



(43) International Publication Date
12 January 2012 (12.01.2012)

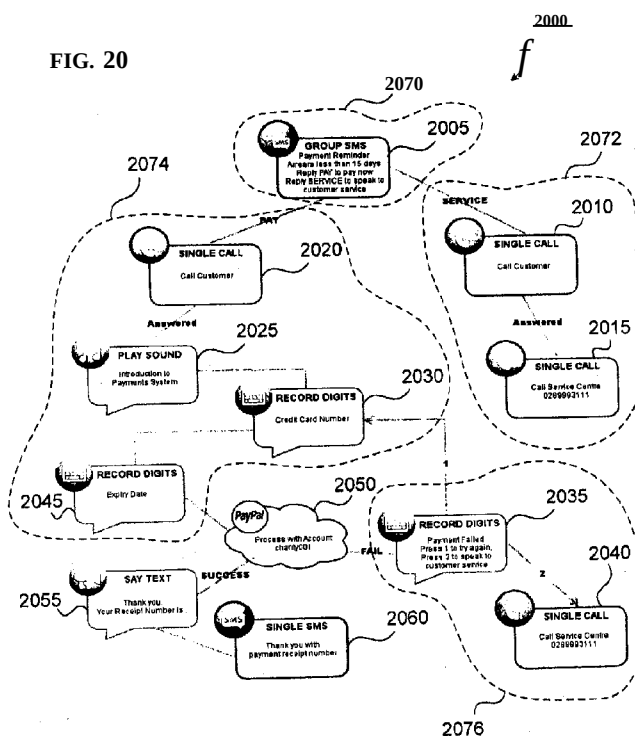
PCT

- | | |
|---|--|
| <p>(51) International Patent Classification:
 <i>H04B 5/04</i> (2006.01) <i>H04W 4/16</i> (2009.01)</p> | <p>[CN/AU]; 244 Pennant Hills Road, Carlingford, NSW 2118 (AU).</p> |
| <p>(21) International Application Number:
 PCT/AU201 1/000846</p> | <p>(74) Agent: SPRUSON & FERGUSON; GPO Box 3898, Sydney, NSW 2001 (AU).</p> |
| <p>(22) International Filing Date:
 5 July 201 1 (05.07.201 1)</p> | <p>(81) Designated States (<i>unless otherwise indicated, for every kind of national protection available</i>): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, 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, PE, PG, PH, PL, PT, RO, RS, RU, 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.</p> |
| <p>(25) Filing Language: English</p> | |
| <p>(26) Publication Language: English</p> | |
| <p>(30) Priority Data:
 2010902979 5 July 2010 (05.07.2010) AU
 201 1901844 13 May 201 1 (13.05.201 1) AU</p> | |
| <p>(71) Applicant (<i>for all designated States except US</i>): IP-SCAPE PTY LTD [AU/AU]; Level 8, 140 Arthur Street, North Sydney, NSW 2060 (AU).</p> | |
| <p>(72) Inventors; and</p> | |
| <p>(75) Inventors/ Applicants (<i>for US only</i>): YOUNGER, Darren [AU/AU]; 152B Alfred Street, Narrabeena, NSW 2099 (AU). KOLIYAT, Vinuraj [IN/AU]; 32/35-37 Darcy Road, Westmead, NSW 2145 (AU). NIU, John</p> | |
| | <p>(84) Designated States (<i>unless otherwise indicated, for every kind of regional protection available</i>): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, BR, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,</p> |

[Continued on next page]

(54) Title: CALL CONVERSATION MANAGER

FIG. 20



(57) Abstract: Disclosed herein are a call conversation mapping method and system. The method creates a call conversation map (2000) that includes a plurality of objects (2005, 2010, 2015) and at least one connecting link to define a call handling process, wherein the objects (2005, 2010, 2015) include at least one task (2010) relating to receiving an incoming communication or generating an outgoing communication. The method activates the call conversation map (2000) in response to a trigger (2140), and processes communications in accordance with the call handling process of the call conversation map (2000).

WO 2012/003534 A1



SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, **Published:**
GW, ML, MR, NE, SN, TD, TG).

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

- 1 -

CALL CONVERSATION MANAGER

REFERENCE TO RELATED PATENT APPLICATIONS

This application is related to Australian Provisional Patent Application No. 2010902979, entitled "Systems and methods for managing communication data", filed
5 5 July, 2010 in the name of IPscape Pty Limited, the entire contents of which are hereby incorporated by reference as if fully set forth herein.

This application is also related to Australian Provisional Patent Application No. 201 1901843, entitled "Contact centre system and method", filed on 13 May, 201 1 in the name of IPscape Pty Limited, the entire contents of which are hereby incorporated by
10 reference as if fully set forth herein.

This application is also related to Australian Provisional Patent Application No. 201 1901844, entitled "Call conversation manager", filed on 13 May, 201 1 in the name of IPscape Pty Limited, the entire contents of which are hereby incorporated by reference as if fully set forth herein.

15 TECHNICAL FIELD

The present disclosure relates to systems and methods for managing communication data. The arrangements presently disclosed have been particularly developed for providing communication management functionalities via a scalable distributed architecture, for example in the context of a call centre or dial-out campaign
20 arrangement. While some embodiments will be described herein with particular reference to that application, it will be appreciated that the present disclosure is not limited to such a field of use, and is applicable in broader contexts.

BACKGROUND

Any discussion of the background art throughout the specification should in no
25 way be considered as an admission that such art is widely known or forms part of common general knowledge in the field.

There are various situations where a party has a need to utilise a plurality of communication lines, for example in the context of operating a call centre, or implementing a dial-out campaign, such as occurs in conjunction with marketing activities.
30 However, the nature of existing technology limits scalability and flexibility. For example, it is difficult for a party to upscale resources for a relatively short time period.

- 2 -

Furthermore, there are inherent inefficiencies in terms of resource utilisation.

There is a need in the art for improved systems and methods for managing communications data.

SUMMARY

5 It is an object of the present invention to overcome substantially or at least ameliorate at least one of the disadvantages of the prior art, or to provide a useful alternative.

According to a first aspect of the present disclosure, there is provided a system for managing communication data including:

10 a managing core configured for centrally managing a plurality of distributed nodes; and

a plurality of distributed nodes, each node being configured to execute:

a channel driver module that provides access to a plurality of communications channels including at least an incoming call channel and outgoing call channel;

15 a communications gateway component for managing communications between the node and the managing core; and

a node application module that provides logic to drive node functionalities.

One embodiment provides a system including a load balancing layer for load balancing incoming calls, wherein the load balancing component receives data indicative
20 of an incoming call from an external gateway, processes that data thereby to select a node for that incoming call, and provides an instruction for the external gateway to communicate directly with the selected node in respect of that incoming call.

One embodiment provides a system wherein the load balancing layer includes a plurality of distributed load balancing components, each load balancing component being
25 configured to receive data indicative of an incoming call from an external gateway, process that data thereby to select a node for that incoming call, and provide an instruction for the external gateway to communicate directly with the selected node in respect of that incoming call.

One embodiment provides a system wherein the managing core is defined by a
30 plurality of distributed sub-cores.

One embodiment provides a system wherein the plurality of distributed sub-cores are configured for intercommunication thereby to allow synchronisation of

- 3 -

predefined selection of data aspects.

One embodiment provides a system wherein the plurality of nodes includes a first node within a first local system and a second node within a second local system, the second local system being distributed with respect to the first local system.

5 One embodiment provides a system wherein the plurality of communications channels include ingoing and outgoing Short Message Service (SMS) messages.

One embodiment provides a system wherein inbound SMS messages are matched to an SMS conversation database thereby to determine whether a particular inbound SMS message is a reply to a current conversation or a new conversation initiated
10 by the sender.

One embodiment provides a system wherein, when an outbound SMS message is initiated, the outbound SMS message is placed into a conversation data store thereby to monitor conversations to which subsequent inbound SMS messages are matched upon receipt.

15 One embodiment provides a system wherein the plurality of communications channels include web-based chat channels.

One embodiment provides a system wherein the plurality of communications channels include channels for interfacing with social media.

One embodiment provides a system wherein each node performs a node
20 heartbeat process by which the nodes each communicate with the managing core on a regular basis thereby to allow the managing core to manage the resources of the node.

One embodiment provides a system wherein, via the heartbeat process, each node repeatedly sends a request to the managing core to determine if there are any tasks for that node to perform.

25 One embodiment provides a system wherein, if there is a task to be performed, the managing core responds with that task, and the node notifies the managing core upon completion of that task.

One embodiment provides a system wherein each node performs its own self maintenance by requesting any latest changes from the ICC core and, if there is an upgrade
30 to be performed, the node downloads and installs the required files from the managing core.

One embodiment provides a system wherein the media driver configuration for

- 4 -

each communications channel is maintained by an automated process within each node, wherein the managing core publishes driver configurations and applications for release and each node checks the latest versions against its local settings and, if the node requires an update for a given channel, the node transfers the update from the managing core and
5 re-initialises the channel.

According to a second aspect of the present disclosure, there is provided a node for use in a system for managing communication data, the node including:

a channel driver module that provides access to a plurality of communications channels including at least an incoming call channel and outgoing call channel;

10 a communications gateway component for managing communications between the node and a managing core configured for centrally managing a plurality of distributed nodes; and

a node application module that provides logic to drive node functionalities.

According to a third aspect of the present disclosure, there is provided a
15 method for operating a node for use in a system for managing communication data, the node including:

executing a channel driver module that provides access to a plurality of communications channels including at least an incoming call channel and outgoing call channel;

20 executing a communications gateway component for managing communications between the node and a managing core configured for centrally managing a plurality of distributed nodes; and

executing a node application module that provides logic to drive node functionalities.

25 According to a fourth aspect of the present disclosure, there is provided a computer program product for performing a method as described herein.

According to a fifth aspect of the present disclosure, there is provided a non-transitive carrier medium for carrying computer executable code that, when executed on a processor, causes the processor to perform a method as described herein.

30 One embodiment provides a system configured for performing a method as described herein.

- 5 -

According to a sixth aspect of the present disclosure, there is provided a contact centre system comprising: a plurality of nodes, each node having a plurality of call channels, the call channels being connectable to at least one communications network and including at least one incoming channel and at least an outgoing channel; and a managing
5 core coupled to each of the nodes, the core comprising a data store with at least one call architecture of a merchant defining a call handling process for the merchant, the call handling process being operable to detect receipt by a first said node of an incoming call on a corresponding incoming channel, and to couple the call from said first node via one of said call channels thereof to a second said node for connection to an agent associated with
10 the merchant.

According to a seventh aspect of the present disclosure, there is provided a distributed communication management system comprising: a plurality of nodes, each node having a plurality of call channels[^] the call channels being connectable to at least one communications network and including at least one incoming voice channel and at least an
15 outgoing voice channel; and a managing core coupled to each of the nodes, the core comprising a data store with at least one call architecture of a merchant defining a call handling process for the merchant, the call handling process being operable to detect receipt by a first said node of an incoming voice call on a corresponding incoming voice channel, and to couple the voice call from said first node via one of said call channels
20 thereof to a second the node for connection to an agent associated with the merchant.

According to an eighth aspect of the present disclosure, there is provided a call conversation mapping method, comprising the steps of: creating a call conversation map that includes a plurality of objects and at least one connecting link to define a call handling process, wherein the objects include at least one task relating to receiving an incoming
25 communication or generating an outgoing communication; activating the call conversation map in response to a trigger; and processing communications in accordance with the call handling process of the call conversation map.

According to a ninth aspect of the present disclosure, there is provided a call conversation mapping system comprising: a conversation map for creating a call
30 conversation map that includes a plurality of objects and at least one connecting link to define a call handling process, wherein the objects include at least one task relating to receiving an incoming communication or generating an outgoing communication; a data

- 6 -

store for storing a call conversation map; a conversation triggers module defining a trigger for activating the stored call conversation map; and a conversation engine for activating the stored call conversation map in response to the trigger and processing communications in accordance with the call handling process of the stored call conversation map.

5 According to another aspect of the present disclosure, there is provided an apparatus for implementing any one of the aforementioned methods.

 According to another aspect of the present disclosure, there is provided a computer program product including a computer readable medium having recorded thereon a computer program for implementing any one of the methods described above.

10 Other aspects of the invention are also disclosed.

 Reference throughout this specification to "one embodiment", "some embodiments", or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases "in one
15 embodiment", "in some embodiments", or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment, but may. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner, as would be apparent to one of ordinary skill in the art from this disclosure, in one or more embodiments.

20 As used herein, unless otherwise specified, the use of the ordinal adjectives "first", "second", "third", etc., to describe a common object, merely indicate that different instances of like objects are being referred to, and are not intended to imply that the objects so described must be in a given sequence, either temporally, spatially, in ranking, or in any other manner.

25 In the claims below and the description herein, any one of the terms "comprising", "comprised of, or "which comprises" is an open term that means including at least the elements/features that follow, but not excluding others. Thus, the term comprising, when used in the claims, should not be interpreted as being limitative to the means or elements or steps listed thereafter. For example, the scope of the expression
30 "a device comprising A and B" should not be limited to devices consisting only of elements A and B. Any one of the terms "including", or "which includes", or "that includes" as used herein is also an open term that also means including at least the

- 7 -

elements/features that follow the term, but not excluding others. Thus, "including" is synonymous with and means "comprising".

BRIEF DESCRIPTION OF THE DRAWINGS

At least one embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

FIG. 1 is a schematic block diagram representation that illustrates a logical software architecture.

FIG. 2 is a schematic block diagram representation that illustrates a load balancing arrangement for incoming calls.

FIG. 3A is a schematic block diagram representation that illustrates an arrangement for managing incoming SMS messages.

FIG. 3B is a schematic block diagram representation that illustrates an arrangement for managing outgoing SMS messages.

FIG. 4 is a schematic block diagram representation that illustrates an arrangement.

FIG. 5 is a schematic block diagram representation that illustrates a live data arrangement.

FIG. 6 is a schematic block diagram representation that illustrates a physical architecture for a distributed system.

FIG. 7 is a schematic block diagram representation that illustrates load balancing for a distributed system.

FIG. 8 is a schematic block diagram representation that illustrates a data clustering arrangement.

FIG. 9 is a schematic block diagram representation that illustrates a node architecture.

FIGs 10A and 10B collectively form a schematic block diagram representation of a computing system upon which described arrangements can be practised.

FIG. 11 is a schematic block diagram representation illustrating a contact centre system in accordance with an embodiment of the present disclosure.

FIG. 12 is a schematic block diagram representation illustrating connection of a call from a customer to an agent using the contact centre system of FIG. 11.

- 8 -

FIG. 13 is a flow diagram illustrating functionality of a node during the call connection of FIG. 12.

FIG. 14 is a flow diagram illustrating functionality of the contact centre system during the call connection of FIG. 12.

5 FIG. 15 is a schematic block diagram representation illustrating functionality of the ICC Core 101 of FIG. 1.

FIG. 16 is a schematic block diagram representation illustrating a communication management system, embodied using the contact centre system of FIG. 11.

10 FIG. 17 is a flow diagram illustrating a method of utilising the communication management system of FIG. 16 to establish a voice call from a customer to a merchant agent.

FIG. 18 is a flow diagram illustrating a method of utilising the communication management system of FIG. 16 to establish a web-based chat session between a customer and a merchant agent.

15 FIG. 19 is a flow diagram illustrating a method of utilising the communication management system of FIG. 16 to establish an SMS message conversation between a customer and a merchant agent.

FIG. 20 is a screenshot of a call conversation map.

20 FIG. 21 is a schematic block diagram representation illustrating a logical architecture of a call mapping system.

FIG. 22 is a schematic block diagram representation illustrating an example of a contact library.

FIG. 23 is a flow diagram illustrating a conversation mapping method.

25 FIG. 24 is a flow diagram illustrating a conversation mapping method 2400 utilising the example of FIG. 20.

FIG. 25 is a schematic block diagram representation illustrating activation of tasks from the conversation map of FIG. 20.

DETAILED DESCRIPTION

Where reference is made in any one or more of the accompanying drawings to steps and/or features that have the same reference numerals, those steps and/or features have for the purposes of this description the same function(s) or operation(s), unless the
5 contrary intention appears.

Described herein are systems and methods for managing communication data and mapping call conversations. In overview, the present embodiments are directed to a cloud-based distributed architecture for managing incoming and outgoing communications. These communications occur over a range of channels, including voice-based channels
10 (telephonic, for example), and messaging-based channels (such as email, SMS, multimedia messaging service (MMS), web-based chat, and so on). The purposes of communications vary between embodiments, and include the likes of providing customer support, such as through a contact centre system, and delivering marketing materials. A contact centre system manages communications between an end user and a customer service
15 representative, such as an agent of a merchant, and may handle voice calls, web requests, SMS messages, and other communication formats. A call is an exchange of at least one communication between an end user and the contact centre system and may utilise one or more communication media, including, but not limited to, voice, SMS messages, emails, and web chat sessions. The end user may be, for example, a customer, prospective
20 customer, a merchant manager, or an agent of the merchant. In one arrangement, a contact centre is a call centre for handling voice calls.

One aspect of the present disclosure provides a communication management system in the form of a contact centre system for managing communication data. The contact centre system includes a plurality of nodes and a managing core coupled to each of
25 the nodes. Each node has a plurality of call channels, including at least one incoming channel and at least one outgoing channel. Each of the call channels is connectable to at least one communications network, which may include, for example, a wired or wireless communications link, or combinations thereof. The managing core includes a data store with at least one call architecture of a merchant defining a call handling process for the
30 merchant, the call handling process being operable to detect receipt by a first node, of the plurality of nodes, of an incoming call on a corresponding incoming channel, and to couple the call from the first node via one of said call channels thereof to a second node of the

- 10 -

plurality of nodes for connection to an agent associated with the merchant.

In one arrangement, the at least one incoming channel is a voice channel, the at least one outgoing channel is a voice channel, and the incoming call received by the first node is a voice call received on a corresponding incoming voice channel.

5 In one arrangement, the agent associated with the merchant advises the contact centre system of the availability of the agent through registration and login processes. The agent may communicate with the contact centre system using multiple communication media, including a voice connection and a web connection.

One aspect of the present disclosure provides call conversation mapping
10 method. The method creates a call conversation map that includes a plurality of objects and at least one connecting link to define a call handling process. The objects include at least one task relating to receiving an incoming communication or generating an outgoing communication. The method activates the call conversation map in response to a trigger and processes communications in accordance with the call handling process of the call
15 conversation map.

Another aspect of the present disclosure provides a call conversation mapping system. The system includes a conversation map application, a data store for storing a call conversation map, a conversation triggers module defining a trigger for activating the stored call conversation map, and a conversation engine. The conversation map
20 application is for creating a call conversation map that includes a plurality of objects and at least one connecting link to define a call handling process, wherein the objects include at least one task relating to receiving an incoming communication or generating an outgoing communication. The conversation engine activates the stored call conversation map in response to the trigger and processing communications in accordance with the call
25 handling process of the stored call conversation map.

An exemplary contact centre system includes a managing core, also referred to throughout this description as an intelligent contact cloud (ICC) core. This ICC core is configured for centrally managing a plurality of distributed nodes. In some embodiments, the ICC core is implemented as a distributed component, defined by a plurality of
30 synchronised sub-cores. The system includes a plurality of distributed nodes, these nodes providing a distributed scalable resource for handling communications. Each node is configured to execute a channel driver module that provides access to a plurality of

- 11 -

communications channels. In the present example, these communication channels include at least one incoming call channel and one outgoing call channel. Each node is additionally configured to execute a communications gateway component for managing communications between the node and the ICC core, and a node application module that provides logic to drive general node functionalities. Collectively, the ICC core and nodes provide a distributed software application.

FIG. 11 is a schematic block diagram representation of a distributed communication management system in the form of a contact centre system 1100, in accordance with an embodiment of the present disclosure. The contact centre system 1100 includes a firewall 1102 for regulating communications between the contact centre system 1100 and the external world, wherein the firewall 1102 is capable of handling various types of communications messages. The types of communications messages may include, for example, telephony, Internet Protocol (IP) messages, and Short Message Service (SMS) messages. Internet Protocol (IP) and Session Initiation Protocol (SIP) communications to and from the contact centre system 1100 pass through the firewall 1102. The firewall 1102 couples via at least one communication link 1103 to a communication network (not illustrated) for exchanging communication messages between the contact centre system 1100 and externally located computing devices and telephone handsets. The firewall 1102 directs incoming communications messages to one of a Session Initiation Protocol (SIP) Gateway 1104 and a Web Gateway 1116, based on the type of the incoming communications messages. In particular, requests for voice and video calls over Internet Protocol (IP) are directed to the SIP Gateway 1104 and requests for data exchanges, such as web-based requests, are directed to the Web Gateway 1116.

The SIP Gateway 1104 is coupled to a SIP Load Balancer 1106. The SIP Load Balancer 1106 distributes incoming communications messages received by the SIP Gateway 1102 among a plurality of nodes 1108a, ..., 1108n. As indicated above, each of the plurality of nodes is equipped with a plurality of call channels, including at least one incoming channel and at least one outgoing channel. In one arrangement, the call channels include at least one incoming voice channel and the at least one outgoing voice channel. In one embodiment, the SIP Load Balancer 1106 distributes new incoming SIP requests across the plurality of nodes 1108a, ..., 1108n utilising a round robin method.

The Web Gateway 1116 is coupled to a Web Load Balancer 1118. The Web

- 12 -

Load Balancer 1118 distributes incoming communications messages received by the Web Gateway 1116 among a plurality of nodes 1120a, ..., 1120n. Each of the nodes 1120a, ..., 1120n is equipped with at least one channel for handling Internet Protocol (IP) communication messages. In one embodiment, the Web Load Balancer 1118 distributes
5 new incoming web requests across the plurality of nodes 1120a, ..., 1120n utilising a round robin method.

Voice communications between the outside world and the plurality of nodes 1108a, ..., 1108n and 1120a, ..., 1120n may be exchanged via a Public Switched Telephony Network (PSTN) 1180 or a mobile telephony network 1186. Voice over
10 Internet Protocol (VOIP) communications may also be exchanged through the firewall 1102.

Each of the nodes 1108a, ..., 1108n and 1120a, ..., 1120n is coupled via an optional internal firewall 1122 to a managing core 1130. The internal firewall 1122 regulates communication between each of the nodes 1108a, ..., 1108n and
15 1120a, ..., 1120n and the managing core 1130. The managing core 1130 receives inbound service requests 1126 from the plurality of nodes 1108a, ..., 1108n and 1120a, ..., 1120n and sends outbound instructions 1128 to the plurality of nodes 1108a, ..., 1108n and 1120a, ..., 1120n. In one implementation, the inbound service requests and outbound instructions between the nodes 1108, 1120 and the managing core 1130 utilise the
20 extensible Markup Language (XML). It will be appreciated, however, that other metalanguages and coding formats may equally be used. In the example of FIG. 11, the managing core 1130 is implemented using a plurality of sub-cores 1132a, 1132b. In the example of FIG. 11, the internal firewall 1122 is coupled to an optional core load balancer 1124, which distributes communications received from the nodes 1108a, ..., 1108n and
25 1120a, ..., 1120n among the plurality of sub-cores 1132a, 1132b. In implementations that utilise a single managing core 1130, rather than a plurality of sub-cores 1132a, 1132b, it is not necessary to include the core load balancer 1124.

Each sub-core 1132a, 1132b is coupled to a data store 1170 for storing at least one call architecture 1175 of a merchant, wherein the call architecture 1175 defines a call
30 handling process for that merchant. The data store 1170 may be integral with the managing core 1130 or may alternatively be coupled to the managing core 1130. The data store 1170 may be implemented as a single entity or may be a distributed entity, such as an

- 13 -

array of computer disk drives. Each call handling process defined in a call architecture 1175 of a merchant is operable to detect receipt by a first node in the contact centre system 1100 of an incoming call on a corresponding incoming channel, and to handle the call in a predefined manner. In one embodiment, the call handling process couples the call from the
5 first node via one of the call channels thereof to a second of the plurality of nodes in the contact centre system 1100 for connection to an agent associated with the merchant.

Node 1108a is an exemplary node and includes a plurality of communication channels 1109a, a channel driver 1110a, a node application 1112a, and a communications gateway 1114a. The functionality of the node architecture is described in further detail
10 below with reference to FIG. 9. Each of the nodes 1108n and 1120a, ..., 1120n includes a corresponding plurality of communication channels, a channel driver, a node application, and a communications gateway, which for the sake of clarity are not illustrated.

Each of the nodes 1108a, ..., 1108n and 1120a, ..., 1120n and each of the sub-cores 1132a, 1132b may be implemented using a computing system 1000 of the type
15 described below with reference to FIGs 10A and 10B.

The contact centre system 1100 of FIG. 11 is arranged to establish call connections between customers and agents that represent one or more merchants. The contact centre 1100 is able to receive input requests from a number of different sources and connect those requests to a plurality of agents. The agents may be physically located
20 together in a structured contact centre arrangement or may be disparately located, such as contact centre agents working from home or contact centre agents located across multiple time zones so as to provide 24 hour customer service.

The contact centre system 1100 includes call architectures 1175 that define call handling processes for merchants utilising the contact centre system 1100. Merchants
25 define call handling processes that establish how incoming requests are to be handled. Incoming requests may generate responsive emails or SMS messages, connection of the incoming request to a recorded voice announcement or interactive voice response system, or connection of the incoming request to a computing device or telephone attended by an agent of the merchant. Thus, a call handling process may utilise a plurality of different
30 communication types, including, for example, but not limited to, voice, SMS, email, web chat, recorded announcements, and interactive voice responses.

In one embodiment, a merchant defines a call architecture 1175 for a dial-out

- 14 -

campaign, wherein the call architecture 1175 includes a list of contacts, the material that is to be sent to each contact, and timing relating to the transmission of outbound messages. The timing may include a start time, a stop time, a call rate, or any combination thereof. For example, a dial-out campaign may be defined to send an outgoing SMS message with
5 the content "Send reply SMS for a chance to win" to each contact in a defined set of 10,000 contacts, wherein the campaign runs from 11:00am on Monday to 5:00pm on Friday with a call rate of 100 SMS messages per hour. The outbound messages may include, for example, SMS messages, emails, video or voice calls. Merchants are able to modify associated call architectures 1175 to provide personalised customer service.

10 Merchants can provide different services to valued customers or potential customers, based on existing information known about those customers. The call architecture 1175 of a merchant can be defined to provide offers or services in response to a keyword received in an incoming SMS message or web request or in response to an incoming telephone request received on a predefined number associated with the merchant.

15 FIG. 16 is a schematic representation of a communication management system 1600, embodied using an instance of the ICC system 1100. The communication system 1600 includes a merchant agent 1610, Customer A 1620, Customer B 1630, Customer C 1640, and Merchant Manager 1650, each of which is able to communicate with the ICC system 1100 via a telecommunications network 1660. The telecommunications network
20 1660 may be implemented using a public switched telephone network (PSTN), a mobile telephony network, a wireless access network, a local access network, the Internet, or any combination thereof.

The merchant manager 1650 utilises a personal computer 1652 to access the ICC system 1100 via the telecommunications network 1660 and configure a call
25 architecture 1175 associated with that merchant. The merchant manager 1650 is thus able to change parameters of the call architecture 1175 to modify an existing call handling process or configure a new call handling process, such as may arise in conjunction with targeted marketing activities.

The merchant agent 1610 utilises a personal computer 1612, a telephone 1614,
30 or a combination thereof, to access the ICC system 1100 via the telecommunications network 1660. The merchant agent 1610 is employed by the merchant or an operator of the communication management system 1600 to handle enquiries received from customers

- 15 -

and potential customers relating to services offered or provided by the merchant. The merchant agent 1610 registers an availability for handling enquiries by logging into the ICC system 1100 through the use of the personal computer 1612 or the telephone 1614. In one implementation, the merchant agent 1610 remains coupled to the ICC System 1100 via the telecommunications network 1662 and waits for enquiries to arrive. In an alternative implementation, the merchant agent 1610 registers an availability with the ICC system 1100 and waits for the ICC system 1100 to subsequently connect calls to the merchant agent 1610.

Customer A 1620 utilises a telephone 1622, such as a conventional telephone handset, a mobile telephone handset, or a smartphone, to access the ICC system 1100 via the telecommunications network 1660. Customer A 1620 may generate an incoming call to the ICC System in relation to an account, a service enquiry, in response to a promotion, or in response to a received telephone call or message received from a merchant. The ICC System 1100 identifies a merchant from the incoming call, retrieves a call architecture 1175 of the identified merchant, and handles the incoming call in accordance with the retrieved call architecture 1175.

Customer B 1630 utilises a personal computer (PC) 1632 to access the ICC system 1100 via the telecommunications network 1660. In one implementation, a node of the ICC System 1100 pushes web content to the personal computer 1632 via the telecommunications network 1660. Depending on the implementation, Customer B is able to provide information relating to the enquiry, which may include a customer identifier or an account number. Further, Customer B 1630 is able to complete data fields in any pro-forma data sheets that are hosted on the website. Customer B can send emails or data requests from the personal computer 1632. The ICC System 1100 identifies a merchant from the incoming web request or email, retrieves a call architecture 1175 of the identified merchant and handles the incoming call in accordance with the call architecture 1175.

Customer C 1640 utilises an SMS engine 1642 to interact with the ICC system 1100 via the telecommunications network 1660. The SMS engine 1642 may be implemented using a mobile telephone handset, a computing device, or a smartphone. Customer C sends an SMS message from the SMS engine 1642 to the ICC System 1100 via the telecommunications network 1660. The ICC System 1100 identifies a merchant

- 16 -

from the incoming SMS message, retrieves a call architecture 1175 of the identified merchant and handles the incoming call in accordance with the call architecture 1175.

In a first example, the merchant manager 1650 is running a promotional campaign. The merchant manager 1650 utilises the personal computer 1652 to access the ICC system 1100 via the telecommunications network 1660 and configure a call architecture 1175 of the merchant manager 1650. The merchant manager defines a call handling process in the call architecture 1175 whereby calls received on a telephone number associated with the promotional campaign are connected to the merchant agent 1610. In this example, the telephone number associated with the promotional campaign is a smart-number in the form of "1800 CAMPAIGN".

Customer A 1620 becomes aware of promotional material relating to the promotional campaign being run by merchant manager 1650 and wants to find out further information about the offer. FIG. 17 is a flow diagram illustrating a method 1700 for establishing a telephone call from Customer A 1620 to the merchant agent 1610. The method 1700 begins at a start step 1705 and proceeds to step 1710, in which Customer A 1620 utilises the telephone 1622 to dial a number associated with the merchant manager 1650. In this example, Customer A dials "1800 CAMPAIGN" and the telecommunications network 1660 routes the call from the telephone 1622 to the ICC System 1100. In step 1720, the ICC system 1100 receives the inbound call from Customer A 1620 via the PSTN 1180 and the call is connected to a voice communication channel on a receiving node 1108, 1120. The receiving node 1108, 1120 sends an inbound service request via the internal firewall 1122 to the managing core 1130. The managing core 1130 utilises a service system gateway to validate that the number is valid and identifies that the inbound service request relates to merchant manager 1650, based on the dialed number. Control passes from step 1720 to step 1730, in which the managing core 1130 of the ICC system 1100 interrogates the data store 1170 within the ICC system 1100 to retrieve a call architecture 1175 associated with the merchant manager 1650. Control passes from step 1730 to step 1740, in which the ICC system 1100 handles the call in accordance with the retrieved call architecture 1175 associated with the identified merchant. In this example, the call architecture 1175 of merchant manager 1650 defines that the call is to be routed to merchant agent 1610, who has previously registered an availability for handling enquiries relating to merchant manager 1650. The managing core 1130 sends an outbound

- 17 -

instruction to the receiving node 1108, 1120 to connect the call to the telephone 1614 of the merchant agent 1610. The receiving node 1108, 1120 connects the call to the second node 1108, 1120, which receives the incoming call from the receiving node 1108, 1120 and sends an inbound service request to the managing core 1130 seeking instructions for handling the call. The managing core 1130 sends an outbound instruction, based on the retrieved call architecture, to the second node 1108, 1120 to connect the incoming call via the telecommunications network 1660 to the telephone 1614 of the merchant agent 1610. Control passes to step 1750 in which Customer A is connected to the Merchant Agent 1610. Control passes from step 1750 to an End step 1760 and the process 1700 terminates.

10 In a second example, Customer B 1630 becomes aware of the promotional campaign being run by merchant manager 1650 and Customer B 1630 utilises the PC 1632 to send an inbound web request to the ICC system 1100 via the telecommunications network 1660. FIG. 18 is a flow diagram illustrating a method 1800 for establishing a chat session between Customer B 1630 and the merchant agent 1610. The method 1800 begins
15 at a Start step 1810 and proceeds to step 1810, in which Customer B 1630 utilises the personal computer 1632 to send an incoming web request via the telecommunications network 1660 to the ICC system 1100. In step 1820, the ICC system 1100 receives the incoming web request from Customer B 1630 via the firewall 1102. The incoming web request passes to the Web Gateway 1116 and then to the Web Load Balancer 1118. The
20 Web Load Balancer 1118 utilises a round robin algorithm or other algorithm to allocate the incoming web request to a receiving nodes 1108, 1120 that is equipped with a web communication channel. The receiving node 1108, 1120 sends an inbound service request to the managing core 1130. The managing core 1130 utilises the service system gateway to validate that the request is valid and identifies that the inbound service request received
25 from Customer B 1630 relates to the merchant manager 1650, based on an incoming internet protocol (IP) address, a hyperlink, or other metadata associated with the inbound service request from Customer B 1630. Control passes from step 1820 to step 1830, in which the ICC system 1100 utilises the managing core 1130 to retrieve from the data store 1170 a call architecture 1175 associated with the merchant manager 1650 and determine
30 how to process the inbound service query request. Control passes to step 1840, in which the ICC system 1100 sends an outbound instruction to the receiving node 1108, 1120 to establish a connection to the Merchant Agent 1610, in accordance with the retrieved call

- 18 -

architecture 1175. In this case, the ICC system 1100 transmits the request received from Customer B 1630 via the telecommunications network 1660 to the personal computer 1612 of the merchant agent 1610. In step 1850, Customer B 1630 is connected to Merchant Agent 1610 and the merchant agent 1610 is then able to establish a chat session with
5 Customer B 1630. Alternatively, the ICC System 1100 may push web content via the telecommunications network 1660 to the personal computer 1632. Control passes from step 1850 to an End step 1860 and the method 1800 terminates.

In a third example, Customer C 1640 utilises an SMS engine 1642 to transmit an SMS message via the telecommunications network 1660 to the ICC system 1100. In
10 this example, the SMS message includes a keyword identified in the promotional material associated with merchant manager 1650. Fig. 19 is a flow diagram illustrating a method 1900 for establishing an SMS conversation between Customer C 1640 and the ICC system 1100. The method 1900 begins at a Start step 1900 and proceeds to step 1910, in which Customer C 1640 utilises the SMS engine 1642 to generate an incoming SMS message to
15 the ICC system 1100. In step 1920, the ICC system 1100 receives the inbound service SMS message from Customer C 1640 via the firewall 1103 and the SIP Gateway 1104 and SIP Load Balancer 1106. The SIP Load Balancer utilises a round robin algorithm or other algorithm to allocate the inbound SMS message to a receiving node 1108, 1120 equipped with an SMS communication channel. The receiving node 1108, 1120 sends an incoming
20 service request to the managing core 1130, which determines that merchant manager 1650 is associated with the inbound SMS text message. Control passes from step 1920 to step 1930, in which the ICC system 1100 utilises the managing core 1130 to retrieve from the data store 1170 a call architecture 1175 associated with the merchant manager 1650. Control passes from step 1930 to step 1940, in which the ICC system 1100 handles the call
25 in accordance with the retrieved call architecture 1175 associated with the identified merchant. In this example, the merchant manger 1650 has configured the call architecture 1175 of the merchant manger 1650 to transmit a predefined SMS message in response to any received SMS messages that contain the keyword associated with the promotional campaign. Accordingly, in step 1940 the ICC system 1100 transmits, via the
30 telecommunications network 1660, an SMS message to Customer C 1640, by sending an outbound instruction from the managing core 1130 to the receiving node 1108, 1120 with

- 19 -

the content of the SMS message. Control passes from step 1940 to an End step 1950 and the method 1900 terminates.

Whilst FIG. 16 refers to the use of personal computers 1612, 1632, and 1652, a person skilled in the art will understand that equivalent functionality may be delivered by other computing devices, without departing from the spirit and scope of the present disclosure. Such computing devices may include, for example, but are not limited to, a data-capable mobile telephone handset, a smartphone, or a personal digital assistant.

COMPUTER IMPLEMENTATION

FIGs 10A and 10B depict a general-purpose computer system 1000, upon which the various arrangements described can be practised. In particular, embodiments of the communication management systems 1100, 1600 and call conversation mapping system utilise the general purpose computer system 1000 to implement one or more of the functional components, including the nodes 1108, 1120, the managing core 1130, the gateways 1104, 1116, the load balancers 1106, 1118, 1124, the personal computers 1612, 1632, 1652, the telephones 1614, 1622, and the SMS engine 1642.

As seen in FIG. 10A, the computer system 1000 includes: a computer module 1001; input devices such as a keyboard 1002, a mouse pointer device 1003, a scanner 1026, a camera 1027, and a microphone 1080; and output devices including a printer 1015, a display device 1014 and loudspeakers 1017. An external Modulator-Demodulator (Modem) transceiver device 1016 may be used by the computer module 1001 for communicating to and from a communications network 1020 via a connection 1021. The communications network corresponds to the telecommunications network 1660 of FIG. 16 and may be a Public Switched Telephone Network (PSTN), a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the connection 1021 is a telephone line, the modem 1016 may be a traditional "dial-up" modem. Alternatively, where the connection 1021 is a high capacity (e.g., cable) connection, the modem 1016 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 1020.

The computer module 1001 typically includes at least one processor unit 1005, and a memory unit 1006. For example, the memory unit 1006 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 1001 also includes an number of input/output (I/O) interfaces including:

- 20 -

an audio-video interface 1007 that couples to the video display 1014, loudspeakers 1017 and microphone 1080; an I/O interface 1013 that couples to the keyboard 1002, mouse 1003, scanner 1026, camera 1027 and optionally a joystick or other human interface device (not illustrated); and an interface 1008 for the external modem 1016 and printer 1015. In one example, the personal computer 1630 is implemented using the computing device 1000 and Customer B is able to utilise the printer 1015 to print a record of a receipt or voucher issued by the ICC System 1100. The loudspeakers 1017 and microphone 1080 may be utilised in conjunction with instructions executing on the processor unit 1005 to effect a telephone handset for VOIP communications. Such a VOIP telephone handset may be utilised to implement the telephones 1614 and 1622 of FIG. 16. The camera 1027 may be utilised to enhance a chat session established between a customer 1630 and a merchant agent 1610 by facilitating the connection of a video call, in conjunction with software instructions executing on the processor 1005. In one arrangement, the personal computer 1652 is implemented using the computing device 1000 and software instructions execute on the processor 1005 to provide the merchant manager 1650 with a user interface displayed on the display device 1014. The Merchant Manager 1650 utilises one or more input devices, such as the keyboard 1002 and mouse 1003 to send commands to the ICC System 1100 to configure the call architecture 1175 associated with the Merchant Manager 1650, including the creation and modification of a call conversation map. In another arrangement, the personal computer 1632 is implemented using the computing device 1000 and web content is displayed on the display device 1014. Customer C can utilise any one or more of the keyboard 1002, mouse 1003, microphone 1080, and camera 1027 to establish and maintain a web call utilising the ICC System 1100, which may include a chat session with the Merchant Agent 1610. In some implementations, the modem 1016 may be incorporated within the computer module 1001, for example within the interface 1008. The computer module 1001 also has a local network interface 1011, which permits coupling of the computer system 1000 via a connection 1023 to a local-area communications network 1022, known as a Local Area Network (LAN). As illustrated in FIG. 10A, the local communications network 1022 may also couple to the wide network 1020 via a connection 1024, which would typically include a so-called "firewall" device or device of similar functionality. The local network interface 1011 may comprise an EthernetTM circuit card, a BluetoothTM wireless

- 21 -

arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practised for the interface 1011.

The I/O interfaces 1008 and 1013 may afford either or both of serial and parallel connectivity, the former typically being implemented according to the Universal
5 Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 1009 are provided and typically include a hard disk drive (HDD) 1010. Such storage devices 1009 may be utilised to implement the data store 1170 of FIG. 11, which is used to store call architectures 1175 associated with merchants. Such call
architectures 1175 can include or be coupled to a stored list of contacts and other customer
10 information stored on the same or other storage devices 1009. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 1012 is typically provided to act as a non-volatile source of data. Portable memory devices, such optical disks (e.g., CD-ROM, DVD, Blu-ray Disc™), USB-RAM, portable, external hard drives, and floppy disks, for example, may be used as
15 appropriate sources of data to the system 1000.

The components 1005 to 1013 of the computer module 1001 typically communicate via an interconnected bus 1004 and in a manner that results in a conventional mode of operation of the computer system 1000 known to those in the relevant art. For example, the processor 1005 is coupled to the system bus 1004 using a connection 1018.
20 Likewise, the memory 1006 and optical disk drive 1012 are coupled to the system bus 1004 by connections 1019. Examples of computers on which the described arrangements can be practised include IBM-PCs and compatibles, Sun Sparcstations, Apple Mac™, or alike computer systems.

The communication management system and call conversation mapping
25 system may be implemented using the computer system 1000, wherein the processes of FIGS 1 to 9 and 11 to 25, described herein, may be implemented as one or more software application programs 1033 executable within the computer system 1000. In particular, the steps of the method of managing data communications are effected by instructions 1031 (see FIG. 10B) in the software 1033 that are carried out within the computer system 1000.
30 The method of managing data communications includes the functionality performed by the managing core 1130, including the sub-functions of an Inbound Service Manager, an Outbound Job Manager, a Service Security Gateway, a Job Scheduler 140, a Maintenance

- 22 -

application 180, a Management Application 150, a Conversation Map application, a contact library, a conversation triggers module, a conversation management application programming interface, and a conversation engine. The method of managing data communications further includes the functionality performed by the nodes, including the channel driver 127, the node application 126, and the ICC Communication Gateway 102. The software instructions 1031 may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part and the corresponding code modules performs the communication management methods and a second part and the corresponding code modules manage a user interface between the first part and the user.

The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system 1000 from the computer readable medium, and then executed by the computer system 1000. A computer readable medium having such software or computer program recorded on the computer readable medium is a computer program product. The use of the computer program product in the computer system 1000 preferably effects an advantageous apparatus for communication management.

The software 1033 is typically stored in the HDD 1010 or the memory 1006. The software is loaded into the computer system 1000 from a computer readable medium, and executed by the computer system 1000. Thus, for example, the software 1033 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 1025 that is read by the optical disk drive 1012. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer system 1000 preferably effects an apparatus for communication management.

In some instances, the application programs 1033 may be supplied to the user encoded on one or more CD-ROMs 1025 and read via the corresponding drive 1012, or alternatively may be read by the user from the networks 1020 or 1022. Still further, the software can also be loaded into the computer system 1000 from other computer readable media. In particular, the nodes 1108, 1120 are configured to load application programs from other computer readable media coupled to the managing core 101. In this way, the distributed nodes are maintained with the same software. Computer readable storage

- 23 -

media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 1000 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, CD-ROM, DVD, Blu-ray Disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 1001. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computer module 1001 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

The second part of the application programs 1033 and the corresponding code modules mentioned above may be executed to implement one or more graphical user interfaces (GUIs) to be rendered or otherwise represented upon the display 1014. Through manipulation of typically the keyboard 1002 and the mouse 1003, a user of the computer system 1000 and the application may manipulate the interface in a functionally adaptable manner to provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user interfaces may also be implemented, such as an audio interface utilising speech prompts output via the loudspeakers 1017 and user voice commands input via the microphone 1080.

FIG. 10B is a detailed schematic block diagram of the processor 1005 and a "memory" 1034. The memory 1034 represents a logical aggregation of all the memory modules (including the HDD 1009 and semiconductor memory 1006) that can be accessed by the computer module 1001 in FIG. 10A.

When the computer module 1001 is initially powered up, a power-on self-test (POST) program 1050 executes. The POST program 1050 is typically stored in a ROM 1049 of the semiconductor memory 1006 of FIG. 10A. A hardware device such as the ROM 1049 storing software is sometimes referred to as firmware. The POST program 1050 examines hardware within the computer module 1001 to ensure proper functioning and typically checks the processor 1005, the memory 1034 (1009, 1006), and a basic input-output systems software (BIOS) module 1051, also typically stored in the ROM 1049, for correct operation. Once the POST program 1050 has run successfully, the

- 24 -

BIOS 1051 activates the hard disk drive 1010 of FIG. 10A. Activation of the hard disk drive 1010 causes a bootstrap loader program 1052 that is resident on the hard disk drive 1010 to execute via the processor 1005. This loads an operating system 1053 into the RAM memory 1006, upon which the operating system 1053 commences operation. The
5 operating system 1053 is a system level application, executable by the processor 1005, to fulfil various high level functions, including processor management, memory management, device management, storage management, software application interface, and generic user interface.

The operating system 1053 manages the memory 1034 (1009, 1006) to ensure
10 that each process or application running on the computer module 1001 has sufficient memory in which to execute without colliding with memory allocated to another process. Furthermore, the different types of memory available in the system 1000 of FIG. 10A must be used properly so that each process can run effectively. Accordingly, the aggregated memory 1034 is not intended to illustrate how particular segments of memory are allocated
15 (unless otherwise stated), but rather to provide a general view of the memory accessible by the computer system 1000 and how such is used.

As shown in FIG. 10B, the processor 1005 includes a number of functional modules including a control unit 1039, an arithmetic logic unit (ALU) 1040, and a local or internal memory 1048, sometimes called a cache memory. The cache memory 1048
20 typically include a number of storage registers 1044 - 1046 in a register section. One or more internal busses 1041 functionally interconnect these functional modules. The processor 1005 typically also has one or more interfaces 1042 for communicating with external devices via the system bus 1004, using a connection 1018. The memory 1034 is coupled to the bus 1004 using a connection 1019.

25 The application program 1033 includes a sequence of instructions 1031 that may include conditional branch and loop instructions. The program 1033 may also include data 1032 which is used in execution of the program 1033. The instructions 1031 and the data 1032 are stored in memory locations 1028, 1029, 1030 and 1035, 1036, 1037, respectively. Depending upon the relative size of the instructions 1031 and the memory
30 locations 1028-1030, a particular instruction may be stored in a single memory location as depicted by the instruction shown in the memory location 1030. Alternatively, an instruction may be segmented into a number of parts, each of which is stored in a separate

- 25 -

memory location, as depicted by the instruction segments shown in the memory locations 1028 and 1029.

In general, the processor 1005 is given a set of instructions which are executed therein. The processor 1005 waits for a subsequent input, to which the processor 1005
5 reacts to by executing another set of instructions. Each input may be provided from one or more of a number of sources, including data generated by one or more of the input devices 1002, 1003, data received from an external source across one of the networks 1020, 1002, data retrieved from one of the storage devices 1006, 1009 or data
10 retrieved from a storage medium 1025 inserted into the corresponding reader 1012, all depicted in FIG. 10A, The execution of a set of the instructions may in some cases result in output of data. Execution may also involve storing data or variables to the memory 1034.

The disclosed communication management arrangements use input variables 1054, which are stored in the memory 1034 in corresponding memory
15 locations 1055, 1056, 1057. The communication management arrangements produce output variables 1061, which are stored in the memory 1034 in corresponding memory locations 1062, 1063, 1064. Intermediate variables 1058 may be stored in memory locations 1059, 1060, 1066 and 1067.

Referring to the processor 1005 of FIG. 10B, the registers 1044, 1045, 1046,
20 the arithmetic logic unit (ALU) 1040, and the control unit 1039 work together to perform sequences of micro-operations needed to perform "fetch, decode, and execute" cycles for every instruction in the instruction set making up the program 1033. Each fetch, decode, and execute cycle comprises:

- 25 (a) a fetch operation, which fetches or reads an instruction 1031 from a memory location 1028, 1029, 1030;
- (b) a decode operation in which the control unit 1039 determines which instruction has been fetched; and
- (c) an execute operation in which the control unit 1039 and/or the ALU 1040 execute the instruction.

30 Thereafter, a further fetch, decode, and execute cycle for the next instruction may be executed. Similarly, a store cycle may be performed by which the control unit 1039 stores or writes a value to a memory location 1032.

- 26 -

Each step or sub-process in the processes of FIGs 1 to 9 and 11 to 25 is associated with one or more segments of the program 1033 and is performed by the register section 1044, 1045, 1047, the ALU 1040, and the control unit 1039 in the processor 1005 working together to perform the fetch, decode, and execute cycles for every instruction in the instruction set for the noted segments of the program 1033.

The method of communication management and call conversation mapping may alternatively be implemented in dedicated hardware, such as one or more integrated circuits performing the functions or sub functions of call handling, inbound service management, outbound job management, service security gateway, and load balancing. Such dedicated hardware may include graphic processors, digital signal processors, or one or more microprocessors and associated memories.

LOGICAL SOFTWARE ARCHITECTURE

Embodiments are described by reference to a system referred to as a communication management system or an Intelligent Contact Cloud (ICC) system. The system features a distributed "cloud" of semi-autonomous nodes, each of which is configured to handle incoming and outgoing communications over a range of communication channels. As indicated above, the communication channels may include, for example, voice channels, SMS channels, chat channels, and so on.

FIG. 1 illustrates, from a logical perspective, a communication system 100, for example for implementing multiple contact-centre processes, with an application architecture split into a plurality of major functional components. The physical architecture of the system 100 is discussed separately further below. The components shown in FIG. 1 are independent of each other, yet combine to realise the complete system. This gives a high level of distribution and scalability, as the location of individual hardware and/or software components becomes relatively unimportant. In this manner, the logical location of the components within the application is structured, however the physical location of the components can be diverse.

In order to have the components stand alone, the logic that drives those components and ultimately identifies how those components work and interact with other components is governed by a central process and body of information describing how the component functions within the system. This service is herein described as a managing core, or an "ICC core", and is illustrated in FIG. 1 as ICC core 101.

- 27 -

The ICC core 101 is a software application executable within a computing device, such as the computing device 1000 of FIGs 10A and 10B, and defines a set of processes and procedures that provide the logic to power a contact centre process for a merchant. In the present embodiment, components within the ICC core are built from base
5 class modules with inheritance and standard communications methods providing the backbone for a scalability and flexibility for multiple merchant processes. Some major functional processes of the ICC core 101 are described below.

The ICC Core 101 includes an Inbound Service Manager (ISM) 110, an Outbound Job Manager (OJM) 120, a Service Security Gateway (SSG) 130, a Job
10 Scheduler 140, a Maintenance Application 180, a Merchant 1 Call Architecture 160, and a Merchant 2 Call Architecture 170, each of which is coupled to a Management Application 150. The Management Application 150 controls communication among the functional processes of the ICC Core 101.

The Inbound Service Manager (ISM) 110 is an application that is responsible
15 for allowing and responding to inbound service requests 1126 received by the ICC Core 101. Inbound service requests 1126 can be received on many different channels, at any time, and the ISM 110 interprets each request and delivers an appropriate response. When an inbound service request is received by the ICC Core 101, the ISM 110 performs a first check with the Service Security Guard 130 to ensure that the request is for a valid
20 service, and that the source of the request is a valid source for the requested service. After the request is validated by the service security guard, the ISM 110 proceeds to check the strategy of the requested service and requests any actions to be carried out by the ICC core 101. For example, a requested service may require the playing of a recorded message and the ISM 110 retrieves from the data store 1170 information relating to the event of playing
25 the recorded message.

The Outbound Job Manager (OJM) 120 controls all the node requests for jobs to be completed (for example, the sending of a particular message via SMS to a set of recipients) and sends outbound instructions 1128 to the plurality of nodes 1108, 1120. A queue of jobs to be performed is distributed across tenants, organisation units, broadcasts
30 ,and services as evenly as possible and in accordance with any predefined priorities. A tenant may correspond, for example, to an organisation or company that has commissioned an operator of the communication management system 1100 to handle customer service

- 28 -

contact centre operations, marketing operations, or a combination thereof. The communication management system of the present disclosure is adapted to handle multiple tenants through the logical architecture built into the OJM 120. This logical architecture allows a plurality of tenants (such as different companies) to share the physical components of the distributed architecture, such as communication channels of nodes, based on varying requirements over time. For example, the share of overall resources utilised by a given tenant varies over time based on the requirements of that particular tenant. The logical architecture is configured to balance priorities of outbound service requests to ensure there is minimum lag or delay for real-time services, such as webdial.

The ICC Service Security Guard 130 adds a protection layer to the communication management system in the same way a firewall adds protection to a network. Every inbound request received from an external source is validated by the Service Security Guard 130 to ensure that the request is a valid request. The ICC Service Security Guard 130 also optionally maintains a blacklist of invalid request sources to provide an extra level of protection against possible threats. If the ICC Service Security Guard 130 identifies a potential security breach, the ICC Security Guard 130 automatically adds the source of the request to the blacklist. Only system administrators have the level of access to manually change the entries in the blacklist.

The ICC Job Scheduler 140 allows scheduled jobs to be activated when required. Scheduled jobs are planned in advance and need to be monitored and activated when the schedule is due.

The Merchant 1 Call Architecture 160 defines a call handling process for calls relating to Merchant 1. The call handling process may define, for example, how incoming telephone or web requests are to be routed, when to generate and transmit SMS messages or emails as part of a promotional campaign for Merchant 1 and to whom those messages and emails are to be sent. Merchant 1 is able to access the Merchant 1 Call Architecture 160 to define parameters of the call handling process via a connection 165. In one arrangement, the connection 165 couples the ICC Merchant 1 Call Architecture 160 of the ICC Core 101 to a management gateway of a node.

Similarly, the Merchant 2 Call Architecture 170 defines a call handling process for calls relating to Merchant 2. Merchant 2 is able to access the Merchant 2 Call Architecture 160 to define parameters of the call handling process via a connection 175. In

- 29 -

one arrangement, the connection 175 couples the ICC Merchant 2 Call Architecture 170 of the ICC Core 101 to a management gateway of a node.

The Maintenance application 180 is utilised to maintain and monitor the functional processes of the ICC Core 101. An authorised user accesses the Maintenance application 180 via a connection 185, which may be via a user input interface to the ICC Core 101, via a direct communications link, or via communications link with appropriate security parameters, such as a user identifier and an associated password. In one arrangement, the connection 185 couples the Maintenance application 180 of the ICC Core 101 to a management gateway of a node.

FIG. 15 is a schematic representation illustrating functional components of the ICC core 101. As described above with reference to FIG. 1, the ICC Core 101 includes an Inbound Service Manager 110, an Outbound Job Manager 120, a Service Security Gateway 130, a Job Scheduler 140, and a Maintenance Application 180, each of which is coupled to the Management Application 150.

The ICC Core 101 further includes at least one call architecture of a merchant. In the example of FIG. 15, the ICC Core 101 includes a plurality of call architectures, each associated with a merchant and defining a call handling process for that merchant. In particular, the ICC Core 101 includes a Merchant 1 Call Architecture 160 through to Merchant "n" Call Architecture 170, each of which is coupled to the Management Application 150. Depending on the application and the requirements of the particular merchant, the call architecture associated with that merchant may include a plurality of call handling processes for voice calls. This may occur, for example, when a merchant utilises a plurality of different telephone numbers to identify incoming calls with different services offered by that merchant. A first telephone number may identify a call as a new customer enquiry and a second telephone number may identify a call as an existing customer with a payment query.

Merchant 1 Call Architecture 160 includes a plurality of call handling processes associated with Merchant 1: Call Processing A 162, Call Processing B 164, and Call Processing C 166. In this example, Call Processing A 162 defines a call handling process for inbound voice calls relating to Merchant 1, Call Processing B 164 defines a call handling process for inbound web requests relating to Merchant 1, and Call Processing C 166 defines a call handling process for inbound SMS messages relating to Merchant 1.

- 30 -

Merchant "n" Call Architecture 170 includes a plurality of call handling processes associated with Merchant "n": Call Processing A 172, Call Processing B 174, Call Processing C 176, Call Processing D 178, and Call Processing E 179. In this example, Call Processing A 172 defines a call handling process for inbound SMS messages relating to a first promotion being run by Merchant "n", Call Processing B 174 defines a call handling process for inbound web requests relating to Merchant "n", Call Processing C 176 defines a call handling process for inbound SMS messages in response to SMS payment reminders relating to Merchant "n", Call Processing D 178 defines a call handling process for inbound inbound voice calls relating to a first telephone number associated with Merchant "n", and Call Processing E 179 defines a call handling process for inbound voice calls relating to a toll-free Smart-number relating to Merchant "n".

ICC COMMUNICATION GATEWAY

Returning to FIG. 1, coupled to the ICC core 101 is an ICC Communication Gateway 102. The ICC Communication Gateway 102 facilitates the media communication for all communications channels (including voice calling and SMS/MMS/email), and is flexible to support future channels as such channels are developed (e.g., social media, such as integration to channels provided via Facebook, Twitter, or the like).

The ICC Communication Gateway 102 is embodied within each node 1108, 1120 in the communication system 1100. A node application 126 within each node 1108, 1120 provides logic to manage each of the ICC Communication Gateway 102 and a channel driver 127, and thus control communication between the ICC Communication Gateway 102 and the ICC Core 101 and between a plurality of communication channels 1109 and the outside world.

The ICC Communication Gateway 102 includes a plurality of gateways, corresponding to the defined system architecture. These gateways and corresponding applications within the system are described in brief below and are detailed further in later areas of this document.

- *Conversation Gateway 103* - provides connectivity to external gateway services (Voice, SMS, MMS, Email, Chat, Video).
- *Contact Centre Gateway 104* - provides agent interaction, 3rd Party Computer-Telephony Integration (CTI) functionality, and a one channel presence. A "one channel presence" treats multiple communication modes as a single

- 31 -

channel, such that an agent of a merchant can communicate with a customer in a single conversation channel using different communication modes, such as voice, SMS messages, web chat, and emails.

- *Management Gateway 105* - provides a gateway for managers, supervisors and administrators to configure, monitor, and otherwise administer the contact centre system.
- *Live Data Gateway 106* - provides a gateway for providing live data feeds, wallboards, and alerts.
- *CRM Gateway 107* - provides for 3rd party Customer Relationship Management (CRM) communications. This is typically delivered through a set of Application Programming Interfaces (APIs).
- *Data Vault Gateway 108* - provides access to historical data, voice recording files, sound files, images.

The conversation gateway 103 provides a connection to the outside world of the ICC system 1100. This includes the services that provide a gateway to external provider services, thereby to allow the present system to have a real world application. Exemplary types of service gateways are detailed below. These services are provided by each of the distributed nodes of the ICC system 1100.

An Inbound Calls service provided by the conversation gateway 103 provides for connection directly to multiple telecommunication service providers ("Telcos") via Session Initiation Protocol (SIP) over Ethernet or El. These service gateways are limited by the hardware that is being used for the traversal of calls and the Ethernet bandwidth or availability of El termination from the telecommunication service provider. A Smart-Number is a telephone number presented as a phone word and is typically utilised in conjunction with a freephone or toll-free telephone service. A Smart-Number typically includes a freephone or toll-free prefix, such as 1800, 13, or 1300, and one or more words relating to a service. For example, a Smart-Number relating to a plumber may be 1800 PLUMBER, which corresponds to the telephone number 1800 7586237 on a standard telephone handset. Using Smart-Numbers from telecommunication service providers, the inbound call traffic is load balanced across multiple Inbound Call services to allow for a very large number of concurrent inbound calls, if required. This is useful when handling call loads generated in response to promotional materials, competitions, and in conjunction

- 32 -

with live or broadcast events.

Inbound calls which are provided via SIP are distributed via a load balancing process performed by the ICC Core 101. This load balancing is separate from the incoming load balancing performed by the SIP Load Balancer 1106 and Web Load Balancer 1118. In the load balancing performed by the ICC Core 101, the ICC Core 101 interprets the SIP control information associated with an incoming service request derived from an inbound SIP call and the ICC Core 101 selects a node to handle the call. The ICC Core 101 sends an outbound communication message back to the SIP gateway 1104 instructing the SIP gateway 1104 to redirect the call to the selected node. This then allows the selected node to take the call directly and then the selected node communicates directly with the SIP gateway 1104. The Real-Time Transport Protocol (RTP) stream is always between the SIP gateway 1104 and the selected Node.

FIG. 2 illustrates the inbound call load balancing process mentioned above. Inbound calls are provided by a SIP gateway 201, corresponding to the SIP Gateway 1104 of FIG. 11, and arrive at a SIP call load balancer 202, corresponding to the SIP Load Balancer 1106 of FIG. 11. The SIP load balancer 202 provides instructions to each of a plurality of nodes 203. Specifically, for each call an instruction is provided by the SIP load balancer 202 to a selected one of the nodes 203. That selected node then communicates directly with the SIP gateway 201 via an RTP stream. The nodes 203 are all in communication with the ICC core 101, which manages the allocation and scheduling of jobs among the nodes 203.

In a manner similar to the inbound calls service, the outbound calls service is connected to the telecommunication (Telco) service providers via SIP over Ethernet or E1. The SIP and Web gateways 1104, 1116 are limited only by the hardware used for the traversal of the calls and the available bandwidth or availability of E1 termination. The Inbound and outbound calls services are functionally separate within the managing core 101, 1130 to allow for the confirmed availability of both inbound and outbound channels for the system users.

Outbound calls are load balanced by the ICC core 101. Each outbound voice call is directed by the core 101, 1130 to a selected node 203 from the plurality of nodes 1108, 1120 in the system 1100, based on predefined criteria. In one example, the managing core 1130 selects a node 203 with the least load at the relevant point in time.

- 33 -

This direction of outbound calls ensures that the load is spread evenly across the node base.

Inbound emails are received from a customer via a third-party service, such as a web server provided by an Internet Service Provider (ISP), wherein the web server couples to one or more of the communication channels 1109 via the firewall 1102, Web Gateway 1116, and Web Load Balancer 1118. Each of the inbound email services has non-contended Ethernet connectivity to the Internet. Each of the inbound email services may optionally have point-to-point links to mail services of one or more particular client, which assists in implementations in which the volume of email is very large. The inbound email services are limited by the hardware and bandwidth available, and a load balancing protocol may be implemented.

Outbound email services, much like the inbound email services, use a third party provider, such as a web server provided by an Internet Service Provider and coupled to one or more of the communication channels 1109 via the firewall 1102, Web Gateway 1116, and Web Load Balancer 1118. Each of the outbound email services has non-contended Ethernet connectivity to the Internet. In a manner similar to the inbound email services, each of the outbound email services may optionally have point-to-point links to mail services of one or more particular client, which assists in implementations in which the volume of email is very large. The outbound email services are limited by the hardware and bandwidth available. Inbound and outbound email services are functionally separate to allow for capacity confirmation to an end customer for the inbound and outbound email traffic for that end customer.

Inbound SMS services are connected to an SMS service provider via an Application Programming Interface (API). The inbound SMS service has non-contended Ethernet connectivity to the Internet and may have direct connectivity to the SMS provider. The inbound SMS service is again limited only to the hardware and available bandwidth. Using load balancing provided by the SMS provider, the inbound SMS system can optionally be grown across multiple services. The SMS service provider couples to one or more of the communication channels 1109 in a node 1108, 1120 via the mobile network 1186.

FIG. 3A is a schematic block diagram representation illustrating handling of an inbound SMS message. Inbound SMS messages are received by the communication management system 1100. The managing core 1130 analyses information derived from

- 34 -

the SMS message, such as the called number, calling number, customer identifier, customer account number, keyword, or a combination thereof, to search a conversation data store 301 stored in the data store 1170. The conversation data store 301 stores a record for each conversation that is presently active. This allows the managing core 1130
5 to determine if the inbound SMS is a reply to a current conversation or a new conversation initiated by the sender. The managing core 1130 applies any work flow that is applicable to the current conversation. In one implementation, the work flow is established by a call handling process defined in a call architecture 1175 of a merchant relating to the SMS message. FIG. 3 shows a user utilising a remote computing device 302, such as a mobile
10 telephone or smartphone, to send an inbound SMS. The inbound SMS is received by a channel bank of the communication management system 1100, represented by module 304 and implemented by the plurality of communication channels on the nodes 1108, 1120. Information derived from the inbound SMS is passed to the conversation data store 301. Module 306 interrogates the conversation data store 301 to identify whether the inbound
15 SMS relates to a current conversation and subsequently apply any relevant work flow. The data store 301 is stored in the datastore 1170 or in another storage medium 1010, 1009.

Like the inbound SMS services, outbound SMS services are connected to the SMS provider via an API. The outbound SMS service also has non-contended Ethernet connectivity to the internet and may have direct connectivity to the SMS provider. The
20 outbound SMS service is also limited only to the hardware and available bandwidth. Inbound and outbound SMS services are functionally separate to allow for capacity confirmation to an end customer for inbound and outbound SMS traffic for that end customer.

When an outbound SMS message is initiated, the managing core 1130 creates a
25 conversation record for the outbound SMS in the conversation data store 301, which the system 1100 uses to monitor conversations when inbound SMS messages are received. FIG. 3B is a schematic block diagram representation illustrating initiation of an outbound SMS message. In the example of FIG. 3B, the communication management system 1100 is configured to implement a dial-out campaign.

30 An SMS channel bank is a set of telephone numbers that can be utilised in a dial-out campaign. A merchant can utilise a subset of the SMS channel bank for different campaigns. In one arrangement, each number in the SMS channel bank is associated with

- 35 -

an identifier that associates the number with a particular merchant, campaign, contact, or any combination thereof. One arrangement also associates each number in the SMS channel bank with an operating timeframe, which defines a time period during which that number is active for the identifier associated with that number. As the communication management system 1100 knows which numbers from the SMS channel bank are being
5 utilised in the sending of outbound SMS messages, the communication management system 1100 can match incoming SMS replies to the outgoing SMS messages in an existing conversation record. For example, a phone number "555 1234" is utilised to send an SMS message to Customer C 1640. Phone number "555 1234" is reserved for
10 communications with Customer C 1640 for a predefined operating timeframe, so an identifier associated with that number is marked "Customer C 1640". When an incoming SMS message is received by the system 1100, the system 1100 checks information pertaining to the SMS message to determine whether the system 1100 has sent an outbound SMS message to that number within the operating timeframe and, if so, matches the
15 incoming SMS message to the outbound SMS message. In another example, a single telephone number from the SMS channel bank is utilised in the sending of an outbound SMS message to a selected list of contacts. When customers reply to that single number, the system 1100 utilises a combination of the single number and the number of the customer to identify uniquely each conversation.

20 Module 310 selects a phone number from an SMS channel bank and passes the number to the conversation data store 301. The conversation data store 301 creates a record for a new conversation relating to the selected phone number and then passes the selected phone number to module 320, which sends an SMS message from the selected phone number to a remote computing device 330, such as a mobile telephone.

25 A chat service is an online messaging system provided over a communications network, such as the Internet, and may be implemented as a web-based interface to enable a customer accessing a computing device to communicate with a customer service representative, or agent, associated with a particular merchant. Chat services in the present embodiment include multiple chat provider interfaces in place, thereby to allow
30 interoperation with various providers. This connectivity is provided via APIs into respective engines of the chat providers. The chat services have non-contended Internet bandwidth. An end user interface of the ICC system 1100 is propagated from the system

- 36 -

1 100 through the firewall to a remote computing device accessed by a customer, wherein the end user interface is a dialog box or "popup" window on a website of a merchant.

When a chat message is received, the chat engine sends the interpretation of the message to the ICC system's end user interface, while if a message is sent from the end user interface
5 desktop, the message is translated into the required chat engine required format by the chat services. The scalability of the chat services is limited to hardware and available bandwidth and load balancing can be performed by limiting the number of concurrent connections allowed for each API.

Video services are configured to operate substantially in the same way that the
10 voice services work, either to be delivered via the SIP Gateway 1104 using SIP over Ethernet (using an available video codec such as H.263) or be delivered over ISDN, transcoded, and delivered via SIP over Ethernet to the customer. Availability of video over SIP is heavily dependent on the availability of a telecommunications service provider that can handle the traffic. Aside from the telecommunications service provider, the scalability
15 of these services is limited only by the hardware available to perform functionality associated with the video services.

Outbound video is provided through the use of SIP over Ethernet to the end user or customer. The outbound video may be transcoded to video over ISDN if the service is the only one available, but ultimately the video traffic will be passed, using SIP,
20 over one or multiple telecommunications service providers with video capabilities. Inbound video and outbound video services are kept separate at all times, so as to allow for the confirmed allocation of available services for clients, i.e., to make sure that a large influx in either types of traffic would not affect the other service.

Contact centre gateway 104 of FIG. 1 manages agents of merchants, wherein
25 those agents have logged into the system 100 and are active, and manages the components of the system architecture that facilitate communication between agents and customers. These components of the system architecture include communication channels implemented using web servers, voice servers, SMS servers, etc. The components enable the use and interfacing with the application components by way of running critical
30 components such as World Wide Web "web" services or Private Automatic Branch Exchange (PABX) Services. In one arrangement, an agent of a merchant logs into the system 1100 and establishes a voice connection and a web connection to the system 1100.

- 37 -

The contact centre gateway 104 manages the voice connection and web connection. The key services offered by the contact centre gateway 104 are detailed below.

Load Balance. The load balancing of web requests is the process that allows the scalability of the contact centre gateway 104. This is illustrated in FIG. 4. A request
5 405 for a web session is first received by a load balancer 401, 1118 via a public facing firewall 1102, and the load balancer 401 then allocates a node 402 from the plurality of distributed nodes to connect that web session. The plurality of nodes 402 are implemented in this example as an expandable web server cluster that is coupled to an internal firewall
1 122, which in turn is coupled to the ICC Core 101. Each new web session is balanced by
10 the load balancer 401 across all the online nodes that have web session capability. Every request after that session initiation that relates to the same session is then handled by the same allocated node.

Login. In order for the login service to be scalable within the application, the login service needs to be an extremely light process. That is, the login process
15 preferably consumes relatively few resources and time. The login process in the present embodiment makes the login service a "one touch" process. This login process acts as a redirection process for when a user attempts a connection to the ICC system. The login process is used by an agent of a merchant to indicate an availability for communication with a customer through the system 1100. The agent of the merchant accesses a computing
20 device to visit a predetermined Uniform Resource Locator (URL) access portal and enters a user identifier and password associated with that agent. The user identifier and password are then securely posed to the login process, which does not deal with the actual login, but rather sends a request to an Allocation Service to request a physical service, with enough resource, to which the user can connect. The Allocation Service is resident on the
25 managing core 101. Once the request is answered, a login request is sent to the allocated Security/Authentication service for validation. The Security/Authentication service forms part of the node application 126. Whilst this embodiment relates to the login process followed by an agent of a merchant, a similar login process can be implemented for enabling authorised access to a merchant manager or customer.

30 *Security/Authentication.* This service receives a login request and authenticates the login request for access to the remainder of the services. Once authenticated, a certificate is assigned to the user for a limited access time and a request is

- 38 -

then sent to the Allocation Service for the relevant Interface Service and Extensions Service (if it is a voice system) to which the user is sent.

Interface Service. Once the user (agent) has successfully authenticated into the system, that user needs to be able to interface with the system via a web page. These web pages are hosted on web servers. The Interface System Service is provided by the web
5 servers used for the system, wherein the web servers are implemented by the nodes 1108, 1120.

Extensions Service. The ICC system, in a particular embodiment, offers a hosted PABX system. With the abstraction of the Interactive Voice Response (IVR)
10 process into the Application Services and the abstraction of the Inbound & Outbound call services into the Gateway and Application Services, what is left is the requirement for Extension services for the agents to be able to connect, so that the agents can receive and make calls. An Extensions Service is assigned to the agent when the agent logs in, via a request to the Allocation services. Thus, the ICC system delivers PABX functionality to
15 voice calls handled by the system, wherein the PABX functionality includes, but is not limited to, routing of calls to different numbers, and voicemail.

Logging. When a transaction is performing associated tasks, the transaction needs to log the status and outcomes of those tasks. Each transaction in the system is allocated a Logging service, which the transaction requests from the Allocation service.
20 Once a transaction has an event that needs to be logged, the transaction sends a request to the Logging service, which logs that information as required. The Logging service logs information that will be used for all levels of reporting.

Management Gateway. The Management Gateway is the interface that enables the management of the ICC system platform. This gateway is preferably delivered
25 via a web based application.

Live Data Gateway. The components of the Live Data Gateway give real time views of all the data, as shown in FIG. 5. FIG. 5 shows the ICC core 501 coupled to a Live Data Gatherer 501. The real-time Live Data Gatherer engine 501 is a process which repeatedly polls the ICC core 101 for information to be available for any of the Live
30 Gateway data feeds 502. The live data feeds are a constant stream of data in XML format that are broadcast to any listening processes. The listening processes can then update visual interfaces, such as live reports 504 and wallboards 503.

PHYSICAL SYSTEM ARCHITECTURE

The distributed communication management system of the present disclosure makes use of a distributed application, and therefore the associated hardware is also distributed. The hardware can be split across multiple physical systems and data centres.

5 For example, a first group of one or more nodes may form part of a first local system, whilst a second group of one or more nodes may form part of a second local system distributed with respect to the first system.

The distributed architecture model enables substantially infinite scalability of the communication management system. This is achievable using a single managing
10 "ICC" core, which in some embodiments is a distributed component in itself, and many nodes which each perform work for the overall system.

Each node repeatedly submits, as a "heartbeat" to the ICC core 101, 1130, requests for a task to perform. If the ICC core 101, 1130 has no jobs for allocation to a given node, that node remains dormant until the next heartbeat from that node is due. The
15 ICC core 101, 1130 manages all the tasks performed by the system 1100 and distributes tasks based on the capacity, load, and capabilities of respective nodes.

All the processing of media gateways (such as voice) is performed at the individual nodes. The nodes essentially pool together to create the overall processing power of the ICC system 1100. The nodes are generally substantially identical to one other
20 at a functional level, and are capable of performing the same tasks. However, individual nodes may be configured with different communication channels. The tasks performed by the nodes are managed by the ICC core 101, 1130. This model allows for substantially infinite scalability as the bulk of the processing is handled by the nodes. FIG. 6 is a schematic representation of a Distributed Architecture model, which includes a plurality of
25 distributed ICC sub-cores 601, and a plurality of distributed nodes 602. The distributed architecture 600 of FIG. 6 is split into three local systems 603A, 603B and 603C, each of which includes a sub-core 601 and a corresponding set of nodes 602. The sub-cores 601 communicate with each other to perform data synchronization across the distributed architecture of the local systems 603A, 603B, and 603C. Each local system may be
30 allocated to a particular merchant, campaign, communication medium, or any combination thereof. Alternatively, one or more of the local systems may operate together.

The physical realisation of this model occurs with load balancing of all the

- 40 -

components which make up the overall system. The distributed nature of the system provides resilience through redundancy and every process is expandable. FIG. 7 is a schematic representation illustrating a physical layout of a distributed communication management system 700 in accordance with the present disclosure. The system

5 700 includes a public facing firewall 701 which initially receives requests 702. These requests 702 are passed by the firewall 701 to load balancers 703, including primary and secondary web load balancers (for web requests) and primary and secondary SIP load balancers (for SIP requests). In the illustration of FIG. 7, only the primary balancers are in use and the secondary load balancers are in reserve. The load balancers 703 direct requests
10 702 to nodes in an expandable node cluster 705, which includes one or more nodes as described elsewhere herein. These nodes communicate with a managing ICC core 101 via an internal firewall 707 and primary load balancer 708. The managing core 101 is defined by expandable database (DB) server clusters and expandable file server clusters 706. These clusters 706 have respective disk arrays 709.

15 Data Synchronisation allows multiple cores to be replicated across locations. The use of Data Clusters allows the ICC core itself to scale to any size without the need of any architecture changes. The ICC Data Cluster is managed at the database level and allows the database to be clustered across servers. With a managed cluster the data is split internally with a map of the data managed by the database system.

20 The Data Cluster is segregated into a database load balancer, database application, and physical data storage. These three layers combine to allow the scalable data store that the ICC system requires, as shown in FIG. 8. FIG. 8 is a schematic representation illustrating a data clustering arrangement. FIG. 8 shows incoming core database requests being received by the internal firewall 707, 1122. The internal firewall
25 707, 1122 passes the requests to a load balancer 708, 1124 for allocation to a server in an expandable server cluster 706 implementing the core 101, 1130. The core 101, 1130 is coupled to a data store 709, 1170.

NODE OVERVIEW

The Nodes 1108, 1120 are individual servers which are commissioned to work
30 for the ICC core 101, 1130. In one arrangement, each node 1108, 1120 is implemented using the computing system 1000 of FIGs 10A and 10B. Each node communicates with the ICC core 101, 1130 and communicates with the outside world via its own media

- 41 -

gateway. The plurality of nodes 1108, 1120 may be referred to as a "cloud" of distributed nodes for actioning tasks for the core 101, 1130.

The node heartbeat process mentioned above allows each node to communicate with the ICC core on a regular basis and provides a constant pulse allowing the ICC core to manage the resources of the node. Each node is repeatedly asking the ICC core if there are any tasks for that node to perform. If there is a task to be performed, the ICC core responds to the requesting node with that task. The node performs that task, and then notifies the ICC core with the outcome of the task. The heartbeat is a variable process and can speed up and slow down, as required.

Each node performs its own self maintenance. This is achieved by requesting any latest changes from the ICC core. Such changes may include, for example, software or firmware updates or changes to parameters. Any such change is notified to the node by the ICC core in response to a heartbeat message. If there is an upgrade to be performed, the Node downloads the required files from the core and upgrades the appropriate modules on the node. Self maintenance is extremely important in this scalable model to keep the effort required for maintenance to a minimum. It is also important for each node to be substantially the same as each other node at all times, so that tasks to be performed may be allocated across as much of the distributed architecture as possible.

In terms of node driver configuration, the media driver configuration for each communications channel on a node is maintained by an automated process within each node. The ICC core publishes the latest driver configurations and applications for release and each node checks the latest versions against its local settings. If the node requires an update, the node transfers the update from the ICC core and re-initialise the affected channel(s).

Extra media channels are in some embodiments added using the same general process. This allows for additional media drivers to be deployed as those media drivers are developed for new technologies or media interfaces, such as social networking.

Certain channels require media to be located locally on the node. Some examples of this are sound files for Interactive Voice Response (IVR) and images for web pages. As part of the heartbeat process, all nodes are repeatedly checking if any files are required to be copied from the ICC core and stored locally on the nodes. If there are new or updated files, the node downloads the file(s).

- 42 -

The channel capacity of each node is configured by the ICC core, based on the build and specifications of the node. The nodes may differ greatly in channel capacity. For example, one node may be configured with a high number of voice channels, but a low number of SMS channels. Each node is configured independently and managed from the ICC core to ensure the most even distribution of channel activity across the entire platform.

NODE ARCHITECTURE

The node architecture model of the present embodiment is shown in FIG. 9, which shows an exemplary node 900, and is discussed below.

The node 900 includes an ICC Communication Gateway 901, an ICC Node application 902, an ICC Channel Driver 903, and a plurality of communication channels 904.

The communication between each node 900 and the ICC core 101, 1130 goes through the ICC Communication Gateway 901. The ICC Communication Gateway 901 is responsible for all the communications between the ICC Node application 902 and the ICC core 101, 1130. In one implementation, all messaging between the ICC Node application 902 and the ICC Communication Gateway 901 uses standard XML. Other metalanguages and programming languages may equally be practised without departing from the spirit and scope of the present application. The Communication gateway 901 converts all messages from the ICC core 101, 1130 to XML to be passed to the ICC Node Application 902.

The ICC Node application 902 controls functionality of the ICC Nodes, however the ICC Node application 902 is only a communication tool. There is no business logic within the ICC Node Application 902, as the business logic resides in the ICC core 101, 1130. All the node functionalities mentioned in the section above entitled "Node Overview" are carried out by the ICC Node Application 902.

The ICC Channel driver 903 is the interface layer to the communication channels 904. The channel driver 903 is responsible for communications with a selected channel of the communication channels 904. The channel driver 903 translates the commands from the ICC Node application 902 to the appropriate commands for the selected channel. The communication channels depend on the configuration of the particular node. In the example of FIG. 9, the communication channels include an SMS channel, a voice channel, an email channel, a World Wide Web channel, a video channel, and a Salesforce.com customer relationship management (CRM) channel. The

- 43 -

communication channels also include a plurality of other channels available for other communication types provided by third-parties, including, for example, but not limited to, twitter™, facebook™, and NetSuite™, wherein the channels function as an application programming interface between the node and the third-party.

5 FIG. 12 is a schematic block diagram representation illustrating the use of the contact centre system 1100 of FIG. 11 to establish a call from a customer to an agent of a merchant, wherein the contact centre system 1100 has a data store 1170 that stores a call architecture 1175 of that merchant. In the example of FIG. 12, the data store 1170 is presented as a first data store 1170a integral with the managing core 1130 and a second
10 data store 1170b coupled to the managing core 1130. In this example, the call architecture 1175 defines a call handling process for the merchant, wherein the call handling process is operable to detect receipt by a first node of an incoming voice call on a corresponding incoming voice channel of that first node, and to couple the voice call from the first node via one of the call channels of the first node to a second node for connection
15 to an agent associated with the merchant. In this example, the contact centre system 1100 includes five nodes 1108a, ..., 1108e.

In the example of FIG. 12, the customer accesses a telephone handset 1206 or a computing device 1205, which may be a general purpose computer implemented using the general purpose computing device 1000 of FIGs 10A and 10B, to send a request 1215 to
20 the contact centre system 1100 seeking to establish a connection to the merchant. The request 1215 may, for example, be based on a telephone number or World-Wide- Web hyperlink associated with the merchant. The telephone handset 1206 may be a conventional telephone handset, a mobile telephone handset, a smartphone, or the computing device 1000 of FIG. 10, wherein the customer utilises the microphone 1080 and
25 speaker 1017 to effect a telephone, such as when making a voice or video call utilizing Skype™. In this example, the request is from the computing device 1205 for a voice over Internet Protocol (VOIP) call to the merchant.

The contact centre system 1100 receives the request 1215 at the firewall 1102, which determines that the request 1215 is for a voice call. The firewall 1102 sends a
30 request 1220 to the SIP Gateway 1104. The SIP Gateway 1104 forwards the request 1225 to the SIP Load Balancer 1106 for allocation to a node that is able to handle the call request. In this example, the SIP Load Balancer 1106 forwards the request 1230 to node

- 44 -

1108c. Node 1108c receives the request 1230 on an incoming voice channel and sends an incoming service request 1235 to the managing core 1130 seeking clarification of how the request is to be handled. The managing core 1130 receives the incoming service request 1235 and identifies the merchant associated with the request. The managing core 1130
5 then retrieves from the data store 1170 the call architecture associated with the identified merchant and determines the required handling process. In this example, the call is to be routed to an agent associated with the merchant. The agent has previously logged-in to the contact centre system 1130 using a computing device 1210, to advise of the availability of that agent. The managing core 1130 maintains a conversation data store 301 for each new
10 incoming and outgoing request.

The managing core 1130 sends an outbound instruction in the form of a call handling message 1240 to the node 1108c, which in this example instructs node 1108c to couple the call to node 1108a, based on the retrieved call architecture 1175 of the merchant. Node 1108c sends a request 1245 to establish a connection with node 1108a.
15 Node 1108a receives the request 1245 and sends an incoming service request 1250 to the managing core 1130 seeking clarification of how the request 1245 is to be handled. The managing core 1130 receives the incoming service request 1250 and identifies the merchant associated with the request. The managing core 1130 then retrieves the call architecture 1175 associated with the identified merchant and determines the required
20 handling process. In this example, the call is to be routed to an agent associated with the merchant, wherein the agent has previously logged-in to the contact centre system 1100 to advise of the availability of that agent.

The managing core sends an outbound instruction 1255 to the node 1108a advising node 1108a to connect the call to the computing device 1210 associated with the
25 agent, based on the call architecture of the merchant. Node 1108a sends a request 1260 to the SIP Load Balancer 1106, which in turn forwards a request 1265 to the SIP Gateway 1104. The SIP Gateway 1104 forwards a request 1270 to the firewall 1102, which in turn forwards a request 1275 to establish a connection to the computing device 1210.

The distributed nature of the contact centre system 1100 allows for the system
30 to be readily scaled through the addition of further nodes, sub-cores, or a combination thereof. When a node 1108, 1120 receives a request to establish a connection, the node 1108, 1120 sends an incoming service request 1235 to the managing core 1130 to seek

- 45 -

instructions for how the connection is to be established. The managing core 1130 utilises information in the incoming service request 1235 from the node 1108, 1120 to identify a merchant architecture 1175 stored in a data store 1170 associated with the managing core 1130, wherein the merchant architecture 1175 defines a call handling process for that
5 merchant. The data store 1170 may be integral with the managing core 1130 or alternatively may be located remotely from the managing core 1130 and coupled to the managing core 1130 by a communications link. The data store 1170 may be a database implemented, for example, using the disk storage medium 1025, storage devices 1009, or a combination thereof.

10 Storing an associated merchant architecture 1175 for each merchant enables the contact centre system 1100 to handle a plurality of merchants, with each merchant architecture defining a call handling process for call requests associated with the respective merchant.

FIG. 13 is a flow diagram illustrating a call handling process 1300 performed by
15 the node 1108c in the example described with reference to FIG. 12. The node call handling process 1300 begins at a start step 1305 and proceeds to step 1310, in which the node 1108c receives an incoming request 1230 from a customer. This incoming request is typically a voice call initiated by the customer wanting to communicate with a merchant. In one example, the customer dials a toll-free "1800" number associated with the
20 merchant. The call may be initiated, for example, by a customer seeking clarification of services offered by a merchant or may be in response to a billing query or payment, or may be in response to a communication sent from the merchant or on behalf of the merchant to the customer.

In a next step 1315, the node 1108c sends an incoming service request 1235 to the
25 managing core 1130, wherein the incoming service request 1235 seeks instructions for handling the request received from the customer. The incoming service request 1235 may include information pertaining to the call, such as the incoming caller number associated with the customer or the dialed number, or a combination thereof.

Control passes from step 1315 to step 1320, in which the node 1108c receives an
30 outbound call handling communication 1240 from the managing core 1130. The outbound call handling information 1240 may include, for example, routing information and may be based on the caller number associated with the customer or the dialed number. The

- 46 -

outbound call handling information may include, for example, routing information and instructions to connect the call to an external agent or one of the other nodes 1108a, ..., 1108n in the distributed system 1100. In the example of FIG. 12, the node 1108c receives outbound call handling information 1240 instructing the node 1108c to connect the call to node 1108a.

Control passes from step 1320 to step 1325 and the node 1108c connects the call to node 1108a, based on the outbound call handling information 1240 received from the managing core 1130. In this way, the call from the customer is now routed from the computing device 1205 accessed by the customer to the first node 1108c then onto the second node 1108a. Control passes to an End step 1330 and the process 1300 terminates.

In establishing the call from the customer to node 1108c and then to node 1108a to be coupled to a merchant agent, the call is not connected to the managing core 1130.

Rather, node 1108c sends the incoming service request communication 1235 to the managing core 1130 and receives the outbound call handling communication 1240 from the managing core 1130. The incoming service request communication 1235 and the outbound call handling communication 1240 contain information relating to the call, but there is no voice path to or from the managing core 1130. The distributed architecture of the contact centre system 1100 is arranged to enable the managing core 1130 to manage the distributed nodes 1108, 1120, wherein the nodes 1108, 1120 handle direct communication with the external world through a plurality of communication channels.

FIG. 14 is a flow diagram illustrating a call handling process 1400 performed by the managing core 1130 in the example described with reference to FIG. 12. The managing core call handling process 1400 begins at a start step 1405 and proceeds to step 1410, in which the managing core 1130 receives an incoming service request communication 1235 from a node 1108c. The incoming service request communication 1235 may be related to an incoming call received by the node 1108c from a customer 1205. The incoming service request communication 1235 may include, for example, information pertaining to the call received by the node 1108c such as, for example, the caller number associated with the customer, a customer reference number, a customer account number, a called number associated with a merchant, Internet Protocol (IP) address, or any combination thereof. Control passes from step 1410 to step 1415, in which the managing core 1130 identifies a merchant, based on the received incoming service

- 47 -

request communication 1235. Control passes from step 1415 to step 1420, in which the managing core 1130 retrieves from the data store 1170 a call architecture 1175 of the identified merchant. The call architecture 1175 of the identified merchant defines a call handling process for calls relating to identified merchant.

5 Control passes from step 1420 to step 1425, in which the managing core 1130 sends outbound call handling information 1240 to the node 1108c. The outbound call handling information 1240 instructs the node 1108c how to proceed with handling of the call. In one example, the outbound call handling information 1240 instructs the node 1108c to connect the call to an outgoing channel of that same node 1108c to establish a
10 connection to an agent associated with the identified merchant. In another example, the outbound call handling information 1240 instructs the node 1108c to connect the incoming call from the customer to a recorded voice announcement (RVA) or interactive voice recording (IVR) system. In a further example, the call handling information 1240 instructs the node 1108c to connect the incoming call from the customer to a second node
15 in the system. That second node subsequently contacts the managing core 1130 to seek further instructions for handling the call. Control passes from step 1425 and returns to step 1410, where the managing core 1130 awaits a further communication from another node.

CALL CONVERSATION MAPPING

One aspect of the present disclosure provides a conversation mapping method
20 and system. In one arrangement, the conversation mapping method and system are embodied utilising the communication management system 1100 of FIG. 11, wherein a conversation map application and conversation engine form part of the management application 150 of the managing core 101. The conversation mapping method and system enable a merchant to create a call conversation map that establishes a call workflow in a
25 multi-channel environment. The call conversation map is modifiable by the merchant and is able to handle communications on multiple media, including voice, data, SMS messages, email, and calendar entries. The mapping method may be implemented as an application for execution as a portion of the management application 150 in the managing core 101, 1130 of the communication system 100, 1100.

30 A call conversation map utilises a plurality of linked objects to define a logical call workflow for a given scenario. Each object corresponds to a state in the conversation map representing an event in a call handling process. Objects that involve the generation

- 48 -

of an outbound message 1128 or receipt of an inbound request 1126 are termed "tasks". One implementation of the conversation mapping system utilises a graphical user interface to provide a visual tool for designing a call conversation map. A call conversation map can be linear or alternatively can include one or more branches. Once a call conversation map has been created, the call conversation map can be activated by a trigger, which may be a timing event, an outbound broadcast from the communication management system 1100, or an inbound service to the communication management system 1100. Once triggered, a conversation map becomes active and a conversation engine handles incoming communications and generates outbound communications in accordance with the call workflow defined by the active conversation map.

The call conversation map is stored in the data store 1170 of FIG. 11 as part of a call architecture 1175 of an associated merchant. The call architecture 1175 includes one or more call handling processes for communications associated with the merchant. In the example of FIG. 15, a Merchant 1 Call Architecture 1160 includes three call handling processes associated with Merchant 1: Call Processing A 162, Call Processing B 164, and Call Processing C 166. In accordance with this embodiment, the merchant establishes a call conversation map to define the workflow for each call handling process 162, 164, 166. Whilst the example of FIG. 15 demarcated call handling processes based on the communication medium, the call conversation map is able to handle multiple communication media, such as voice, video, SMS messages, MMS messages, emails, recorded announcements, and interactive voice responses. However, multiple call handling processes may still be appropriate for a merchant under some circumstances, such as separate call handling processes for different marketing campaigns.

The conversation mapping method and system provide a merchant with a flexible mechanism for defining and controlling call handling processes. A merchant can define a separate conversation map for each call scenario, including, but not limited to, calls relating to marketing campaigns, different types of customers, different services, or different accounts. Each conversation map is optionally associated with one or more triggers that activate the conversation map. The merchant can modify each conversation map to update or change the call handling process, contact details associated with the merchant, a list of contacts to whom materials are to be sent, or triggers associated with the conversation map.

- 49 -

In one example, a trigger defines a time at which a conversation map becomes active. In another example, a trigger defines that a conversation map becomes active upon receipt of an incoming message 1126 from a particular source. In one example, a merchant defines a call conversation map to handle the payment of overdue accounts, wherein an initial task in the conversation map relates to the sending of a group SMS message to all customers with overdue accounts. The group SMS message includes predefined content provided by the merchant. The merchant defines a trigger to activate that conversation map on the first day of the month at 9am, so that every month the communication management system 1100 running an instance of the conversation mapping method generates outgoing SMS messages with predefined content.

FIG. 21 is a schematic block diagram representation illustrating a logical architecture of a call mapping system 2100. The call mapping system 2100 includes a conversation map module 2130 coupled to each of a conversation engine 2120, a conversation triggers module 2140, a contact library 2150, and a conversation management application programming interface (API) 2160. The conversation engine 2120 also couples to a media gateway 2110.

The conversation management API 2160 couples to each of an external API gateway 2170 and a conversation management interface 2180 to exchange communications with merchants, administrators, and other authorised persons. The conversation management interface 2180 provides a web-based interface for a merchant, administrator, or other authorised person to access the conversation mapping system 1100 to create or modify a call conversation map. The external API gateway 2170 provides an alternative interface for the merchant, administrator, or other authorised person to access the conversation mapping system 1100, whereby the external API gateway 2170 receives instructions for creating or modifying a call conversation map in accordance with a third party programming interface. For example, a merchant may utilise a third party application to provide via the external API gateway 2170 a batch set of instructions to configure a contacts list, a conversation map, triggers associated with one or more conversation maps, or other parameters of the conversation mapping system 1100.

The conversation management API 2160 couples to each of the contact library 2150, the conversation map module 2130, and the conversation triggers module 2140. Instructions received from the external API gateway 2170 and the conversation

- 50 -

management interface 2180 are passed to the conversation management API 2160 and then to one or more of the contact library 2150, the conversation map module 2130, and the conversation triggers module 2140. The instructions may relate, for example, but are not restricted to, adding, deleting, or modifying contacts in the contact library 2150, creating,
5 deleting, or modifying a conversation map via the conversation map module 2130, or creating, deleting, or modifying a trigger associated with a conversation map via the conversation triggers module 2140.

The media gateway 2110 represents a logical module for managing communications between the conversation mapping system 2100 and the outside world. In
10 one arrangement, the media gateway is implemented by one or more nodes 1108, 1120 in the communication management system 1100.

The conversation engine 2120 is a software application containing instructions for execution on a processor 1005 to effect a process that utilises information from a conversation map to manage communications with an end customer. The conversation
15 engine 2120 continually monitors and acts on events, based on the objects and links defined in the conversation map. Thus, the conversation engine 2120 retrieves information from a conversation map 2130 that has been activated and interprets current events, such as the receipt of an inbound request 1126 or transmission of an outbound communication
1 128, to determine when to activate an outbound communication, how to respond to any
20 inbound communications, and also when to proceed to a next object in the conversation map. The conversation engine 2120 is able to manage multiple active conversation maps.

The contact library 2150 is a data store of information for contacts that have been added to the call mapping system 2100. The call mapping system 1100 utilises the information stored in the contact library 2150 to communicate with the contacts stored
25 therein. Depending on the implementation, contacts may be classified into different types, such as customer, potential customer, service provider, and merchant manager, wherein different information is stored for each type of contact. In one example, each contact is stored as a record with one or more fields. For a customer, the customer contact record includes fields for recording each of a name, address, telephone number, email address,
30 and customer number. For a merchant manager, the merchant manager contact record includes fields for recording each of a name, company number, mailing address, billing address, telephone number, email address, and merchant number. It will be appreciated by

- 51 -

a person skilled in the art that the actual type of information recorded for each contact will depend on the particular implementation and changes to the number and types of fields may be practised without departing from the spirit and scope of the present disclosure.

FIG. 23 is a flow diagram illustrating a conversation mapping method 2300. In one embodiment, the conversation mapping method 2300 utilises a conversation map application 2130 implemented as software instructions forming part of a management application 150 of a managing core 101, wherein the software instructions execute on a processing unit 1005 of a computing system 1000 implementing the communication management system 100. The conversation mapping method 2300 begins at a Start step 2305 and proceeds to step 2310, in which a merchant manager 1650 utilises a computing device 1652 to provide instructions via one of the external API gateway 2170 and the conversation management interface 2180 to the conversation map application 2130. The merchant manager 1650 utilises the conversation map application 2130 to create a call conversation map by allocating a plurality of objects and connecting links. The call conversation map defines a call handling process for a particular call scenario for that merchant. The call conversation map may relate, for example, to account enquiries and payments, promotional offers, or marketing campaigns. Control passes from step 2310 to step 2320, which stores the call conversation map in a data store 1170 of the communications management system 1100. Step 2330 activates a stored call conversation map, based upon a predefined trigger from the conversation triggers module 2140. One or more triggers may be associated with a call conversation map. In one example, a trigger is based on a calendar entry or time of day, wherein activation of a call conversation map is scheduled. In another example, a trigger is based on receiving an incoming communication matching at least one predefined criterion. The at least one predefined criterion may relate, for example, to an inbound calling number, a dialed number, a source identifier, a range of times or dates, or any combination thereof. The source identifier may be, for example, a calling number or an Internet Protocol (IP) address.

Control passes from step 2330 to step 2340, in which a conversation engine 2120 monitors and handles incoming and outgoing communications in accordance with the activated call conversation map. The conversation engine 2120 utilises a Task Memory stored in the data store 1170 to record information relating to each object. This allows the conversation mapping system to utilise information from previously executed objects

- 52 -

within the same conversation map or from other conversation maps. Control passes to decision step 2350, which determines whether the call conversation map has been completed. If the call conversation map has reached an end of a branch, Yes, control passes to an End step 2360 and the method 2300 terminates. The call conversation map is
5 then marked as inactive. If at step 2350 the map has not reached an end of a branch and further actions are expected in the call conversation map, No, control returns from step 2350 to step 2340 to monitor and handle further incoming and outgoing communications.

One implementation of the call conversation mapping method and system utilises Extensible Markup Language (XML) to define a call conversation map. Objects
10 and links within the call conversation map are defined separately and the call handling workflow is defined as an XML file. One embodiment optionally provides a second file to define display properties associated with the call conversation map. The second file includes parameters relating to the layout properties of the objects and links.

FIG. 20 is a screenshot of a call conversation map 2000 created using a
15 graphical user interface. The graphical user interface may be implemented using text and graphics presented to a computing device 1652 accessed by a merchant manager 1650.

The computing device may be implemented using a computing device 1000, whereby the graphical user interface is displayed on a video display unit 1014 and the merchant manager provides input via one or more of the keyboard 1002, microphone 1080,
20 and mouse pointing device 1003. The conversation map 2000 is stored in call architecture 1175 of a merchant and establishes a call handling process for that merchant. In one arrangement, the merchant creates or modifies a conversation map by utilising the mouse 1003 or keyboard 1002 to position one or more objects on a window displayed in the video display 1014 and connect those objects in a desired workflow by connecting the objects
25 with links. The links are utilised by the conversation engine 2120 to determine a next step in a conversation map from a current object. Each link can utilise one or more of an outcome, time schedule, or task expiry to determine a workflow within the call conversation map. The merchant manager 1650 provides information relating to each object, wherein objects relating to the handling of incoming or outgoing communications
30 are denoted as tasks and each conversation map includes at least one task. In one embodiment, different objects are displayed in different colours to provide a visual distinction in the conversation map. For example, all tasks are displayed in green to

- 53 -

indicate a communication action to be performed and objects relating to payment are displayed in orange to indicate an accounting transaction. One embodiment provides a task master template for grouping together objects that relate to a particular phase of a call handling process. Such a task master template can be stored either locally on the
5 computing device 1652 of the merchant manager 1650 or on the data store 1170 of the communication management system 1100 for retrieval at a later time. Task master templates facilitate the later creation or modification of the same or other conversation maps in which the same or similar functionality is required.

In the example of FIG. 20, the conversation map 2000 relates to collection of
10 payments for customer accounts that are overdue by less than 15 days. The call handling process 2000 begins at an initial object 2005, which is a task for sending an outbound group SMS message from the communication management system 1100 to a list of selected customers having overdue accounts, wherein those selected customers have mobile telephone contact numbers suitable for receiving SMS messages and the contact
15 details are stored in the contact library 2050. In this example, the SMS message has the content *"Payment Reminder. Arrears less than 15 days. Reply PAY to pay now. Reply SERVICE to speak to customer service"*. When the conversation map 2000 is triggered, the communication management system 1100 generates an outbound SMS message to each of the selected customers 1640 via one or more of the nodes 1108, 1120.

20 The conversation map 2000 defines two possible responses to the sending of the group SMS message: (i) the customer 1640 receiving the outbound SMS message sends an incoming SMS message with "PAY" in the content; or (ii) the customer 1640 receiving the outbound SMS message sends an incoming SMS message with "SERVICE" in the content.

25 If the customer 1640 sends an incoming SMS message with "SERVICE" in the content, the workflow of the call conversation map 2000 passes from object 2005 to object 2010, which is a task for establishing a call from an agent 1610 of the merchant to the customer 1640. The call may be implemented using an exchange of further SMS messages, a web chat session, a video call, or a voice call, depending on the
30 implementation of the system and the capabilities of both the agent 1610 and the customer 1640. Object 2010 links to a further object 2015, which is a task indicating that the call has been answered by the customer 1640, thus confirming the establishment of a call

- 54 -

between the agent 1610 and the customer 1640. The customer 1640 is then able to query the account with the agent 1610, discuss payment options, or provide payment details.

Returning to object 2005, if the customer 1640 sends an incoming SMS message with "PAY" in the content, the workflow of the call conversation map 2000
5 passes from object 2005 to object 2010, which is a task for establishing a call from a node 1108, 1120 of the communication management system 1100 to the customer 1640. In this example, the call is generated by a node 1108, 1120 equipped with IVR capabilities. Object 2010 links to a further object 2025, which indicates that when the call has been answered by the customer 1640, the node 1108, 1120 is to connect the IVR to play a
10 recorded message introducing the customer 1640 to the payment system. The IVR may utilise dual tone multi-frequency (DTMF) tones to interact with the keypad of a telephone being utilised by the customer 1640. This enables the customer 1640 to navigate a menu hosted by the IVR. Object 2025 links to a further object 2030 that records payment details provided by the customer 1640. In this example, object 2030 records digits of a credit
15 card. Object 2030 links to another object 2045 that records digits of an expiry date of the credit card recorded in object 2030.

Object 2045 links to object 2050, which validates the credit card. In this example, object 2050 is linked to a payment facility, such as that provided by PayPal™. There are two possible results from the validation performed in object 2050: (i) the
20 payment is successful; or (ii) the payment fails. Accordingly, there are two links branching from object 2050. If the payment fails at object 2050, a link connects object 2050 to object 2035, which prompts the customer to either try again, by pressing "1" on the keypad, or be connected to customer service, by pressing "2" on the keypad. If the customer 1640 presses "1", the conversation map 2000 proceeds from object 2035 via a link to return to
25 object 2030, in which the customer 1640 again provides payment details. If at object 2035 the customer 1640 presses "2", the conversation map 2000 proceeds from object 2035 via a link to object 2040. Object 2040 connects the call to an agent 1610.

Returning to object 2050, if the payment is successful, a link connects object 2050 to object 2055, which engages the IVR to play a recorded message thanking the
30 customer 1640 for the payment and reciting a receipt number for the payment. Object 2055 links to object 2060, which is a task for sending a single SMS message to customer 1640, wherein the SMS message again thanks the customer 1640 for the payment received

- 55 -

in object 2050 and provides the receipt number for that payment.

In the example of FIG. 20, the merchant manager groups objects in a plurality of task master templates. In particular, task master template A 2070 includes object 2005 relating to the task of generating an outbound SMS message. Task master template B 2072 includes objects 2010 and 2015 relating to that part of the conversation map 2000 that arises if the customer 1640 replies to the group SMS message with an SMS message containing the word "SERVICE". Task master template C 2074 includes objects 2020, 2025, 2030, and 2045 relating to the acquisition of credit card details to effect payment. Task master template D 2076 includes objects 2035 and 2040 relating to a failed attempt to pay the account. The task master templates are optional and it is not essential for all objects to belong to a task master template. In the example of FIG. 20, objects 2050, 2055, and 2060 do not belong to a task master template.

FIG. 22 is a schematic block diagram representation illustrating an example of the contact library 2150. The contact library 2150 includes one or more contact lists. Each contact list may correspond, for example, to customers, suppliers, geographical locations, services, merchant managers, marketing campaigns, or any combination thereof. In the example of FIG. 22, the contact library includes a plurality of contact lists: Contact List A 2052, ..., Contact List "n" 2058. In this example, Contact List A 2052 stores customer contacts for Merchant Manager 1650 and Contact List "n" 2058 stores agent contacts for Merchant Manager 1650. Contact List A 2052 includes Contact Fields 2053, which defines the types of fields in each record for the contacts in Contact List A 2052. In this example, Contact Fields 2053 defines fields for storing name, address, telephone number, email address, and account number. Contact List A 2052 also includes Contacts 2054, which is a list of the customer contacts for Merchant Manager 1650. Contacts can be added individually or through a bulk importation process. A bulk importation of contacts may be performed, for example, by the merchant manager 1650 utilising the external gateway 2170 to upload an existing database of contacts or by the merchant manager 1650 utilising the conversation management interface 2180 to access a web interface to enter a new contact.

The Contacts 2054 form at least one contact pool, based on at least one matching criterion. In the simple case, all of the contacts in Contacts 2054 form a contact pool. Merchant manager 1650 can apply a contact pool filter to the Contacts 2054 to form

- 56 -

one or more further contact pools, based on the parameters of the contact pool filter. A contact pool filter can include one or more rules. Multiple contact pool filters can be applied to generate multiple contact pools. Contact pools are used to group contacts that match the contact pool filter. This is useful for organizing a marketing campaign using a filtered contact pool. Different contact pools can overlap. In the example of FIG. 22, 5 Contacts 2054 includes Contact Pool A 2055, Contact Pool B 2056, and Contact Pool C 2057. Some of the contacts in Contacts 2054 are common to each of Contact Pool A 2055, Contact Pool B 2056, and Contact Pool C 2057. There are also contacts in Contact Pool A 2055 that are common to Contact Pool C 2057, but not to Contact Pool B 2056. Similarly, 10 there are contacts in Contact Pool A 2055 that are common to Contact Pool B 2056, but not to Contact Pool C 2057. Further, there are contacts in Contact Pool B 2056 that are common to Contact Pool C 2057, but not to Contact Pool C 2055.

Contact List "n" 2058 includes Contact Fields 2059, which defines the types of fields in each record for the contacts in Contact List "n" 2058. In this example, Contact 15 Fields 2059 defines fields for storing name, address, telephone number, email address, and agent number for each agent associated with merchant manager 1650. Contact List "n" 2058 also includes Contacts 2060, which is a list of the agent contacts for Merchant Manager 1650. The Contacts 2060 includes Contact Pool D 2061, Contact Pool E 2062, and Contact Pool F 2063, wherein there is some overlap between the contacts in Contact 20 Pool D 2061 and Contact Pool E 2062.

When a conversation map is activated by a conversation engine 2120 in response to a trigger, the conversation engine 2120 migrates tasks within the conversation map from the conversation map into an Active Task Template. The Active Task Template includes task templates, which are the actual tasks to be actioned by the conversation 25 engine 2120. The task templates are the individual steps that require an outbound communication to be triggered. In one example, a task template includes information relating to an incoming or outgoing communication channel that is to be used, the content that is to be sent or received, and the source or destination of the task. A Task Scheduler is responsible for monitoring all the task templates. When a task template is triggered to be 30 actioned, that task template is migrated to a Task Online area to be actioned. The Task Online area is a staging area where the actual communication is performed. When a task is online, that task is marked as being currently active and is referred to as an online task.

- 57 -

An online task has five phases: scheduled, allocated, started, waiting, and completed. An online task that is "scheduled" is active and waiting to be allocated. An online task is "allocated" when the task is waiting to be allocated to a resource, such as a node 1108, 1120. When an online task is allocated to a node 1108, 1120 in response to a heartbeat message from the node 1108, 1120, the online task changes to a "started" phase. A "started" phase indicates that an initial outbound instruction 1128 has been sent from the managing core 1130 to a node 1108, 1120 and the managing core 1130 is waiting for a response from the node 1108, 1120. When the managing core 1130 receives a response from the node 1108, 1120, the online task enters a "waiting" phase, in which the conversation engine 2120 executing on the managing core 1130 waits for notification from the node 1108, 1120 that the task has been completed. Once a task has been completed, the online task is marked as "completed" and the task is moved from the Online Task area and is stored in an archived history. Such a history may be recorded, for example, on the data store 1170 of the managing core 1130, and form part of a conversation record for that instance of the call conversation map 2000. The conversation records can be used for later analysis of the call handling process and for managing subsequent objects within the call conversation map 2000. An initial phase of an online task is to create an outbound communication from the conversation mapping system 2100 to a designated contact. The designated contact may be a customer, a merchant manager, or other contact. This step uses Router Control to allocate a specific outbound communications router.

In one arrangement, Router Control is a function of the Node Application 125, 1112 within a node 1108, 1120 of the system 1100. The Router Control selects a service provider for a required service. In one example, an outbound communication is a voice call and an allocated node has three outgoing voice channels, each channel corresponding to a different service provider. Router control selects one of the outgoing voice channels on which to send the outbound communication. Router Control may be implemented as a manual process in which an administrator allocates outbound communications among the plurality of communication channels. In an alternative implementation, Router Control is an automated process that allocates outbound communications to channels. One embodiment assigns a priority to each router and utilises the priorities when allocating communications. Once a specific outbound communications router has been allocated, the task will be available to be allocated in response to a heartbeat request from one of the

- 58 -

distributed nodes 1108, 1120. A node 1108, 1120 will then actually perform the communication by generating the required outbound communication, such as a phone call, SMS message, email, etc. The communication is sent out via the media gateways.

FIG. 25 is a schematic block diagram representation illustrating activation of tasks from the conversation map of FIG. 20. Fig. 25 shows the conversation map 2000 of FIG. 20, which includes a first task 2060 relating to the sending of a Group SMS message. The task 2060 includes a set of task properties that define the actions to be performed in relation to task 2060. In this example, the task 2060 includes task properties indicating that the Group SMS message is to be sent to all of the contacts in Contact List A 2052 at a rate of 500 calls per minute. The conversation map 2000 is associated with one or more triggers in a triggers module 2140, wherein the triggers define when the conversation map 2000 becomes active.

When a trigger activates the conversation map 2000, the conversation engine 2120 migrates the tasks of the conversation map 2000 to corresponding task templates in an Active Task Template 2520. In the example of FIG. 25, task 2060 migrates to Task Template A 2522. Other tasks in the conversation map 2000 migrate to Task Template B 2524, ..., Task Template n 2526. Task Template A 2522 includes the set of task properties associated with task 2060. A Task Scheduler 2530 forms part of the conversation engine 2120 and manages the task templates in the Active Task Template 2520. In particular, the Task Scheduler monitors events associated with the call conversation map and determines when to activate a next task template in the Active Task Template 2520.

When a task template is triggered to be actioned, as a result of a predefined trigger or the completion of a related event in the conversation map 2000, the Task Scheduler 2530 migrates the task template to a task in a Task Online area 2540. The Task Scheduler 2530 populates the task with all of the information required to perform that task. Depending on the task, the Task Scheduler may retrieve information from the contacts library 2150 to populate a destination for an outbound communication from the system 1100. In this example, the Task Scheduler 2530 migrates Task Template A 2522 to the Task Online area 2540 as an online task 2542 and the online task 2542 is in the "scheduled" phase. As indicated above, the online task 2542 includes all of the information to perform the task. In this example, the online 2542 includes information

- 59 -

indicating that the task to be performed is the sending of an SMS message to Customer C 1640 with content "Payment overdue. Reply 'PAY' to pay account or 'SERVICE' to speak with an agent".

The Active Task Template 2520, Task Scheduler 2530 and Task Online Area 2540 combine to perform the logical functionality of the Outbound Job Manager 120. As described above, the nodes 1108, 1120 send heartbeat messages to the managing core 1130 asking for online tasks that are to be performed. The managing core 1130 allocates the online tasks from the Task Online area 2540 to the nodes 1108, 1120 in response to the heartbeat messages. The task 2542 then enters the "allocated" phase. When the task 2542 is performed by the allocated node, the online task 2542 becomes "started" and then progresses to "waiting" until a timer expires or part of the online task 2542 completes. Once the timer expires or all aspects of the online task 2542 have been performed, the online task 2542 becomes "completed". Completed online tasks are removed from the Task Online area 2540.

FIG. 24 is a flow diagram illustrating a conversation mapping method 2400 utilising the example of FIG. 20 and the communication management system 1600 of FIG. 16, which is embodied using an instance of the system 1100 of FIG. 11. In one implementation, a merchant manager 1650 first logs into the system 1100 through a login process running on the node application 126 of a node 1108, 1120. The conversation mapping method 2400 begins at a Start step 2405 and proceeds to step 2410 in which the merchant manager 1650 utilises a computing device 1652 to communicate with the communication management system 1100. The merchant manager 1650 utilises a keyboard 1002, mouse pointer device 1003, or other input device to provide input to the computing device 1652, implemented using a computing system 1000 as described with reference to FIGs 10A and 10B. The computing device 1652 exchanges communications with the communication system 1100 via the telecommunications network 1660. The communication system 1100 includes a conversation map application 2130 and conversation engine 2120 as part of a management application 150 in the managing core 1130. In one arrangement, the managing core 1130 is implemented using a computing system 1000 as described with reference to FIGs 10A and 10B and the conversation map application 2130 and conversation engine 2120 include a set of software instructions executing on a processing unit 1005.

- 60 -

In this example, the merchant manager 1652 utilises the conversation map application 2130 to create a call conversation map in step 2410 corresponding to the call conversation map 2000 of FIG. 20, which relates to a call handling process for communications relating to customers having accounts that are overdue by less than 15 days. The call conversation map 2000 is stored in the data store 1170. The method 2400 proceeds from step 2410 to step 2415, in which the merchant manager 1650 defines a trigger associated with the conversation call map 2000. In this example, the trigger is 9am on the first Monday of each month. Step 2415 also stores the trigger in the data store 1170 as part of the conversation triggers module 2140.

Control passes from step 2415 to step 2420, in which the conversation engine 2120 activates the call conversation map 2000, based on the trigger. Thus, at 9am on the first Monday of the next month, the conversation engine 2120 activates the call conversation map 2000. In step 2425, the conversation engine 2120 migrates tasks from the call conversation map 2000 to an Active Task Template 2520, which includes a set of task templates 2522, ..., 2526. The task templates 2522, ..., 2526 are the tasks that are to be actioned in respect of the call conversation map 2000. A Task Scheduler 2530 within the conversation engine 2120 manages the task templates 2522, ..., 2526. Control passes from step 2425 to step 2430, in which the Task Scheduler 2530 migrates the task templates to online tasks in a Task Online area 2540 and allocates each task template to an outbound communications router, being an outgoing communication channel on a node 1108, 1120. In migrating the task templates to online tasks, the Task Scheduler 2530 retrieves from the contact library 2050 a set of contacts to whom the group SMS message is to be sent. Depending on the implementation, the set of contacts having overdue accounts may be stored in the contact library 2050 as a separate contact list 2052, 2058. Alternatively, the Task Scheduler 2530 applies a contact pool filter to Contacts 2054 to retrieve the set of contacts having overdue accounts, wherein the pool filter includes parameters that identify customers of merchant manager 1650 that have accounts that are overdue by less than 15 days. The task templates are now arranged as online tasks in a pipeline and are ready to be allocated to nodes 1108, 1120 in response to heartbeat communications from the nodes 1108, 1120 that seek jobs to perform.

Control passes from step 2430 to step 2435, which receives a heartbeat request from a node 1108, 1120. The initial object in the call conversation map 2000 is the task of

- 61 -

sending a group SMS message to each customer having an overdue account.

In step 2440, the conversation engine 2120 assigns the initial task 2005 in the call conversation map 2000 to one or more of the nodes 1108, 1120 by sending outbound instructions 1128 from the managing core 1130 to one or more of the plurality of nodes 1108a, ..., 1108n and 1120a, ..., 1120n, and those nodes send outgoing SMS messages via the conversation gateway 103 and the plurality of communication channels 1109 to the set of contacts. The group SMS messages may be implemented by utilising a bank of telephone numbers. As the conversation engine 2120 knows which outgoing SMS messages relate to which of the telephone numbers in the bank of numbers, the conversation map application matches incoming and outgoing communications to one or more active call conversation maps by utilising the telephone numbers that appear in calling numbers, called numbers, or signaling information.

The next steps in the call conversation map 2000 branch from object 2005, depending on whether a customer replies to the group SMS message with an SMS message containing the content "PAY" or "SERVICE". The conversation engine 2120 monitors incoming communications received after the Group SMS message has been sent and determines whether the incoming communications relate to an existing conversation. Control passes from step 2440 to step 2442, in which a customer 1640 receives the group SMS message and utilises an SMS engine 1642, which in this example is implemented as a mobile telephone handset, to send a reply SMS message. The reply SMS message is received by the system 1100 as an incoming message and is allocated to one of the nodes 1108, 1120. The node 1108, 1120 sends an incoming request 1126 to the managing core 1130 seeking instructions for handling the incoming SMS message. Control passes to step 2445, in which the conversation engine 2120 monitors incoming communications relating to this call conversation map and identifies the incoming request 1126 as relating to the call conversation map 2000. Control passes to decision step 2450, which determines whether the incoming message includes the content "PAY" or "SERVICE".

If the incoming message includes the content "SERVICE", control passes from step 2450 to step 2455, which corresponds to object 2010 of the call conversation map 2000, in which the conversation engine 2120 generates an outgoing call from an agent associated with the merchant manager 1650 to the customer 1640. The conversation engine 2120 sends an outbound instruction 1128 to the node 1108, 1120 that handled the

- 62 -

incoming SMS message, wherein the outbound instruction 1128 instructs the node to establish a voice link with an agent 1610 of the merchant manager. The instruction 1128 may inform the node 1108, 1120 to effect the connection to the agent 1610 directly or to forward instructions to another one of the plurality of nodes 1108, 1120. The managing
5 core utilises the nodes to establish a voice connection between the customer 1640 and the agent 1610, in the manner described above with reference to FIG. 12. The customer is connected to the agent to discuss the outstanding account balance, corresponding to object 2015 of the call conversation map 2000. Control passes from step 2455 to step 2460 and the method 2400 terminates.

10 If the incoming message from the customer 1640 includes the content "PAY", control passes from step 2450 to step 2465, which corresponds to object 2020 of the call conversation map 2000, in which the conversation engine 2120 generates an outgoing call to the customer 1640. The conversation engine 2120 sends an outbound instruction 1128 to the node 1108, 1120 that handled the incoming SMS message, wherein the outbound
15 instruction 1128 instructs the node to establish a voice link with the customer 1640. The instruction 1128 may inform the node 1108, 1120 to effect the connection to the customer 1640 directly or to forward instructions to another one of the plurality of nodes 1108, 1120. Control passes from step 2465 to step 2470, in which the managing core 1130 utilises the nodes to establish a voice connection between the customer 1640 and then utilises an IVR
20 coupled to a node 1108, 1102 to play a recorded announcement welcoming the customer 1640 to the payment system, corresponding to object 2025 of the call conversation map 2000. Control passes from step 2470 to step 2475, in which the customer 1640 utilises a keypad of the mobile telephone handset to enter credit card details, corresponding to object 2035. In step 2475, the customer 1640 utilises the keypad of the mobile telephone handset
25 to enter credit card expiry details, corresponding to object 2040.

Control passes to step 2485, which processes the credit card payment, corresponding to object 2050. Control then passes to decision step 2490, which determines whether or not the payment is successful. If the payment is successful, YES, control passes to step 2495, in which the conversation engine 2120 instructs a node to play a
30 recorded announcement from the IVR thanking the customer 1640 for the payment and advising the customer 1640 of the receipt number, corresponding to object 2055. Control passes to step 2496, in which the conversation engine 2120 instructs a node 1108, 1120 to

- 63 -

send an SMS message to the customer 1640 confirming payment and again providing the receipt number, corresponding to object 2060. Control passes from step 2496 to step 2460 and the method 2400 terminates.

If at step 2490 the payment is not successful, NO, control passes to step 2497, in which the conversation engine 2120 instructs a node 1108, 1120 to play a recorded message using the IVR, corresponding to object 2035. The recorded message provides the customer 1640 with the option of pressing "1" to re-attempt the payment process and pressing "2" to speak to an agent 1610. Control passes to decision step 2498, which determines whether the customer 1640 replies with "1" or "2", corresponding to whether the customer wants to retry the payment process or be connected to a customer service agent 1610. If the customer presses "1", indicating that the customer 1640 wants to retry the payment process, control passes from step 2498 and returns to payment step 2475. If the customer presses "2", indicating that the customer 1640 wants to speak with an agent, control passes from step 2498 to step 2499, in which the conversation engine 2120 sends outbound instructions 1128 to a node to establish a call to an agent 1610 and subsequently to establish a voice call between the agent 1610 and the customer 1640, corresponding to object 2040. Having established the voice call between the agent 1610 and the customer 1640, control passes from step 2499 to End step 2460 and the method 2400 terminates.

CONCLUSIONS AND INTERPRETATION

It will be appreciated that the disclosure above provides various significant systems and methods for managing communication data. For example, the present embodiments allow for a distributed system architecture that allows for high levels of scalability and adaptability.

Unless specifically stated otherwise, as apparent from the following discussions, it is appreciated that throughout the specification discussions utilising terms such as "processing", "computing", "calculating", "determining", "analysing", or the like, refer to the action and/or processes of a computer or computing system, or similar electronic computing device, that manipulate and/or transform data represented as physical, such as electronic, quantities into other data similarly represented as physical quantities.

In a similar manner, the term "processor" may refer to any device or portion of a device that processes electronic data, e.g., from registers and/or memory to transform that

- 64 -

electronic data into other electronic data that, e.g., may be stored in registers and/or memory. A "computer" or a "computing machine" or a "computing platform" may include one or more processors.

The methodologies described herein are, in one embodiment, performable by
5 one or more processors that accept computer-readable (also called machine-readable) code containing a set of instructions that when executed by one or more of the processors carry out at least one of the methods described herein. Any processor capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken are included. Thus, one example is a typical processing system that includes one or more processors.
10 Each processor may include one or more of a CPU, a graphics processing unit, and a programmable DSP unit. The processing system further may include a memory subsystem including main RAM and/or a static RAM, and/or ROM. A bus subsystem may be included for communicating between the components. The processing system further may be a distributed processing system with processors coupled by a network. If the processing
15 system requires a display, such a display may be included, e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT) display. If manual data entry is required, the processing system also includes an input device such as one or more of an alphanumeric input unit such as a keyboard, a pointing control device such as a mouse, and so forth. The term memory unit as used herein, if clear from the context and unless explicitly stated
20 otherwise, also encompasses a storage system such as a disk drive unit. The processing system in some configurations may include a sound output device, and a network interface device. The memory subsystem thus includes a computer-readable carrier medium that carries computer-readable code (e.g., software) including a set of instructions to cause performing, when executed by one or more processors, one of more of the methods
25 described herein. Note that when the method includes several elements, e.g., several steps, no ordering of such elements is implied, unless specifically stated. The software may reside in the hard disk, or may also reside, completely or at least partially, within the RAM and/or within the processor during execution thereof by the computer system. Thus, the memory and the processor also constitute computer-readable carrier medium carrying
30 computer-readable code.

Furthermore, a computer-readable carrier medium may form, or be included in a computer program product.

- 65 -

In alternative embodiments, the one or more processors operate as a standalone device or may be connected, e.g., networked to other processor(s), in a networked deployment, the one or more processors may operate in the capacity of a server or a user machine in server-user network environment, or as a peer machine in a peer-to-peer or distributed network environment. The one or more processors may form a personal computer (PC), a tablet PC, a set-top box (STB), a Personal Digital Assistant (PDA), a cellular telephone, a web appliance, a network router, switch or bridge, or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine.

Note that while some diagrams only show a single processor and a single memory that carries the computer-readable code, those in the art will understand that many of the components described above are included, but not explicitly shown or described in order not to obscure the inventive aspect. For example, while only a single machine is illustrated, the term "machine" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any or more of the methodologies discussed herein.

Thus, one embodiment of each of the methods described herein is in the form of a computer-readable carrier medium carrying a set of instructions, e.g., a computer program that is for execution on one or more processors, e.g., one or more processors that are part of web server arrangement. Thus, as will be appreciated by those skilled in the art, embodiments of the present invention may be embodied as a method, an apparatus such as a special purpose apparatus, an apparatus such as a data processing system, or a computer-readable carrier medium, e.g., a computer program product. The computer-readable carrier medium carries computer readable code including a set of instructions that when executed on one or more processors cause the processor or processors to implement a method. Accordingly, aspects of the present invention may take the form of a method, an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, the present invention may take the form of carrier medium (e.g., a computer program product on a computer-readable storage medium) carrying computer-readable program code embodied in the medium.

The software may further be transmitted or received over a network via a network interface device. While the carrier medium is shown in an exemplary

- 66 -

embodiment to be a single medium, the term "carrier medium" should be taken to include a single medium or multiple media (e.g., a centralised or distributed database, and/or associated caches and servers) that store the one or more sets of instructions. The term "carrier medium" shall also be taken to include any medium that is capable of storing, encoding or carrying a set of instructions for execution by one or more of the processors and that cause the one or more processors to perform anyone or more of the methodologies of the present invention. A carrier medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical, magnetic disks, and magneto[^]optical disks. Volatile media includes dynamic memory, such as main memory. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise a bus subsystem. Transmission media also may also take the form of acoustic or light waves, such as those generated during radio wave and infrared data communications. For example, the term "carrier medium" shall accordingly be taken to include, but not be limited to, solid-state memories, a computer product embodied in optical and magnetic media; a medium bearing a propagated signal detectable by at least one processor of one or more processors and representing a set of instructions that, when executed, implement a method; a carrier wave bearing a propagated signal detectable by at least one processor of the one or more processors and representing the set of instructions a propagated signal and representing the set of instructions; and a transmission medium in a network bearing a propagated signal detectable by at least one processor of the one or more processors and representing the set of instructions.

It will be understood that the steps of methods discussed are performed in one embodiment by an appropriate processor (or processors) of a processing (i.e., computer) system executing instructions (computer-readable code) stored in storage. It will also be understood that the invention is not limited to any particular implementation or programming technique and that the invention may be implemented using any appropriate techniques for implementing the functionality described herein. The invention is not limited to any particular programming language or operating system.

Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present

- 67 -

invention. Thus, appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment, but may. Furthermore, the particular features, structures or characteristics may be combined in any suitable manner, as would be apparent to one of ordinary skill in the art from this disclosure, in one or more embodiments.

Similarly it should be appreciated that in the above description of exemplary embodiments of the invention, various features of the invention are sometimes grouped together in a single embodiment, FIG., or description thereof for the purpose of streamlining the disclosure and aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims following the Detailed Description are hereby expressly incorporated into this Detailed Description, with each claim standing on its own as a separate embodiment of this invention.

Furthermore, while some embodiments described herein include some but not other features included in other embodiments, combinations of features of different embodiments are meant to be within the scope of the invention, and form different embodiments, as would be understood by those skilled in the art. For example, in the following claims, any of the claimed embodiments can be used in any combination.

Furthermore, some of the embodiments are described herein as a method or combination of elements of a method that can be implemented by a processor of a computer system or by other means of carrying out the function. Thus, a processor with the necessary instructions for carrying out such a method or element of a method forms a means for carrying out the method or element of a method. Furthermore, an element described herein of an apparatus embodiment is an example of a means for carrying out the function performed by the element for the purpose of carrying out the invention.

In the description provided herein, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practised without these specific details. In other instances, well-known methods, structures and techniques have not been shown in detail in order not to obscure an understanding of this description.

Similarly, it is to be noticed that the term coupled, when used in the claims,

- 68 -

should not be interpreted as being limited to direct connections only. The terms "coupled" and "connected," along with their derivatives, may be used. It should be understood that these terms are not intended as synonyms for each other. Thus, the scope of the expression a device A coupled to a device B should not be limited to devices or systems wherein an output of device A is directly connected to an input of device B. It means that there exists a path between an output of A and an input of B which may be a path including other devices or means. "Coupled" may mean that two or more elements are either in direct physical or electrical contact, or that two or more elements are not in direct contact with each other but yet still co-operate or interact with each other.

Thus, while there has been described what are believed to be the preferred embodiments of the invention, those skilled in the art will recognise that other and further modifications may be made thereto without departing from the spirit of the invention, and it is intended to claim all such changes and modifications as falling within the scope of the invention. For example, any formulas given above are merely representative of procedures that may be used. Functionality may be added or deleted from the block diagrams and operations may be interchanged among functional blocks. Steps may be added or deleted to methods described within the scope of the present invention.

INDUSTRIAL APPLICABILITY

The arrangements described are applicable to the computer and data processing industries and particularly for the telecommunications, call centre, and data management industries.

The foregoing describes only some embodiments of the present invention, and modifications and/or changes can be made thereto without departing from the scope and spirit of the invention, the embodiments being illustrative and not restrictive.

- 69 -

Claims:

1. A call conversation mapping method, comprising the steps of:
creating a call conversation map that includes a plurality of objects and at least one
connecting link to define a call handling process, wherein said objects include at least one
5 task relating to receiving an incoming communication or generating an outgoing
communication;
activating said call conversation map in response to a trigger; and
processing communications in accordance with the call handling process of said
call conversation map.
10
2. The call conversation mapping method according to claim 1, wherein said
processing comprises the further steps of:
sending an outgoing communication, based on a current object in said call handling
process; and
15 utilising said outgoing communication and a link from said current object to
identify a new current object in said call handling process.
3. The call conversation mapping method according to claim 1, wherein said
processing comprises the further steps of:
20 receiving an incoming communication;
mapping said incoming communication to a current object in said call handling
process; and
identifying a further object connected to said current object by one of said links;
performing an action associated with said further object; and
25 utilising said action and a link from said current object to identify a new current
object in said call handling process.
4. The call conversation mapping method according to claim 3, wherein said mapping
is based on at least one of a called number, a calling number, and signaling information
30 associated with the incoming communication.

- 70 -

5. The call conversation mapping method according to any one of claims 1 to 4, comprising the further step of:

storing in a history file information relating to an object in said call handling process, once an action associated with said object has been completed.

5

6. The call conversation mapping method according to any one of claims 1 to 5, wherein creating said call conversation map includes the steps of:

positioning said objects relative to one another; and

placing said links to establish connections between selected ones of said objects,

10 said placement of said links establishing a flow among said objects to define said call handling process.

7. The call conversation mapping method according to claim 6, wherein said call conversation map is represented graphically and positioning of said objects and placement
15 of said links is performed using a graphical user interface.

8. The call conversation mapping method according to any one of claims 1 to 7, wherein said trigger is one of a timing event, an incoming communication, or a scheduled event.

20

9. The call conversation mapping method according to any one of claims 1 to 8, wherein said call conversation map includes at least one task master template, each task master template including at least of said plurality of objects to define a reproducible portion of said call conversation map.

25

10. The call conversation mapping method according to any one of claims 1 to 8, wherein activating said call conversation map includes the steps of:

migrating tasks from said call conversation map to an active task template that includes at least one task template;

30

allocating a task template to an outbound communication router; and

- 71 -

distributing said task template to one of a plurality of distributed nodes, each node including at least one incoming communication channel and an outgoing communication channel.

- 5 11. The call conversation mapping method according to claim 10, wherein processing communications includes the step of:

executing said task template by said node to initiate an outgoing communication.

12. The call conversation mapping method according to any one of claims 1 to 11,
10 wherein processing communications utilises a distributed management communication system comprising:

a plurality of nodes, each node having a plurality of call channels, the call channels being connectable to at least one communications network and including at least one incoming voice channel and at least an outgoing voice channel; and

- 15 a managing core coupled to each of the nodes, the core comprising a data store for storing said call conversation map as a call architecture of a merchant, said call conversation map defining a call handling process for the merchant, the call handling process being operable to detect receipt by a first said node of an incoming voice call on a corresponding incoming voice channel, and to couple the voice call from said first node via
20 one of said call channels thereof to a second said node for connection to an agent associated with the merchant.

13. A call conversation mapping system comprising:

- 25 a conversation map for creating a call conversation map that includes a plurality of objects and at least one connecting link to define a call handling process, wherein said objects include at least one task relating to receiving an incoming communication or generating an outgoing communication;

a data store for storing a call conversation map;

- 30 a conversation triggers module defining a trigger for activating said stored call conversation map; and

- 72 -

a conversation engine for activating said stored call conversation map in response to said trigger and processing communications in accordance with the call handling process of said stored call conversation map.

5 14. The call conversation mapping system according to claim 13, further comprising:
a contacts library including at least one contact associated with said stored call conversation map.

10 15. The call conversation mapping system according to claim 14, wherein said contacts library includes at least one contact list, each contact list including contacts and contacts fields, said contacts fields defining at least one field of information to be stored in respect of each contact.

15 16. The call conversation mapping system according to any one of claims 13 to 15, further comprising:
a conversation management application programming interface for receiving instructions for said conversation map to create said stored call conversation map.

20 17. The call conversation mapping system according to any one of claims 13 to 16, further comprising:
a media gateway for managing communications with an external computing device.

25 18. The call conversation mapping system according to claim 17, wherein said external computing device is selected from the group consisting of: a general purpose computer, a mobile telephone handset, an SMS engine, a smartphone.

30 19. The call conversation mapping system according to claim 17, wherein said media gateway includes at least one node, each node having a plurality of call channels, the call channels being connectable to at least one communications network and including at least one incoming voice channel and at least an outgoing voice channel.

- 73 -

20. The call conversation mapping system according to claim 19, wherein said conversation map and conversation engine form part of a managing core of a distributed management communication system having a plurality of nodes, the core comprising a data store for storing said call conversation map as a call architecture of a merchant, said
5 call conversation map defining a call handling process for the merchant, the call handling process being operable to detect receipt by a first said node of an incoming voice call on a corresponding incoming voice channel, and to couple the voice call from said first node via one of said call channels thereof to a second said node for connection to an agent associated with the merchant.

10

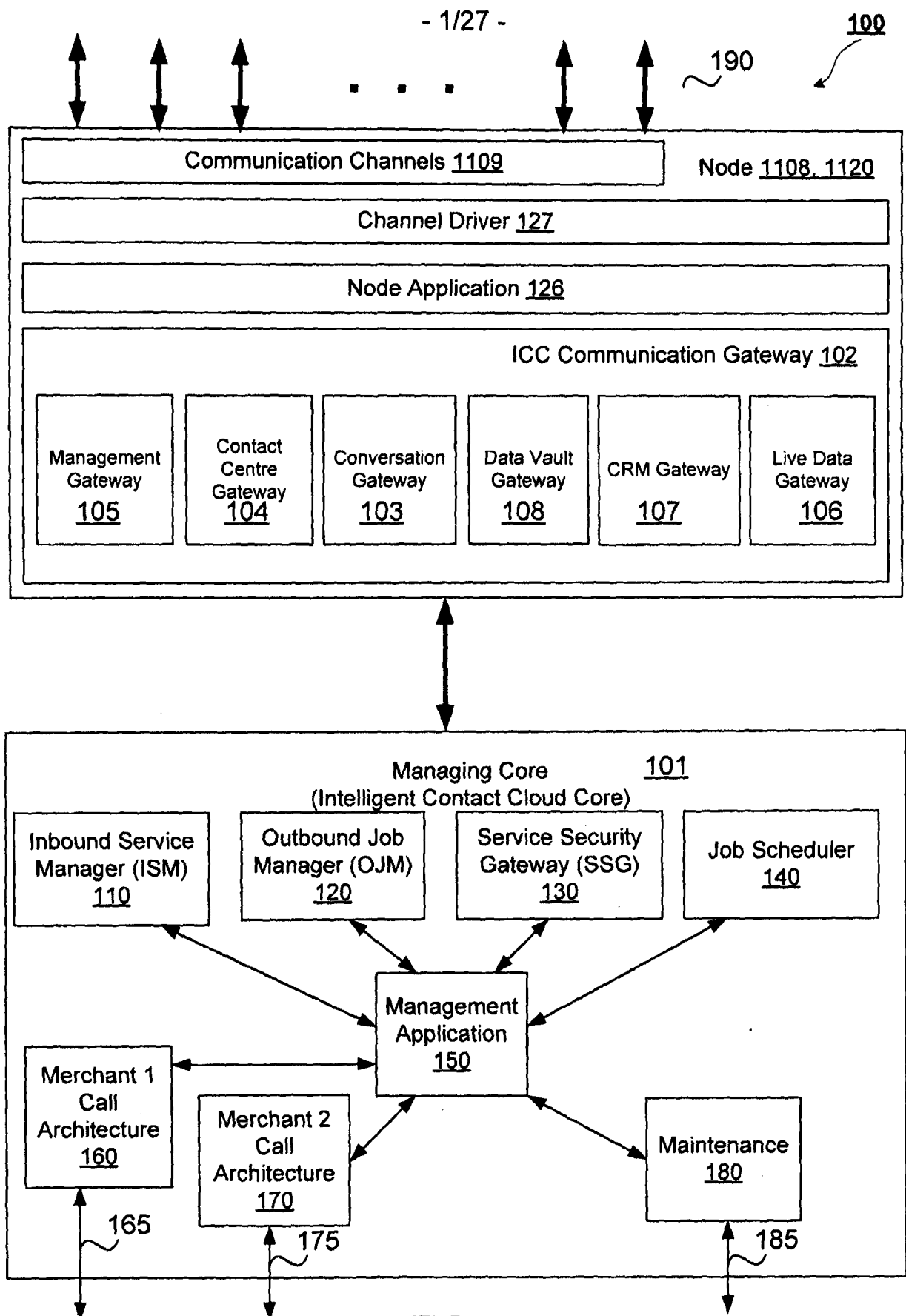
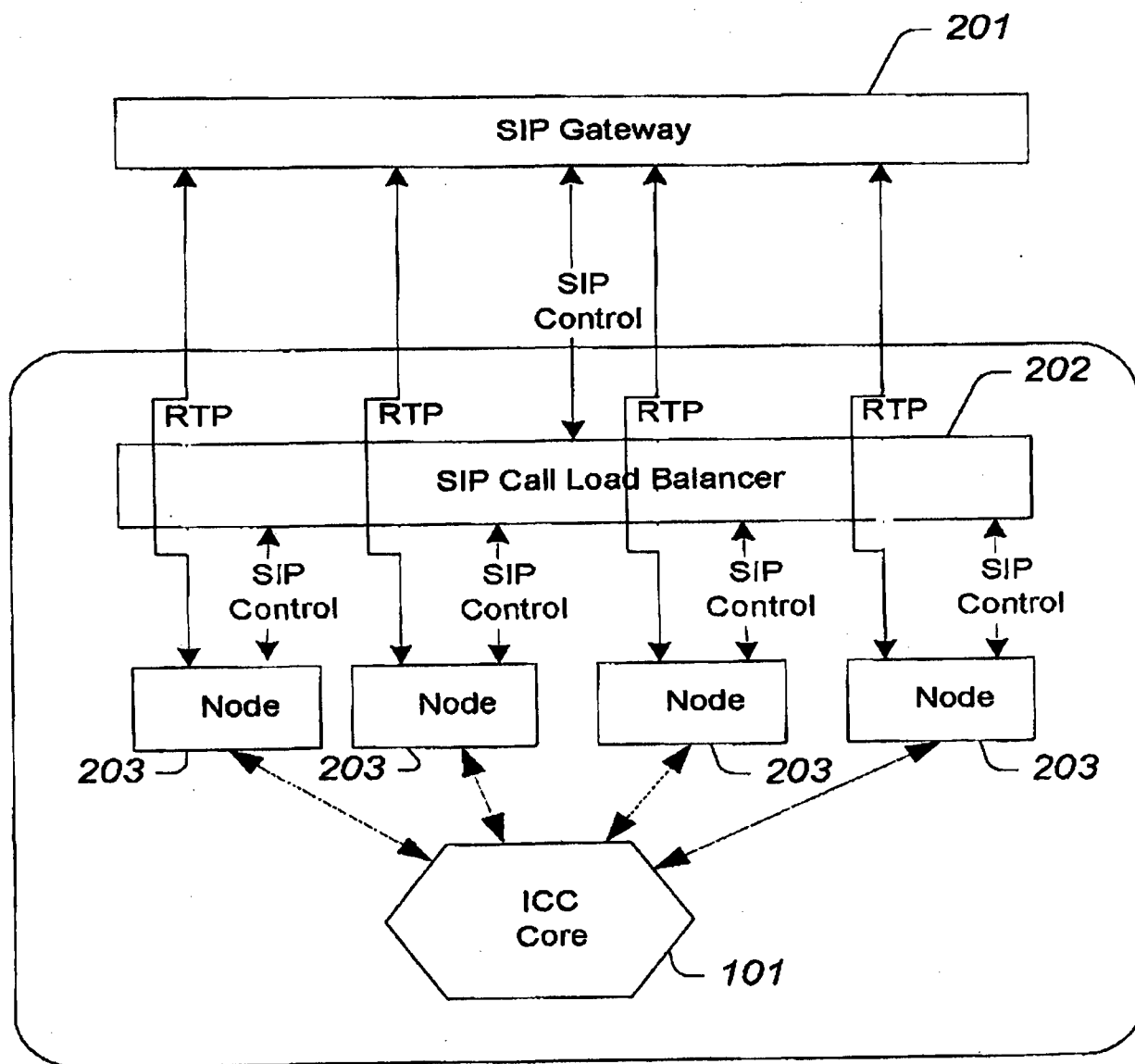
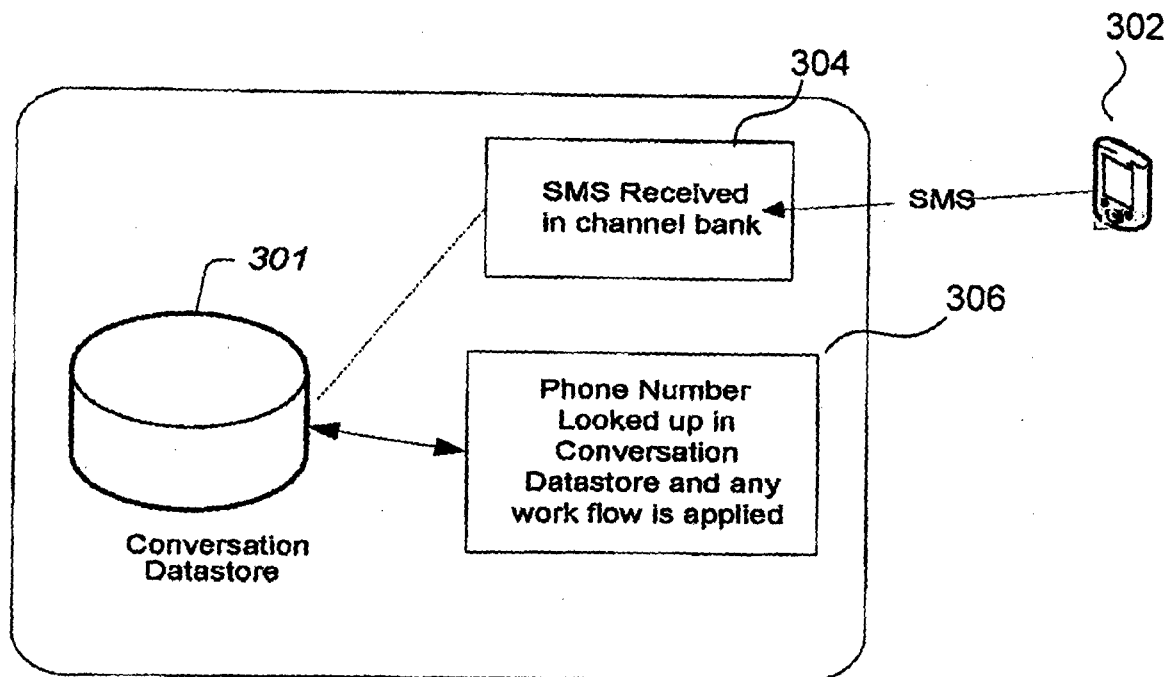


FIG. 1

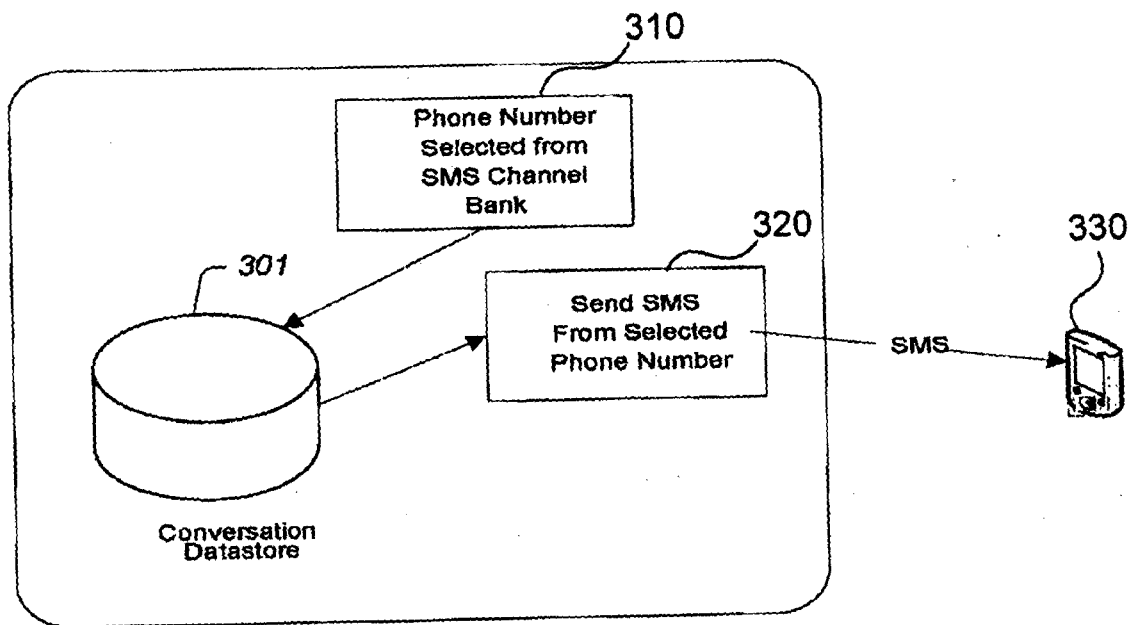
- 2/27 -

**FIG. 2**

- 3/27 -

**FIG. 3A**

- 4/27 -

**FIG. 3B**

- 5/27 -

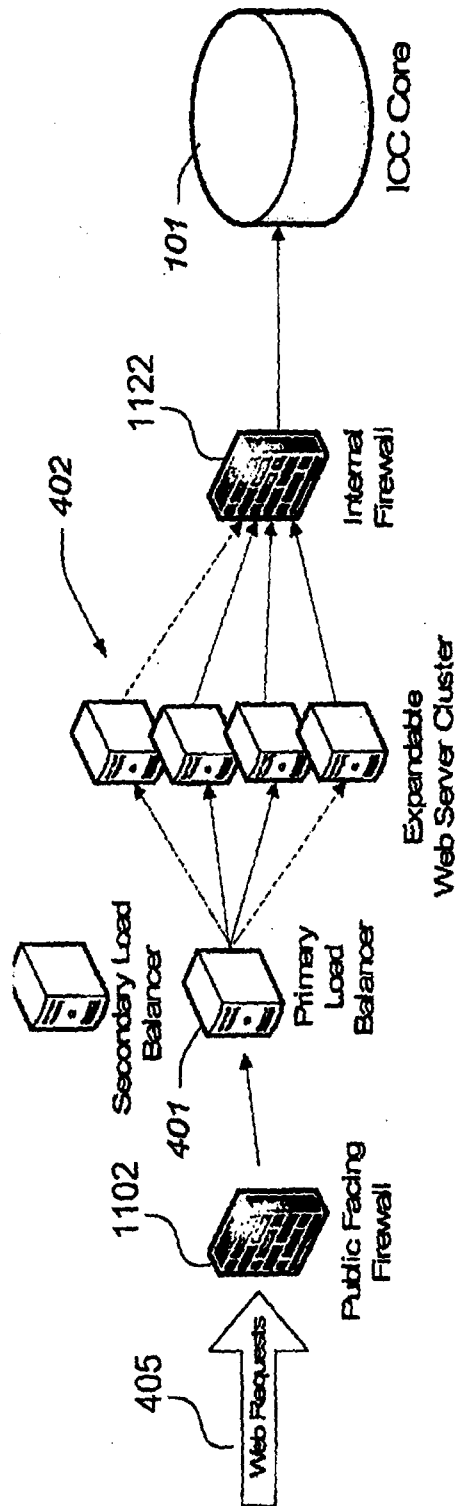
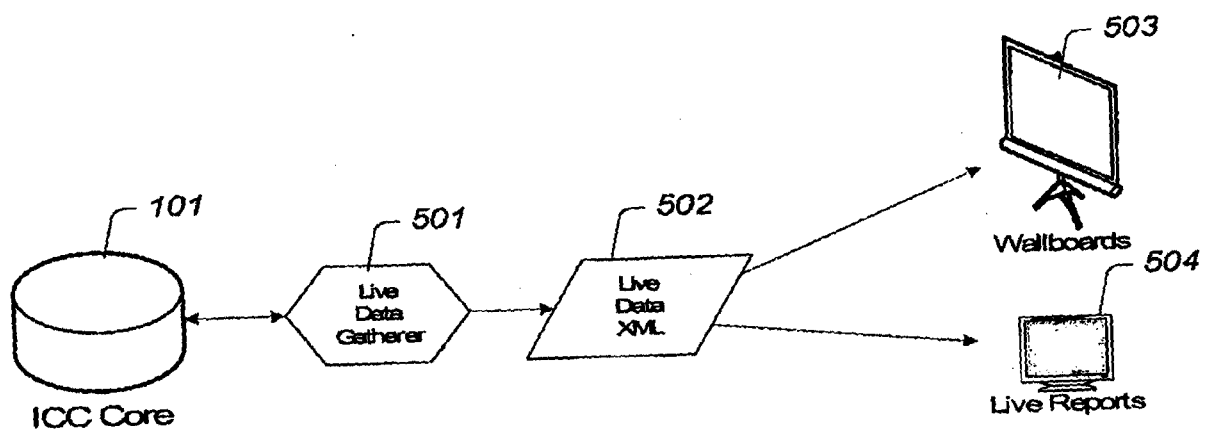
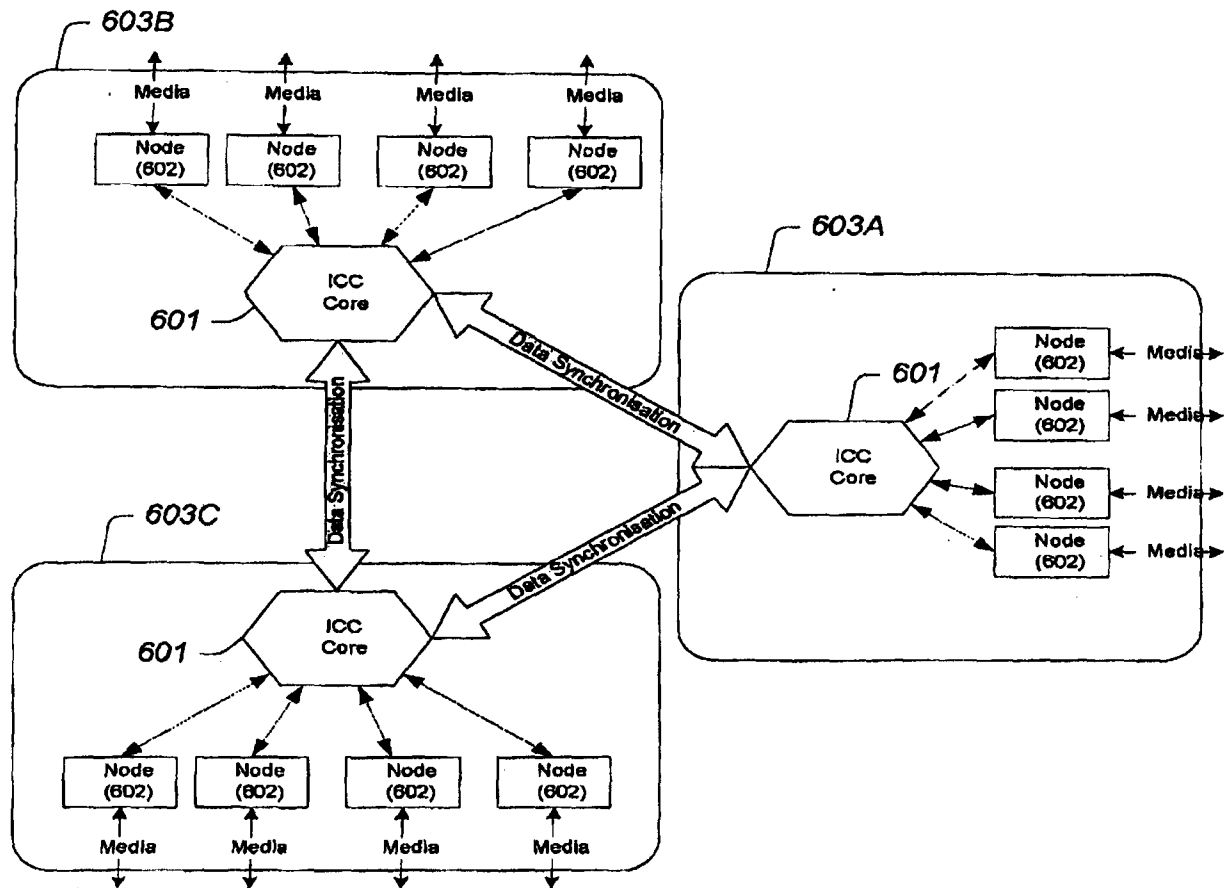


FIG. 4

- 6/27 -

**FIG. 5**

- 7/27 -

**FIG. 6**

- 8/27 -

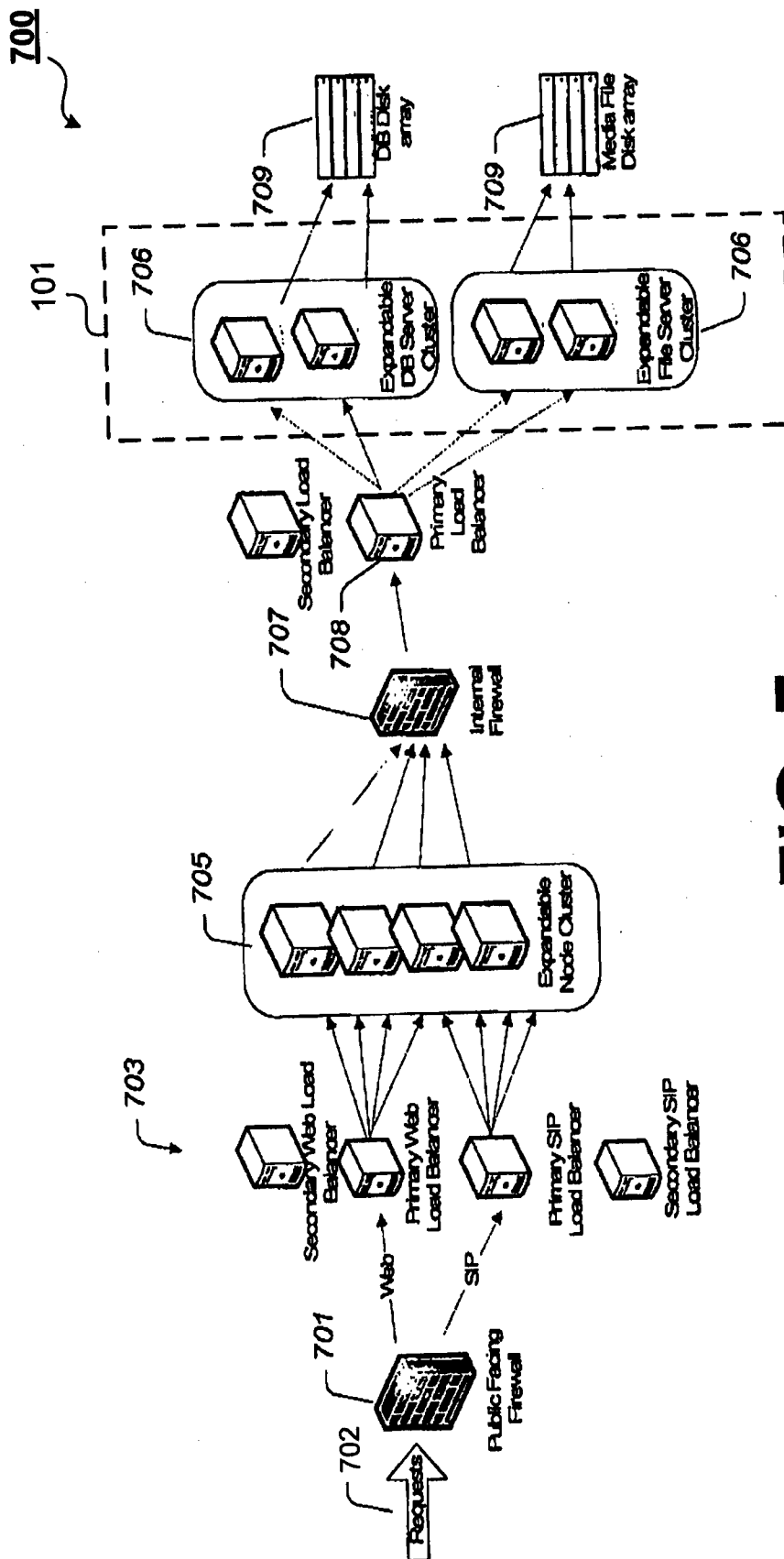


FIG. 7

- 9/27 -

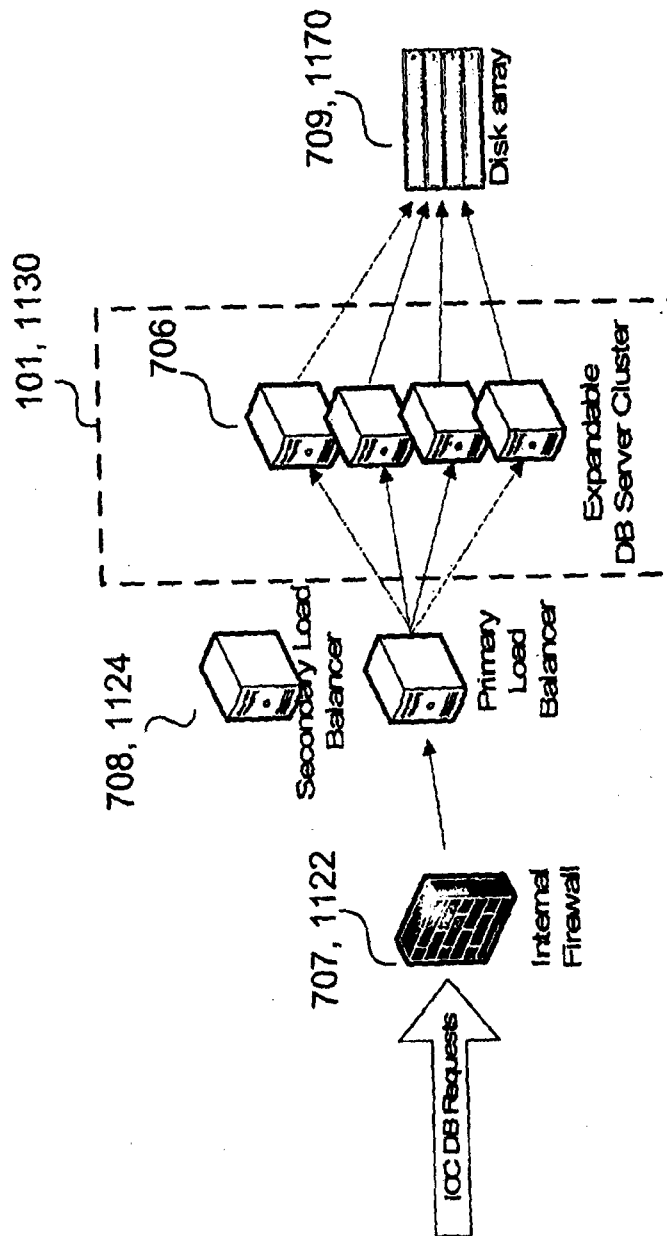
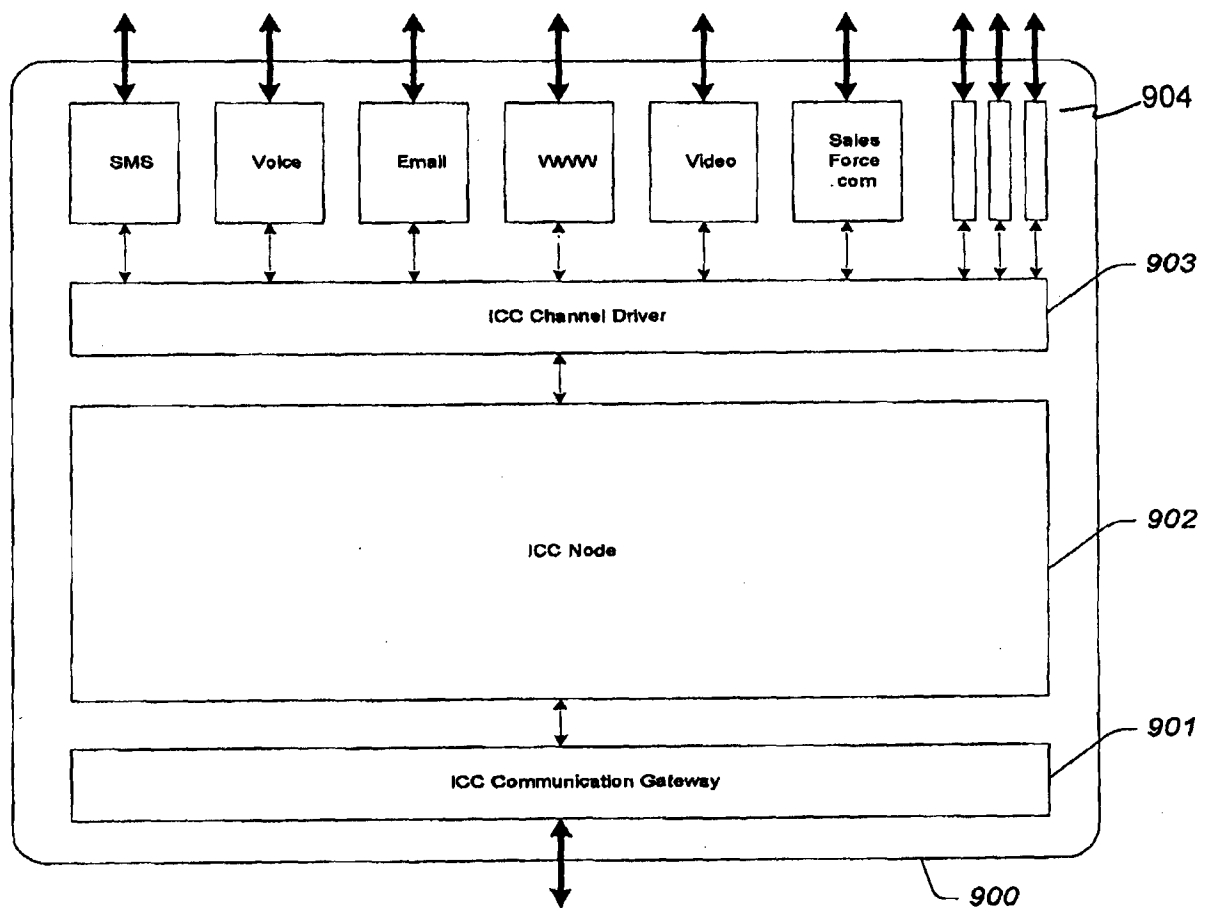


FIG. 8

- 10/27 -

**FIG. 9**

- 11/27 -

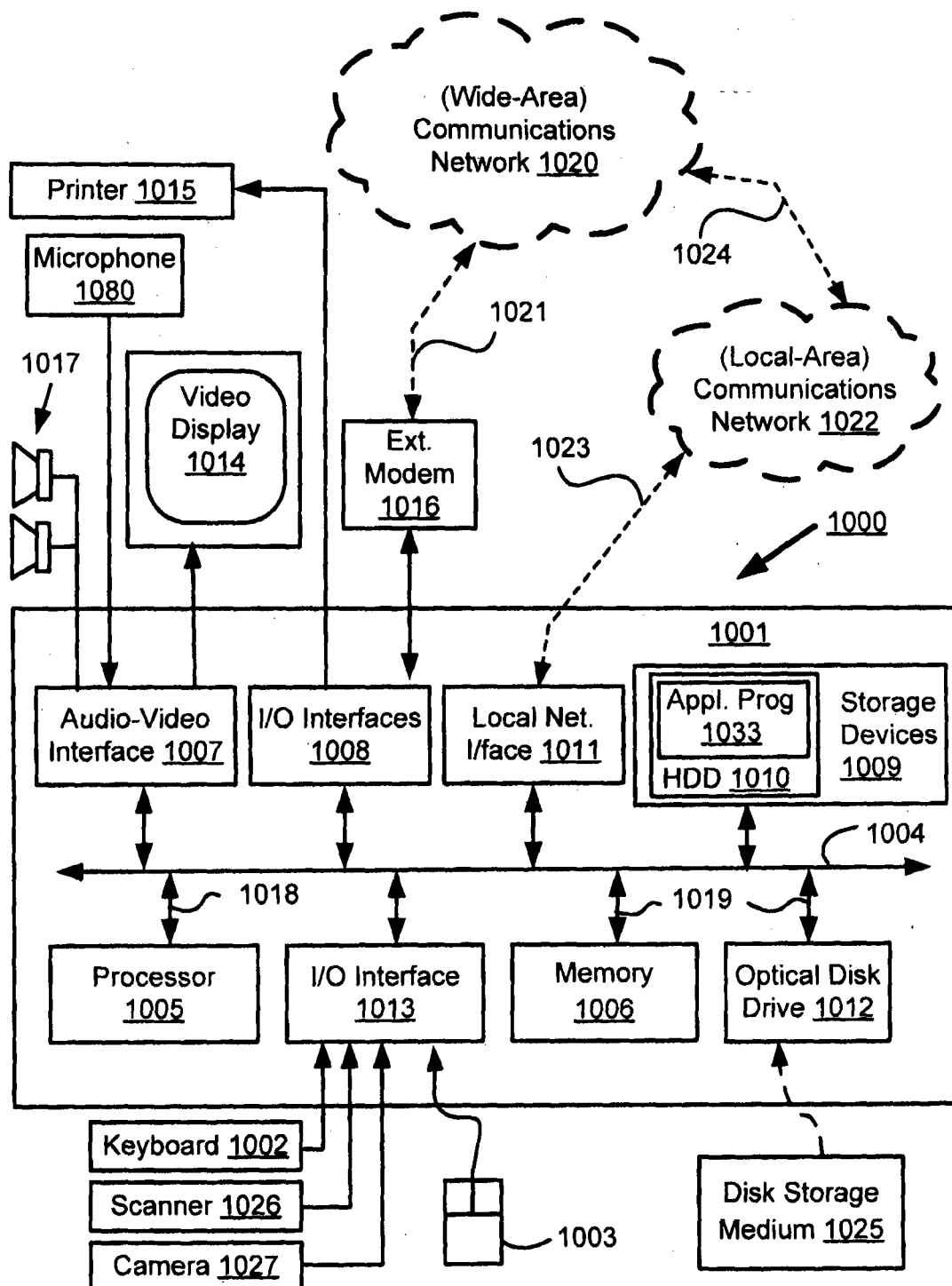


FIG. 10A

- 12/27 -

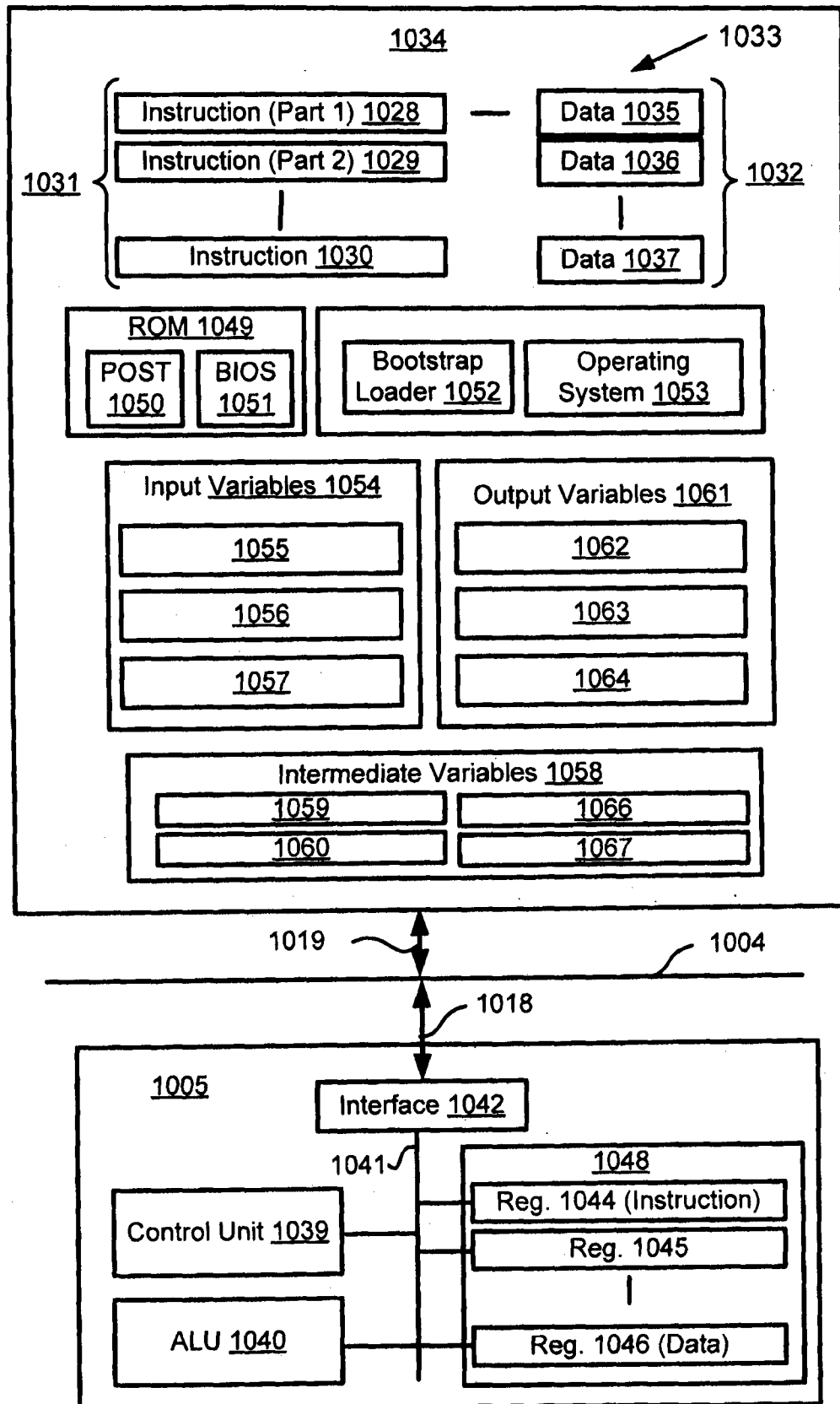


FIG. 10B

- 13/27 -

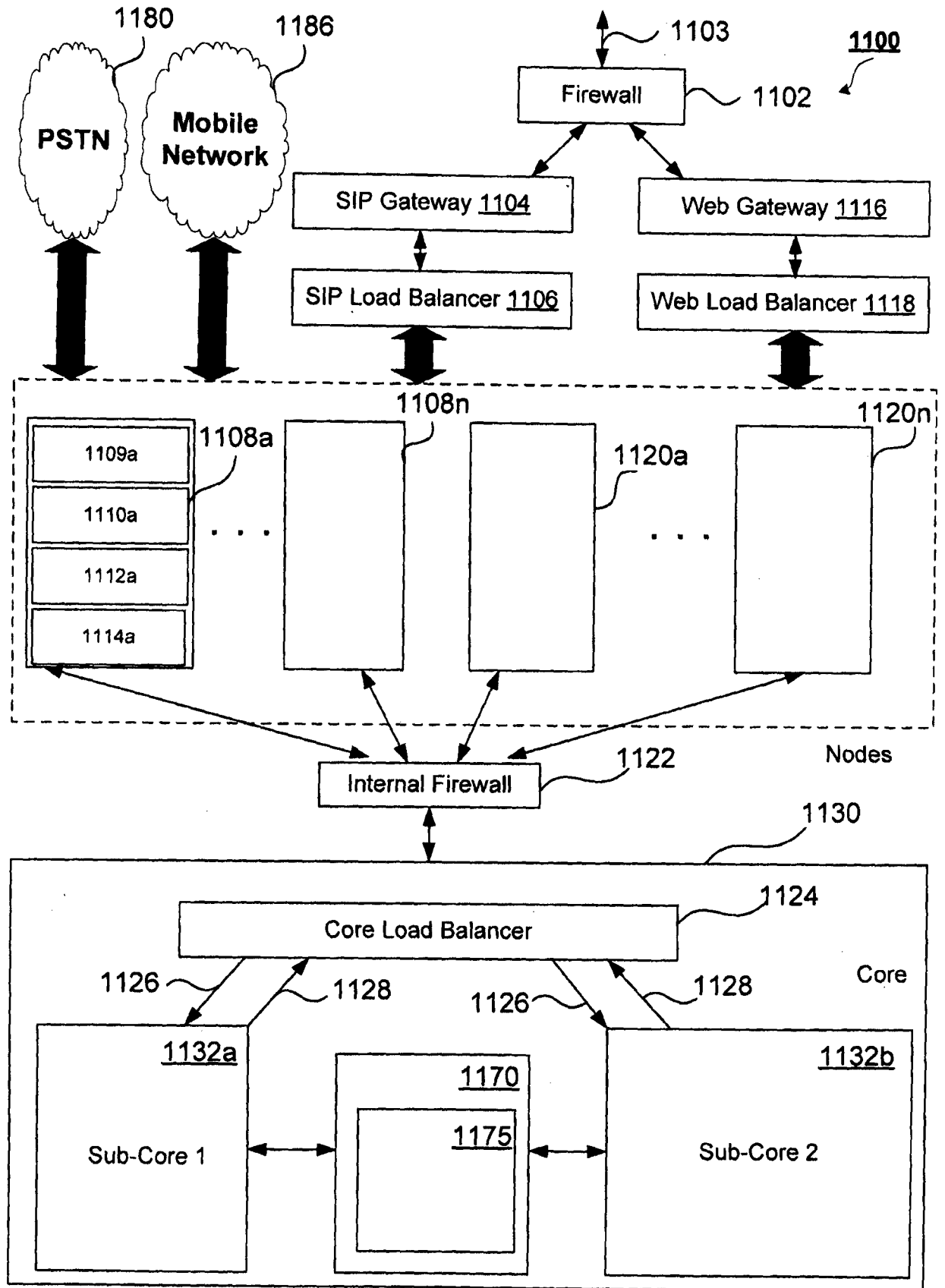


FIG. 11

- 14/27 -

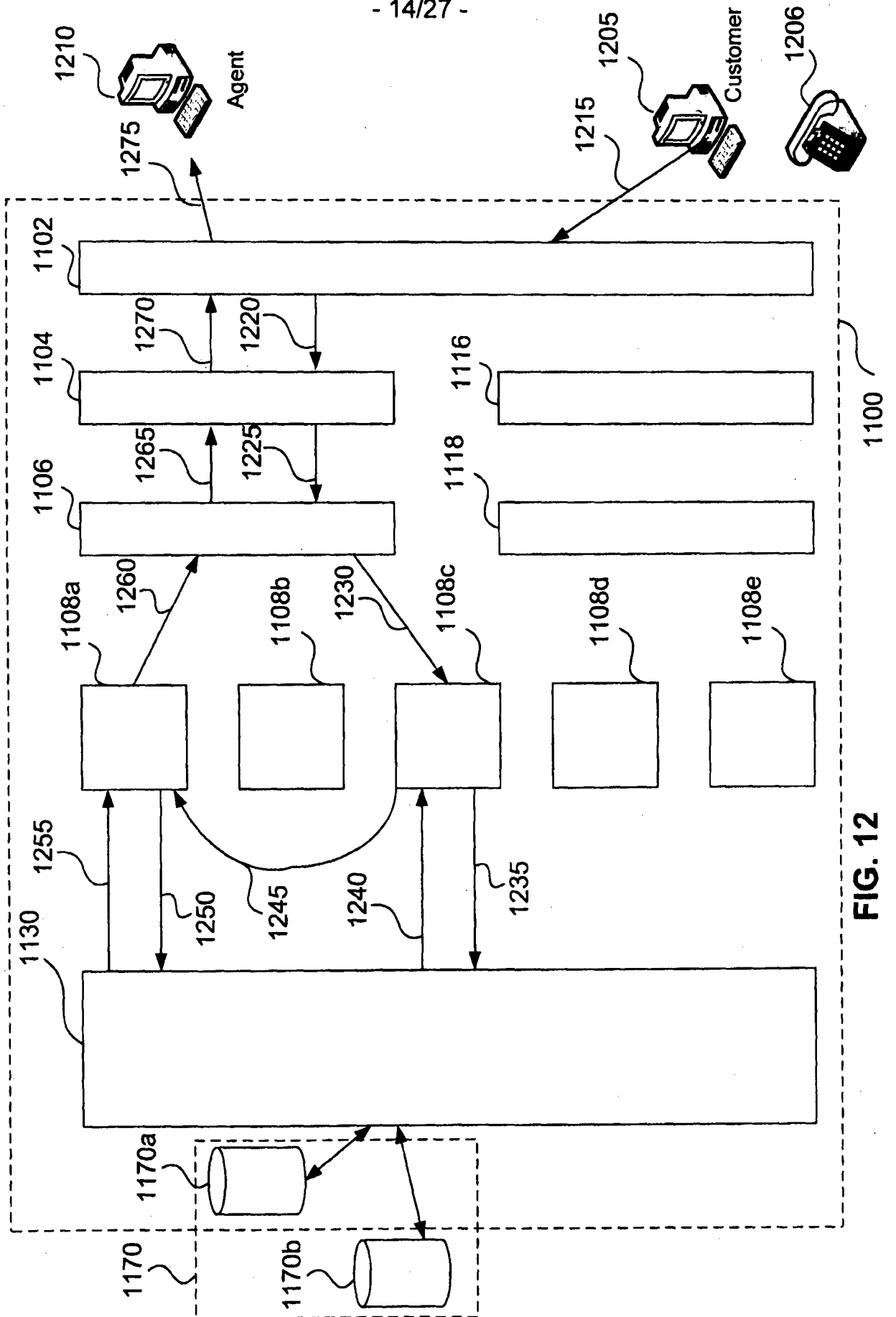
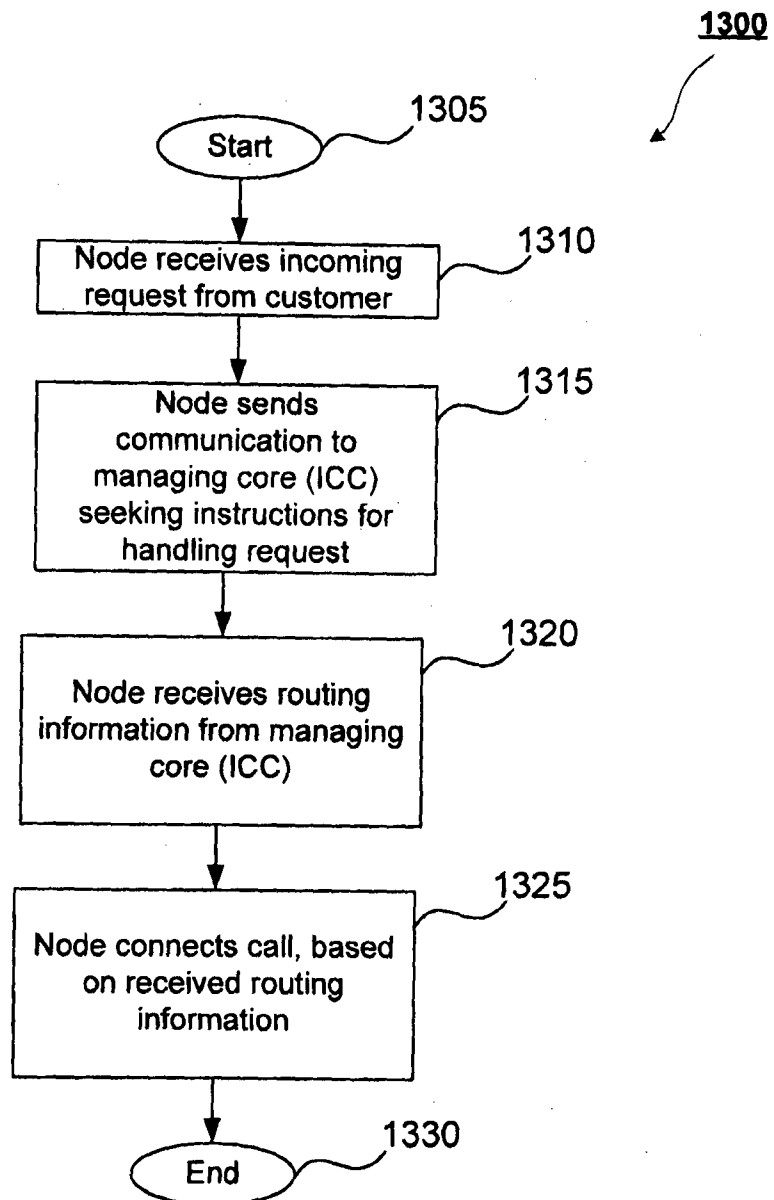
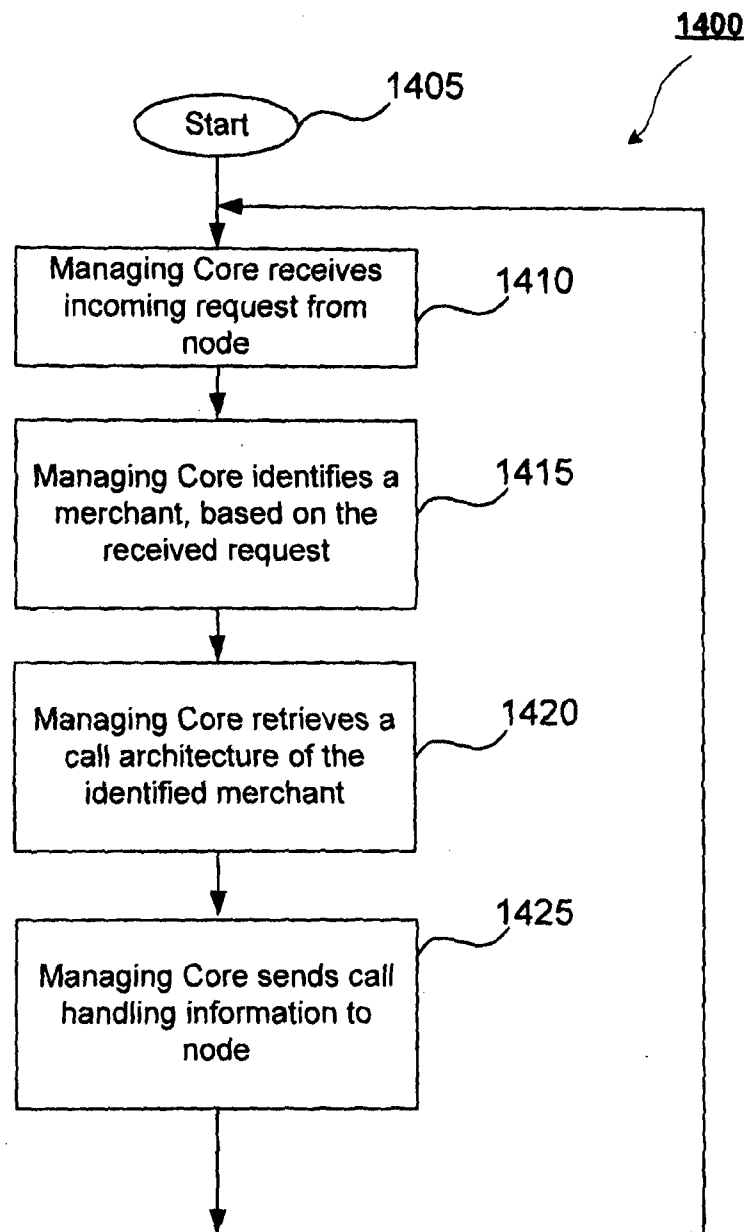


FIG. 12

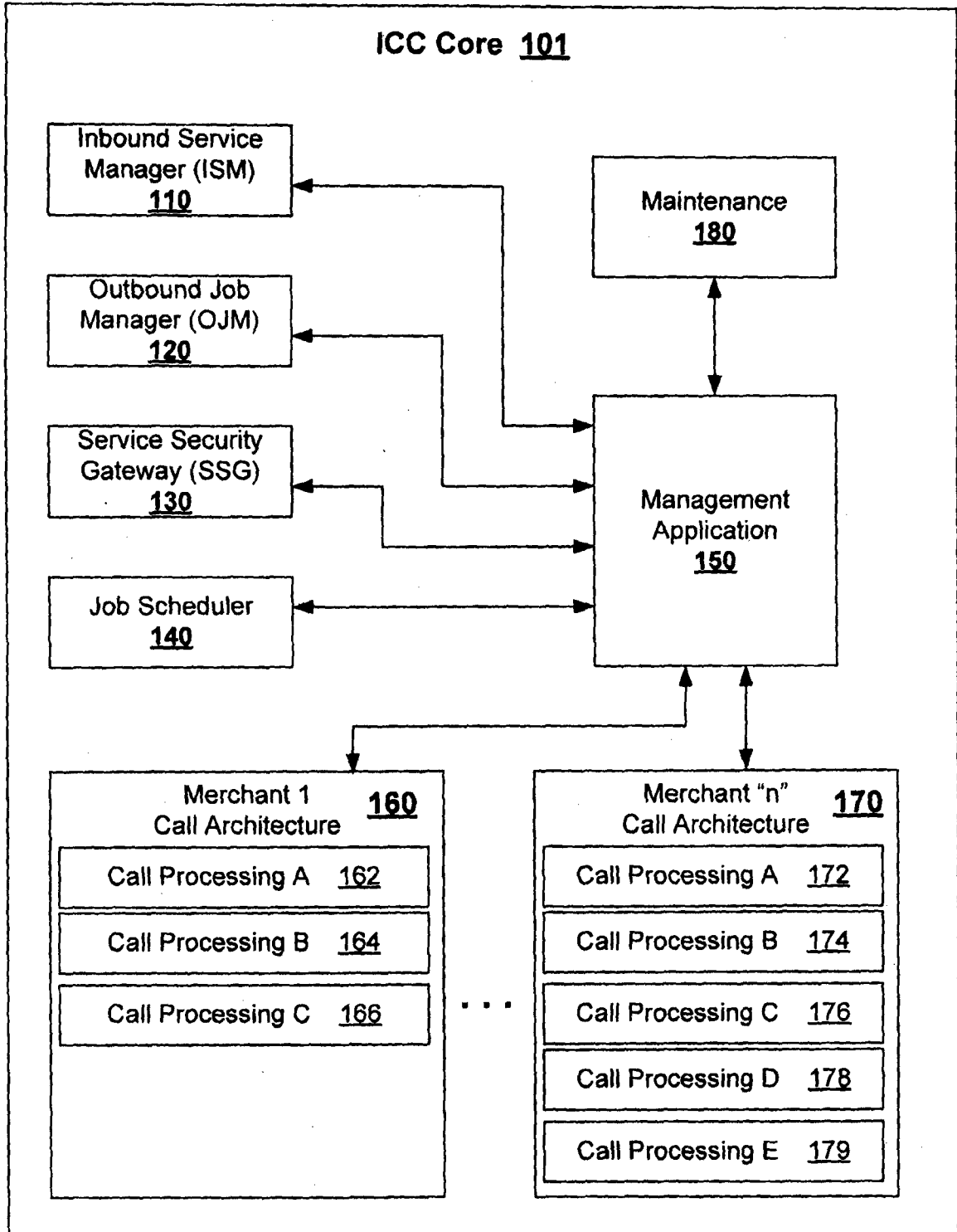
- 15/27 -

**FIG. 13**

- 16/27 -

**FIG. 14**

- 17/27 -

**FIG. 15**

- 18/27 -

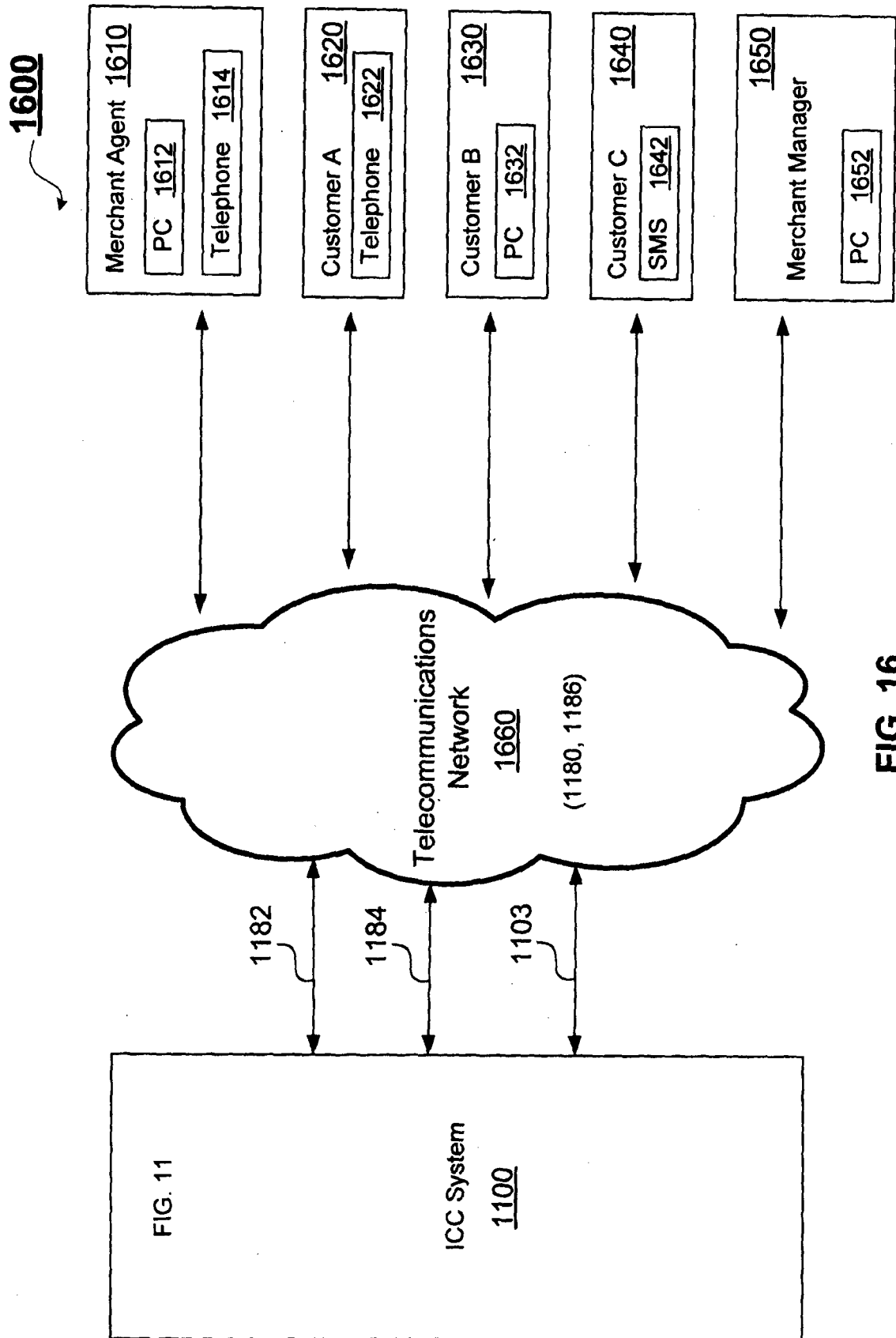


FIG. 16

- 19/27 -

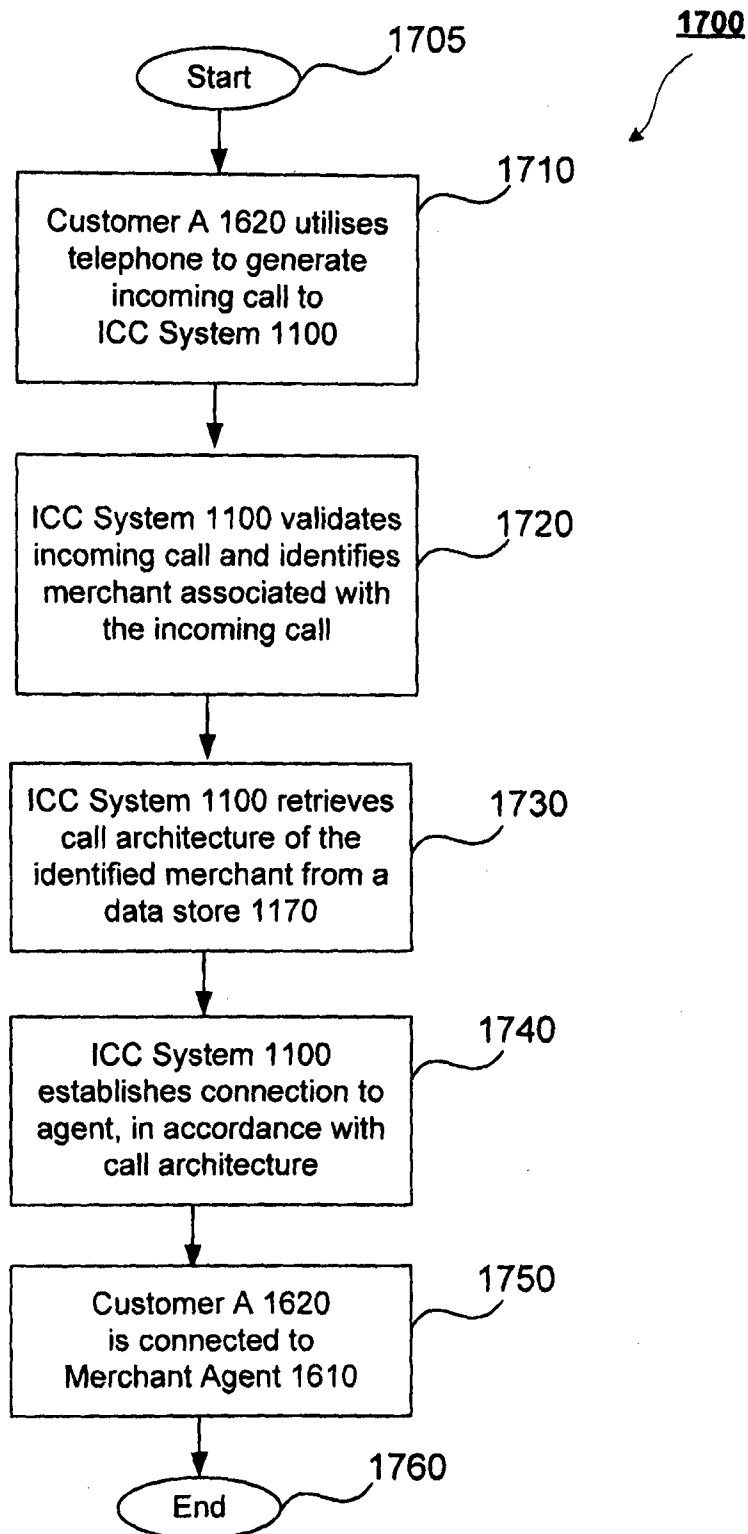
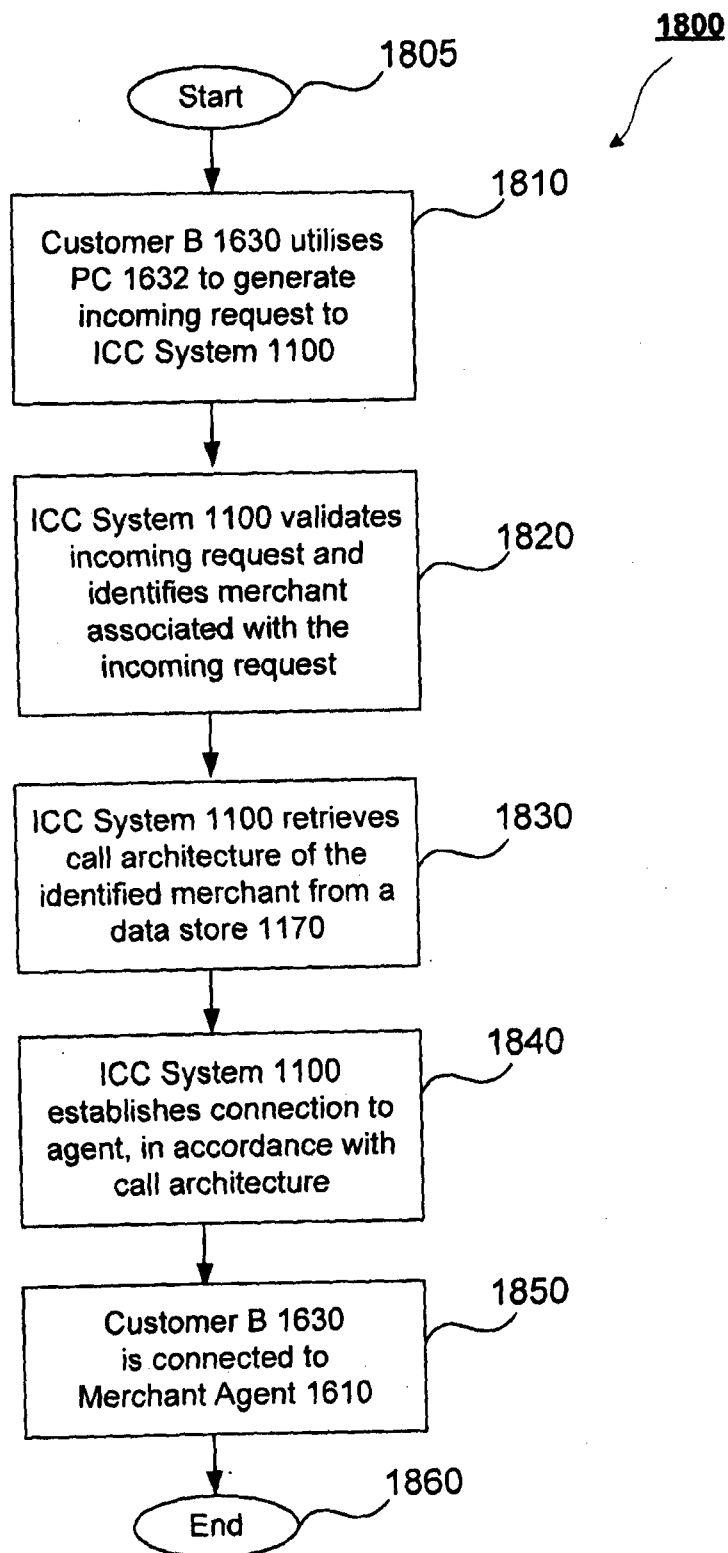
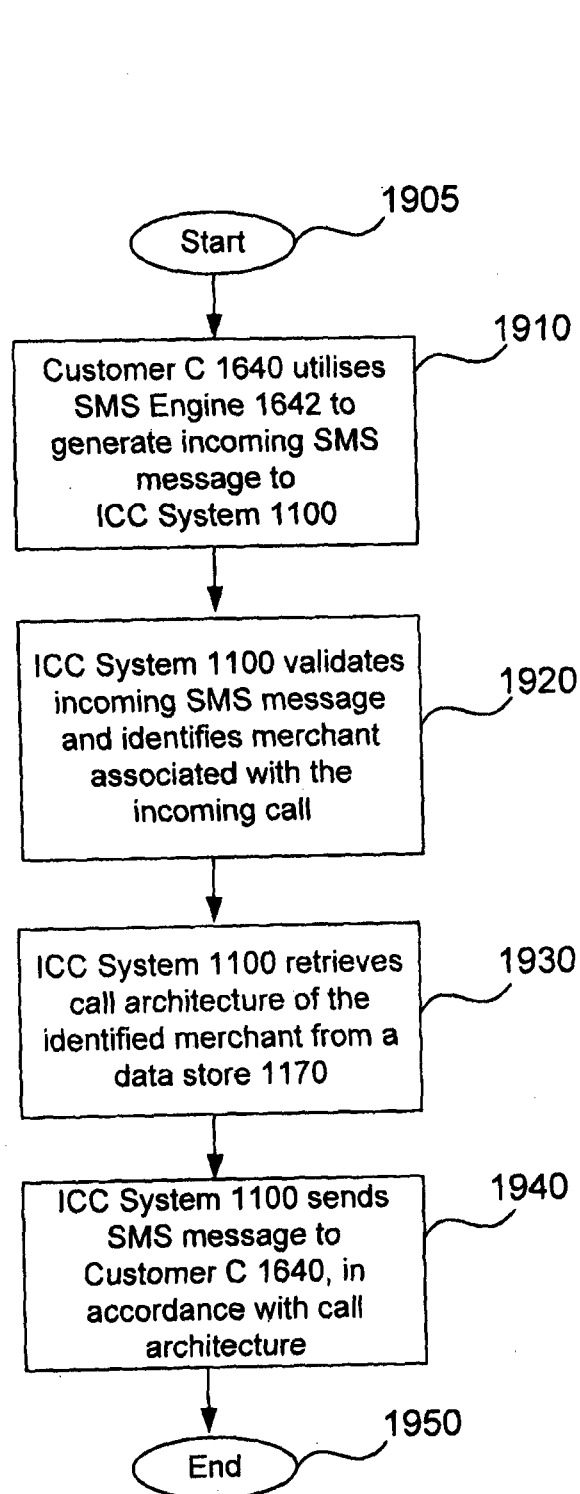


FIG. 17

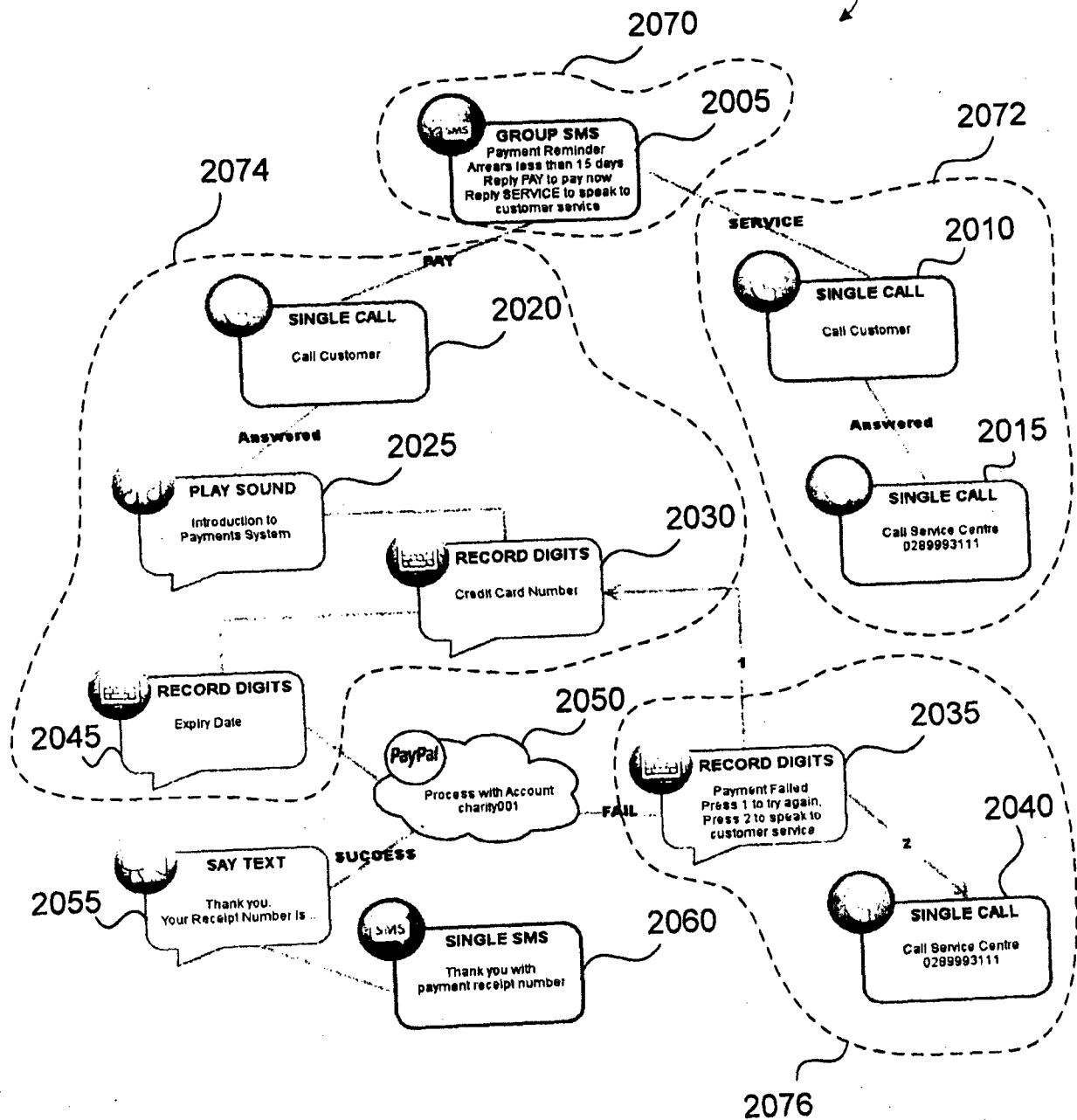
- 20/27 -

**FIG. 18**

- 21/27 -

**FIG. 19**

- 22/27 -

2000**FIG. 20**

- 23/27 -

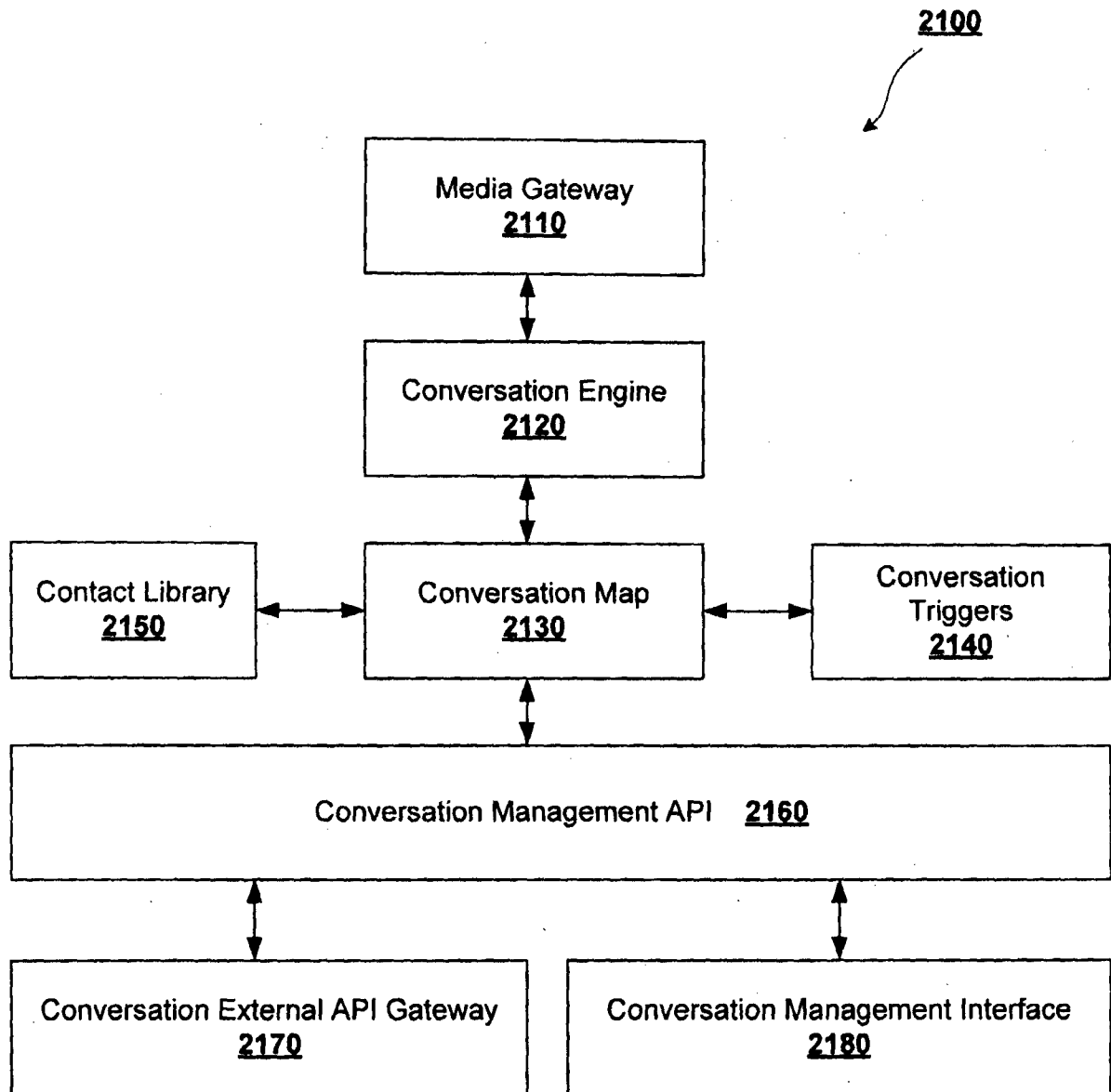
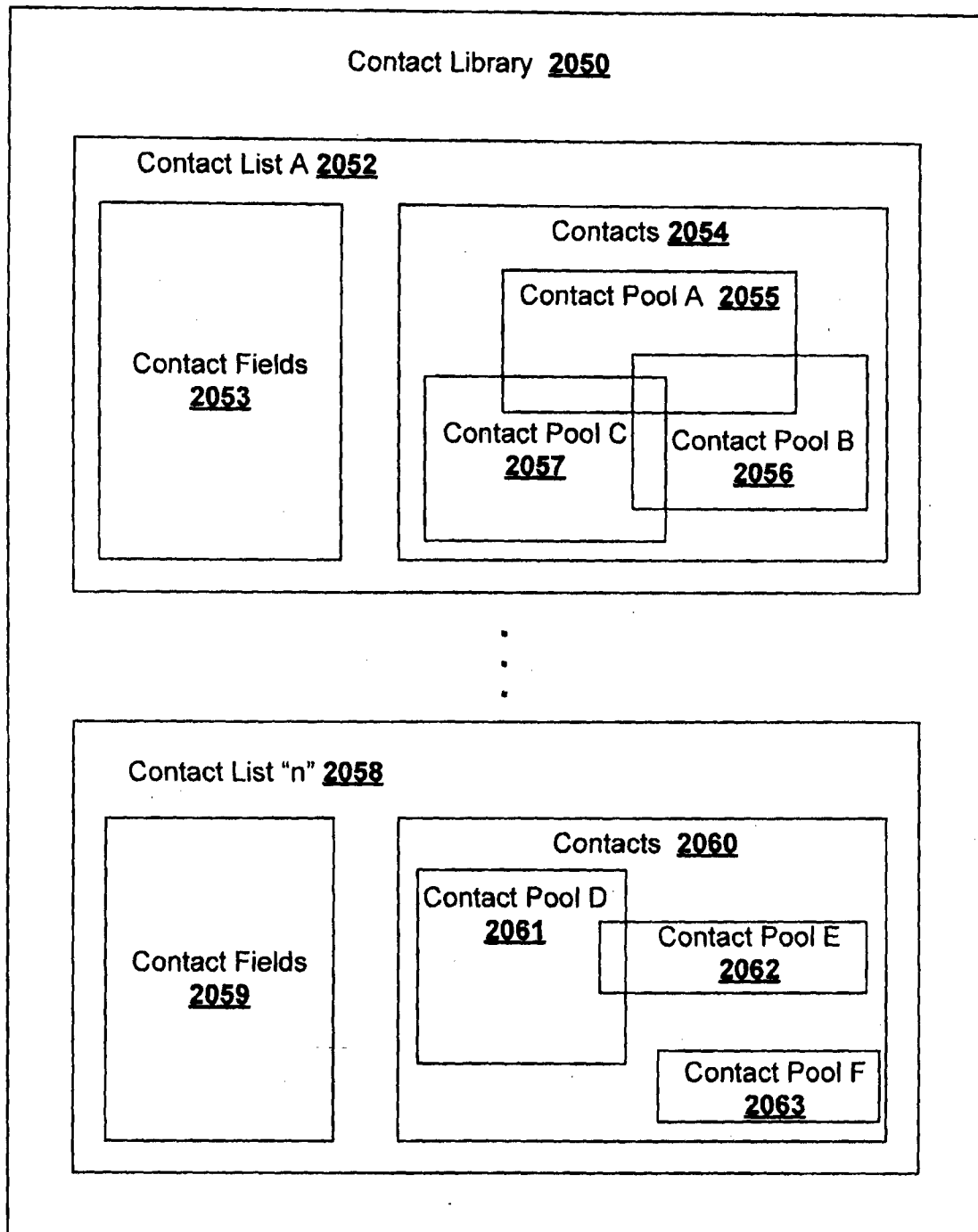


FIG. 21

- 24/27 -

**FIG. 22**

- 25/27 -

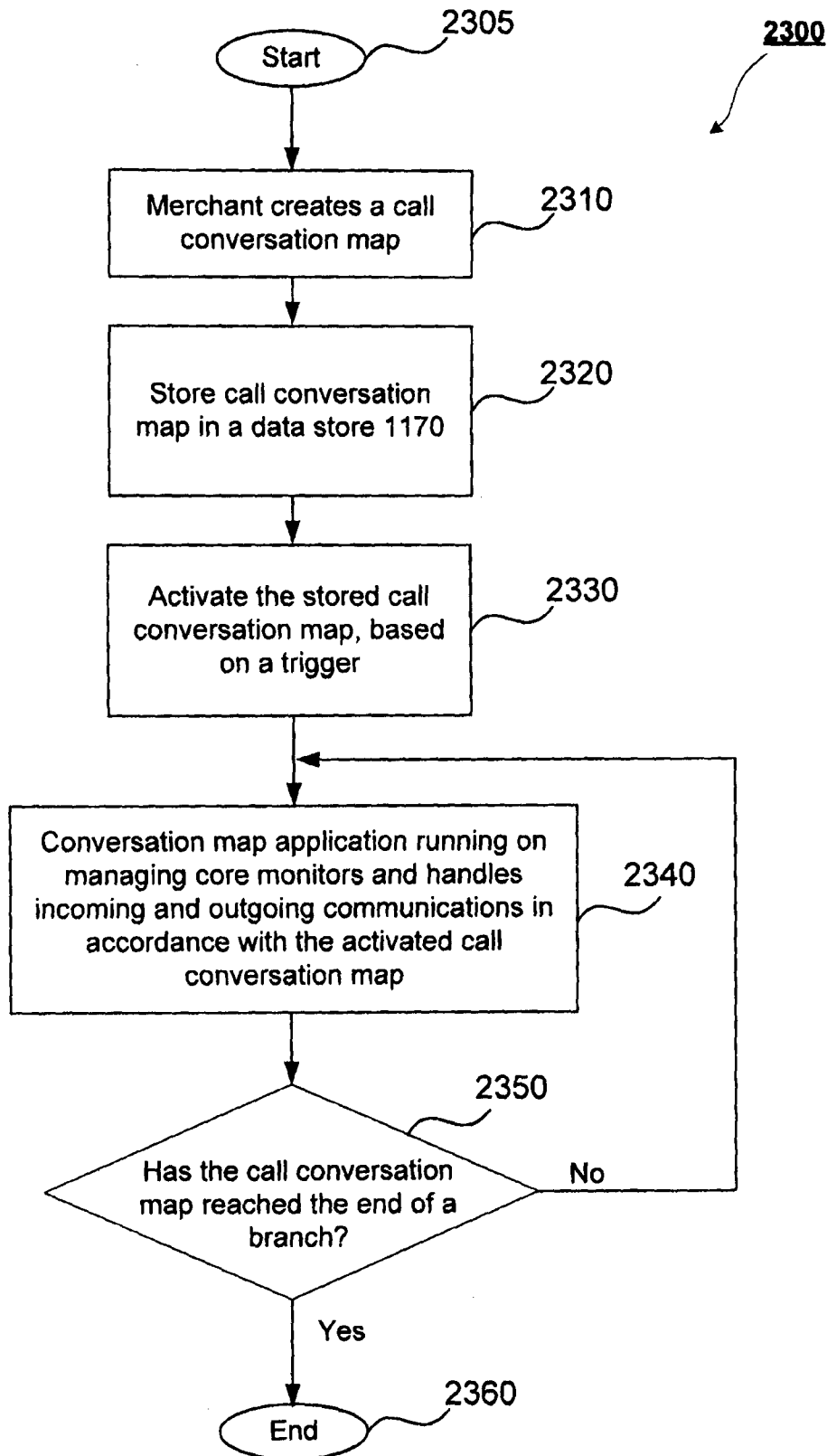


FIG. 23

- 26/27 -

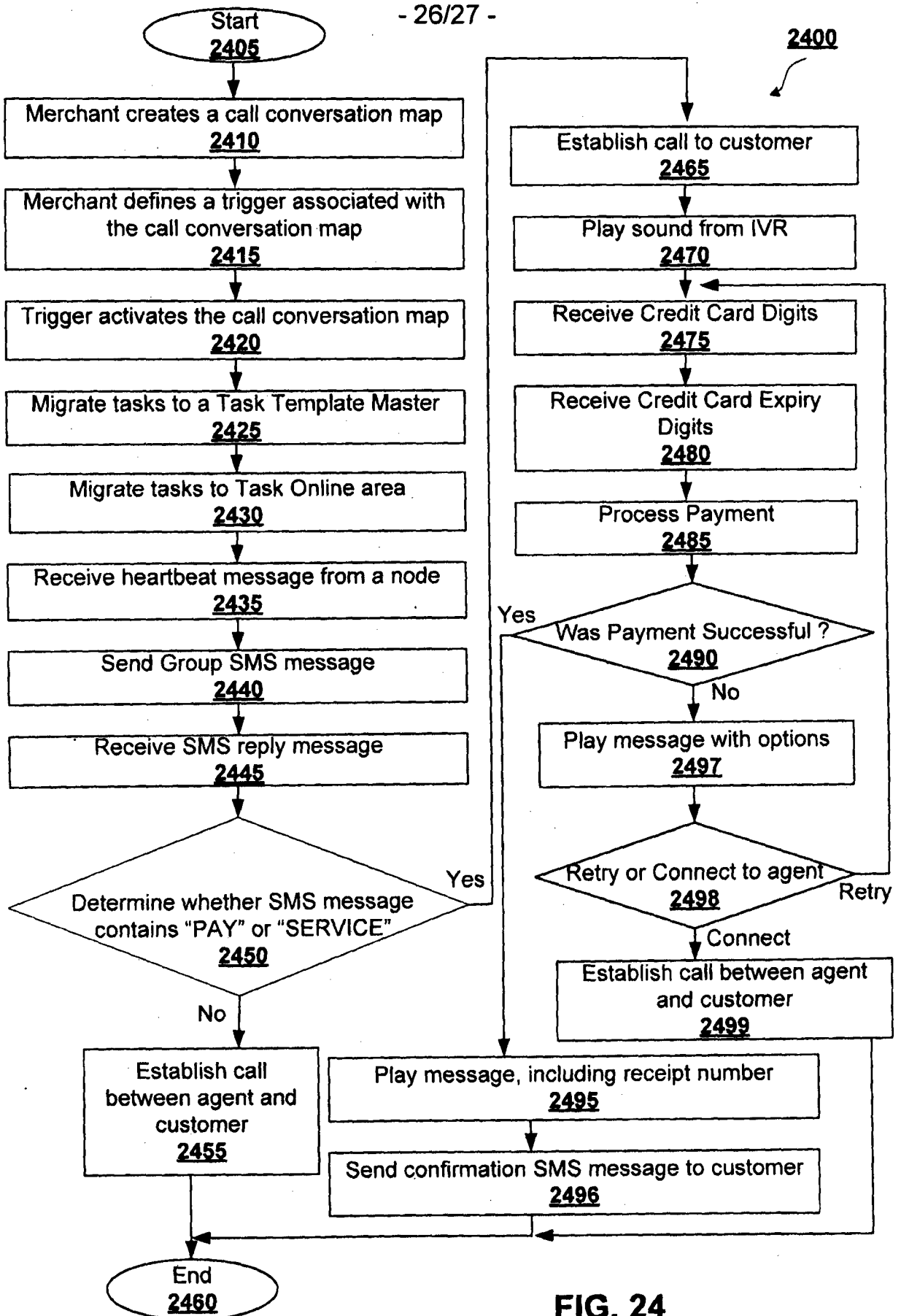


FIG. 24

- 27/27 -

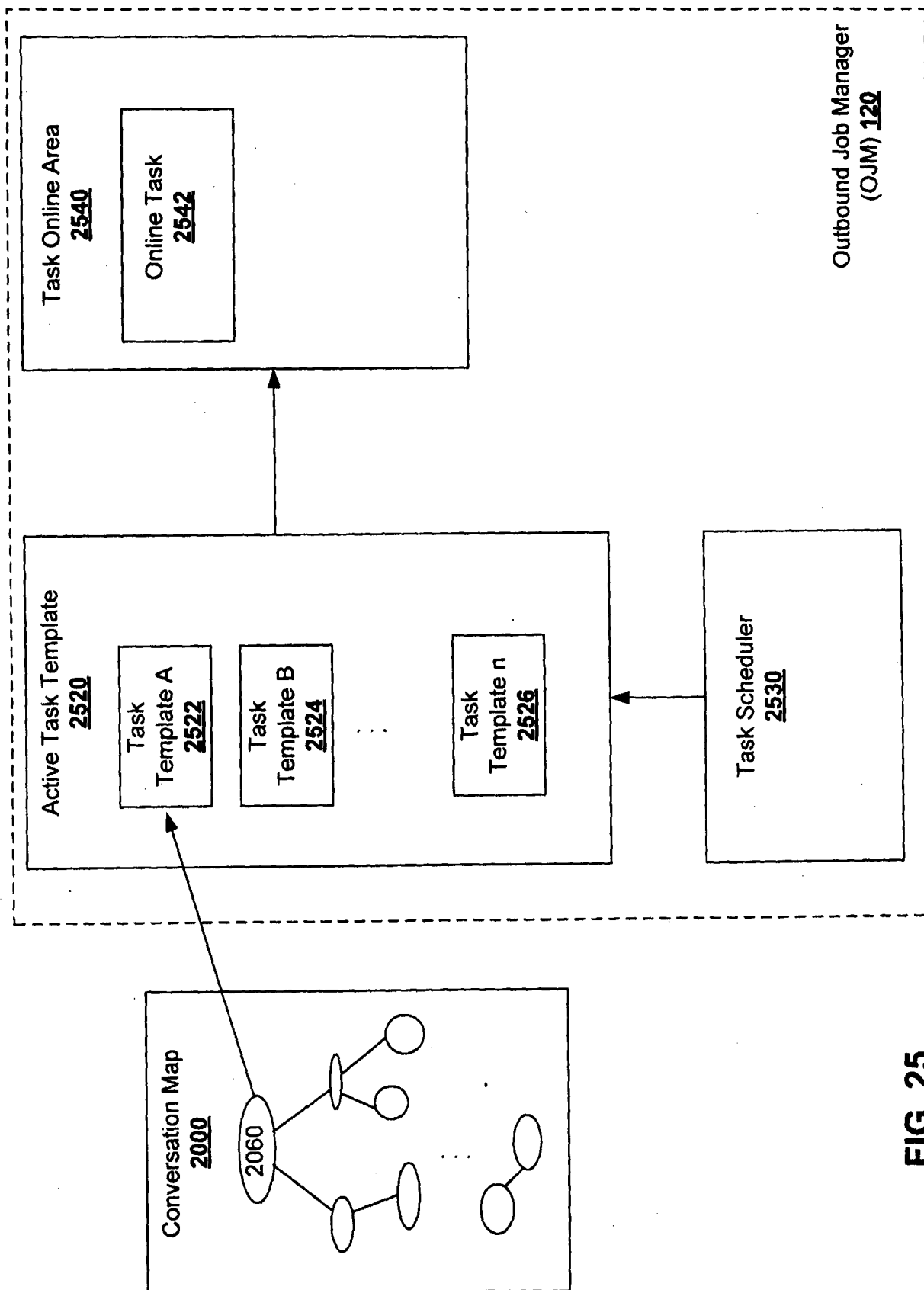


FIG. 25

INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU20 11/000846

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

H04B 5/04 (2006.0 1)**H04W 4/16 (2009.0 1)****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)

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

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

EPODOC, WPI, IPC marks with keywords: call mapping, conversation, object, link, incoming, outgoing, outbound, inbound and similar keywords. ESP(CENET, Google Scholar, Google Patents and Google with similar keywords as above.**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 20 10/01 53 106 A1 (FRAZIER et al.) 17 June 201 0 See whole document	1-20
A	WO 2006/03 1200 A1 (CRIMSONLOGIC PTE LTD) 23 March 2006 See whole document	1-20
A	US 2007/01 24740 A1 (BRUNSWIG et al.) 31 May 2007 See whole document	1-20



Further documents are listed in the continuation of Box C



See patent family annex

* Special categories of cited documents:

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

"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

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

"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

"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)

"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

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

"&" document member of the same patent family

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

Date of the actual completion of the international search
7 September 2011Date of mailing of the international search report
08.09.201 1Name and mailing address of the ISA/AU
AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaustalia.gov.au
Facsimile No. +61 2 6283 7999Authorized officer
AVANTHI GEEDIPALLY
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No : +61 2 6283 2806

INTERNATIONAL SEARCH REPORT**Information on patent family members****International application No.****PCT/AU2011/000846**

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report			Patent Family Member	
US	20 10/0 153 106	NONE		
WO	2006/03 1200	SG	120996	
US	2007/0 124740	NONE		
Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.				
END OF ANNEX				