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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
3 June 2010 (03.06.2010)

PCT

(10) International Publication Number
WO 2010/060465 A1

(51) International Patent Classification:
H04L 29/06 (2006.01) *H04L 29/08* (2006.01)

(21) International Application Number:
PCT/EP2008/066196

(22) International Filing Date:
26 November 2008 (26.11.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **NOKIA SIEMENS NETWORKS OY** [FI/FI]; Karaportti 3, FI-02610 Espoo (FI).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **RAJANIEMI, Jaakko** [FI/FI]; Vesakkotie 3 A 71, 00630 Helsinki (FI).

(74) Common Representative: **NOKIA SIEMENS NETWORKS OY**; COO RTP IPR, Patent Administration, 80240 Munich (DE).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

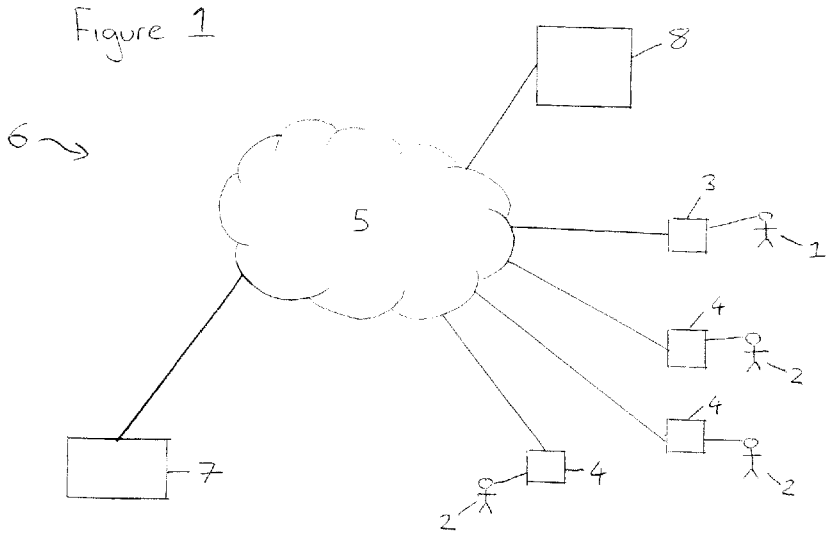
Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

WO 2010/060465 A1

(54) **Title**: REAL-TIME COMMUNICATION



(57) **Abstract**: A method and apparatus for providing real-time communication where a first user 1 can download a web page and content from a web server 8 to the first user's terminal device 3. A first and second interface embedded in the web page enable the first user 1 to log in to a real-time server 7 to initi- ate a real-time session for the content of the web page. The real-time session can then be logged into by one or more further users 2 via their terminal devices 4. The first user 1 and further users 2 can then communicate in real-time via the real-time session hosted by the real-time server 7.

Real-Time Communication

The present invention relates to providing a real-time communication in a network system and in particular for providing real-time communication between at least two terminals using the same web application. The present invention may also be used to provide real-time communication between a web application hosted on a web server and at least one terminal.

Typically, a user uses a web browser on the user's terminal to access and browse connected computer networks, such as the Internet. A web server on a computer network can host several web pages and content e.g. web applications which can be downloaded and displayed by a web browser on the user's terminal, which is typically a personal computer but may be any suitable terminal device including mobile terminal devices.

The user typically downloads the web page and content from a web server to the user's terminal device. The user may then interact with the web page and the content using a web browser on the user's terminal device.

An example of a web page and content is a web page that includes a map application as the content. The user downloads the web page including the map application as the content which is displayed by a web browser on the user's terminal device. The user may then search, view and interact with a map of an area of the user's choice. As will be appreciated, there are many different types of web applications that the user can choose to use by downloading the relevant web page and content from the web server over the Internet.

Typically, the communication between the web browser on the user's computer and the web server hosting the web pages and

content is via the well-known HyperText Transfer Protocol (HTTP).

Using HTTP the relationship between the web server hosting the content being displayed by a web browser on the user's terminal device and the user's terminal device is one of a request - response relationship. In other words, the web browser on the user's terminal device has to request, or poll, the web server in order to receive any data from the web server, i.e. the web server can only respond to a request from the web browser.

In terms of the above example, where the user downloads a web page with a map application as the content, the user may input a search term, for example, a postcode, place name etc. and the web browser sends the search term to the web server hosting the map application as a request using HTTP. The web server responds to this request via HTTP with the appropriate data relating to the requested map which is then displayed by the web browser on the user's terminal device. If the map application is designed to keep the data up-to-date on the web browser on the user's terminal device, then, at predetermined time intervals the web browser will poll the web server hosting the map web application.

However, HTTP has several limitations and disadvantages. The main disadvantage with the response - request arrangement is that content which requires real-time communication either between at least two users interacting with the same content or between the web server hosting the content and at least one user is very difficult to implement.

For example, in the case where two users are to interact with content hosted on a web server such as a map web application

then both users connect to the same session of the map web application which is displayed by each user's web browser. In the conventional systems, if one user zooms in on the map the remaining user will not see this occur until after a delay caused by having to wait for their respective web browsers to poll the web server and request any changes that may have occurred. Therefore, each of the users connected to the same session of the map application may not be viewing the same synchronized and up-to-date view of the map at any given time.

Previously, in order to implement a version of real-time communication for content hosted on a web server it has been known to implement advanced server side mechanisms, such as HTTP Streaming, HTTP Longpolling, or a solution using Adobes™ Flash technology for the individual content, e.g. for each web application hosted by the web server. However, the problem with these mechanisms and solutions is that they have to be developed on a per content, e.g. web application basis which increases the complexity and difficulty in developing such content. The developer of the content will have to understand and implement a real-time solution which increases the development time and cost of the content. Moreover, these per content solutions have a further disadvantage in that they can only be implemented on proprietary network systems and therefore cannot be re-used with any other content.

The preferred embodiments of the present invention provide a framework for real-time communication that is generic, independent and re-usable for the varying and different content where the developer of the content does not need to know or understand the real-time communication but can invoke and use the generic real-time communication of the preferred embodiments.

In particular, the preferred embodiments provide a framework for enabling real-time communication between at least two user terminals who are using or sharing the same content, e.g. the same content, and for enabling real-time communication between a between a web server hosting the content and at least one user terminal.

In a first aspect of the present invention there is provided an interface embedded in a web page, wherein the interface interfaces between content of the web page and a second interface, wherein the second interface interfaces between the interface and a real-time server; the interface is adapted to send a log in request to the second interface in order to establish a real-time session on the real-time server; send data to the real-time session via the second interface; and receive data from the real-time session via the second interface.

Accordingly, in the first aspect of the present invention the interface that is embedded in a web page acts as an interface between the content of the web page and the real-time server side of the network. This enables the preferred embodiments of the present invention to pass data from the content of the web application to the real-time server in real-time to be handled by the real-time session initiated by the interface. The real-time session can then distribute the data in real-time to the relevant parties. The preferred embodiments of the present invention therefore effectively bypass the web server hosting the content of the web page which avoids the drawbacks and disadvantages associated with HTTP and the request - response model described hereinabove. The interface is also therefore a generic interface which can be used with any content of a web page that requires real-time communica-

tion which means the developer of the content of a web page does not have to develop any bespoke method for handling the real-time communication from the content of the web page.

The interface may be an Application Program Interface (API) and the API may be a JavaScript API.

The content of the web page may be downloaded to a first terminal device on request of a first user.

The interface may receive an identifier of the the real-time session.

The interface may further be adapted to send the unique identifier to the content on a web server; and receive a link from the content on the web server wherein said link identifies the real-time session.

The interface may further be adapted to receive a link from the second interface wherein the link identifies the real-time session.

The first user via the first terminal device may send the link to one or more further users at one or more further terminal devices; the interface embedded in the web page is then downloaded to the one or more further terminal devices when the one or more further users activates the link; and the interface is further adapted to: send a log in request to the second interface for the one or more further users; receive a log in response from the second interface; and send a message to the first user indicating that the one or more further users has joined the real-time session.

Accordingly, in the preferred embodiments the first user initiates a real-time session for the content of the web page they are accessing by logging into the real-time server. The first user receives a link that identifies the web page and the real-time session they have initiated for the content of the web page. The first user forwards the link to any number of further users who can then join the real-time session that the first user has initiated which enables all of the connected users to pass data between on another in real-time via the real-time session.

The content of the web page may comprise content of a web application, wherein the web application is hosted by a web server.

The log in request may comprise a username and password. Alternatively, the log in request may be an anonymous log in request.

The link may comprise an identification of the web page and the link may also comprise a unique identifier, wherein the unique identifier identifies the real-time session. The unique identifier may be a Jabber Identity (JID).

The real-time session may be hosted by a real-time server and preferably, the real-time session handles real-time communication for the content of the web page.

Data may be sent to the real-time session via the second interface comprises data to be sent in real-time to a terminal device that is logged into the real-time session. Data may be received from the real-time session via the second interface comprises data sent in real-time from a terminal device that is logged into the real-time session.

Accordingly, in the preferred embodiments any data that is to be transmitted in real-time to another user using another terminal device that is also logged into the same real-time session is sent via the second interface to the real-time session for distribution to the further terminal devices logged in to the real-time session. Similarly, the interface of the preferred embodiments receives data from the real-time session where the data is preferably information sent from another terminal device that is logged into the same real-time session. By enabling the users who are logged into the real-time session for the content of the web page they are both using to send data in real-time to one another via the real-time server then the disadvantages and drawbacks of the conventional systems may be overcome.

The interface may be further adapted to send a request for an endpoint URL to the second interface; and receive the endpoint URL from the second interface wherein the endpoint URL identifies the real-time session. The interface may be further adapted to send the endpoint URL to a web server hosting the web page. The endpoint URL may comprise an address of the real-time server and an identification of the real-time session.

The interface may b efurther adapted to receive data in real-time from the web server via the real-time session and the second interface.

Accordingly, the preferred embodiments of the present invention have the further advantages that the interface can obtain an endpoint URL which it forwards to the web server that is hosting the web page the user is accessing and the content of the web page. The web server can then push data to the

real-time session identified by the endpoint URL so that any data from the web server relating to the content of the web page, e.g. a web application, can be sent to the user in real-time without the web server having to wait for the user's terminal device to request the data according to the convention request - response model.

According to a second aspect of the present invention there is provided a system for providing real-time communication comprising the interface as claimed in any one of claims 1 to 19 and the second interface.

According to a third aspect of the present invention there is provided a computer program product comprising computer readable executable code for performing the steps of sending a log in request to said second interface in order to establish a real-time session on said real-time server; sending data to said real-time session via said second interface; and receiving data from said real-time session via said second interface.

According to a fourth aspect of the present invention there is provided an interface wherein the interface interfaces between a second interface and a real-time server, wherein the second interface interfaces between a content of a web page and the interface; the interface is adapted to receive a log in request from said second interface; perform a log in to said real-time server in order to establish a real-time session on said real-timer server; receive a unique identifier from the real-time server identifying said real-time session; receive data from said second interface for the real-time session; send said received data from said second interface to said real-time session; receive data from said real-time

session; and send said received data from said real-time session to said second interface.

Accordingly, in the preferred embodiments of the present invention there is provided an interface that acts as an interface between the content of a web page and the real-time server. In the preferred embodiments of the present invention this interface receives data from the content of the web page (via the second interface) and interacts with the real-time server so that data can be handled in real-time by the real-time session initiated by the interface. The real-time session can then distribute the data in real-time to the relevant parties. The interface performs the log in to the real-time server in order to generate a real-time session for the content of the web page that is requesting a real-time session by requesting to log in to the real-time server. Once the real-time session on the real-time server has been created then the interface can receive data from the content of the web page, which preferably is data that is to be sent to a further user in real-time, and forward the received data to the real-time server for distribution. The interface can also receive data from the real-time server, which is preferably data sent from a further user or a web server in real-time, and forward the received data to the web page so that it can preferably be displayed to the user of the web page.

The preferred embodiments of the present invention therefore effectively bypass the web server which preferably hosts the content of the web page which avoids the drawbacks and disadvantages associated with HTTP and the request - response model described hereinabove. The interface is also therefore a generic interface which can be used with any content of a web page that requires real-time communication which means the developer of the content of a web page does not have to

develop any bespoke method for handling the real-time communication from the content of the web page.

Preferably, the interface is a Flash™ interface. Preferably, the interface is hosted on the real-time server. Alternatively, the interface may preferably be embedded in the web page. Alternatively, the interface may preferably be hosted on a web server.

The interface may be further adapted to receive a unique identifier of said real-time session from said real-time server.

The interface may be further adapted to send said unique identifier to said second interface.

The interface may be further adapted to generate a link, wherein said link comprises said unique identifier; and send said link to said second interface. Thus, the interface may also create a link that can be forwarded by the user of the web page to further users to enable those further users to join the real-time session.

The log in request may comprise a username and password. Alternatively, the log in request may comprise an anonymous log in request.

The step of performing the log in may further comprise the step of sending the log in request to the real-time server.

The unique identifier may be a Jabber Identity (JID). The link may further comprise an identification of the web page.

The data received from the second interface may be sent to the real-time session and the data received from the real-time session may be sent to the second interface.

The interface may be further adapted to receive a request for an endpoint URL from the second interface; generate the endpoint URL, wherein the endpoint URL identifies the real-time session; and send the endpoint URL to the second interface. The endpoint URL may comprise an address of the real-time server and an identification of the real-time session. The interface may be further adapted to receive data in real-time from a web server via the real-time session; and send the received data from the real-time session to the second interface.

Accordingly, the interface of the preferred embodiments can also generate an endpoint URL for a web server that hosts the content of the web page, preferably a web application that the user is accessing. The interface may then receive data from the real-time server that originated from the web server which the interface may forward to the web page the user is accessing. Therefore, the web server can send data in real-time to the user of the content of the web page it is hosting using the preferred embodiments of the present invention which overcomes the drawbacks and disadvantages of the conventional methods and systems.

According to a fifth aspect of the present invention there is provided a system for providing real-time communication comprising the interface and the second interface.

According to a sixth aspect of the present invention there is provided a computer program product comprising computer readable executable code for performing the steps of receiving a

log in request from said second interface; performing a log in to said real-time server in order to establish a real-time session on said real-timer server; receiving a unique identifier from the real-time server identifying said real-time session; receiving data from said second interface for the real-time session; sending said received data from said second interface to said real-time session; receiving data from said real-time session; and send said received data from said real-time session to said second interface.

According to a seventh aspect of the present invention there is provided a method of providing real-time communication comprising a first interface embedded in a web page wherein the first interface interfaces between content of the web page and a second interface, wherein the second interface interfaces between the first interface and a real-time server; the method comprising the steps of: the first interface sending a log in request to the second interface; the second interface performs a log in to the real-time server; the second interface receives a unique identifier from the real-time server wherein the unique identifier identifies a real-time session; the first interface sends data to the second interface for the real-time session; the second interface receives the data from the first interface and sends the data to the real-time session; the second interface receives data from the real-time session; the second interface sends the data received from the real-time session to the first interface; and the first interface receives the data from the real-time session from the second interface.

According to an eighth aspect of the present invention there is provided a method of operating a web server comprising the steps of: receiving an endpoint URL from a first interface; wherein the first interface is embedded in a web page and in-

terfaces between the web page and a second interface, wherein the second interface interfaces between the first interface and a real-time session hosted on a real-time server; storing the endpoint URL, wherein the endpoint URL identifies the real-time session; sending data to the endpoint URL such that the data is sent to the web page in real-time via the real-time session.

According to a ninth aspect of the present invention there is provided a web server, wherein the web server is adapted to: receive an endpoint URL from a first interface; wherein the first interface is embedded in a web page and interfaces between content of the web page and a second interface, wherein the second interface interfaces between the first interface and a real-time session hosted on a real-time server; store the endpoint URL, wherein the endpoint URL identifies the real-time session; send data to the endpoint URL such that the data is sent to the web page in real-time via the real-time session.

Accordingly, the preferred embodiments enable a web server to send data in real-time, via a real-time session, to the interface embedded in the web page. This advantageously enables a user, who has downloaded the web page to their terminal device where the web page comprises content such as a web application, to receive data from the web server that relates to the web application the user is accessing via the web page in real-time without the need to wait until the user's terminal, and in particular a web browser on the user's terminal, to request any data from the web server that is relevant to the web application the user is accessing.

The preferred embodiments of the present invention describe a framework for providing real-time communication between mul-

tiple users' accessing the same web page and content, preferably a web application, and between a user and a web server. The preferred embodiments also provide a generic reusable and independent framework for providing real-time communication so that the developer of the content of the web page, e.g. a web application, does not need to understand and develop a bespoke real-time communication handling system for each and every web application generated. The web application developer only needs to know the functionality provided by the interface and how to invoke the functionality of the interface embedded in the web page.

Accordingly, an advantage of the preferred embodiments of the present invention is that a web browser using Hypertext Transfer Protocol (HTTP) on the user's terminal device, e.g. a computer, does not have to poll the web server hosting the content in order to request for an update of any events that have occurred or receive any data from a further user's terminal device, e.g. a computer.

Preferred embodiments of the present invention will now be described, by way of example only, and with reference to the accompanying drawings, in which:

Figure 1 shows a real-time communication system in accordance with the preferred embodiments.

Figure 2 shows the interaction between the interfaces in accordance with the preferred embodiments.

Figure 3 shows the interaction between the content and the real-time communication system in accordance with the preferred embodiments.

Figure 4 shows the steps in creating a multi user real-time session in accordance with the preferred embodiments.

Figure 5 shows a signaling flow diagram for enabling a web server to send data in real-time to a user in accordance with the preferred embodiments.

With reference to Figure 1, which is a simplified diagram of a preferred embodiment of a real-time communication system 6, several users 1, 2 access a computer network 5, such as the Internet, company LAN (local area network), WLAN (wireless local area network), WAN (wide area network) and so on, each using a terminal device 3, 4. The terminal devices 3, 4 are preferably a personal computer but, as will be appreciated, the terminal devices 3, 4 may be any device capable of accessing a computer network, capable of browsing web sites, capable of downloading content from the web sites and capable of executing web applications such as a laptop, a Personal Digital Assistant (PDA), mobile terminal devices (e.g. a mobile phone) and so on.

A web server 8 is preferably operatively connected to the Internet 5 which hosts one or more web sites and one or more web applications associated with the web sites. A first user 1, via their terminal 3, downloads and accesses the content hosted on the web server 8 over the Internet 5 using a web browser on the user's terminal 3. As is known in the art, the web applications and web sites effectively remain on the web server 8 whilst an instance of the content of the web site and/or the web application is displayed by a web browser on the user's terminal device 3. The user 1 can interact with the web application displayed in the web browser even though the web application is executed on the web server 8 is transparent to the user 1.

The real-time communication system 6 also includes a real-time server 7 that is operatively connected to a computer network 5 such as the Internet. Preferably the real-time server 7 is based on the Extensible Messaging and Presence Protocol (XMPP) but can be based on any real-time protocols suitable for handling real-time events and communication such as the IP Multimedia Subsystem (IMS) protocol.

Both the web server 8 and the user terminals 3, 4 are preferably operatively connected to the real-time server 7 so that electronic communication can occur between the user's terminal 3, 4, the web server 8 and the real-time server 7.

Figure 1 shows the real-time server 7 and the web server 8 being separate severs based at different locations in the real-time system of the preferred embodiments. However, as will be appreciated, the real-time server 7 and the web server 8 may alternatively be the same server.

The user terminal device 3 requests a web page 16 from the web server 8. With reference to Figure 2, the web page 16 comprises content 9, for example, a web application which is programmed using, for example, Hypertext Markup Language (HTML), JavaScript, CSS etc. The content 9 is displayed to the user 1 by a web browser on the user's terminal device 3.

The web page 16 downloaded further includes an Application Program Interface (API) 10 which has been coded in a suitable language such as JavaScript. The JavaScript API 10 is the interface between the content 9 of the web page 16 displayed by the user's browser and a real-time component 11 that communicates with the real-time server 7.

As is known in the art, an API 10 enables the developer of a web page and/or a web application to invoke and access the functionality of pre-built software modules through defined data structures by calling subroutines that have been defined in the pre-built software modules. In the preferred embodiment, the JavaScript API 10 defines the interface between the web browser on the user's terminal 3, which is displaying the downloaded content 9 of the web page 16 and associated web applications, and the real-time component 11. Thus, the JavaScript API 10 defines the data structures and message format that the web browser must use to communicate with and invoke the subroutines that handle the real-time communication.

The real-time component 11 in the preferred embodiments is downloaded to the user's terminal device 3 as part of the web page. However, as will be appreciated by a person skilled in the art the real-time component 11 does not need to be embedded in the web page 16 but can either be hosted by a server, for example, the real-time server 7 or downloaded to the user's terminal device 3 separate from the web page 16.

The real-time component 11 is preferably implemented using Adobe's ™ Flash Technology as Flash™ is widely supported, supports socket API, supports an external interface to JavaScript (which is the preferred implementation of the API), has cross domain support and the Flash ™ component can be re-used in many different web sites and web applications.

However, as would be appreciated, the real-time component 11 could be implemented by any means which is capable of communicating with the real-time server 7 and the real-time network, such as Flex.

The real-time component 11 facilitates the log in to the real-time service, e.g. the XMPP service, on the real-time server 7 and also facilitates the exchange of data between multiple user terminals 3, 4.

As shown in Figure 2 the API 10 can send 14 data to and receive 12 data from the content 9, which is preferably a web application. The API 10 can also send 13 data to and receive 15 data from the real-time component 11.

Referring to Figure 3, the content 9 of the web page 16 being displayed by a web browser on the user's terminal device is downloaded from the web server 8 over the computer network 5. The content 9 can also communicate and interact with the web server 8 over the computer network 5, e.g. the Internet. The real-time component 11 can also be downloaded to the user's terminal device as part of the web page from a server 17 and the real-time component 11 can also communicate with the real-time server 7 over the computer network 5.

Referring to Figure 4, a first user in step 18 downloads content from a web server via the Internet to the first user's terminal device. The web page is displayed by a web browser on the first user's terminal device so that the content, e.g. a web application, can be accessed and interacted with by the first user.

The JavaScript API embedded within the web page downloaded to the first user's terminal sends a log in request to the real-time component that is, in the preferred embodiments, also downloaded to the first user's terminal device. The real-time component on receipt of the log in request facilitates the initiation of a real-time session for the content of the web page, e.g. the web application. On receipt of the func-

tion call from the JavaScript API the real-time component sends a login request from the web browser to the real-time server.

The real-time component may perform a log in to the real-time server anonymously, in other words log in to the real-time server without any authentication of the first user, or, if required, log in with a username and password combination in order to authenticate the first user with the real-time server. If a username and password combination is required in order to log in to the real-time server then the web browser on the first user's terminal device may request that the first user inputs a username and password combination. Alternatively, the username and password combination may be stored on the user's terminal device, for example, as a cookie, and provided automatically once the first user downloads the content from the web server to their terminal device and uses the downloaded content.

The real-time server, on receipt of either the anonymous login request or the login request containing the first user's username and password, verifies the login request and returns a login response. The authentication process operates according to the protocol being used, e.g. XMPP or IP Multimedia System or SIP. The login response returned by the real-time server informs the real-time component of the success or failure of the login request.

If the login request has failed then the first user may be prompted to try and log in to the real-time server again. The process for re-trying to log in to the real-time service may continue indefinitely or may prevent further attempts to log in after a set predetermined number of attempts, for example, the first user may be allowed only three attempts to

log in before the first user is prevented from logging in to the real-time service and may be prompted to contact the administrator of the real-time service and/or the content of the web page.

If the login request is successful the response from the real-time server includes a unique identifier for the real-time session that has been initiated. For example, the unique identifier may be a Jabber Identity (JID) or a Session Initiation Protocol (SIP) identity.

The real-time component on receipt of the unique identifier can then generate a link to the real-time session that has been initiated for the content of the web page that the first user is accessing and interacting with. Alternatively, the real-time component may forward the unique identifier to the web application which may then generate the link to the real-time session.

The real-time session may be used to provide a person to person session or a multi-user chat room session or any other session that requires real-time communication between multiple users. For example, if the first user wishes to participate in a person to person session for the content of the web page, e.g. a web application, then the link created by the real-time component could be, for example:

http://www.app.com/?jid=A_anon_jid

However, if the first user wishes to participate in a multi-user chat room session then the link created by the real-time component could be, for example:

http://www.app.com/?roomid=roomjid&password=thepasswordfortheroom

In the above examples, www.app.com is the link to the web application that is the content of the web page downloaded by the first user and is the content that the first user wishes to share with further users. The JID identifiers in the above examples, e.g. roomid=roomjid, enable the initiated real-time session to be identified. In the example of the multi user chatroom the link may further comprise a room password.

In step 19, the first user sends 22 the link to one or more further users so that the further users can download the same content from the web server and log in to the real-time session identified in the link. The first user and the further users will often be at different locations and each of the users will access the web server using their own terminal device, which as described above is preferably a computer but may also be for, example, any mobile terminal device.

The further users often wish to, or need to, interact with each other on particular content of a web page. In order to be able to interact with each other the multiple users join and participate in the same session of the web application that the first user has initiated. However, in conventional systems the web server hosting the content of the web page will control the transmission of information and messages between the multiple users.

However, this conventional approach has several disadvantages. For example, the web server is unable to handle real-time communication between the users effectively as it is based on HTTP, which is a request-response protocol. There-

fore, any data sent by one user will not be transmitted to any of the further users until each of the further user's terminal devices have polled the web server for any new data since the previous time the web server was polled. In conventional systems the developer of the web application would have to implement a bespoke and complex real-time communication handling system based on the HTTP response - request model for the content, e.g. a bespoke and separate real-time communication handling system for each and every web application, which is ineffective and inefficient.

The preferred embodiments of the present invention enable a real-time communication system 6 to be implemented which effectively bypasses the web server in order to send data between the multiple users in real-time. In the preferred embodiments the users log into and join a real-time session hosted by a real-time server which handles the real-time communication independently of the content of the web page and the web server. Moreover, the real-time communication can be initiated by a generic and re-useable JavaScript API embedded in a web page and therefore the developer of the web application does not need any knowledge of the real-time communication when developing the web application.

In step 23, the one or more further users on receipt of the link from the first user activate the link by, for example, clicking on the link using a mouse connected to the one or more further user's terminal device.

Once the link, which includes an identifier of the web page, is activated the identified web page containing the content and the JavaScript API is downloaded to and displayed to the further user by a web browser on their terminal device. The

real-time component is also preferably downloaded to the further user's terminal device as part of the web page.

In step 20, the first user waits for any one of the further user to join the real-time session the first user had initiated.

In step 24, the real-time component downloaded to the further users' terminal devices sends a login request to the real-time server identifying the real-time session that the further user wishes to join. As described above, link sent to the further user from the first user includes an identifier of the real-time session the first user has initiated. The log in request for the further user is preferably an anonymous log in request or, if required, the log in request may comprise a username and password combination for the further user. If the log in request is successful then the further user has joined the real-time session the first user initiated. On joining the real-time session the real-time component downloaded to the further users' terminal device sends 26 a message to the first user informing the first user that the further user has joined the real-time session the first user initiated.

In steps 21 and 25 the first user and any further users are connected to the same real-time session for the content, e.g. web application, which they are all using. Therefore, any data that one or more of the users which needs to be sent to one or more of the other users connected to the same real-time session can transmit that data in real-time to the other users. The data will be sent from the user's terminal device via the JavaScript API and the real-time component, which have been downloaded to the user's terminal device, to the real-time session hosted on the real-time server. The real-

time server on receipt of the data can send the data to one or more of the other users connected to the real-time session by transmitting the data to the real-time components on each of the other user's terminal devices. The data will be displayed by the web browser, via the JavaScript API, on each of the other users' terminal devices.

An example message that may be transmitted from one user to another user via the real-time session and real-time framework described above is

```
<message from='user_a@webapp.net'to='user_b@webapp.com'
xml:lang='en'>
<x xmlns="real-time.extension">
<data>this contains data from the web application provider
using the push messaging service.</data>
</x>
</message>
```

Using the example of a map web application as the content of the downloaded web page, if two or more users share and participate in the same map web application then the preferred embodiments enable any actions taken by one of the multiple users, for example, move the map, zoom in/out of the map, move a pointer around the map and so on, to be displayed in real-time in each of the other user's web browser thereby synchronizing the views of the map web application displayed in each user's web browser. The multiple users can also exchange messages in real-time and interact without any delays being incorporated by the conventional request-response arrangement using HTTP. Moreover, the preferred embodiments describe a mechanism for enabling real-time communication that can be used and re-used by any web application without the developer of the web application needing to develop a be-

spoke real-time communication handling system for each web application.

In a further embodiment the same framework for real-time communication can be used to enable the web server 8 hosting the content, such as a web application, to send data in real-time to one or more users using the content.

As discussed hereinabove, the web server conventionally uses HTTP to communicate with the user's terminal device which is a request-response arrangement. Therefore, any data that needs to be transmitted to the user's terminal device regarding the content they are using can, in conventional systems, be transmitted to the user's terminal device after the web browser on the user's terminal device has polled the web server. This delay in the user receiving data regarding events that have occurred with the content that they are using means that any content which requires real-time communication from the web server to the user's terminal cannot be implemented effectively or efficiently in conventional systems.

Therefore, in an embodiment of the present invention the real-time framework described above can be used to enable the web application on the web server to push in real-time data to one or more user's terminal devices.

In this embodiment the user logs into the real-time server to obtain a unique identifier for the real-time session that is created for the content embedded in a web page that is downloaded to the user's terminal device in the same manner as described above. In other words, and with reference to Figure 5, the user's terminal device 3 requests 27 a web page and its content e.g. a web application from the web server 8.

The web page and its content are downloaded 28 to the user's terminal device 3 from the web server 8 in response to the request. The real-time component is also preferably downloaded to the user's terminal device 3 as part of the web page. The JavaScript API embedded within the web page downloaded to the user's terminal device 3 sends a log in request to the real-time component which, on receipt of the log in request, sends 29 a log in request to the real-time server 7. If the log in request is successful the real-time server 7 sends 30 a log in response to the real-time component on the user's terminal device 3 which includes a unique identifier such as a JID.

Once the user is logged into the real-time server 7 and therefore a real-time session has been created for the content of the web page the real-time component queries 31 the real-time server 7 for an endpoint URL. The real-time component queries 31 the real-time server 7 by sending the real-time server 7 a push message.

The endpoint URL defines an address on the real-time server 7 to which the web server 8 can push, in real-time, data relating to the content the user 1 is accessing to the user's terminal device 3.

In the preferred embodiments the real-time server 7 returns 32 the endpoint URL to the user's terminal device 3 via the real-time component. On receipt of the endpoint URL the web browser on the user's terminal device 3 forwards 33 the endpoint URL to the web server 8 which stores 34 the endpoint URL on the web server 8.

Preferably, the endpoint URL created comprises a base part that identifies the real-time server providing the real-time

session, for example, http://the.real.time.application.provider.com/pushapi.

Preferably, the endpoint URL also comprises a second part that includes the unique identifier of the user, for example, the JID or SIP identifier. Therefore, the endpoint URL generated would be, for example, http://the.real.time.application.provider.com/pushapi/unique JID indentifier.

However, as will be appreciated by those skilled in the art, the endpoint URL may be generated in any number of ways based on the requirements of the real-time server, web server and the protocols used.

Alternatively, instead of the web browser querying the real-time server via the real-time component for an endpoint URL the web server itself may generate the endpoint URL. In this embodiment the web server knows the web address of the real-time server and uses the unique identifier (e.g. the JID identifier) that is created when the user logs into the real-time server to create the real-time session. In this case, the user's terminal device sends only the unique identifier to the web server.

Once an event has occurred 35 with the content that the user 1 is accessing and which is hosted on the web server 8 then the web server needs to transmit data relating to the event in real-time to the user's terminal device 3. The web server 8 retrieves the endpoint URL and pushes 36 the data, preferably in the form of a message, to the address defined by the endpoint URL. The real-time server 7 receives the data from the web server 8 in real-time and transmits 37 the data in real-time to the user's terminal device 3. The web browser

on the user's terminal device 3 receives the data via the real-time component and the JavaScript API and processes the data in order to modify display of the web page and content in the web browser in accordance with the data received from the real-time session.

In the preferred embodiments an example message transmitted from the web server to the real-time server is:

```
POST /pushapi/unique JID indentifier HTTP 1.0
Host: the.real.time.application.provider.com
Content-Type: applicat ion/x-www-form-urlen coded
Content-Length: 92

data= this contains data from the web application provider
using the push messaging service.
```

In the preferred embodiments, as discussed above, the real-time server is an XMPP server and an example message is transmitted from the real-time server to to the user:

```
<message from='webapp@webapp.net' to='browser_user@real-time-
provider.com' xml:lang='en'>
<x xmlns="real-time.extension">
<data>this contains data from the web application provider
using the push messaging service.</data>
</x>
</message>
```

However, a person skilled in the art will appreciate that the messages containing the data may be transmitted using different protocols such as, HTTP FORM POST, XML-RPC, REST or SOAP.

Accordingly, data regarding an event that occurs with the content, e.g. a web application, that is hosted on the web server and that the user is viewing and interacting with is automatically transmitted to the web browser on the user's terminal device in real-time. As such, the user does not have to wait for their web browser to poll the web server to request data relating to an event that may have occurred since the web browser previously polled the web server.

Therefore, the preferred embodiments of the present invention enable real-time communication between multiple users interacting and sharing a web application and between a web server and at least one user. Moreover, the preferred embodiments describe a real-time framework that can be used by any web application so that individual web applications do not require a special bespoke solution to be developed solely for each web application. Therefore, the developer of the web application only needs to know the API embedded in the web page associated with the web application and not how to develop a real-time communication system which is often difficult and complex to achieve.

While preferred embodiments of the invention have been shown and described, it will be understood that such embodiments are described by way of example only. Numerous variations, changes and substitutions will occur to those skilled in the art without departing from the scope of the present invention as defined by the appended claims. Accordingly, it is intended that the following claims cover all such variations or equivalents as fall within the spirit and the scope of the invention.

Claims

1. An interface embedded in a web page, wherein said interface interfaces between content of said web page and a second interface, wherein said second interface interfaces between said interface and a real-time server;

said interface is adapted to:

send a log in request to said second interface in order to establish a real-time session on said real-time server;

send data to said real-time session via said second interface; and

receive data from said real-time session via said second interface.

2. The interface as claimed in claim 1 in which said interface is further adapted to receive an identifier of the said real-time session.

3. The interface as claimed in claim 2 in which said interface is further adapted to:

send said unique identifier to said content on a web server; and

receive a link from said content on said web server wherein said link identifies said real-time session.

4. The interface as claimed in claim 1 in which said interface is further adapted to receive a link from said second interface wherein said link identifies said real-time session.

5. The interface as claimed in any one of the preceding claims in which said interface embedded in said web page is

downloaded to a first terminal device on request of a first user.

6. The interface as claimed in any one of claims 3 to 5 in which said first user via said first terminal device sends said link to one or more further users at one or more further terminal devices;

said interface embedded in said web page is downloaded to said one or more further terminal devices when said one or more further users activates said link; and

said interface is further adapted to:

send a log in request to said second interface for the one or more further users; and

receive a log in response from said second interface; and

7. The interface as claimed in any one of the preceding claims in which said content of said web page comprises content of a web application, wherein said web application is hosted by a web server.

8. The interface as claimed in claim 7 in which said interface interfaces between said content of said web application and said second interface.

9. The interface as claimed in any one of claims 3 to 8 in which said link comprises an identification of said web page.

10. The interface as claimed in any one of claims 3 to 9 in which the link comprises a unique identifier, wherein said unique identifier identifies said real-time session.

11. The interface as claimed in any one of the preceding claims in which said real-time session is hosted by a real-time server.

12. The interface as claimed in any one of the preceding claims in which said data sent to said real-time session via said second interface comprises data to be sent in real-time to a terminal device that is also logged into said real-time session.

13. The interface as claimed in any one of the preceding claims in which said data received from said real-time session via said second interface comprises data sent in real-time from a terminal device that is also logged into said real-time session.

14. The interface as claimed in any one of the preceding claims in which said interface is further adapted to:

send a request for an endpoint URL to said second interface; and

receive said endpoint URL from said second interface wherein said endpoint URL identifies said real-time session.

15. The interface as claimed in claim 14 in which said interface is further adapted to send said endpoint URL to a web server hosting said web page.

16. The interface as claimed in claim 14 or 15 in which said endpoint URL comprises an address of the real-time server and an identification of the real-time session.

17. A system for providing real-time communication comprising said interface as claimed in any one of claims 1 to 16 and said second interface.

18. A computer program product comprising computer readable executable code for performing the steps of:

sending a log in request to said second interface in order to establish a real-time session on said real-time server;

sending data to said real-time session via said second interface; and

receiving data from said real-time session via said second interface.

19. An interface wherein said interface interfaces between a second interface and a real-time server, wherein said second interface interfaces between a content of a web page and said interface;

said interface is adapted to:

receive a log in request from said second interface;

perform a log in to said real-time server in order to establish a real-time session on said real-timer server;

receive a unique identifier from the real-time server identifying said real-time session;

receive data from said second interface for the real-time session;

send said received data from said second interface to said real-time session;

receive data from said real-time session; and

send said received data from said real-time session to said second interface.

20. The interface as claimed in claim 19 further adapted to:

receive a unique identifier of said real-time session from said real-time server.

21. The interface as claimed in claim 20 further adapted to:

send said unique identifier to said second interface.

22. The interface as claimed in claim 20 further adapted to:

generate a link, wherein said link comprises said unique identifier; and

send said link to said second interface;

23. The interface as claimed in any one of claims 19 to 22 in which said step of performing said log in further comprises the step of sending said log in request to said real-time server and receiving a log in response from said real-time server.

24. The interface as claimed in claim 22 or 23 in which said link further comprises an identification of said web page.

25. The interface as claimed in any one of claims 19 to 24 in which said interface is further adapted to send said received data from said second interface to said real-time session.

26. The interface as claimed in any one of claims 19 to 25 in which said interface is further adapted to send said received data from said real-time session to said second interface.

27. The interface as claimed in any one of claims 19 to 26 in which said interface is further adapted to:

receive a request for an endpoint URL from said second interface;

generate said endpoint URL, wherein said endpoint URL identifies said real-time session; and

send said endpoint URL to said second interface.

28. The interface as claimed in claim 27 in which said end-point URL comprises an address of the real-time server and an identification of said real-time session.

29. The interface as claimed in claim 27 or 28 in which said interface is further adapted to:

receive data in real-time from a web server via said real-time session; and

send said received data from said real-time session to said second interface.

30. A system for providing real-time communication comprising the interface as claimed in any one of claims 19 to 29 and said second interface.

31. A computer program product comprising computer readable executable code for performing the steps of:

receiving a log in request from said second interface;

performing a log in to said real-time server in order to establish a real-time session on said real-timer server;

receiving a unique identifier from the real-time server identifying said real-time session;

receiving data from said second interface for the real-time session;

sending said received data from said second interface to said real-time session;

receiving data from said real-time session; and

send said received data from said real-time session to said second interface.

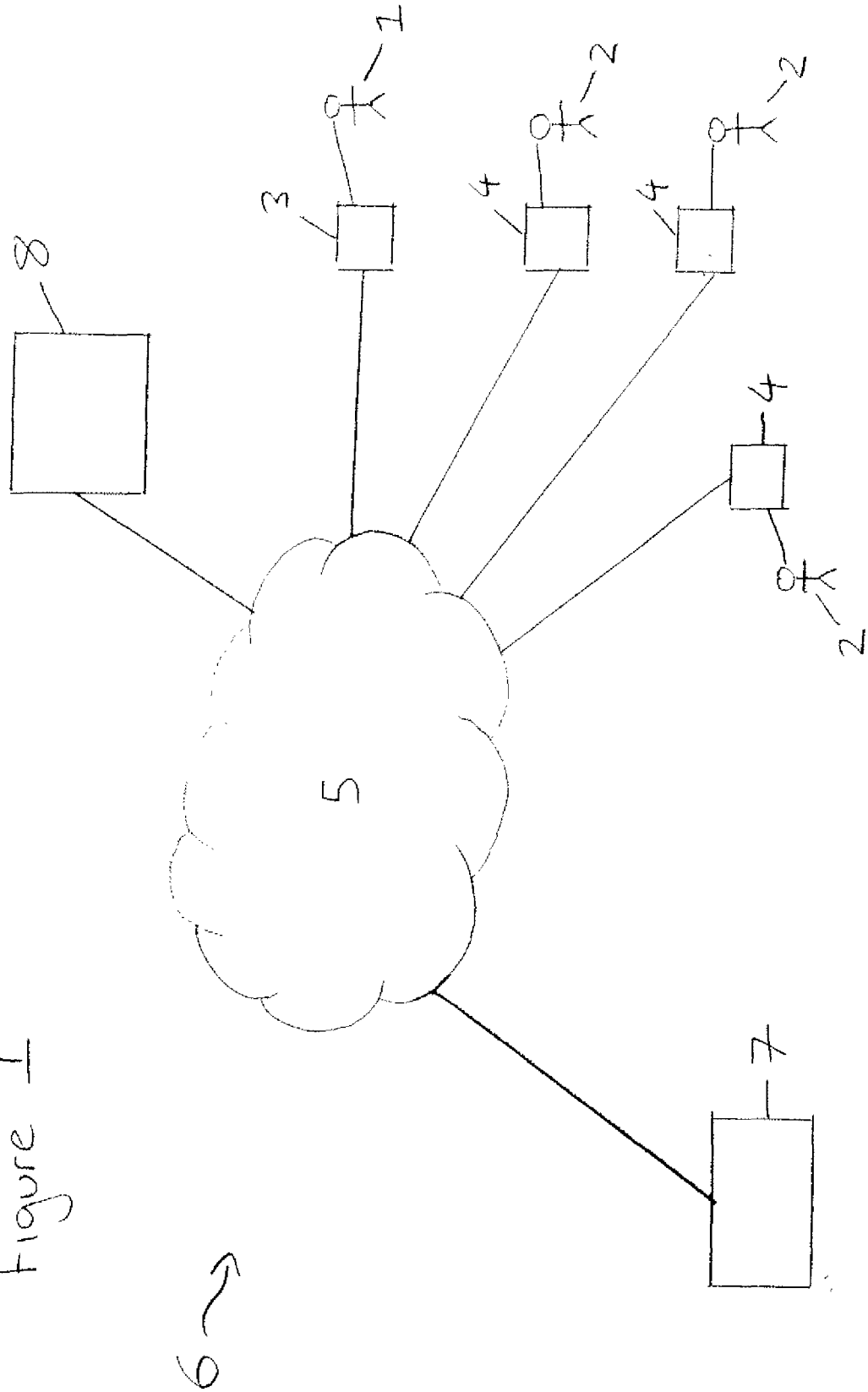


Figure 1

Figure 2

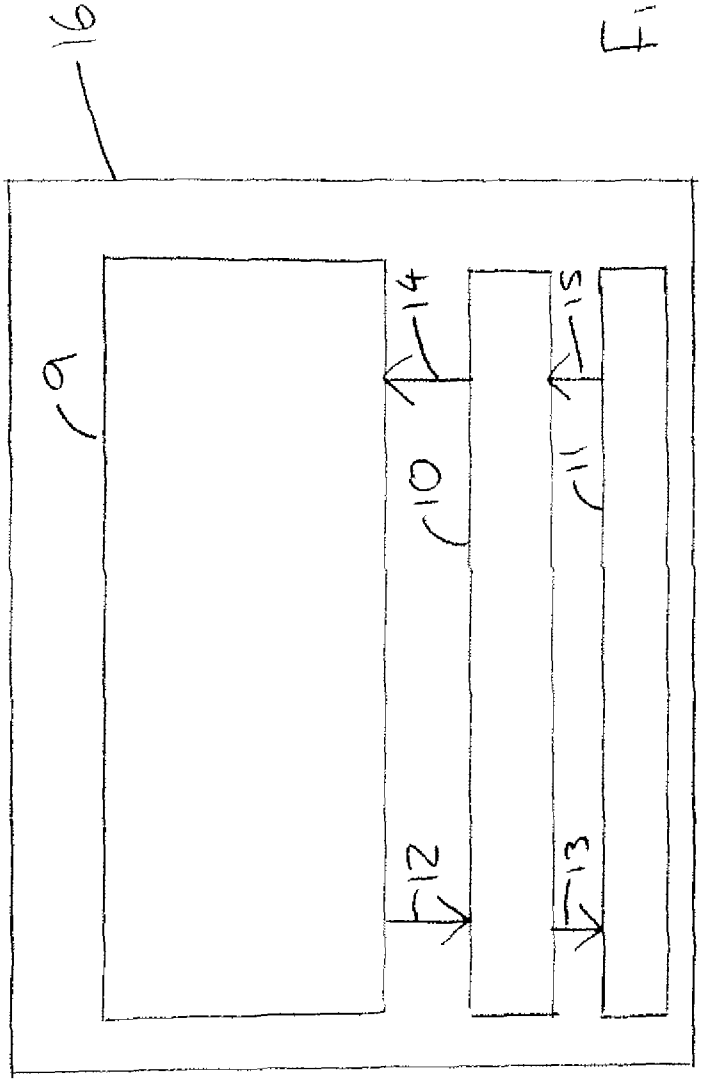


Figure 3

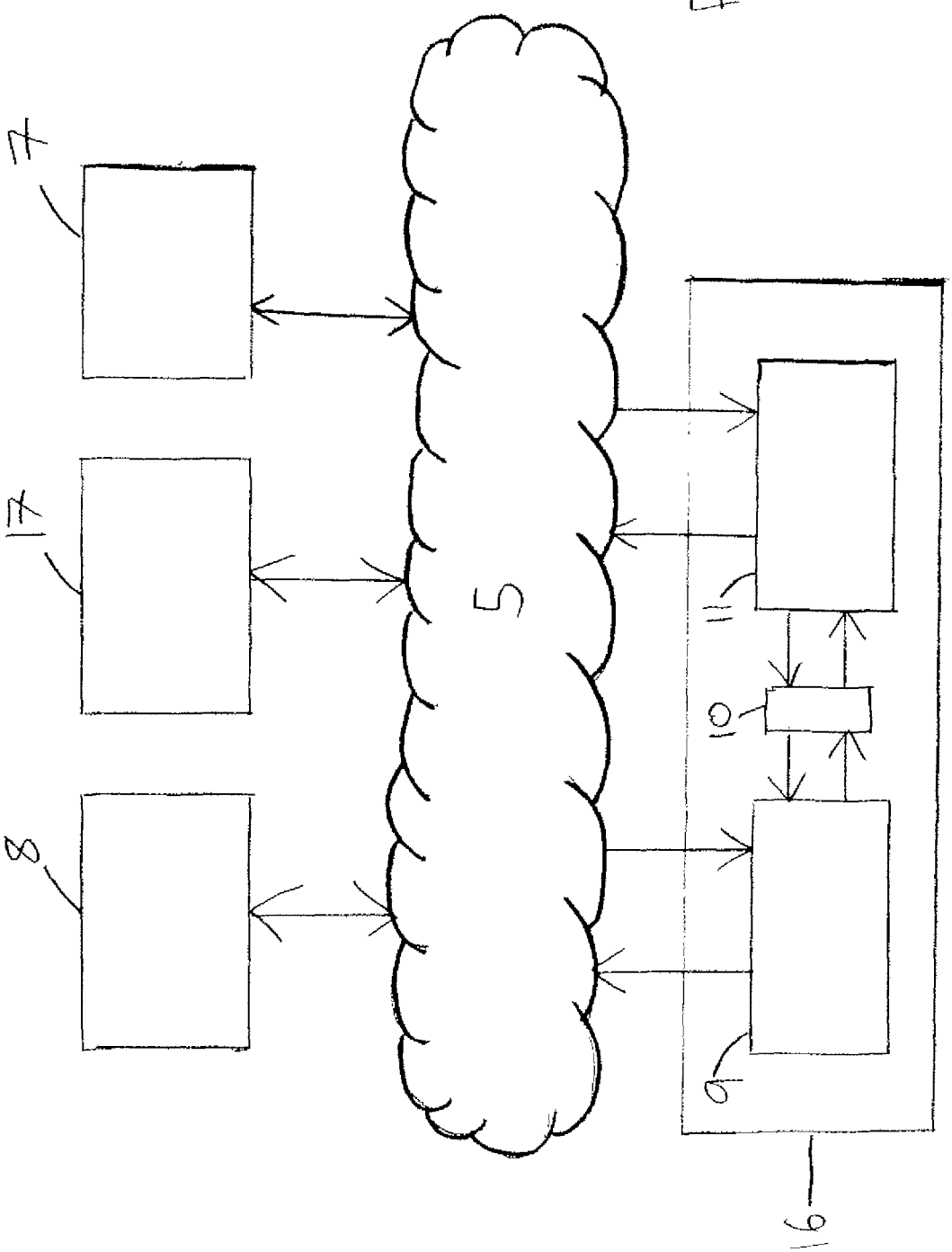


Figure 4

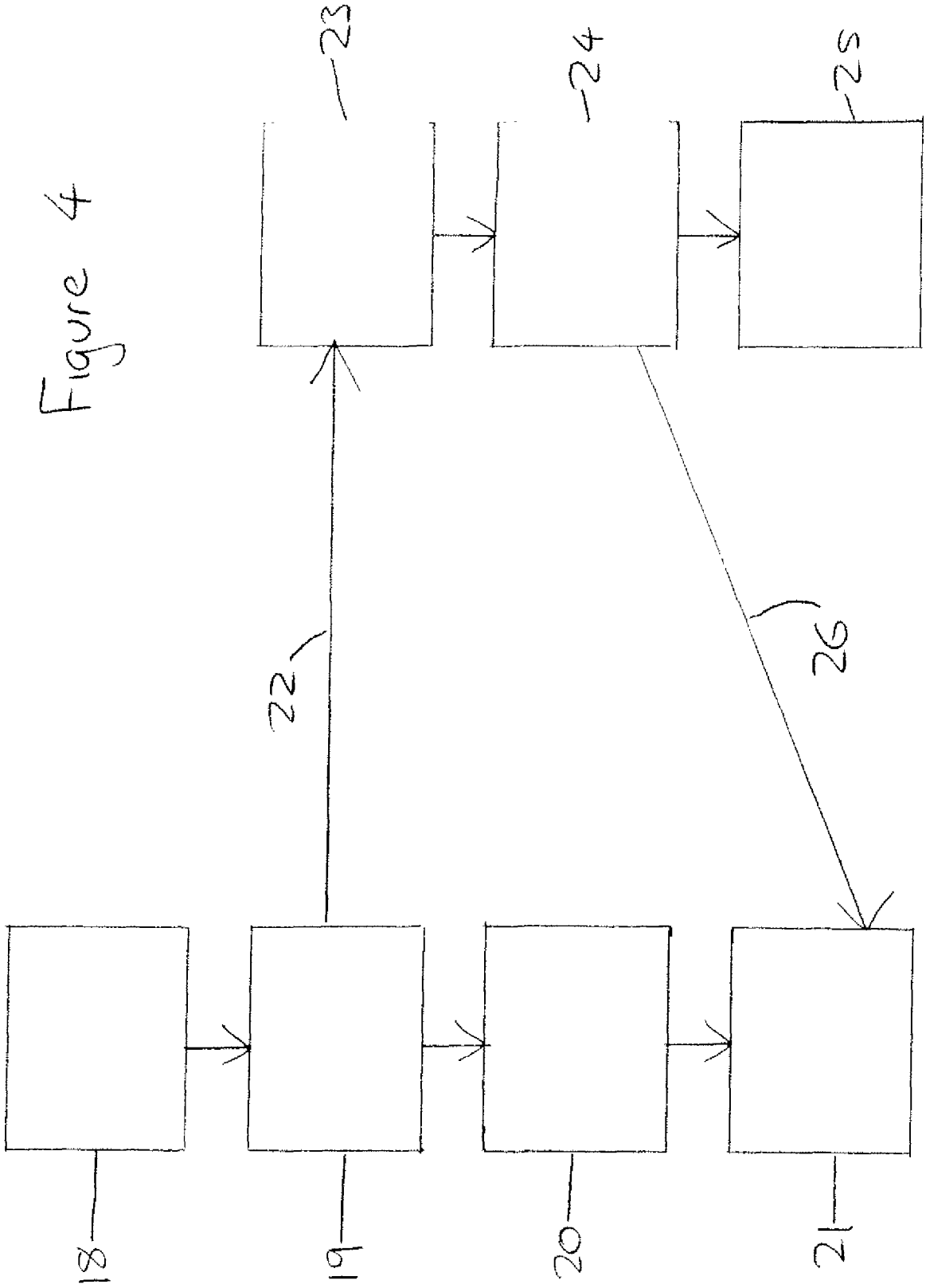
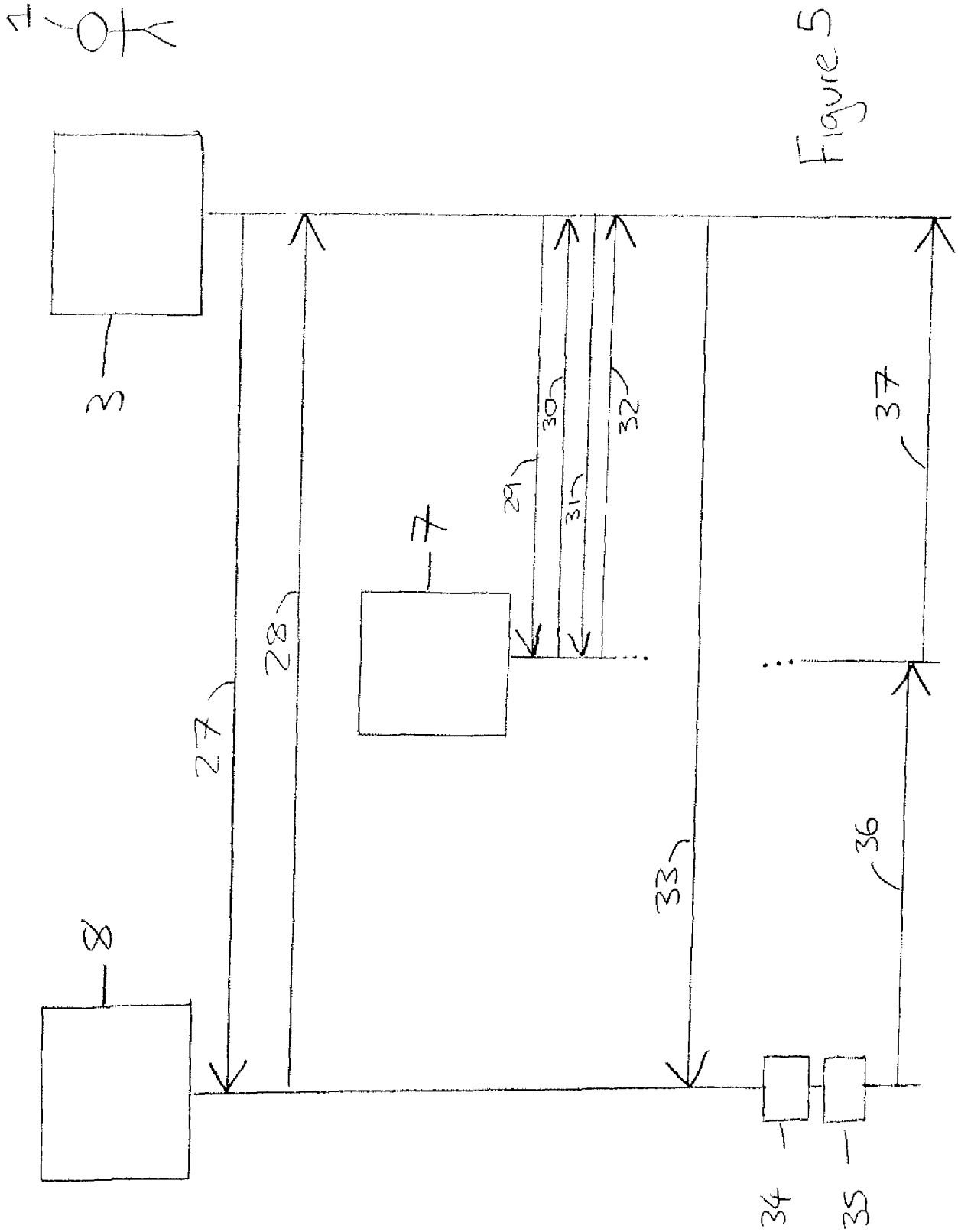


Figure 5



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2008/066196

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/06 H04L29/08
ADD.

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)
H04L

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 practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2006/018707 A1 (NOKIA CORP [FI]; NOKIA INC [US]; YPYA TAPIO [FI]; MITTAL GAURAV [FI]) 23 February 2006 (2006-02-23) abstract page 2, line 22 - page 4, line 11 page 8, line 26 - page 3, line 20 claims 1-16	1-31
X	US 2006/259585 A1 (KEOHANE SUSANN M [US] ET AL) 16 November 2006 (2006-11-16) abstract paragraph [0013] - paragraph [0016] paragraph [0040] - paragraph [0044]	1-31

☐ 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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
22 April 2010	29/04/2010

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Fax: (+31–70) 340–3016	Hilbig, Sophie

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2008/066196

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2006018707 A1	23-02-2006	BR PI0517352 A	07-10-2008
		CA 2577638 A1	23-02-2006
		CN 101032121 A	05-09-2007
		EP 1779594 A1	02-05-2007
		JP 2008511049 T	10-04-2008
		KR 20070039173 A	11-04-2007
		US 2006039561 A1	23-02-2006
US 2006259585 A1	16-11-2006	NONE	