



- (51) **International Patent Classification:**  
*H04J 1/00* (2006.01)
- (21) **International Application Number:**  
PCT/US20 12/0560 18
- (22) **International Filing Date:**  
19 September 2012 (19.09.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**  
61/536,826 20 September 2011 (20.09.2011) US
- (71) **Applicant:** AETHERWORKS, LLC [US/US]; 501 Fifth Avenue, Suite 1803, New York, NY 10017 (US).
- (72) **Inventor:** MACINNIS, Robert, Francis; 7 Sturges Hollow, Westport, CT 06880 (US).
- (74) **Agent:** MAIER, Robert, L.; Baker Botts LLP, 30 Rockefeller Plaza, New York, NY 101 12-4498 (US).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, FT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— with international search report (Art. 21(3))

(54) **Title:** SYSTEMS AND METHODS FOR THE DEMAND-DRIVEN DEPLOYMENT OF LOCATION-NEUTRAL SOFTWARE

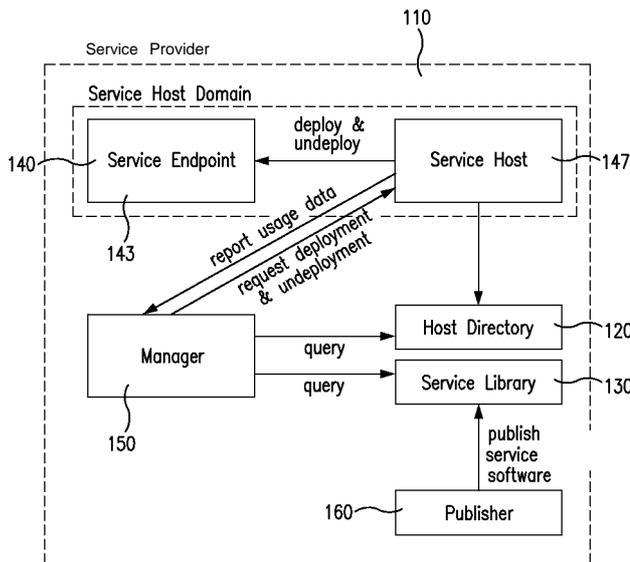


FIG. 1

(57) **Abstract:** Techniques for providing and consuming web services, including a service library configured to store one or more web services and a host directory connected to service hosts, configured to store data related to the service hosts. The service hosts are a network and adapted to receive and fulfill deployment requests for the web services stored in the service library by instantiating one or more endpoints of one of the web services. A manager is configured to query the host directory and the service library, generate a deployment plan, and transmit deployment requests to the one or more service hosts.

WO 2013/043665 A1

## SYSTEMS AND METHODS FOR THE DEMAND-DRIVEN DEPLOYMENT OF LOCATION-NEUTRAL SOFTWARE

### CROSS-REFERENCE TO RELATED APPLICATIONS

5                    This application is related to U.S. Provisional Application Serial no  
61/536,826, filed on September 20, 2011, which is incorporated herein by reference in  
its entirety and from which priority is claimed.

### BACKGROUND

10                    The disclosed subject matter relates to techniques for service-oriented  
computing, and more particularly, but not by way of limitation, to web service  
architecture.

                    Service-oriented computing (SOC) promotes the assembly of  
application components into networks of services that can be loosely coupled to create  
flexible, dynamic business processes and agile applications that span organizations  
15                    and computing platforms. Service-oriented computing has emerged as an approach to  
evolving tightly-coupled, component-based distributed systems into wider networks  
of services which can use uniform techniques to address, bind to and invoke service  
operations.

                    Service-oriented computing can provide a way to create new  
20                    architectures that reflect trends toward autonomy and heterogeneity. Distributed  
architectures which enable the flexible and loosely-coupled processes of service-  
oriented computing can be referred to as service-oriented architectures (SOAs), of  
which software services are a primary component. The guiding characteristics of  
SOAs are the interoperation between loosely coupled autonomous services, the  
25                    promotion of code reuse at a macro (service) level, and architectural composability.

                    SOAs can be composed of three primary entities — a provider, a  
consumer, and a registry. Providers can publish service locations in a registry;  
consumers can use the registry to locate services to program against and invoke. SOA  
implementations can specify the language used to describe services, provide the  
30                    means to publish and discover them, and dictate the protocols and communication  
mechanisms used to interact with them. The deployment of software services on the

Internet is increasingly achieved using one such set of standards collectively known as "web services."

Web services can be based on platform-independent standards, for example those developed through the efforts of the W3C working group. These standards can define the protocols, message formats, and service description language which enable interaction between clients and services on heterogeneous computing platforms across the Internet.

The standards introduced by web services can provide the means to evolve distributed systems from tightly-coupled distributed applications into loosely-coupled systems of services. The standards can enable interoperation between heterogeneous computing platforms through the exchange of messages using well-defined interfaces. By abstracting over implementation and hosting technology, the platform-independent web service standards can provide a means of homogenizing access to existing heterogeneous services while promoting the development of new services without forcing the adoption of anyone particular implementation or hosting technology. While the web services model has been widely adopted, its promise can be undermined by at least two limitations.

First, the web service addressing model can be rigid and only suited to highly reliable networked environments with highly reliable hosts. It can fail to take into account the intrinsic dynamism and fallibility of hosts on the Internet, and applications which aim to be robust to the failure of hosts can become littered with failure-recovery code. Second, an over-burdened and under-specified Service Provider role can lead to the development of proprietary deployment systems and closed-world environments where the use of web services is only incidental. Saddled with these two drawbacks, the wide adoption of the web services model can result in a landscape of software services that is highly populated by applications which expose web service interfaces, but which are largely incompatible in terms of their required deployment systems and hosting environments.

Accordingly, there is a need for an improved web services architecture.

30

## SUMMARY

In one aspect of the disclosed subject matter, a system for providing web services includes a Service Library configured to store one or more web services.

One or more Service Hosts, connected to a network, are adapted to receive and fulfill deployment and undeployment requests for the web services stored in the Service Library. Fulfilling deployment and undeployment requests can include instantiating one or more endpoints of the web services. A Host Directory, connected to the  
5 Service Hosts, is configured to store data related to the Service Hosts. A Manager is configured to query the Host Directory and the Service Library, generate a deployment plan, and transmit deployment and undeployment requests to the Service Hosts.

In one embodiment, the Service Library can be configured to receive  
10 the web services from a Publisher. The Service Hosts can transmit information about usage of the web services to the Manager. The Service Hosts can send the data related to the one or more Service Hosts, for example available resources, to the Host Directory.

In one embodiment, the system can include a discovery service  
15 including a directory, which can be configured to register and maintain entries of the endpoints corresponding to each Service Host. The directory service can store a mapping between the web services and a corresponding set of endpoints on which the web services are deployed. The directory service can be configured to receive a uniform resource identifier (URI) request corresponding to one of the web services,  
20 and request and resolve a URL corresponding to one of the endpoints on which the web service is deployed.

In another aspect of the disclosed subject matter, a method for providing web services includes storing one or more web services in a Service Library and storing data related to one or more Service Hosts in a Host Directory. The  
25 method includes controlling, with a Manager, deployment and undeployment of the one or more web services on one or more service endpoints on each of the one or more Service Hosts. Controlling can include querying the Service Library and Host Directory to retrieve information and data about the web services and the Service Hosts, generating a deployment plan using the information and data, transmitting  
30 deployment and undeployment requests to the Service Hosts, and instantiating one or more endpoints of the web services in response to the deployment and undeployment requests.

In another aspect of the disclosed subject matter, a method for consuming web services includes binding a consumer agent to a point of presence. The point of presence can communicate with a discovery service to resolve a URL corresponding to a service endpoint that corresponds to a desired web service URL. The point of presence can bind to the service endpoint. The discovery service communicates with a Manager, which is configured to query a Host Directory and Service Library, generate a deployment plan, and transmit deployment and undeployment requests to one or more Service Hosts.

## 10 BRIEF DESCRIPTION OF THE DRAWINGS

**Fig. 1** is a schematic diagram of the service provider portion of an embodiment of a system for providing web services in accordance with the disclosed subject matter.

15 **Fig. 2** is a schematic diagram of the discovery service and service provider portions of an embodiment of a system for providing web services in accordance with the disclosed subject matter.

**Fig. 3** is a schematic diagram of the service consumer, discovery service, and service providers of an embodiment of a system for providing and consuming web services in accordance with the disclosed subject matter.

20 **Fig. 4** is a flow chart demonstrating an embodiment of a method for providing and consuming web services in accordance with the disclosed subject matter.

**Fig. 5** is a schematic diagram of exemplary interactions between a service consumer and a local point of presence in accordance with an embodiment of the disclosed subject matter.

**Fig. 6** is a schematic diagram of exemplary interactions between a Publisher and a Service Library in accordance with an embodiment of the disclosed subject matter.

30 **Fig. 7** is a schematic diagram of exemplary interactions between a Service Host and a Host Directory in accordance with an embodiment of the disclosed subject matter.

**Fig. 8** is a schematic diagram of exemplary interactions between a point of presence and an Active Service Directory in accordance with an embodiment of the disclosed subject matter.

**Fig. 9** is a schematic diagram of exemplary interactions between an Active Service Directory and a Manager in accordance with an embodiment of the disclosed subject matter.

**Fig. 10a** is a schematic diagram of exemplary interactions between a Manager and a Service Library in accordance with an embodiment of the disclosed subject matter.

**Fig. 10b** is a schematic diagram of exemplary interactions between a Manager and a Host Directory in accordance with an embodiment of the disclosed subject matter.

**Fig. 10c** and **Fig. 10d** are schematic diagrams of exemplary interactions between a Manager and a Service Host in accordance with an embodiment of the disclosed subject matter.

**Fig. 10e** is a schematic diagram of exemplary interactions between a Manager and an Active Service Directory in accordance with an embodiment of the disclosed subject matter.

**Fig. 11** is a schematic diagram of endpoint deployment in accordance with an embodiment of the disclosed subject matter.

**Fig. 12** is a schematic diagram of service invocation in accordance with an embodiment of the disclosed subject matter.

**Fig. 13** is a schematic diagram of demand-driven dynamic deployment in accordance with an embodiment of the disclosed subject matter.

Throughout the drawings, the same reference numerals and characters, unless otherwise stated or indicated by context, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the disclosed subject matter will now be described in detail with reference to the Figures, it is done so in connection with the illustrative embodiments, which are intended to be non-limiting.

**DETAILED DESCRIPTION**

As used herein, the term "actor" can refer to an entity responsible for taking an action. For example, an actor can include an automated computer program. Alternatively, an "actor" can refer to a computer program with a user interface  
5 suitable to be controlled by an operator of the computer.

As used herein, the term "infrastructure" can refer, collectively, to the described architectural components existing in an operational state, independent of any particular implementation.

As used herein, the term "web service" can refer to the deployment of  
10 software services over a network. As used herein, the term "web service" is not intended to be limited to the deployment of software services over the internet, as one of ordinary skill in the art will appreciate that such services can also be deployed on other networks, such as a local area network LAN or the like.

For purposes of illustration, and not limitation, description will now be  
15 made of certain standards for describing (Web Service Definition Language (WSDL)), advertising and discovering (Universal Description Discovery and Integration (UDDI)), and communicating (Simple Object Access Protocol (SOAP)) with web services in connection with the disclosed subject matter. However, one of ordinary skill in the art will appreciate that other suitable standards, such as  
20 Representational State Transfer (REST), can be used, and the following description is not intended to be limiting.

Web services can interact through the exchange of messages using SOAP. SOAP defines a communication protocol for web services which is independent of programming languages and platforms and can be used over a broad  
25 range of transport protocols. For example, SOAP over HTTP can be used as the reference protocol binding definition for web services.

SOAP messages can include an outer XML element, which can be referred to as an "envelope," which defines the namespace(s) for the message, an optional header element which includes any relevant extensions to the messaging  
30 framework, and a required body element. The body element can provide a mechanism for transmitting information to an ultimate SOAP receiver but unspecified beyond this role, with neither a defined structure or interpretation, nor a means to specify any processing to be done.

WSDL is a language for describing web services in XML. The Component Interface Description of a service can be realized as a WSDL document, which can include information required to locate, bind to, and interact with a web service endpoint. WSDL documents are constructed from XML document elements that describe web service endpoints in terms of their operations, the parameters and return values of each operation (including type definitions), and the protocol and data bindings used for communication.

Requestor agent software can be written against an endpoint's WSDL document, for example with automatic code-generation tools to generate proxy or "stub" code in a particular language. These can be stand-alone tools, such as "WSDL to Java" for Java, as well as IDE-based, such as in Visual Studio and Eclipse, which provide a pluggable framework for generators producing code in Perl, C++, C#, PHP, and others.

Exemplary embodiments of the disclosed subject matter are described below, with reference to **Fig. 1**, **Fig. 2**, **Fig. 3**, and **Fig. 4**, for purposes of illustration, and not limitation.

In an exemplary embodiment, and with reference to **Fig. 1**, a system for providing web services can include a service provider component **110**. The service provider component **110** can be embodied in, for example, a computer program. The computer program can be stored on a computer readable medium, such as a CD-ROM, DVD, Magnetic disk, ROM, RAM, or the like. The instructions of the program can be read into a memory of one or more processors included in one or more computing devices, such as for example a server connected to a network or a distributed computer system. When executed, the program can instruct the processor to control various components of the computing device. While execution of sequences of instructions in the program causes the processor to perform certain functions described herein, hard-wired circuitry may be used in place of, or in combination with, software instructions for implementation of the presently disclosed subject matter. Thus, embodiments of the present invention are not limited to any specific combination of hardware and software.

In connection with an exemplary embodiment, the service provider component **101** can include, for example, a Service Library **130** can be configured to store one or more web services. For example, the Service Library **130** can be

connected to a Publisher **160**, which can "publish" a web service to the Service Library **130** for storage. Service host **147**, connected to a network, can be adapted to receive and fulfill deployment and undeployment requests for the web services stored in the Service Library **130**. For example, fulfilling deployment and undeployment requests can include instantiating one or more endpoints **143** of one of the web services. A Host Directory **120** can be connected to the Service Host **147** and configured to store data related to the one or more Service Hosts **147**. A Manager **150** can be configured to query the Host Directory **120** and the Service Library **130**, generate a deployment plan, and transmit deployment and undeployment requests to the Service Host **147**.

The Service Host **147** can be configured to transmit information about usage of the web services to the Manager **150**. The Service Hosts **147** can also be configured to send data related to their available resources to the Host Directory **120**.

This exemplary embodiment can reduce complexity for participants in the web service lifecycle by partitioning the responsibility of providing a web service into multiple independent roles, reducing the amount of domain-specific knowledge required by each actor and lowering the barriers to participation in the provision of web services. For example the tasks of publishing, deploying and hosting a web service can be treated as distinct, independent activities. Thus, for purposes of illustration and not limitation, the service provider component **101** can be viewed as three distinct "actors": Publisher **160**, Manager **150**, and Service Host **147**. Additionally, two architectural entities: repositories called the Service Library **130** and Host Directory **120**.

Rather than being deployed explicitly, a web service provider agent implementation can instead be described by a Publisher **160** who then "publishes" it into the infrastructure by storing it in a repository called the Service Library **130**. This approach represents an approach to web service deployment by separating service substantiation from actual realization — i.e., the lifecycle of a web service can begin when it is published, not when it is deployed.

In order to participate in the infrastructure, Service Hosts **147** can register their willingness to host web services by describing their available resources and registering with a directory called the Host Directory **120**. Service Hosts **147** can indicate their available web service deployment containers and specify the list of

Publishers **160** whose web service provider agent implementations they are willing to deploy. Service Hosts **147** thus can participate in an infrastructure not by advertising their statically deployed services, but by advertising their hosting capabilities, joining a shared pool of latent hosting resources which can be dynamically consumed (and  
5 reclaimed) by Managers **150** as necessary to meet changing levels of demand.

Managers **160** can be responsible for managing the provisioning level of a single web service (for which there may be zero or more endpoints **140** at any given time). In order to enact deployment, Managers **160** can first query the Service Library **130** and Host Directory **120**, then create deployment plans by pairing web  
10 service implementations with a suitably capable Service Host **147**. Managers **150** can send deployment requests (**433**) to Service Hosts **147** who then can be responsible for instantiating an endpoint **140** of the web service (or denying the request). Service Hosts **147** can provide information about the usage of each web service endpoint deployed within their domain to each web service's Manager **150**. A Manager **150** can  
15 use this usage data to make decisions about the necessary level of provisioning of the web service they manage.

In another exemplary embodiment, and with reference to **Fig. 2**, a system for providing web services can include a discovery service component **210**. The discovery service component **210** can be embodied in, for example and in like  
20 manner to the service provider component **110**, a computer program executed on a computing device. This computer device can be a physically separate device than that of the service provider component **110** coupled to the service provider component **110** via a network such as the internet. Alternatively, the computer device hosting the discovery service component **210** can be the same computer device hosting the  
25 service provider component **110**. Notably, these computing devices and all others described herein may be implemented within the spirit and scope of the invention as single devices, clusters of devices, networks, or the like.

The discovery service component **210** can include an Active Service Directory **220**. The Active Service Directory **220** can be configured, for example, to  
30 store a mapping between the web services stored in the Service Library **130** and a corresponding set of endpoints **143** on which the web services are deployed.

The Active Service Directory **220** can hold mappings between a single web service and a set of active endpoints **143** of the web service. A central

entity in the architecture, the collective mappings held in the Active Service Directory **220** can represent the current state of an infrastructure from all participants" perspectives. Because all infrastructure participants can rely on the Active Service Directory **220** to locate endpoints **143** of their desired web services, this directory can instigate auto deployment procedures for web services that have no currently  
5 deployed endpoints.

The Active Service Directory **220** can provide operations to add, remove and locate (**450**) active endpoints **143** of web services. If the Active Service Directory **220** receives a lookup request for the active endpoints of a particular web  
10 service, but no such entries exist, the Active Service Directory **220** can proactively locate and inform the Manager **150** of the requested web service. The Manager can dynamically initiate the deployment of a new endpoint **143** using the previously described deployment procedure. Managers **150** can be responsible for inserting and maintaining all Active Service Directory **220** entries for any newly deployed  
15 endpoints **143** of the web service it manages. If, for example, demand for its web service drops to zero, a Manager **150** can decide to undeploy one (or all) of the deployed endpoints **143**. If an endpoint **143** is undeployed the Manager **150** can remove the endpoint entry from the Active Service Directory **220**, ensuring that the directory remains up-to-date and as accurate a reflection as possible of the current  
20 state of the infrastructure.

In another exemplary embodiment, and with reference to **Fig. 3**, a system for providing web services can include a service consumer component **310**. The service consumer component **310** can be embodied in, for example and in like manner to the service provider component **110**, a computer program executed on a  
25 computing device. This computer device can be a physically separate device from that of the service provider component **110** coupled to the service provider component **110** via a network such as the internet. Alternatively, the computer device hosting the service consumer component **310** can be part of the same computer device hosting the service provider component **110**.

The service consumer component **310** can include a consumer agent  
30 **320** and point of presence **330**. The discovery service **210** can be configured, for example, to receive a URI corresponding to one or more of the web services in the Service Library **130** from the services consumer component **310** and request and

resolve a one or more URLs corresponding to endpoints **143**, if any, of the deployed web service. In connection with certain embodiments, multiple implementations of a particular web service can be published to the Service Library **130**, each of which corresponds to the same URI. For example, a particular web service can be  
5 implemented using a number of different languages and/or requiring one or more different deployment environments. Each implementation can be stored in the service library **130** and can correspond to the same URI, and can implement the same interface such that a call to the any one of the particular implementations may yield the same return. The consumer agent **320** can be bound to the point of presence **330**.  
10 The point of presence **330** can communicate with the discovery service and can bind to a service endpoint **143** using the URL. The point of presence **330** can be configured to abstract over a location of the service while preserving the consumer agent's **320** view, thereby acting as a gateway for the consumer agent into an instantiation of the web service.

15 In connection with this exemplary embodiment, the disclosed techniques do not require use of URLs to describe web services, as URLs may become invalid over time. Web services can instead be identified with a URI, abstractly describing a service which at any point in time can have zero or more active endpoints. The Service Consumer component **310** can be relieved from the tasks of  
20 locating and binding to web service endpoints **143** through the introduction of a mechanism which performs these tasks on their behalf.

This technique can present a limited-mediation framework for the consumption of web services which can transparently resolve a live endpoint URL from the URI contained in an invocation request. This framework can be realized as  
25 proxy mechanism, residing at the Service Consumer component **310**, which can act as a gateway into an instantiation of the architecture — a "point of presence," as described above and depicted in Fig. **3**. Consumer agent **320** software can be written to bind to this local point of presence **330** and invoke web service operations using the desired web service's URI. The point of presence **330** can be responsible for  
30 transparently resolving a URL from the URI by retrieving a list of active endpoints **143** of the web service from the Active Service Directory **220**, selecting an endpoint for use, invoking the requested operation on behalf of the Service Consumer **310**, and returning any results. It can also be responsible for transparently detecting and

recovering from the failure of web services and the Service Hosts 147 on which they are deployed, and for proactively recovering from these failures by retrying alternative endpoints 143 (according to local policy). Unrecoverable errors can be returned to Service Consumers 310, indicating that given the available resources of the infrastructure, it is not currently possible to fulfill the request.

The point of presence 330 can abstract over the location of a service while still preserving the current Service Consumer component's 310 view upon the system. It can simplify the creation of consumer agent 320 applications, for example, by allowing developers to program against what a web service does, not where it is or whether it is currently deployed. Further, because it consumes all tasks which require interaction with the Discovery Service 210, the point of presence 330 can provide a layer of abstraction over the particular standards versions used in an infrastructure (e.g., UDDI version). This can provide a barrier to obsolescence in the face of evolving standards while enabling consumer agent applications to be portable between environments which use different standards.

In an exemplary embodiment, and with reference to Fig. 4, a method for providing web services can include storing (410) one or more web services in the Service Library. For example, the Service Library can receive (411) published web services from a Publisher. Data related to one or more Service Hosts can be stored (420) in a Host Directory. For example, the Host Directory can receive (422) data related to the Service Hosts, for example data about available resources and which Publishers a Service Host is willing to host a web service from. A Manager can control (430) deployment and undeployment of the web services on sendee endpoints on each of the Service Hosts. For example, the Manager can query (431) the Service Library and Host Directory to retrieve information about the web services and the data related to the Service Hosts. The Manager can then generate (432) a deployment plan using the information about the web services and the data related to the Service Hosts. The Manager can then transmit (433) deployment and undeployment requests to the Service Hosts. The Service Hosts can then instantiate (434) the web services in response to the deployment requests, or unstantiate the web services in response to an undeployment request. Usage data and other information can be fed back 440 to the Manager from the Service Hosts.

The Manager can send **(450)** information about the endpoints on which web services are deployed to the Discovery Service, including the Active Service Directory. A Service Consumer can send a URI request **(460)** to the Directory Service, for example, through a local point of presence. The Discovery Service, in connection with the Manager, can provide a URL response **(470)** in response to the  
5 URI request, and the Service Consumer can bind to the endpoint which hosts a desired web service.

Description will now be made of exemplary interactions between certain components of the system disclosed herein, with reference to **Fig. 5** to **Fig. 10**,  
10 for purposes of illustration, and not limitation.

In accordance with an embodiment of the disclosed subject matter, and with reference to **Fig. 5**, service consumers **310** need not interact directly with web services endpoints. For example, a service consumer **310** can bind to and invoke operations upon a local point of presence **330**. The POPs can be co-located with the  
15 service consumer **310**; consumer agent **320** can be written to statically bind to the URI of a web service, for example prefixed with the protocol, hostname and port of a local URI.

For example, invoking operation "x" of a web service identified by the URI "ServiceA" is illustrated in **Fig. 5**. The consumer agent application can bind to  
20 and invoke the operation on the POP, which then can locate an endpoint of the requested web service. The POP can send a lookup request to the Active Service Directory **220** by invoking a "findAU" operation with the URI "ServiceA" as a parameter and can receive back a list of ServiceInstanceDescriptors describing the currently active endpoints **540** and **550** of ServiceA (**541**, **551**, and **552**). The POP  
25 can apply local policy to select which endpoint to use before connecting to the endpoint and invoking operation "x" on behalf of the Service Consumer. The results of the operation can be returned to the POP and finally returned to the consumer agent **320** application.

In accordance with an embodiment of the disclosed subject matter, and with reference to **Fig. 6**, a Publisher **160** can be, though need not be, a service  
30 consumer who interacts with the Service Library **130** in order to publish and unpublish an implementation of a web service. As with other service consumers (e.g., **310**), the Publisher **160** can include a consumer agent **610** and point of presence **620**,

and need not directly interact with any external web services, instead invoking the desired operations via the local point of presence 620.

For example, in order to publish an implementation of a web service (described, for example, with a ServiceImplementationDescriptor), Publishers 160  
5 can bind to the local point of presence 610 and invoke the "publishServiceImplementation" operation of the Service Library 130 with the desired ServiceImplementationDescriptor. Once an implementation has been published it can become immediately available for use, able to be deployed onto capable Service Hosts as necessary to meet demand.

10 In accordance with an embodiment of the disclosed subject matter, and with reference to Fig. 7, a Service Host 147 can both, though need not, contribute to the role of service provider and also act as service consumers. As with other service consumers (e.g., 310), the Service Host 147 can include a consumer agent 710 and a local point of presence 720. The Service Host 147 can register itself with the Host  
15 Directory 130 by, for example, describing themselves using a HostDescriptor, bind to the local point of presence 720, and invoke the "addAuthorizedPublisher" operation of the host director 130. By operating through the local point of presence 720 the Service Host 147 registration process can thus be performed with the same invocation procedures provided by the infrastructure to all Service Consumers. Upon registration,  
20 a Sendee Host's HostDescriptor can be added to a shared pool of hosting resources, ready to be consumed as necessary to meet demand.

In accordance with an embodiment of the disclosed subject matter, and with reference to Fig. 8, a point of presence 330 associated with any service consumer 310 can be a transparent endpoint resolution and failure recovery mechanism which  
25 can locate, bind to, and invoke operations upon web services on behalf of the service consumer 310. For example, in order to locate an active instance of a web service the point of presence can interact with a web service called the "Active Service Directory" 810, invoking either its "findOne" or "findAU" operations with the URI of the desired web service as a parameter.

30 In order to participate in an instantiation of the architecture, the local point of presence can have prior knowledge of at least one Active Service Directory endpoint 810, Any alternative Active Service Directory endpoints can be listed in the Active Service Directory 220 under the web service URI "Active Service Directory"

and can be located using the "findOne" and "findAU" operations. The POP can periodically retrieve and store a list of alternative Active Service Directory endpoints to use in the event of failure.

In accordance with an embodiment of the disclosed subject matter, and with reference to **Fig. 9**, when the Active Service Directory **220** receives a lookup request for a web service with zero active endpoints, it can locate an instance of that web service's Manager **150** so that an endpoint may be deployed. The Active Service Directory **220** can utilize its own lookup facilities by binding to its local point of presence and invoking the Active Service Directory "findOne" or "findAU" operation using the URI of the Managing entity **920** responsible for the originally requested web sendee. This URI can be constructed, for example, by concatenating "Manager" URI to the URI of the requested web service in the pattern "WebServiceURIJVManagerURI." The Active Service Directory can be returned a ServiceInstanceDescriptor describing a web service endpoint which implements the "Manager" interface.

As a Manager **150** can be responsible for maintaining the directory records of the web service they manage, Managers **150** can insert a record of any endpoint they deploy into the Active Service Directory **220**. Managers **150** can perform this operation in-band with, for example, the "requestFirstInstance" invocation — returning "true" if the entire deployment and registration process completed successfully, or "false" otherwise. The "requestFirstInstance" operation returning "false" can indicate that, given the current state of the system, it is not possible to deploy a new instance of the web service. This situation can represent an unrecoverable error which can be returned to the requesting entity (i.e., the Service Consumer). If the operation returns "true" the Active Service Directory **220** can re-perform the originally requested lookup operation and return the newly-inserted ServiceInstanceDescriptor to the requesting entity.

In accordance with an embodiment of the disclosed subject matter, and with reference to **Fig. 10**, the Manager **150** can communicate with various components, including the Service Library **130**, Host Directory **120**, one or more Service Hosts **147**, and the Active Service Directory **220**, in accordance with the exemplary and non-limiting description that follows. For example, with reference to **Fig. 10a**, Managers **150** can invoke the operations of the Service Library **130** in order

to retrieve a set of implementations of the web service being deployed. These implementations can each be described with a ServiceImplementationDescriptor; the elements of this descriptor can be used later by the Manager 150 in order to select a suitable candidate implementation for deployment.

5 In like manner, with reference to **Fig. 10b**, Managers 150 can invoke the operations of the Host Directory 120 during the deployment process. Managers 150 can use the Host Directory 120 to retrieve a set of HostDescriptors identifying Service Hosts 147 willing to deploy web service implementations published by the indicated Publisher 160 (identified, for example, with a unique PublisherID).

10 Managers 150 can compare the ServiceImplementationDescriptors retrieved from the Service Library 130 with the HostDescriptors retrieved from the Host Directory 120 in order to craft suitable candidate deployment plans based on local policy. Once a plan is selected the Manager 150 can contact the selected Service Host 147 and initiate the deployment process.

15 With reference to **Fig. 10c**, after selecting a suitable web service implementation (described with a ServiceImplementationDescriptor) for deployment on a selected Service Host 147 (described with a HostDescriptor) a Manager 150 can bind to the Service Host 147 included in the HostDescriptor and can invoke the "deploy" operation directly. Upon successful deployment the Service Host 147 can  
20 return the Manager 150 a ServiceInstanceDescriptor describing the newly deployed web service endpoint. The Manager 150 can then register this ServiceInstanceDescriptor in the Active Service Directory 220, completing the deployment process.

Conversely, with reference to **Fig. 10d**, for each web service endpoint  
25 deployed within their domain (e.g., the containers under their control), a Service Host 147 can report usage data to that web service's managing entity. The Service Host 147 can bind to the local point of presence 720 and can invoke the "reportUsageData" operation of the web service identified by the concatenation of a "Manager" URI to the URI of the web service for which data is being reported (e.g.,  
30 "WebServiceURI\_ManagerURi"). The "reportUsageData" operation can return a numerical value indicating the length of time the Service Host 147 should wait before next reporting usage data for this particular web service. A Service Host's 147 local policy can dictate that usage data be returned earlier than requested (e.g., due to local

resource constraints, such as working memory); returning data significantly later than the requested period can indicate to the Manager 150 that there is a problem with the Service Host 147 — information which a Manager 150 can act upon in order to effectively manage the availability of its service.

5                   With reference to **Fig. 10e**, when a Manager 150 of a web service successfully deploys or undeploys endpoints of that web service it can add or remove the endpoint reference in the Active Service Directory 220. Managers 150 can bind to the local point of presence and invoke the "addActiveService" or "removeActiveService" operation of the Active Service Directory 220 web service  
10 using the relevant ServiceInstanceDescriptor.

For example, Managers 150 can be responsible for maintaining the Active Service Directory entries of the web service they manage. Of all entities fulfilling the responsibilities of the Service Provider, Managers 150 can be deemed to be most interested in maintaining an accurate public record of the web service they  
15 each manage as it provides them with accurate usage data from which they may make more informed decisions in the fulfillment of their responsibility to manage the provisioning level of a web service.

#### **EXAMPLE**

20                   The present application is further described by means of an example, presented below. The use of such example is illustrative only and in no way limits the scope and meaning of the invention or of any exemplified term. Likewise, this application is not limited to any particular preferred embodiments described herein. Indeed, many modifications and variations of the invention will be apparent to those  
25 skilled in the art upon reading this specification. The invention is to be understood by the terms of the appended claims along with the full scope of equivalents to which the claims are entitled.

In this example, and with reference to **Fig. 11, Fig. 12, and Fig. 13**, procedures are described using the previously described exemplary component  
30 interactions as single steps for endpoint deployment, service invocation, and demand-driven dynamic deployment for the provision and invocation of a web service referred to as "DayTime."

The deployment of a new web service endpoint can be carried out by a Manager **150** as in **Fig. 11**. The Manager **150** can interact with the Service Library **130**, Host Directory **120**, one or more Service Hosts **147**, and the Active Service Directory **220** in order to make a new "DayTime" web service **1100** endpoint **1120** available for use.

For purposes of illustration and not limitation, in order to deploy a new web service endpoint a Manager **150** can first create and select a valid deployment plan (the deployment plan comprising, for example, a `ServiceImplementationDescriptor` whose "requirements" are matched by the "capabilities" described in a `HostDescriptor`). The Manager **150** can begin by contacting the Service Library **130** and retrieving the set of all published implementations of the "DayTime" web service. Each implementation returned can be described with a `ServiceImplementationDescriptor` **1130**, each of which can have a "PublisherID" element **1131**. For each implementation, the Manager **150** can contact the Host Directory **120** using the `PublisherID` **1131** and retrieve the set of Service Hosts willing to deploy provider agent applications written by the specified Publisher. The Manager **150** then can create a set of valid deployment plans **1140** and select one from amongst the candidates based on local policy. Next the Manager **150** can execute the deployment plan, contacting the Service Host **140** described by the selected `HostDescriptor` **1150** and invoking its "deploy" operation with the selected `ServiceImplementationDescriptor` **1130**. Upon successful deployment, the target Service Host **147** can return the Manager **150** a `ServiceInstanceDescriptor` **1160** describing the endpoint **1120**. The Manager **150** can complete deployment by inserting the `ServiceInstanceDescriptor` **1160** into the Active Service Directory **220**. Once the newly deployed endpoint is listed in the Active Service Directory **220** it can be deemed to be deployed, ready to be located and its operations invoked by Service Consumers.

With reference to **Fig. 12**, service invocation by service consumers **310** can include binding and invoking web service operations on a local point of presence **330**. Once an invocation request is received, the local point of presence **330** can be responsible for locating an active endpoint of the target web service and invoking the requested operation on behalf of the Service Consumer, for example the "getTime" operation of the "DayTime" web service **1110**.

Upon receiving an invocation request from a consumer agent **320** application, the local point of presence **330** can first extract the desired web service URI **1210** from the incoming request. If the point of presence **330** implements an endpoint cache, it can first check locally for a previously retrieved list of endpoints. If  
5 no such entries exist, a new list can be retrieved using the Active Service Directory **220**. Once a list of endpoints is retrieved, the point of presence can select one for use according to local selection policy. The point of presence **330** can then prepare the invocation request (including possible modifications to the message), bind to the selected web service endpoint **1120**, and forward the invocation request. If the  
10 invocation fails for any reason the same procedure can be attempted for the remainder of the active endpoints and, if all endpoints prove unavailable, a generic error can be returned to the Service Consumer **310**. If the invocation is successful, the resulting response can be first prepared (again, possibly involving modification to the message) before being returned to the Service Consumer **310** and concluding the invocation  
15 process.

With reference to **Fig. 13**, in a dynamic demand-driven system, endpoint deployment can occur in response to an existing or anticipated event including, for example, the invocation of an operation of a web service for which there are currently no active endpoints. In such a circumstance, the subject matter  
20 disclosed above with reference to **Fig. 11** and **Fig. 12** can be performed. For example, When the Active Service Directory **220** receives a request for all endpoints **1120** of the "DayTime" web service **1110**, it can find that there are none currently deployed. In this manner, the Active Service Directory **220** can initiate (though not necessarily enact) the deployment of the first instance of the requested web service by  
25 contacting that web service's Manager **150** and invoking the "requestFirstInstance" operation. At this point the deployment process can be executed as described above; when the Manager **150** returns from the "requestFirstInstance" operation, the Active Service Directory **220** can re-perform the lookup and return either a reference to the newly deployed endpoint **1120**, or an error indicating that, given the currently  
30 available resources in the infrastructure, it is not possible to fulfill the request at the given time.

As described herein, the Manager **150** for a web service need not be currently deployed in order for that web service to be deployable. If implemented as a

web service, and with generic and/or custom, service-specific Manager implementations published in the Service Library **130**, the same deployment mechanisms can be used to deploy and manage a web service's Manager as those Managers use to manage their particular web service. This recursive, collapsible model can enable an infrastructure to be "wound up" to provide enough web service endpoints to meet demand, and then "wound down" to a state of zero resource consumption when demand falls to zero.

\* \* \*

As described above in connection with certain embodiments, certain components, e.g., **110**, **210**, and **310**, can include a computer or computers, processor, network, mobile device, cluster, or other hardware to perform various functions. Moreover, certain elements of the disclosed subject matter can be embodied in computer readable code which can be stored on computer readable media and when executed cause a processor to perform certain functions. In these embodiments, the computer plays a significant role in permitting the system and method for providing and/or consuming web services to provide such services over a network, e.g., the internet. For example, the presence of the computer, processor, memory, storage, and networking hardware provides the ability to deploy, consume, and manage a web service environment and dynamically apply and reclaim resources based upon measured consumer demand.

Additionally, as described above in connection with certain embodiments, certain components can communicate with certain other components, for example via a network, e.g., the internet. To the extent not expressly stated above, the disclosed subject matter is intended to encompass both sides of each transaction, including transmitting and receiving. One of ordinary skill in the art will readily understand that with regard to the features described above, if one component transmits, sends, or otherwise makes available to another component, the other component will receive or acquire, whether expressly stated or not.

The presently disclosed subject matter is not to be limited in scope by the specific embodiments herein. Indeed, various modifications of the disclosed subject matter in addition to those described herein will become apparent to those skilled in the art from the foregoing description and the accompanying figures. Such modifications are intended to fall within the scope of the appended claims.

**CLAIMS**

1. A system for providing web services, comprising:
  - a service library configured to store one or more web services;
  - one or more service hosts connected to a network and adapted to
  - 5 receive and fulfill deployment requests for the web services stored in the service library, wherein fulfilling deployment requests includes instantiating one or more endpoints of one of the web services;
  - a host directory connected to the service hosts and configured to store data related to the one or more service hosts; and
  - 10 a manager configured to query the host directory and the service library, generate a deployment plan, and transmit deployment requests to the one or more service hosts.
2. The system of claim 1, wherein the service library is further configured
- 15 to receive the one or more web services from a publisher.
3. The system of claim 1, wherein the one or more service hosts are further configured to transmit information about usage of the web services to the manager.
4. The system of claim 1, wherein the one or more service hosts are
- 20 further configured to send the data related to the one or more service hosts to the host directory, and wherein the data related to the one or more service hosts includes available resources.
5. The system of claim 1, further comprising a discovery service including a directory, configured to register and maintain entries of the endpoints of
- 25 each service host.
6. The system of claim 5, wherein the directory service is further configured to store a mapping between the web services and a corresponding set of the endpoints on which the web services are deployed.
7. The system of claim 6, wherein the directory service is further
- 30 configured to receive a URI corresponding to one of the web services, request and resolve URL corresponding to one of the endpoints on which the one of the web services is deployed.

8. The system of claim 7, further comprising a service consumer including:

a consumer agent; and

a point of presence, to which the consumer agent is bound, the point of presence configured to communicate with the discovery service and configured to bind to the one or more service endpoints.

9. The system of claim 8, wherein the point of presence is configured to abstract over a location of a service while preserving the consumer agent's view, thereby acting as a gateway for the consumer agent into an instantiation of the web services.

10. A method for providing web services, comprising:  
storing one or more web services in a service library;  
storing data related to one or more service hosts in a host directory;  
controlling, with a manager, deployment of the one or more web services on one or more service endpoints on each of the one or more service hosts, wherein controlling deployment includes:

querying the service library and host directory to retrieve information about the one or more web services and the data related to the one or more service hosts;

generating a deployment plan using the information about the one or more web services and the data related to the one or more service hosts;

transmitting deployment requests to the one or more service hosts; and

instantiating the one or more endpoints of one of the web services in response to the deployment requests.

11. The method of claim 10, further comprising receiving, at the service library, the one or more web services from a publisher.

12. The method of claim 10, further comprising transmitting, from the one or more service hosts, information about usage of the web services to the manager.

13. The method of claim 10, further comprising sending, from the one or more service hosts, the data related to the one or more services hosts to the host directory, and wherein the data related to the one or more service hosts includes available resources.

14. The method of claim 10, further comprising registering and maintaining entries of the endpoints of each service host in a discovery service including a directory.

15. The method of claim 14, further comprising storing, with the discovery service, a mapping between the web services and a corresponding set of the endpoints on which the web services are deployed.

16. The method of claim 15, further comprising receiving, at the discovery service, a URI corresponding to one of the web services, and requesting and resolving a URL corresponding to one of the endpoints on which the one of the web services is deployed.

17. A method for consuming web services, comprising:  
binding a consumer agent to a point of presence;  
communicating, via the point of presence, with a discovery service to resolve a URL corresponding to a service endpoint corresponding to a desired web service URI; and

binding the point of presence to the service endpoint;  
wherein the discovery service is configured to communicate with a manager, the manager configured to query a host directory and a service library, generate a deployment plan, and transmit deployment and undeployment requests to one or more service hosts, one of which is configured to instantiate the service endpoint.

18. The method of claim 17, wherein the point of presence is configured to abstract over a location of a service while preserving the consumer agent's view, thereby acting as a gateway for the consumer agent into an instantiation of the web services.

19. The method of claim 17, the host directory is configured to store data related to the one or more service hosts.

20. The method of claim 17, the service directory is configured to store at least the desired web service.

21. A non-transitory computer-readable medium containing computer-executable instructions that when executed cause one or more computer devices to perform a method of providing web services, the method comprising:  
storing one or more web services in a service library;

storing data related to one or more service hosts in a host directory;  
controlling, with a manager, deployment of the one or more web  
services on one or more service endpoints on each of the one or more service hosts,  
wherein controlling deployment includes:

5                    querying the service library and host directory to retrieve  
information about the one or more web services and the data related to the one  
or more service hosts;

                  generating a deployment plan using the information about the  
one or more web services and the data related to the one or more service hosts;

10                   transmitting deployment requests to the one or more service  
hosts; and

                  instantiating the one or more endpoints of one of the web  
services in response to the deployment requests.

22.        The non-transitory computer readable medium of claim 21, further  
15 comprising receiving, at the service library, the one or more web services from a  
publisher.

23.        The non-transitory computer readable medium of claim 21, further  
comprising transmitting, from the one or more service hosts, information about usage  
of the web services to the manager.

20        24.        The non-transitory computer readable medium of claim 21, further  
comprising sending, from the one or more service hosts, the data related to the one or  
more services hosts to the host directory, and wherein the data related to the one or  
more service hosts includes available resources.

25        25.        The non-transitory computer readable medium of claim 21, further  
comprising registering and maintaining entries of the endpoints of each service host in  
a discovery service including a directory.

26.        The non-transitory computer readable medium of claim 25, further  
comprising storing, with the discovery service, a mapping between the web services  
and a corresponding set of the endpoints on which the web services are deployed.

30        27.        The non-transitory computer readable medium of claim 26, further  
comprising receiving, at the discovery service, a URI corresponding to one of the web  
services, and requesting and resolving a URL corresponding to one of the endpoints  
on which the one of the web services is deployed.

28. A non-transitory computer-readable medium containing computer-executable instructions that when executed cause one or more computer devices to perform a method for consuming web services, the method comprising:

binding a consumer agent to a point of presence;

5 communicating, via the point of presence, with a discovery service to resolve a URL corresponding to a service endpoint corresponding to a desired web service URI; and

binding the point of presence to the service endpoint;

10 wherein the discovery service is configured to communicate with a manager, the manager configured to query a host directory and a service library, generate a deployment plan, and transmit deployment and undeployment requests to one or more service hosts, one of which is configured to instantiate the service endpoint.

29. The non-transitory computer readable medium of claim 28, wherein  
15 the point of presence is configured to abstract over a location of a service while preserving the consumer agent's view, thereby acting as a gateway for the consumer agent into an instantiation of the web services.

30. The non-transitory computer readable medium of claim 28, the host  
directory is configured to store data related to the one or more service hosts.

20 31. The non-transitory computer readable medium of claim 28, the service directory is configured to store at least the desired web service.

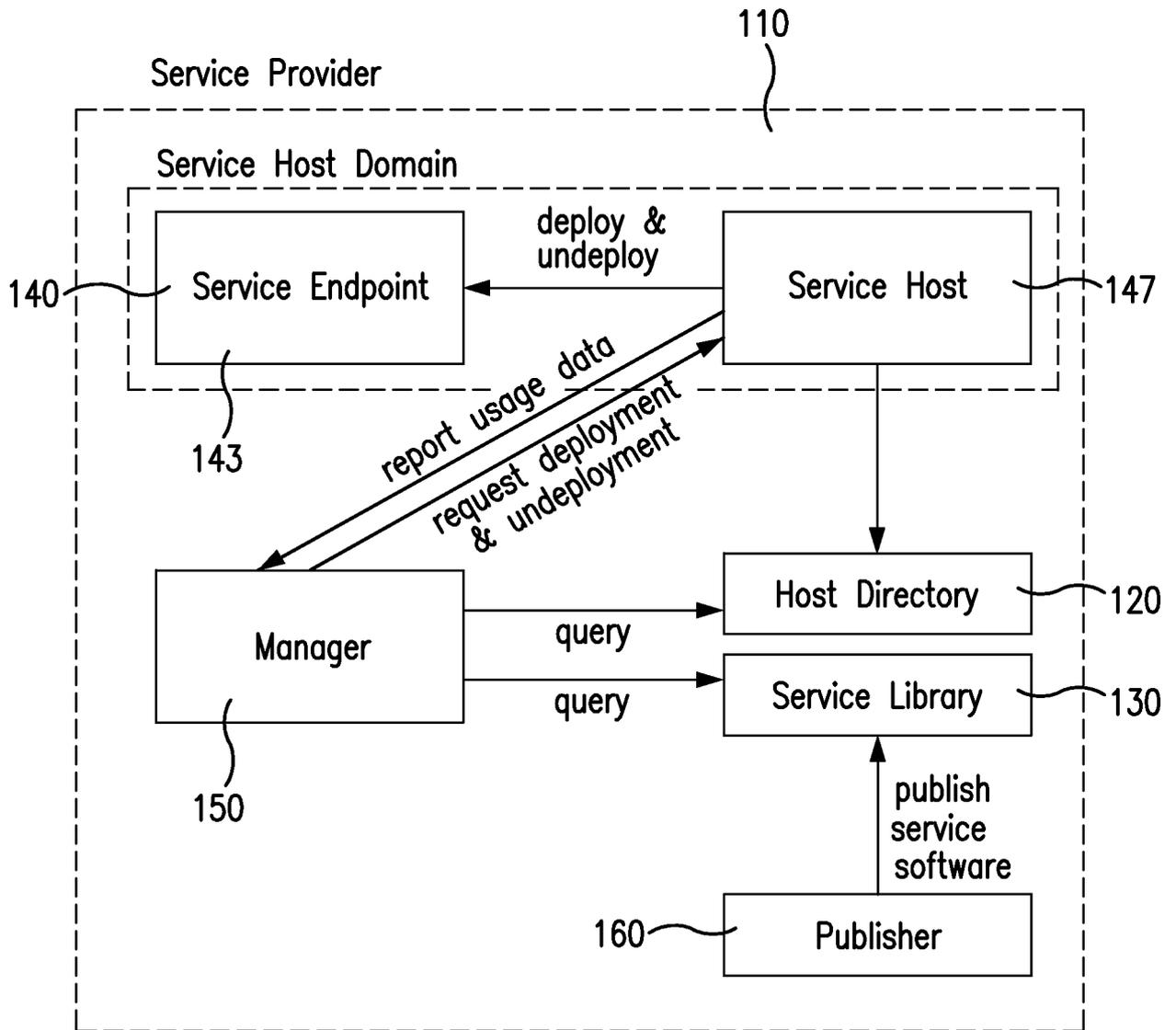


FIG. 1

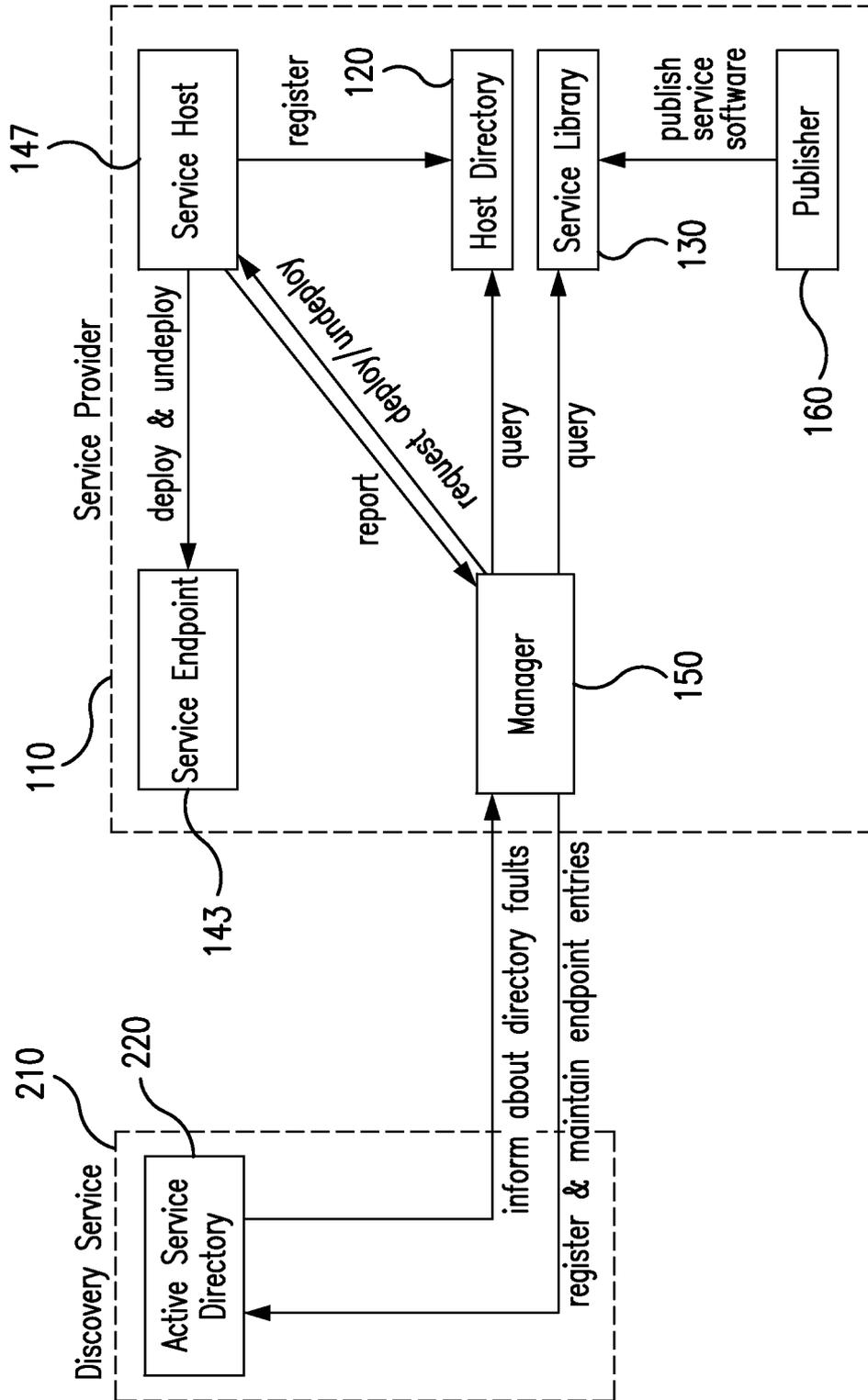


FIG. 2

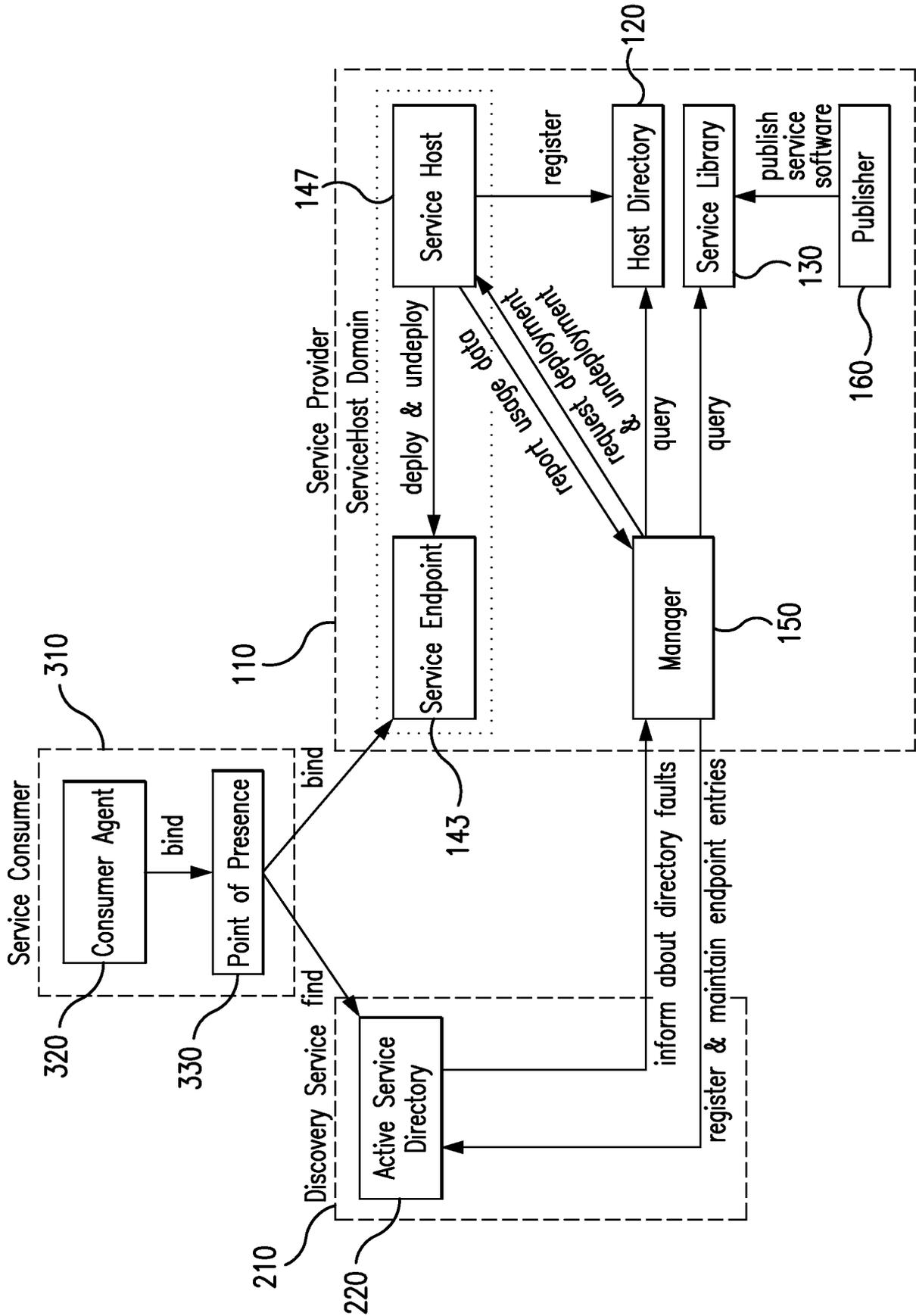


FIG. 3

4/17

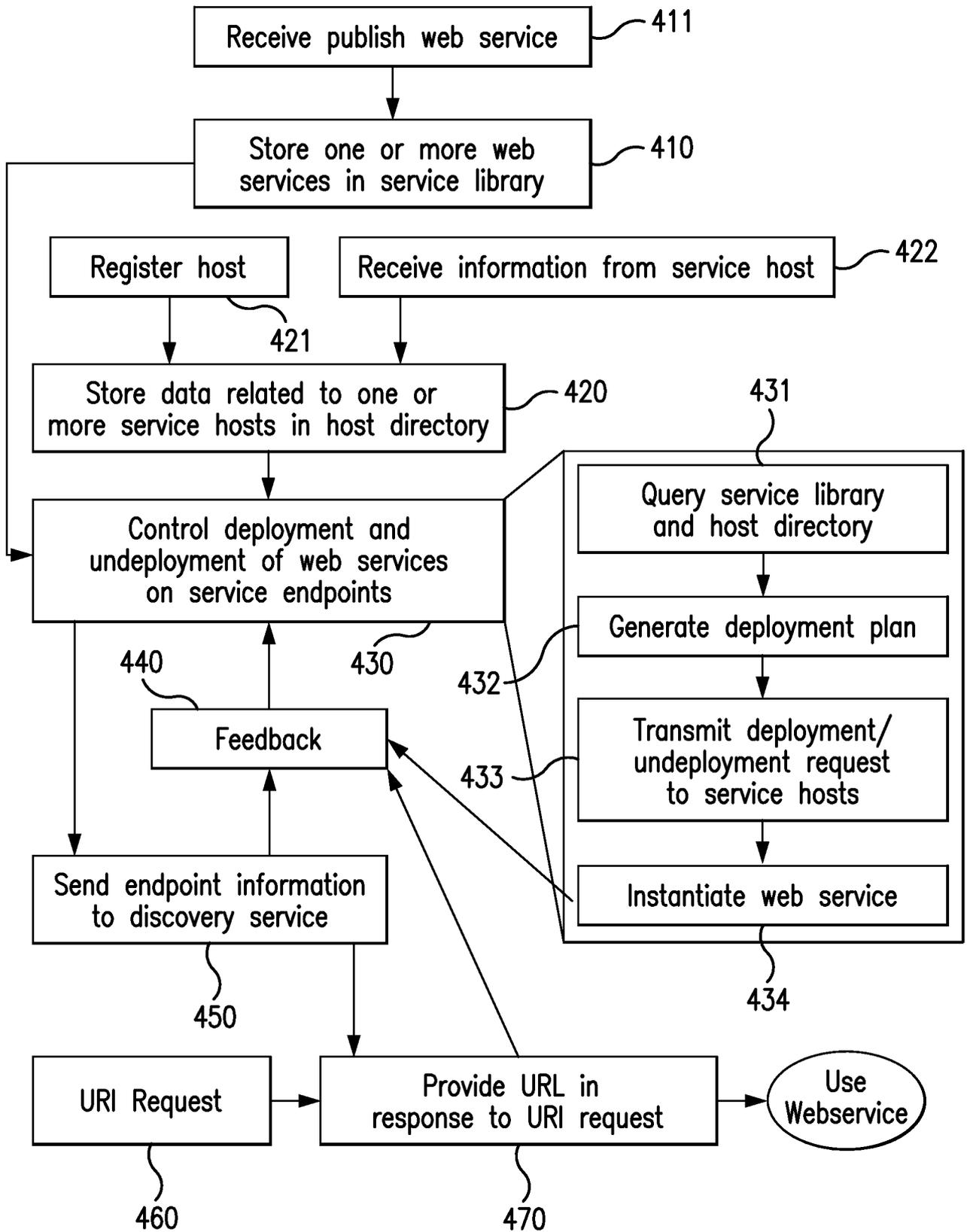


FIG. 4

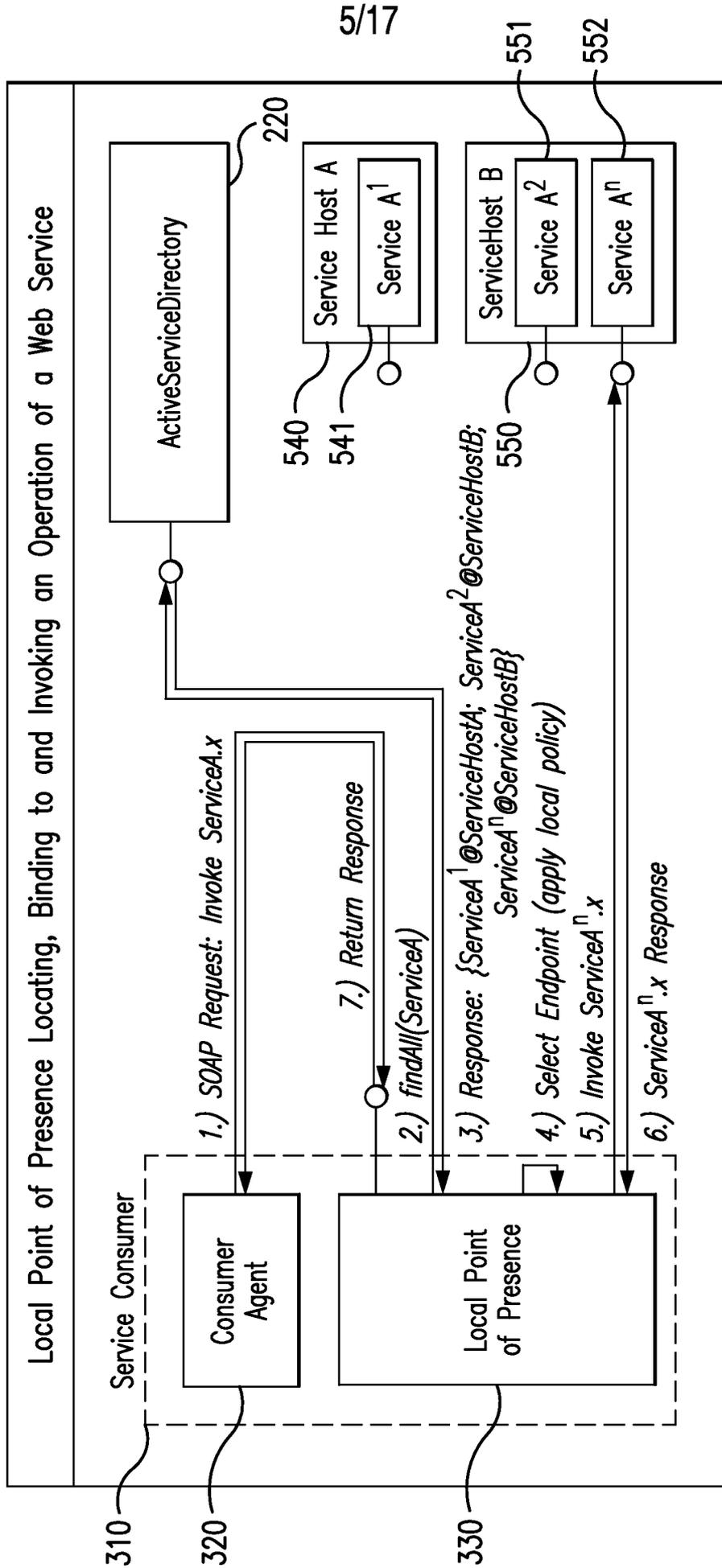


FIG. 5

6/17

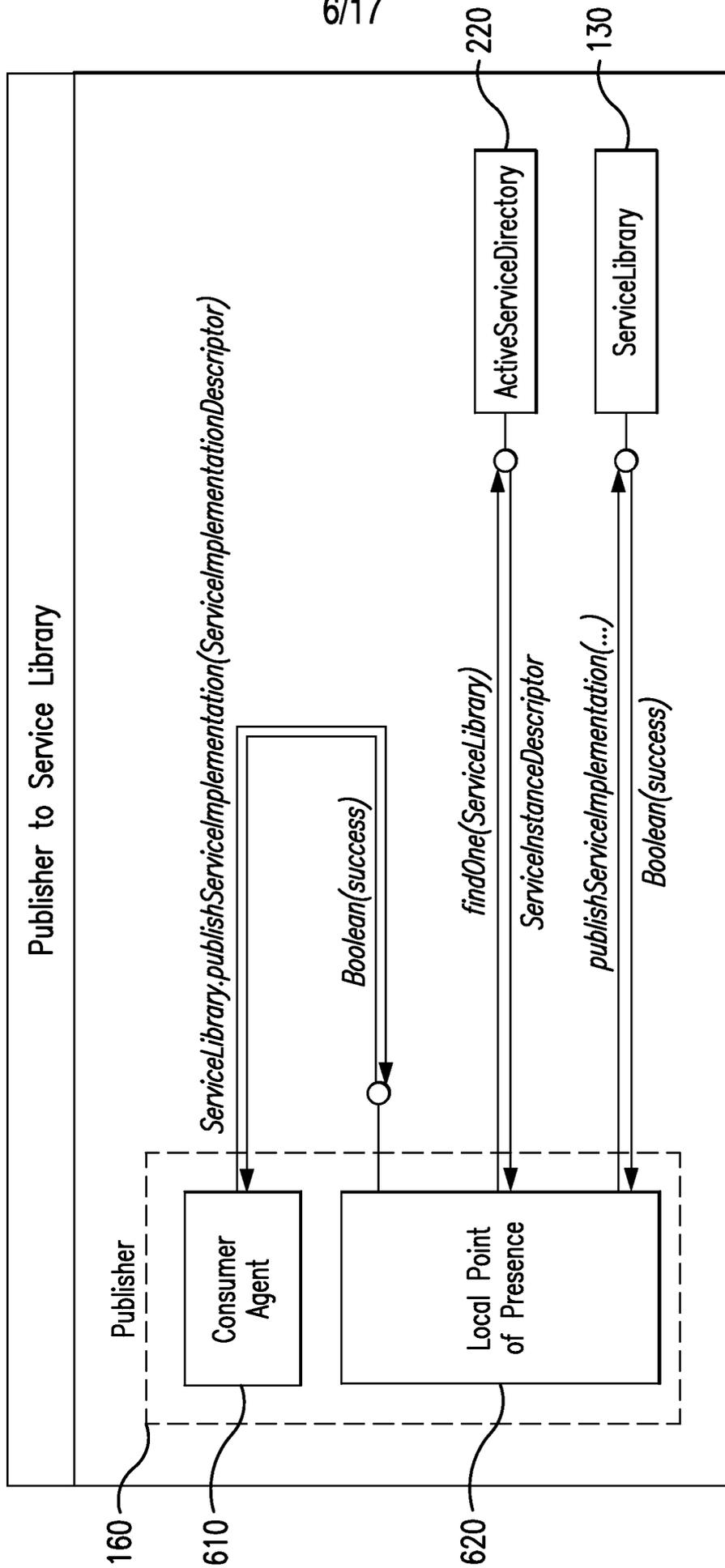


FIG. 6

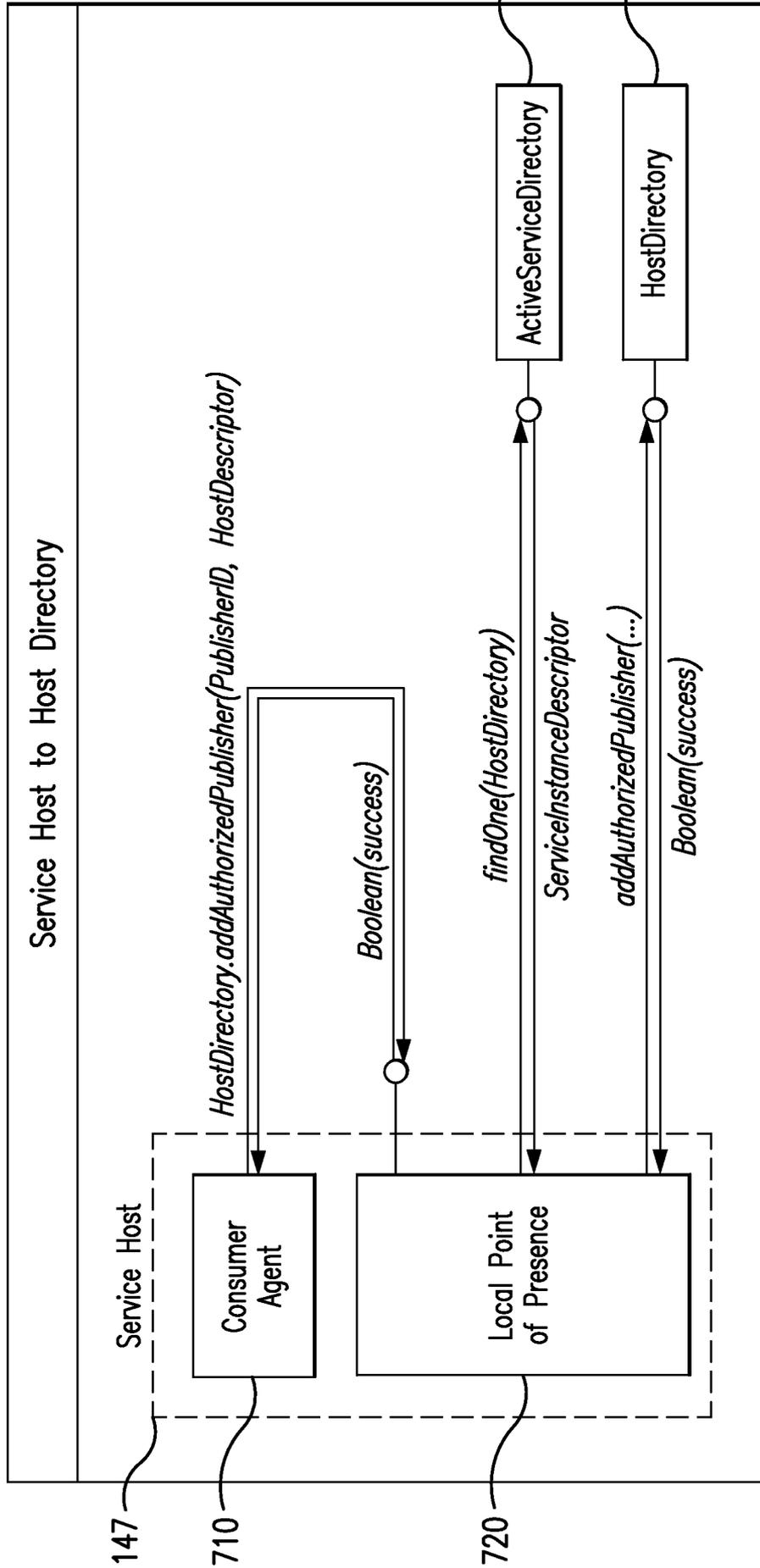


FIG. 7

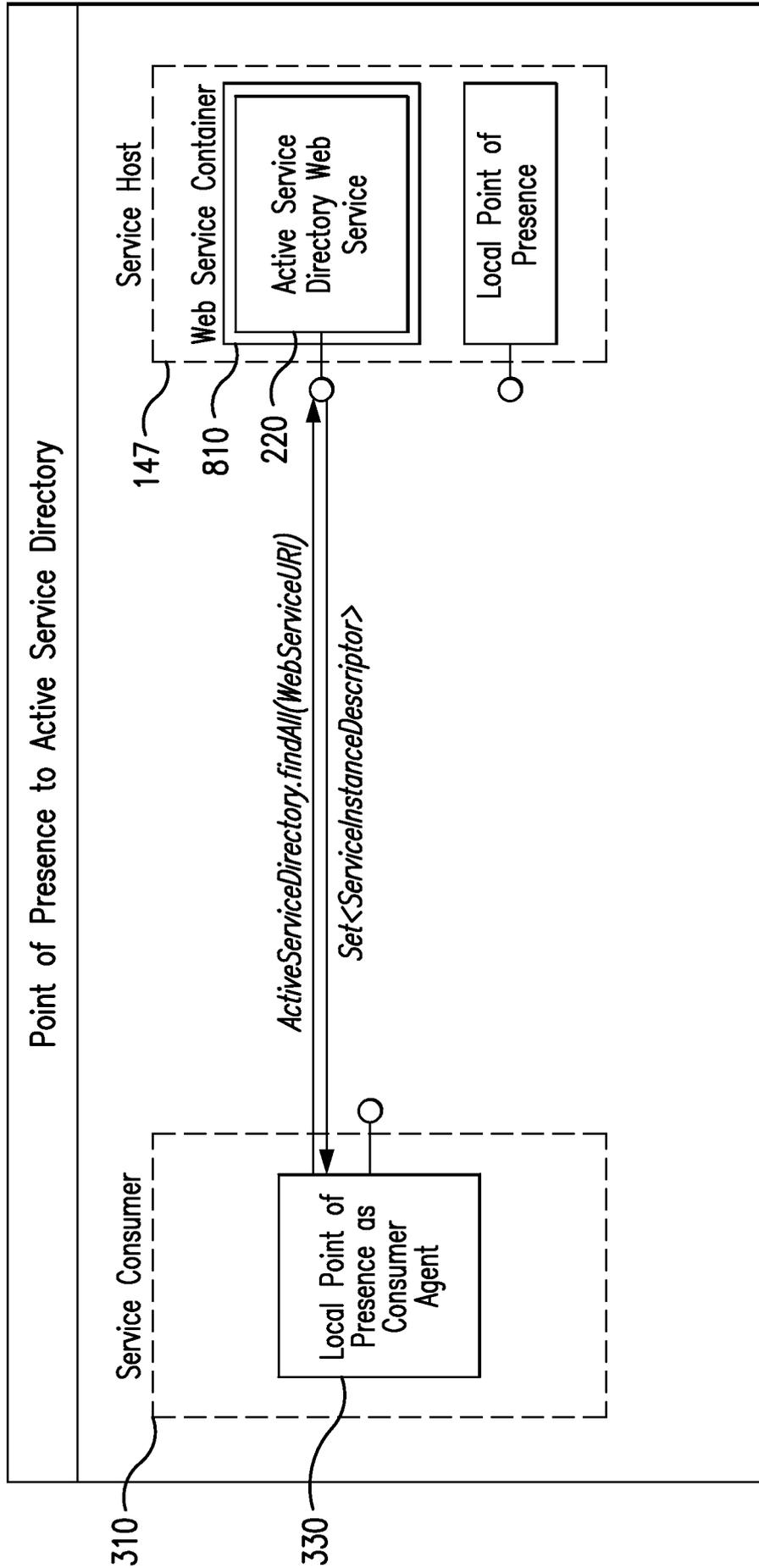


FIG. 8

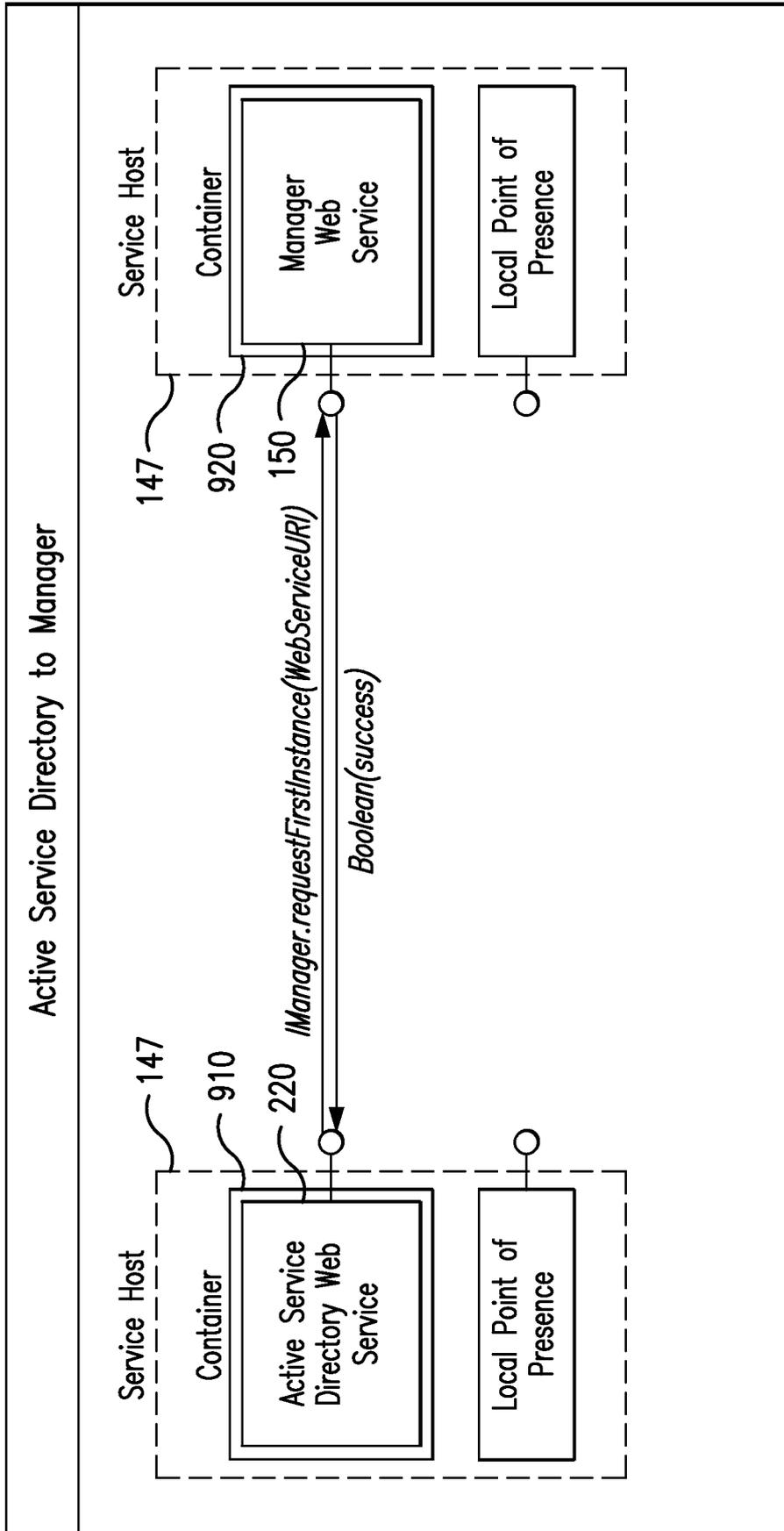


FIG. 9

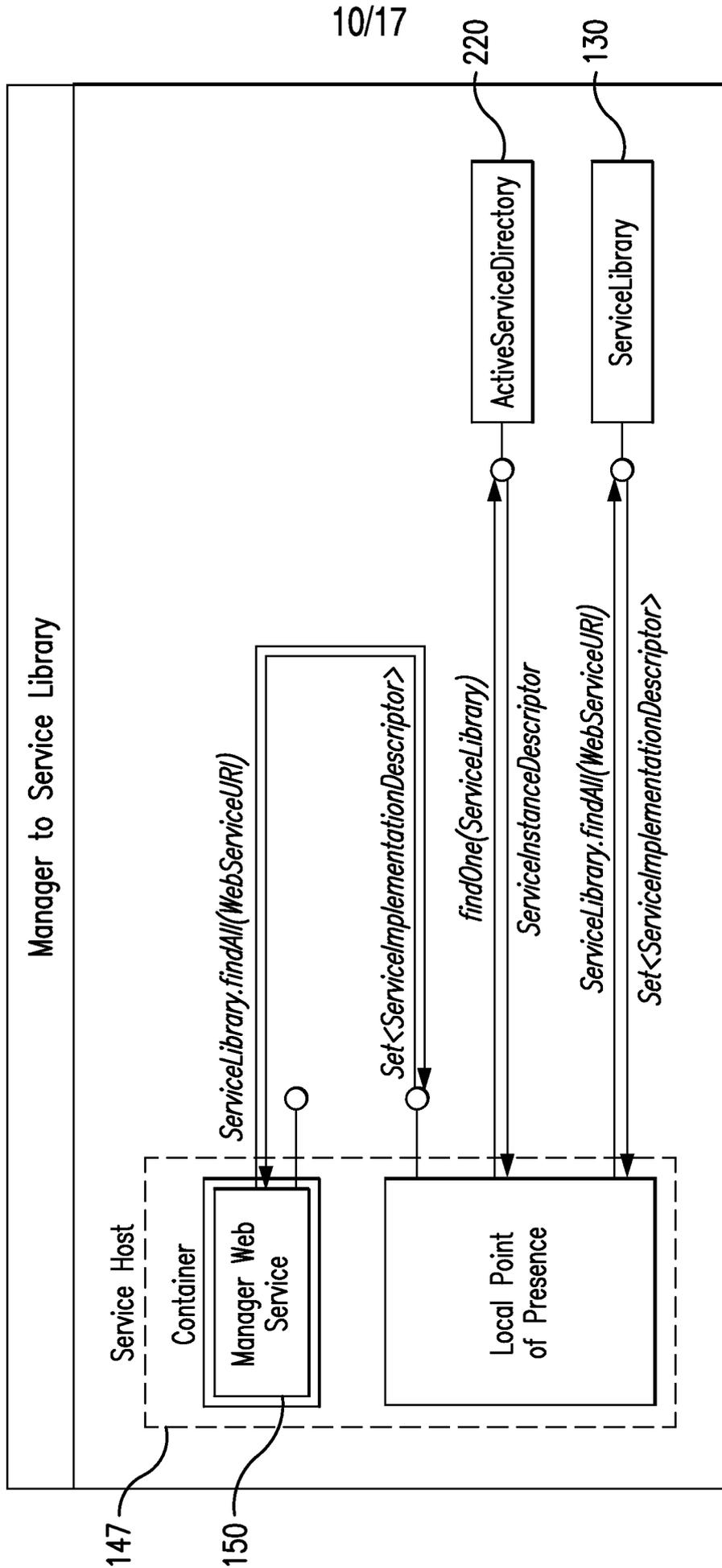


FIG. 10a

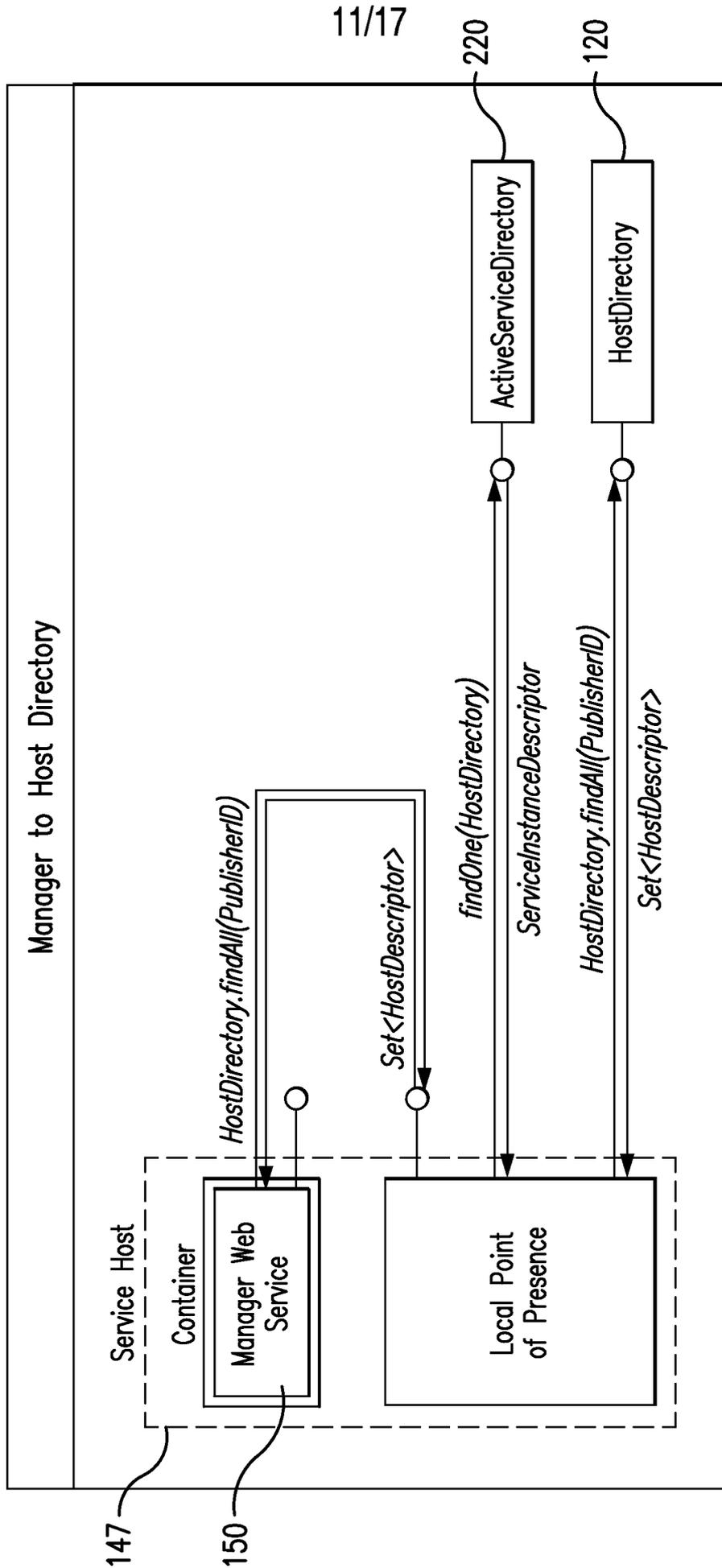


FIG. 10b

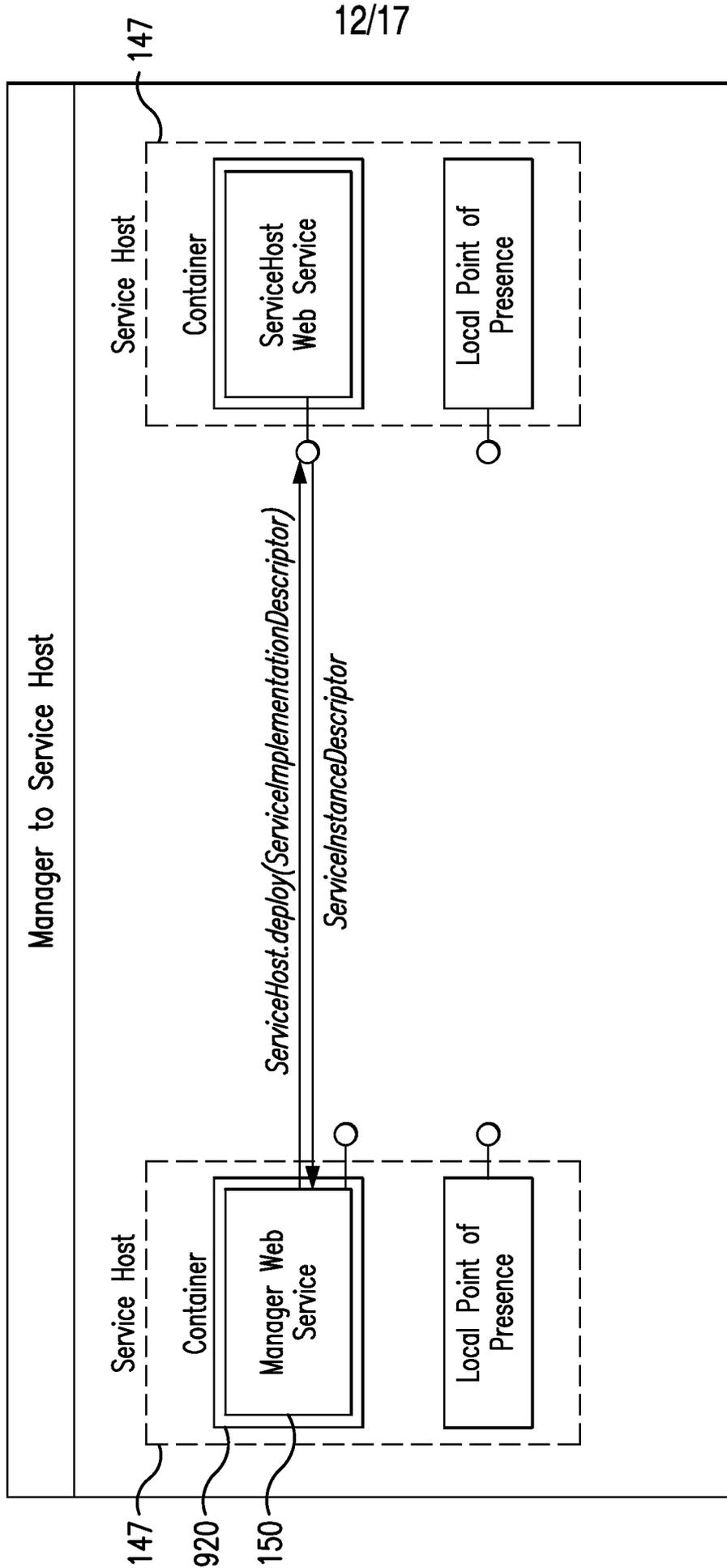


FIG. 10C

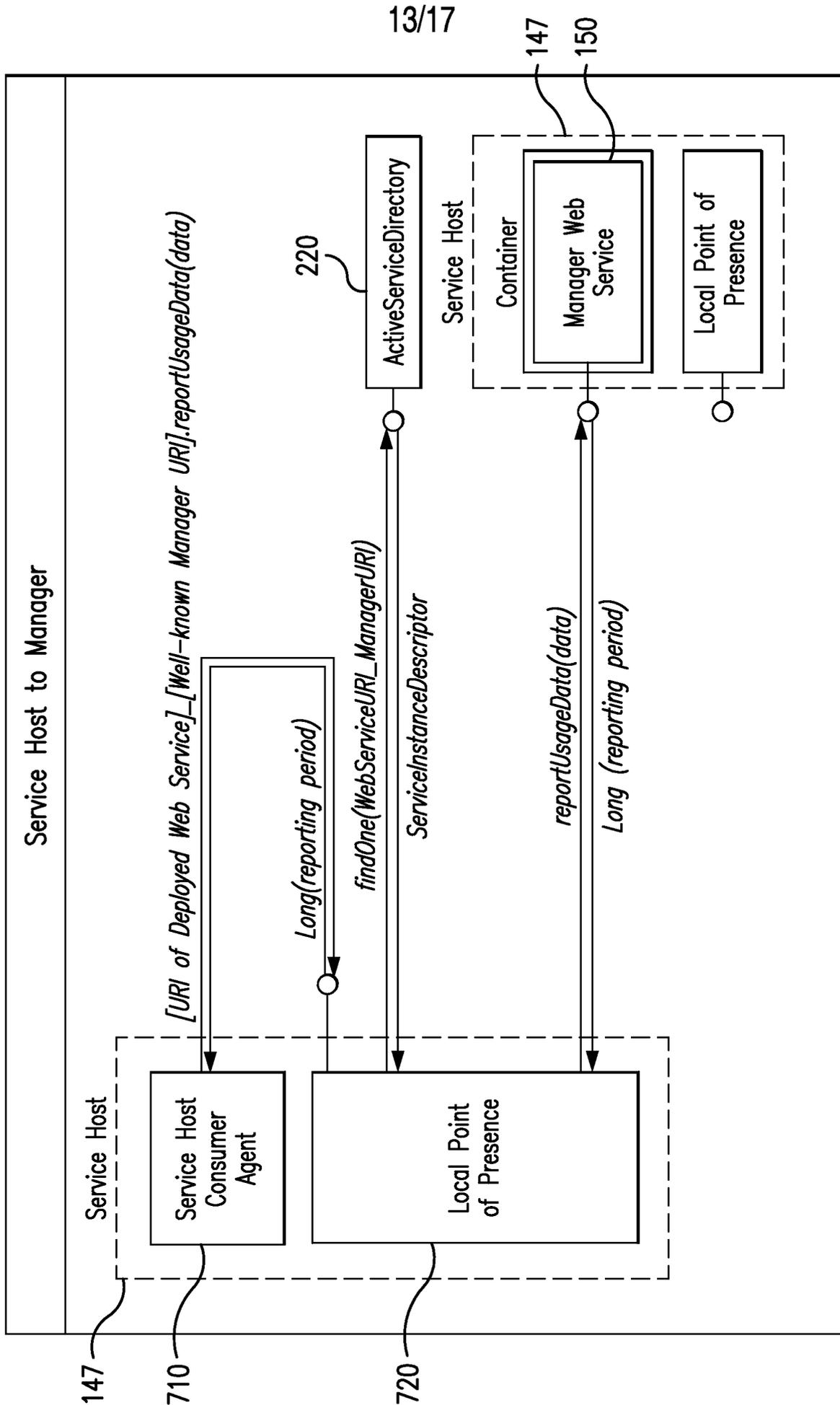


FIG. 10d

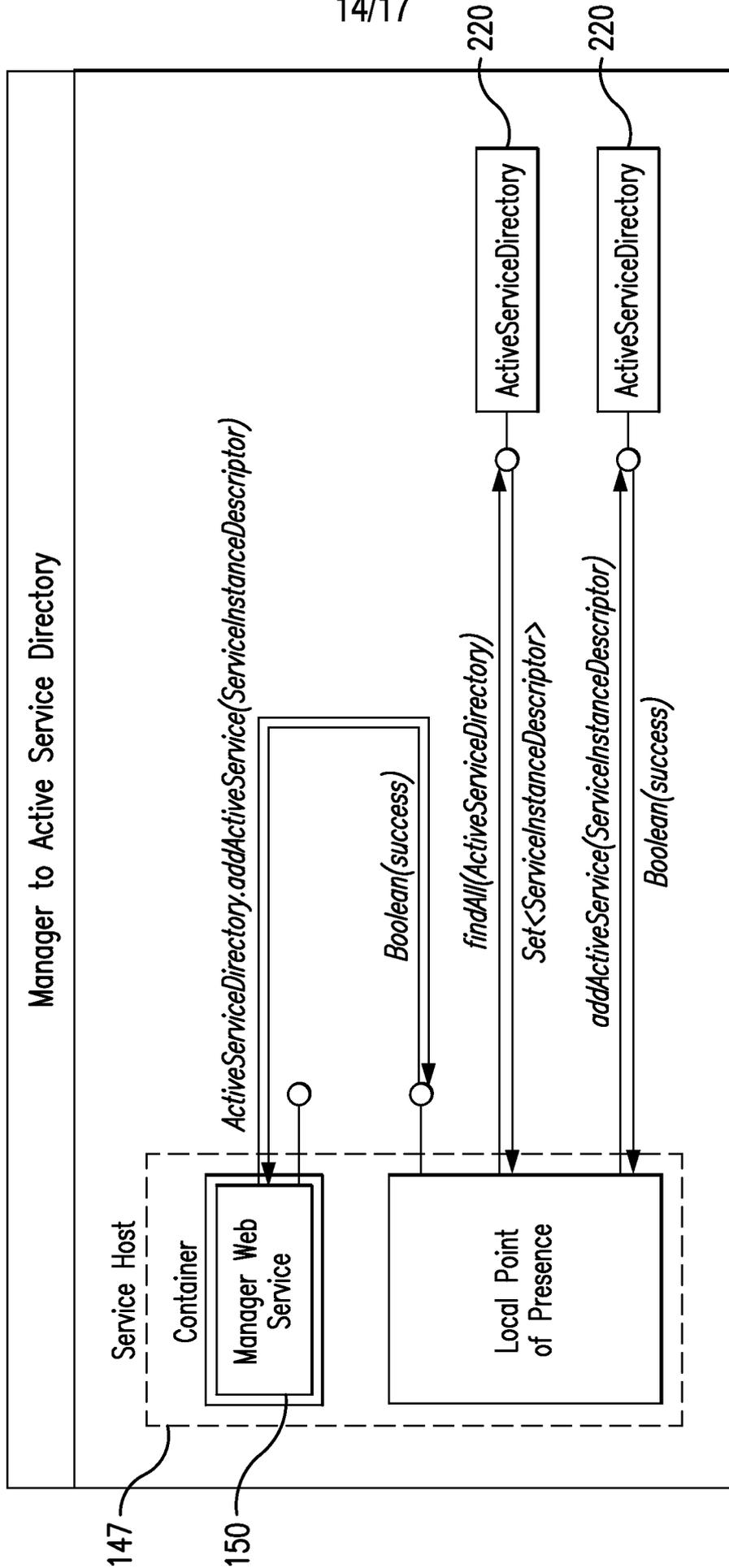


FIG. 10e

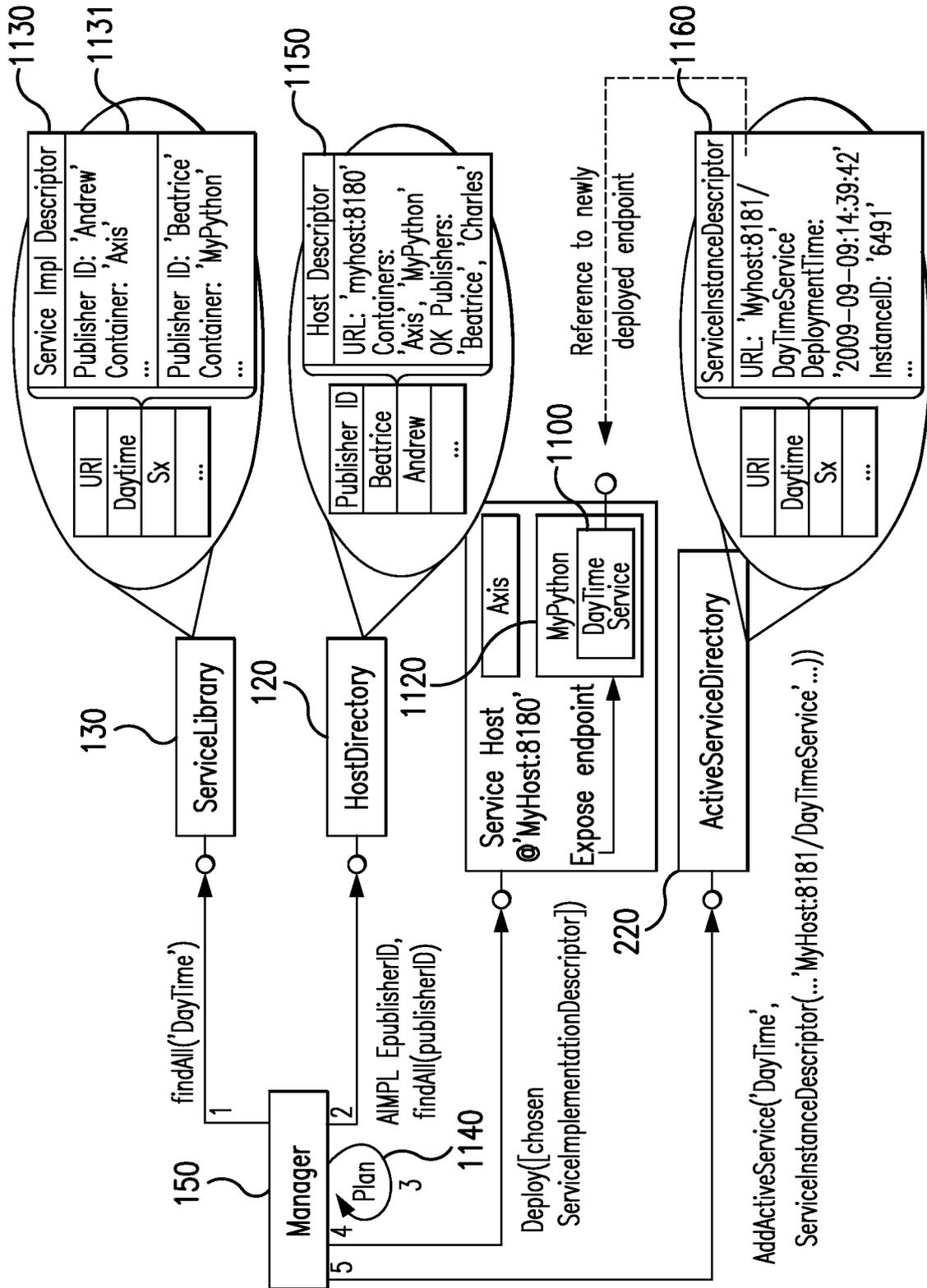


FIG. 11

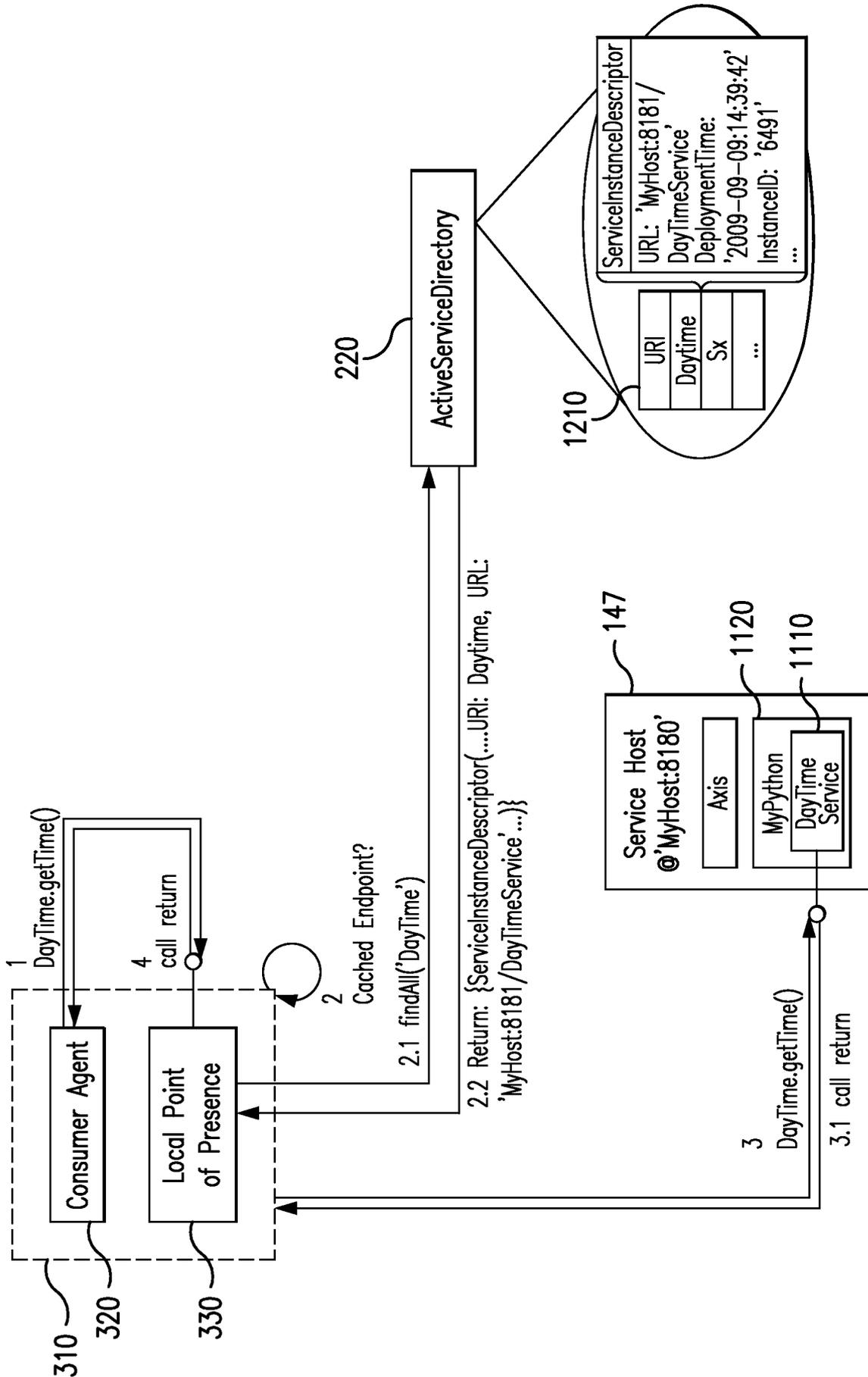


FIG. 12

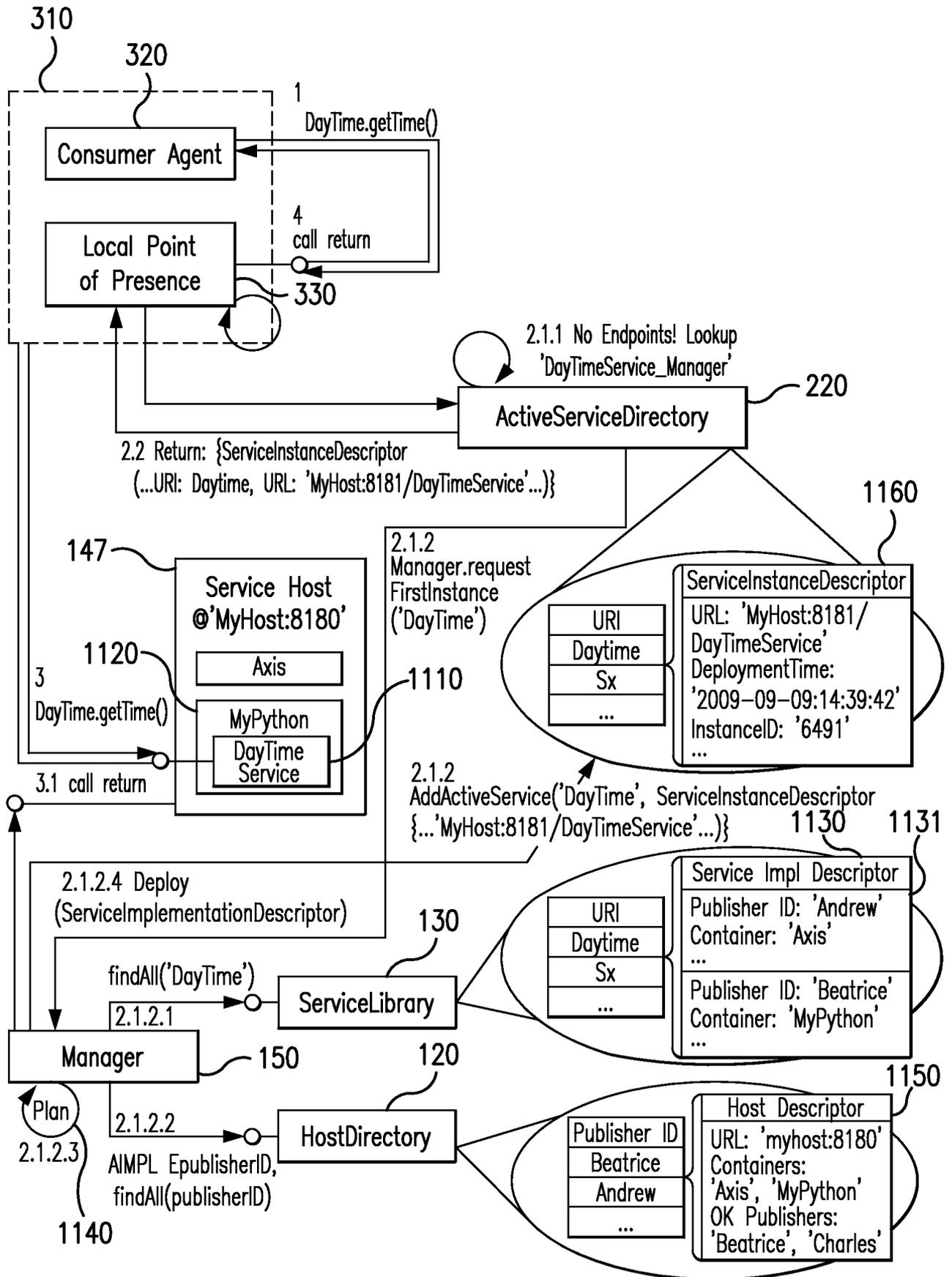


FIG. 13

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 12/56018

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - H04J 1/00 (201 2.01 )

USPC - 370/486

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC(8): H04J 1/00 (2012.01)

USPC: 370/486

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

USPC: 370/326, 335, 336, 342, 367, 442; 709/200, 217, 220; 379/201.05, 221.08; 718/101 ; 719/330 (keyword limited; terms below)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PatBase(All); pubWEST(PGPB, USPT, USOC, EPAB, JPAB); Google(Web); Search terms used: distributed deployment abstraction resolving URI URL endpoint directory library agent proxy gateway SOA SOC point of presence manager bound bind applications location proximity mapping

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006/01 11880 A1 (BROWN et al.) 25 May 2006 (25.05.2006) entire document especially para [0155], [0219J-[0231], [0272]-[0279], [0363]-[0368], [0530], [0622], [0623]	1-31
A	US 7,974,939 B2 (NANJANGUD et al.) 05 July 201 1 (05.07.201 1) entire document	1-31
A	US 2009/0300057 A1 (FRIEDMAN) 03 December 2009 (03.12.2009) entire document	1-31
A	US 2004/0133640 A1 (YEAGER et al.) 08 July 2004 (08.07.2004) entire document	1-31

 Further documents are listed in the continuation of Box C.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

02 November 2012 (02.1 1.2012)

Date of mailing of the international search report

21 NOV 2012

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents  
P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-3201

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300  
PCTOSP: 571-272-7774