error response to be sent to the client application 15.