Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2463050 C 2012/03/13

(11)(21) 2 463 050

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2002/08/29
(87) Date publication PCT/PCT Publication Date: 2003/03/13
(45) Date de délivrance/Issue Date: 2012/03/13
(85) Entrée phase nationale/National Entry: 2004/02/27
(86) N° demande PCT/PCT Application No.: EP 2002/009652
(87) N° publication PCT/PCT Publication No.: 2003/021904
(30) Priorité/Priority: 2001/08/31 (US60/316,619)

(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01), *H04L 12/16* (2006.01), *H04W 4/00* (2009.01)

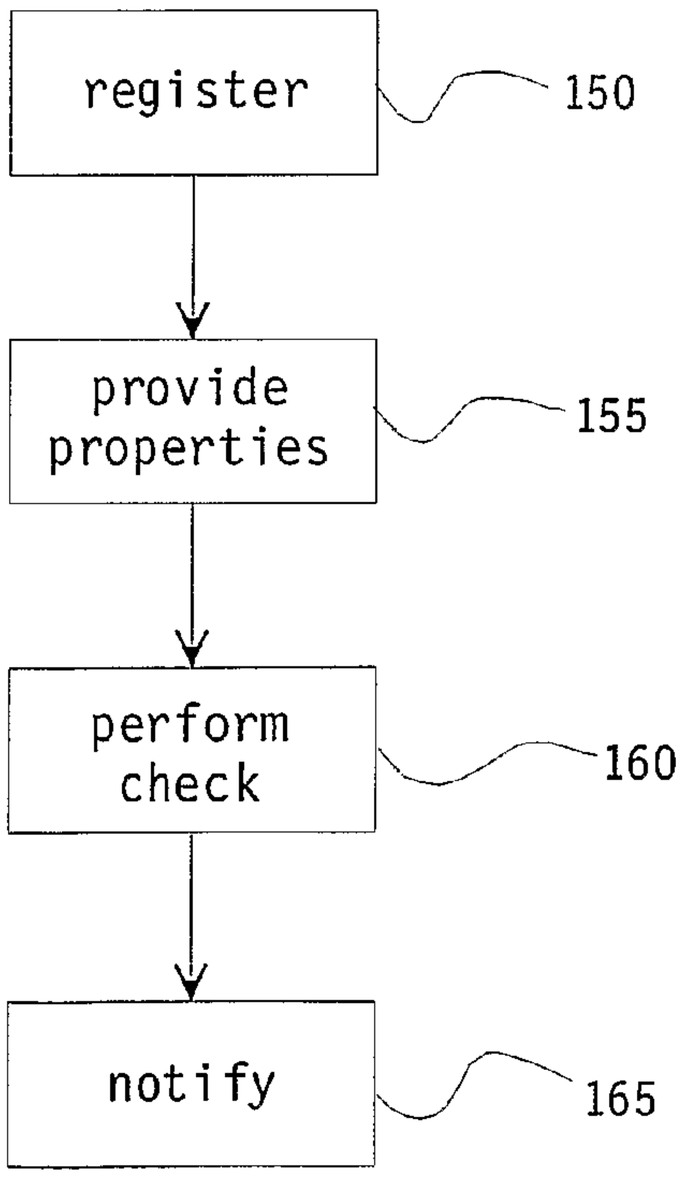
(72) Inventeurs/Inventors:
MOERDIJK, ARD-JAN, NL;
VAN ELBURG, HANS-ERIK, NL

(73) Propriétaire/Owner:
TELEFONAKTIEBOLAGET LM ERICSSON (PUBL), SE

(74) Agent: ERICSSON CANADA PATENT GROUP

(54) Titre : MECANISME DE SUPPORT DE TRANSFERT DANS UNE ARCHITECTURE DE SYSTEMES OUVERTS MOBILES

(54) Title: MIGRATION SUPPORT MECHANISM IN OPEN SERVICE AND OPEN MOBILE ARCHITECTURE



(57) Abrégé/Abstract:
A method for migrating between first and second service enablers in an open service architecture includes registering a second service enabler within open service architecture framework. The properties of the registered second service enabler are compared with the properties of at least one of first service enablers which determines if the second service enabler is backward compatible to the first service enabler. Information concerning whether the second service enabler is backward compatible with the first service enabler is then forwarded to an application that is using the first service enabler. Additional information that can be send to the application in the same message is e.g. the fact that the second service enabler outdates a first one, a migration strategy and references to interfaces on the second service enabler that the application can already start using.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 March 2003 (13.03.2003)

PCT

(10) International Publication Number
WO 03/021904 A3

(51) International Patent Classification[7]: H04L 29/06

(21) International Application Number: PCT/EP02/09652

(22) International Filing Date: 29 August 2002 (29.08.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/316,619 31 August 2001 (31.08.2001) US

(71) Applicant *(for all designated States except US)*: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **MOERDIJK, Ard-Jan** [NL/NL]; Uitoord 42, NL-4824 LE Breda (NL). **VAN ELBURG, Hans-Erik** [NL/NL]; Hagebeemd 5, NL-4907 DM Oosterhout (NL).

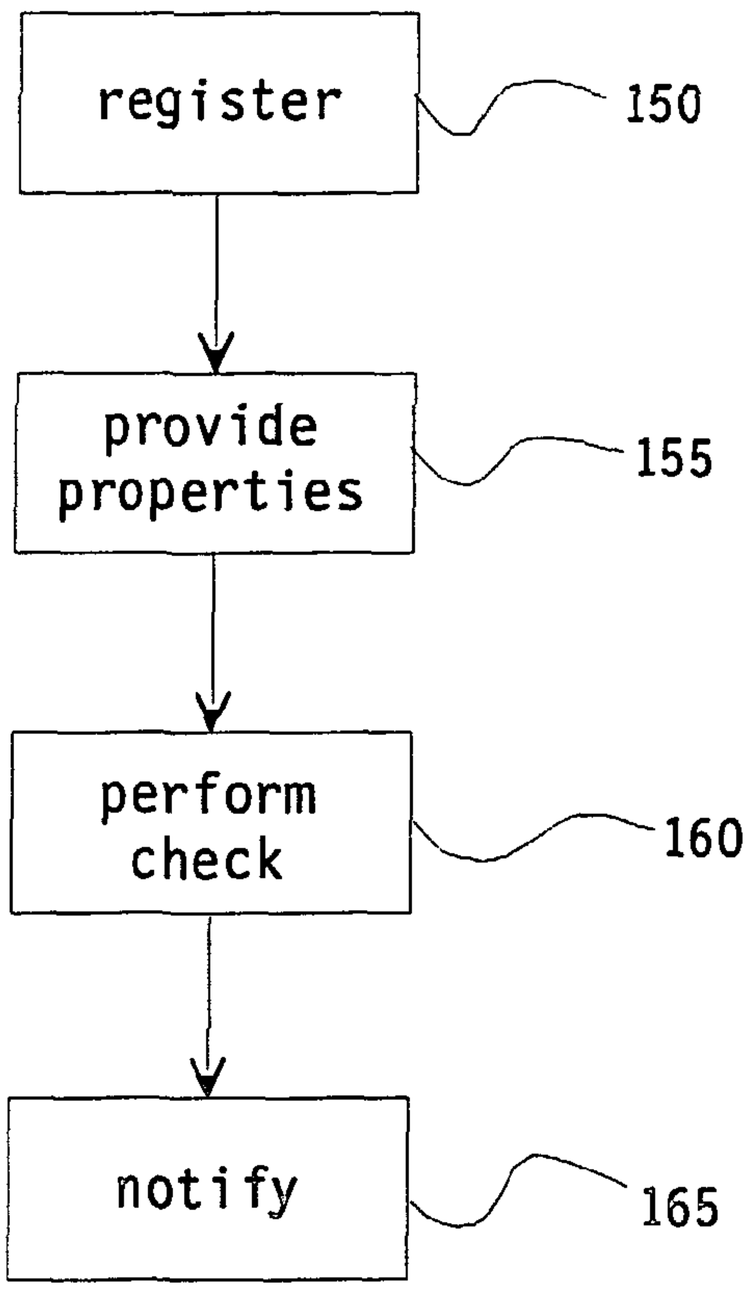
(74) Agent: **VAN KAN J.J.H.**; Algemeen Octrooibureau, World Trade Center, Pastoor Petersstraat 160, 5612 LV Eindhoven (NL).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: MIGRATION SUPPORT MECHANISM IN OPEN SERVICE AND OPEN MOBILE ARCHITECTURE



(57) Abstract: A method for migrating between first and second service enablers in an open service architecture includes registering a second service enabler within open service architecture framework. The properties of the registered second service enabler are compared with the properties of at least one of first service enablers which determines if the second service enabler is backward compatible to the first service enabler. Information concerning whether the second service enabler is backward compatible with the first service enabler is then forwarded to an application that is using the first service enabler. Additional information that can be send to the application in the same message is e.g. the fact that the second service enabler outdates a first one, a migration strategy and references to interfaces on the second service enabler that the application can already start using.

WO 03/021904 A3

WO 03/021904 A3

Published:
— *with international search report*

(88) Date of publication of the international search report:
4 December 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Title

Migration support mechanism in open service and open mobile architecture

Field of the Invention

The present invention relates to the open service architecture, and more particularly, to a method for aiding applications in migrating from one service enabler to another service enabler in the open service architecture.

Background of the Invention

In today's network, applications and services are a part of the network operator's domain and are built using intelligent network technology. This approach is excellent for simple mass-market applications, but with the emergence of mobility and the internet protocol, rapid deployment of innovative applications that combine different features and critical enterprise data becomes a challenge.

A number of industrial forums and standardization bodies, such as Parlay and 3GPP, have addressed this challenge and specified APIs (application program interfaces) that serve as an interface between the applications and core networks. The term Open Service Architecture refers to the set of APIs developed by Parlay, 3GPP and ETSI. Within the open service architecture, a basic mechanism exists to which an application within the OSA can subscribe to be notified when a new service capability feature (SCF) becomes available. However, this mechanism provides no indication as to what extent the new SCF is backward compatible with existing SCFs the application is presently using. Thus, there exists a need within the open service architecture for a mechanism for informing an application of the compatibility of new SCFs with existing SCFs.

Summary of the Invention

The present invention overcomes the foregoing and other problems with a system and method for migrating between a first and

second service in an open service architecture. A second service registers with an open service architecture framework and responsive thereto a comparison is made between the properties of the second service and the properties of the first service to determine if the second service is backward compatible with the first service. Information concerning whether the second service is backward compatible to the first service is forwarded to at least one application using the first service after the comparison.

Brief description of the Drawings

A more complete understanding of the method and apparatus of the present invention may be obtained by reference to the following detailed description when taken in conjunction with the accompanying drawings wherein:

FIGURE 1 illustrates a third generation logical network architecture;

FIGURE 2 illustrates an overview of the open service architecture;

FIGURE 3 illustrates various functionalities of the OSA framework;

FIGURE 4 illustrates a service registration by an application within the open service architecture; and

FIGURE 5 is a flow diagram illustrating the method of the present invention for aiding applications migrating to a new SCF.

Detailed description of the Embodiments

Referring now to FIGURE 1, the network architecture of third generation networks is based upon horizontal layering principles where applications 40 and application servers 45 are logically found in the upper layer, called the application or service network 10. The term service network is used to distinguish it from the core network 15 located in the lower layers. The service network 10 is based upon open

distributed technology (JAVA, CORBA) and applications are able to access core network functionality by means of open and standardized application program interfaces (APIs) 20, through which they may communicate with one or more service capability servers (SCS) 50. SCS 50 may connect to various networks, such as mobile networks 5, IP networks 6 and/or fixed networks 7.

In the layered architecture shown in FIGURE 2, the open service architecture (OSA) comprises an application program interface between the service network 10 and the core network 15. While the invention is described with respect to the Open Service Architecture as defined by Parlay, 3GPP and ETSI, the invention is also applicable for webservice based approaches where the specific implementation details might be a bit different than in OSA. Applications 40 deployed on application servers 45 utilize service capability features that are provided by service capability servers (SCSs) 50. Service capability servers 50 are logical entities that implement the service capability features (SCFs) 35 and interact with the core network 15. The applications 40 and application servers 45 are located within the service network layer 10 as described previously. Thus, it can be seen that the open service architecture 30 acts as an API between the service network layer 10 and the core network layer 15.

The OSA Framework 55 is a registration and discovery server and enables the openness of the open system architecture and makes it possible to go beyond IN (Intelligent Networks) when it comes to openness, discovery and integration of new features as described below. The OSA framework 55 also notifies applications of the addition of new service capability features within the open service architecture 30. The OSA framework 55 provides controlled access to the SCSs 50 which in combination with the distributed technology supports flexibility in application locations in business scenarios. Furthermore, it allows multi-vendorship and even extension of the set of APIs.

As shown in FIGURE 3, the OSA framework 55 is actually a

family of service capability features 35 with the core portion consisting of Trust and Security Management 60 enabling the authentication of domain; Service Discovery 65 enabling the discovery of new SCF provided by the operator; Service Registration 70 providing for the registration of new SCFs to the framework; and Service Factory 75 enabling the creation of new SCF instances. Additionally, APIs are provided for Integrity Management 80 such as load balancing, fault management and heart beat and Event Notification 85 providing notifications for specific events.

Referring now to FIGURE 4, there is illustrated the manner in which an application can start using an SCF 35 provided by a new Service Capability Server (SCS) 50. At a first stage, an SCS 50 will contact the OSA framework 55 and request an authentication and registration interface at steps 90 and 95. Next, the SCS 50 uses the registration interface to publish its capabilities and add a reference to its service factories at step 100. The factory pattern is a general design pattern and allows the OSA framework 55 to request the SCS 50 to create an SCF 35 interface. At this moment, the OSA framework 55 and the SCS 50 know each other.

The application 40 contacts the OSA framework 55 and is authenticated at step 105. The application 40 requests at step 110 a discovery interface. The OSA framework 55 returns a reference or pointer to the discovery interface after which the application 40 uses this interface to request the type of SCF 35 and special capabilities needed by the application 40 at step 115. At this time, the OSA framework 55 tracks whether the application 40 is allowed to use the SCF 35 and under what conditions. This is captured in the service level agreement (SLA) between the network operator and the service provider. If the application is allowed to use the SCF 35, the OSA framework 55 returns all IDs of SCFs 35 that could fulfill the needs of the application.

Next, the application 40 selects at step 120 one of the SCFs 35 and signs the so-called Service Agreement. The OSA framework 55

contacts the service factory of the SCS 50 and forwards the conditions under which the application is allowed to use the SCFs 35 at step 125. The SCS 50 creates an SCF 35 instance that is to be used by this application and is also able to check the conditions at step 130, and the framework returns the reference or pointer to the application at step 135. From this point forward the application is authorized to use the SCF 35. While this described registration and discovery process enables the framework to inform an application of variously available SCFs 35 and via the notification interface the potential availability of a new SCS 50, there exists no mechanism for notifying an application of backward compatibility of a new SCF 35 with preexisting SCFs 35. Applications can use the event notification API on the framework 55 to subscribe to events. One example of an event is when a new SCF is made available.

Referring now to FIGURE 5, here is illustrated a method for determining backward compatibility for an SCF. When a new SCF is made available, the SCF must first register at step 150 with the OSA framework 55 as described previously with respect to FIGURE 4. During this process, the SCF supplies the OSA framework 55 with the properties supported by this implementation of the SCF at step 155. The OSA framework 55 has information about each of the existing SCF implementations available within a specific network operator domain, information about the applications using them and the restrictions applying to the usage of the SCFs by the service level agreements. Using this information the OSA frameworks is able to perform a check at step 160 of the properties of the new SCF implementation against previously existing versions. In this check, an indication is obtained as to what extent the new SCF implementation is backward compatible with other versions of SCFs used by the network. This information is forwarded at step 165 to the applications using previous versions of the SCF together with optional references to the interface of the new SCS. In order to provide references to the interfaces an extension of the current framework notification mechanism would be utilized. The framework notification

mechanism can be guided by dedicated service properties that specify that the SCF implementation replaces or outdates a specified older SCF implementation or specify a migration strategy. The information about level of backward compatibility, the fact that the SCF implementation replaces or outdates an older SCF, migration strategy, etc. might be supplied either by the new SCS and sent to the application via the framework or by the framework after analyzing the properties of the new SCS when it registers or by both in conjunction.

By implementation of the this described extension, seamless migrations or applications between older and newer versions of SCF implementations can be enabled. This would allow an operator to outdate or update an SCF almost automatically. It would further allow an operator to take one SCF out of service for maintenance activities by directing the applications to a backup SCF.

The previous description is of a preferred embodiment for implementing the invention, and the scope of the invention should not necessarily be limited by this description. The scope of the present invention is instead defined by the following claims.

CLAIMS

1. Method comprising steps of:

registering a second service with a registration and discovery server;

comparing properties of the second service with properties of at least one first service to determine if the second service is backward compatible to the at least one first service; and

forwarding to at least one application using the at least one first service, information concerning whether the second service is backward compatible to the at least one first service.

2. The method according to claim 1, wherein the registration and discovery server comprises an open service architecture framework.

3. The method according to anyone of claims 1-2, further comprising a step of providing an open service architecture framework with at least one property supported by the second service.

4. The method according to anyone of claims 1-3, wherein the step of forwarding further comprises a step of forwarding information concerning whether the second service is backward compatible to the at least one first service to a plurality of applications using the at least one first service.

5. The method according to anyone of claims 1-4, wherein the second and at least one first services comprises a service capability feature.

6. The method according to anyone of claims 1-5, wherein the step of forwarding further comprises the steps of forwarding pointers to at least one interface of a functional entity providing the service.

7. The method according to Claim 6, wherein the functional entity comprises a service capability server.

8. The method according to anyone of claims 1-7, wherein the information specifies if the second service replaces the at least one first service.

9. The method according to anyone of claims 1-8, wherein the information specifies if the second service outdates the at least one first service.

10. The method according to anyone of claims 1-9, wherein the information specifies a migration strategy from the second service to the at least one first service.

11. The method according to anyone of claims 1-10, wherein the information is provided by the second service.

12. The method according to anyone of claims 1-10, wherein the information is provided by the registration and discovery service.

13. The method according to anyone of claims 1-9, further comprising a step of replacing the at least one first service with the second service.

14. Open system architecture, comprising:

at least one service capability server providing at least a first service capability feature and a second service capability feature; and

an open system architecture framework configured to register the second service capability feature with the open service architecture framework, compare properties of the second service capability feature with properties of the first service capability feature to determine if the second service capability feature is backward compatible to the first service capability feature and forward to at least one application using the first service capability feature, information concerning whether the second service capability feature is backward compatible to the first service capability feature.

15. The open system architecture according to claim 14, wherein the open system architecture framework is further configured to determine at least one property supported by the second service capability feature.

16. The open system architecture according to any of the claims 14 and 15, wherein the open system architecture framework is further configured to forward information concerning whether the second service capability feature is backward compatible to the first service capability feature to a plurality of applications using the first service capability feature.

17. The open system architecture according to any of the claims 14-16, wherein the open system architecture framework is further configured to forward pointers to at least one interface of the at least one service capability server providing the first service capability feature.

18. The open system architecture according to any of the claims 14-17, wherein the information specifies if the second service capability feature replaces the first service capability feature.

19. The open system architecture according to any of the claims 14-18, wherein the information specifies if the second service capability feature outdates the first service capability feature.

20. The open system architecture according to any of the claims 14-19, wherein the information specifies a migration strategy from the second service capability feature to the first service capability feature.

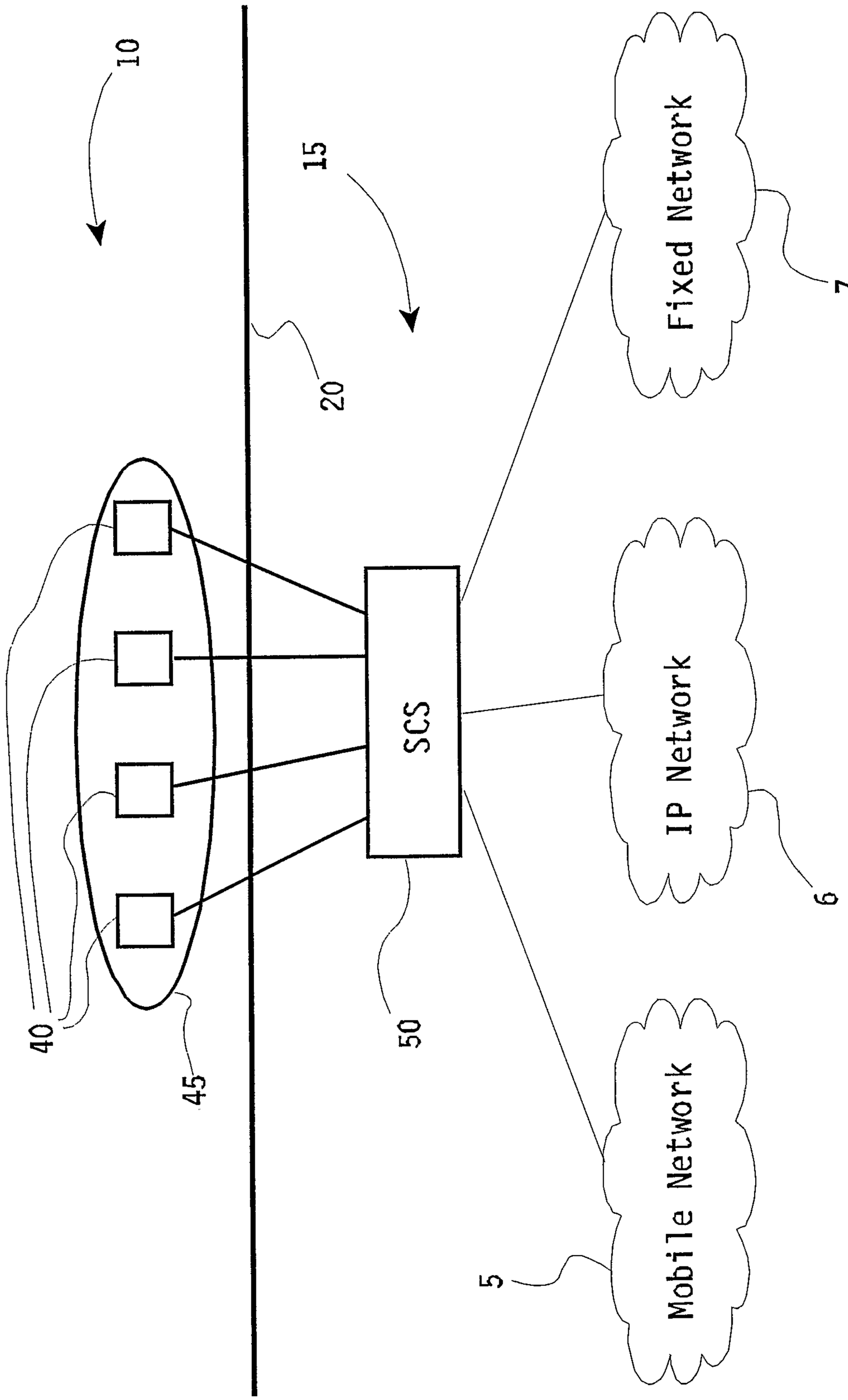

10
15
20
Fixed Network
7
SCS
IP Network
6
40
45
50
Mobile Network
5

FIG. 1

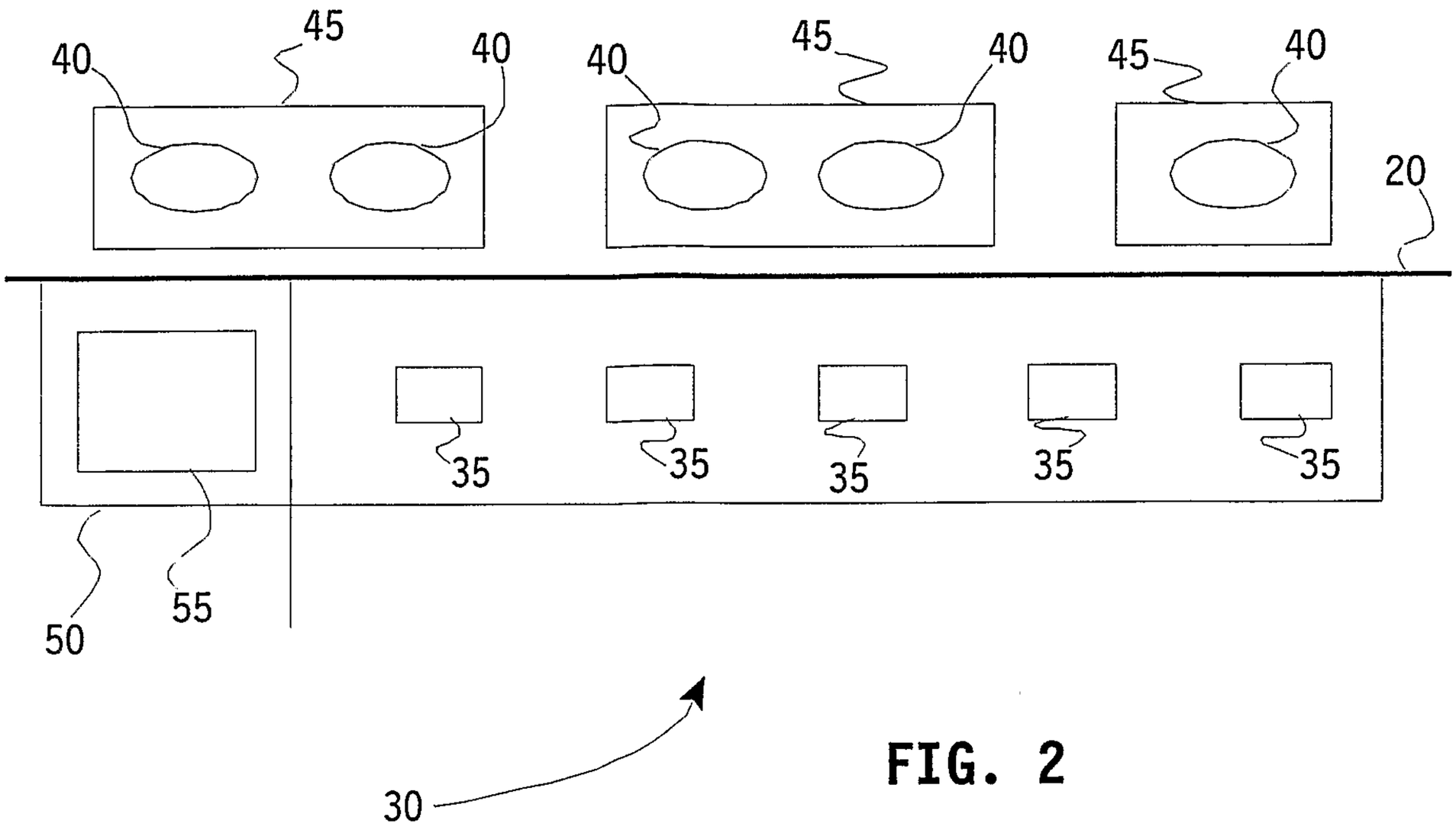

45
40
40
45
40
40
45
40
20
35
35
35
35
35
50
55
30

FIG. 2

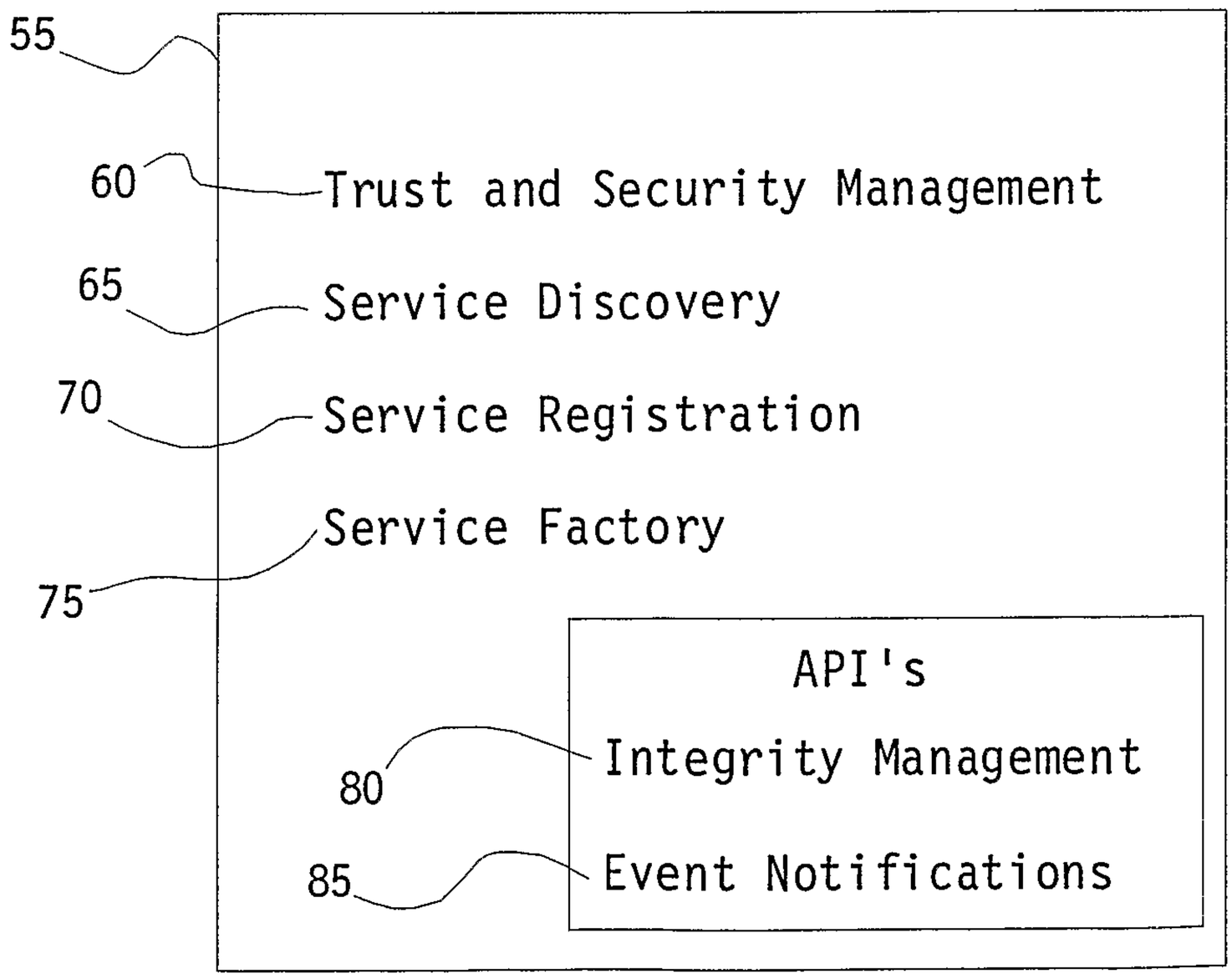

55
60
Trust and Security Management
65
Service Discovery
70
Service Registration
Service Factory
75
API's
80
Integrity Management
85
Event Notifications

FIG. 3

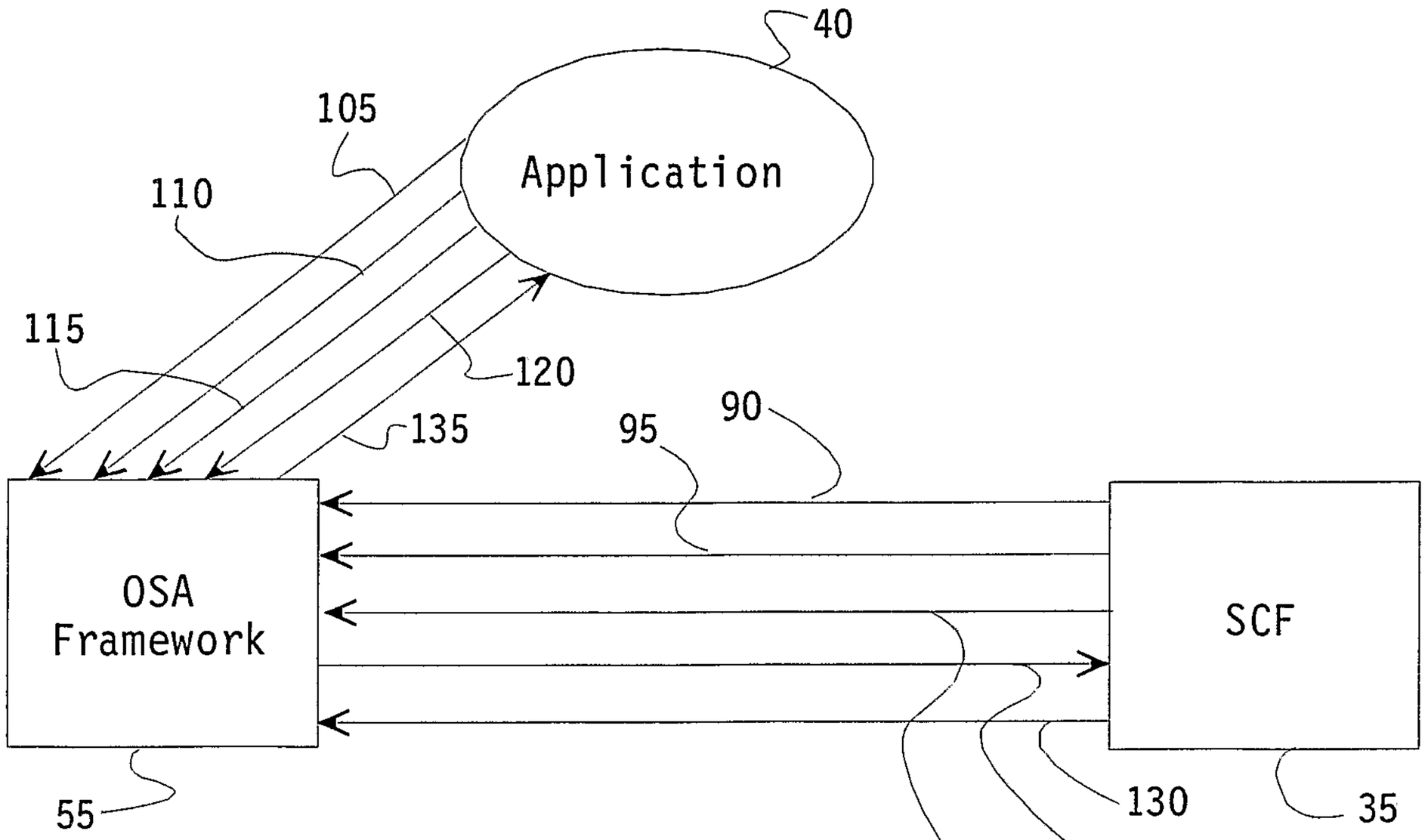

40
Application
105
110
115
120
135
95
90
OSA
Framework
SCF
55
100
125
130
35

FIG. 4

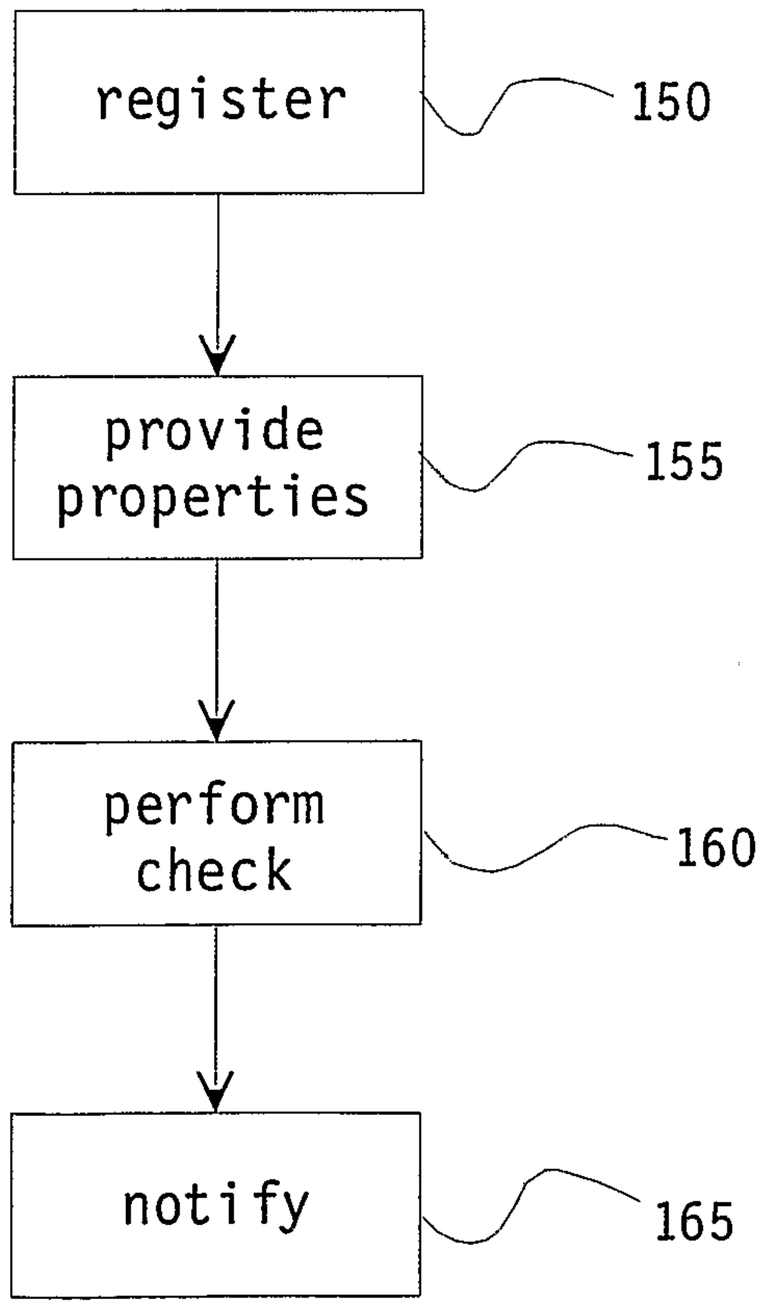
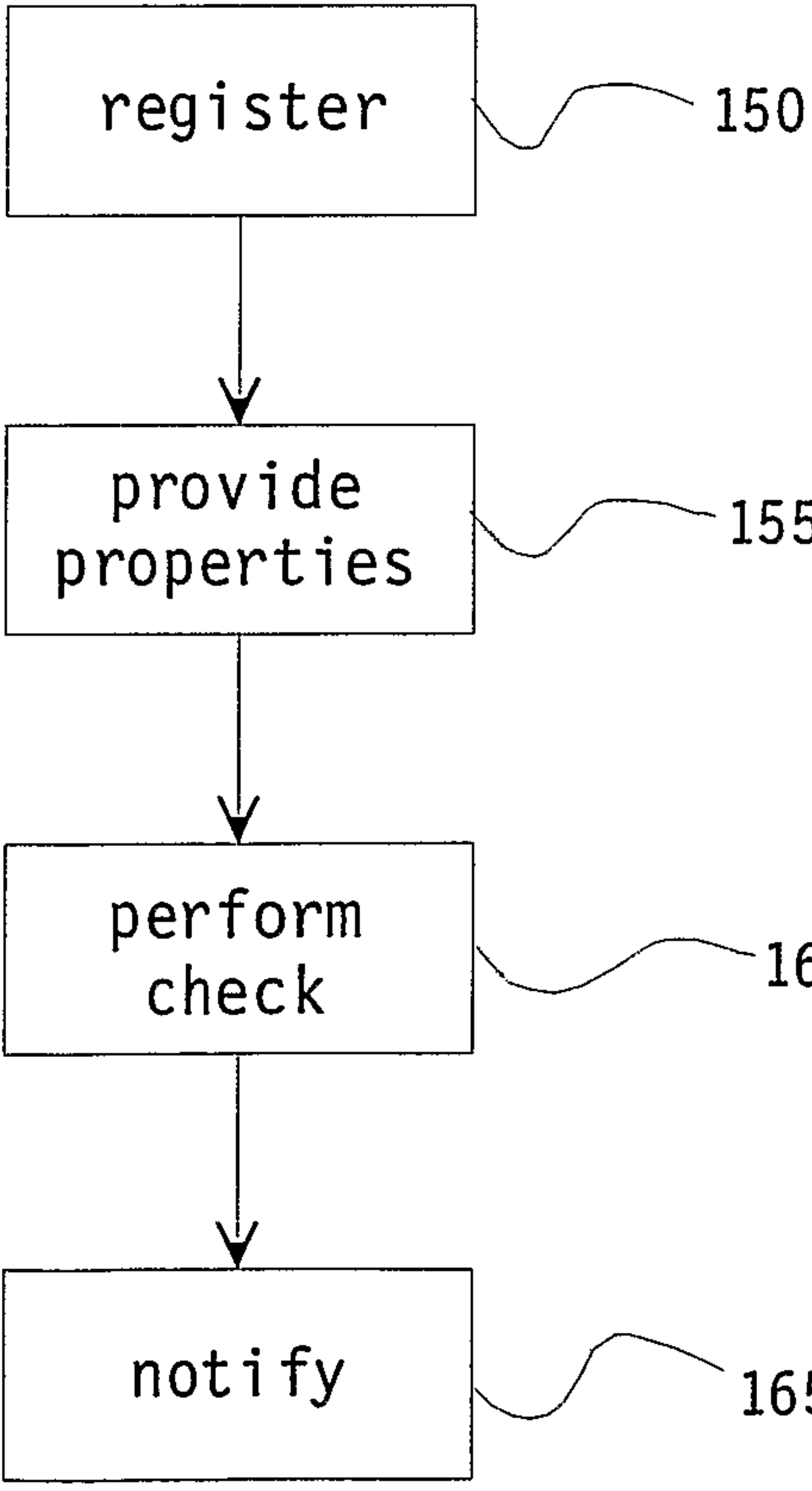

register
150
provide
properties
155
perform
check
160
notify
165

FIG. 5


register
150
provide properties
155
perform check
160
notify
165