



- (51) **International Patent Classification:**
G06F 12/00 (2006.01) *H04N 21/218* (2011.01)
G06F 9/312 (2006.01)
- (21) **International Application Number:**
PCT/FI2013/050567
- (22) **International Filing Date:**
24 May 2013 (24.05.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** NOKIA CORPORATION [FI/FI]; Karakaari
7, FI-02610 Espoo (FI).
- (72) **Inventor:** PAK, Yongbeom; Ylioppilaantie 8 B 20, FI-
90130 Oulu (FI).
- (74) **Agents:** NOKIA CORPORATION et al.; IPR Depart-
ment, Jussi Jaatinen, Karakaari 7, FI-02610 Espoo (FI).
- (81) **Designated States** (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU,
RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ,
TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA,
ZM, ZW.

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

(54) **Title:** MEMORY USAGE OPTIMIZATION METHOD

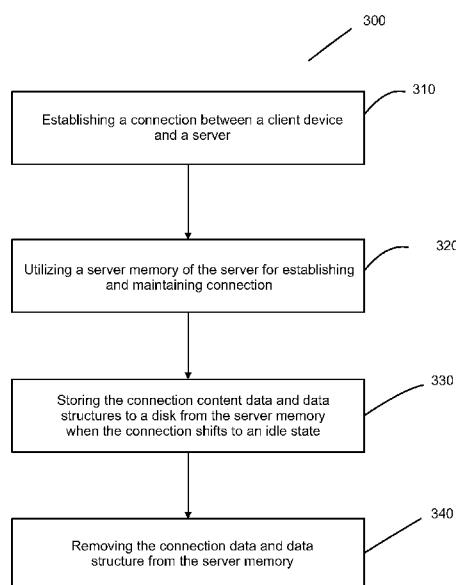


Fig. 3

(57) **Abstract:** The invention relates to memory usage optimization. A method for memory usage optimization comprises establishing a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices, utilizing a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data and data structures, storing the connection content data and data structures to a disk from the server memory when the connection shifts to an idle state, and removing the connection data and data structure from the server memory.



Memory usage optimization method

Background

5 Today a mobile user expects immediate access to fresh, relevant data in the internet. Fresh data can be brought to users, for example, by real-time push notifications. Before an application can receive notifications, the user has to subscribe to information channels provided by a server (i.e. a connection is established between the user's device [i.e. a client] and the server);
10 whenever new content is available on one of those channels, the server pushes that information out to the client. Real-time push notifications are a battery saving technology that allows instant updates without polling. Such a service may utilize data compression algorithm to reduce size of message that need to be delivered to user device.

15 However, the increasing use of internet services demands a high number of concurrent connections from the server. The high number of concurrent connections requires high resource usage in the server. Server memory is utilized for establishing and maintaining connection of a user with the server.
20 Generally the server does not terminate a connection after response data has been served to a user. The server leaves the connection open so that if an event is received by the server or service, it can be sent immediately to one or multiple users; otherwise, the data would have to be queued until the client's next request is received. This high memory utilization limits the
25 number of active users served by a single server. Each user connection may require 300kB or more memory. The memory is required for user authentication, for example, through ZLIB compression algorithm (i.e. 300KB memory) and for real-time push notification service, for example through ZLIB and/or Transport Layer Security (TLS) stream compression.

30

Summary

The present application relates generally to memory usage optimization of a server, for example, a real-time push notification server, a video or audio
35 streaming server or any other suitable server that uses persistent connections. Now there has been invented an improved method and technical equipment implementing the method. Various aspects of the invention include a method, an apparatus, and a computer readable medium

comprising a computer program stored therein, which are characterized by what is stated in the independent claims. Various embodiments of the invention are disclosed in the dependent claims.

5 According to a first aspect of the invention, there is provided a method, comprising establishing a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices, utilizing a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data
10 and data structures, storing the connection content data and data structures to a disk from the server memory when the connection shifts to an idle state, and removing the connection data and data structure from the server memory.

15 According to an embodiment, the server is a real-time push notification server. According to an embodiment, the method further comprises restoring the connection data and data structure to the server memory when server is indicated about a real-time push notification to be delivered to the client device. According to an embodiment, the server is a video or audio
20 streaming server. According to an embodiment, the connection content data and data structures are stored to the disk after the connection has been in the idle state for a predetermined time. According to an embodiment, the connection content data and data structures are stored to the disk in the idle state if the server has more than a predetermined number of concurrent
25 connections with client devices. According to an embodiment, the connection content data and data structures are stored to the disk in every second or third idle state. According to an embodiment, the connection content data and data structures are stored to the disk for idle state if the server memory capacity is running low.

30 According to a second aspect of the invention, there is provided an apparatus comprising at least one processor, memory including computer program code, the memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the
35 following: establish a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices, utilize a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data

and data structures, store the connection content data and data structures to a disk from the server memory when the connection shifts to an idle state, and remove the connection data and data structure from the server memory.

5 According to an embodiment, the server is a real-time push notification server. According to an embodiment, the computer program code configured to, when executed on at least one processor, further cause the apparatus to restore the connection data and data structure to the server memory when server is indicated about a real-time push notification to be
10 delivered to the client device. According to an embodiment, the server is a video or audio streaming server. According to an embodiment, the connection content data and data structures are stored to the disk after the connection has been in the idle state for a predetermined time. According to an embodiment, the connection content data and data structures are stored
15 to the disk in the idle state if the server has more than a predetermined number of concurrent connections with client devices. According to an embodiment, the connection content data and data structures are stored to the disk in every second or third idle state. According to an embodiment, the connection content data and data structures are stored to the disk for idle
20 state if the server memory capacity is running low.

According to a third aspect of the invention, there is provided a computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at
25 least one processor, cause an apparatus to establish a connection between a client device and a suitable to have a plurality of connections with a plurality of client devices, utilize a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data and data structures, store the connection
30 content data and data structures to a disk from the server memory when the connection shifts to an idle state, and remove the connection data and data structure from the server memory.

35 According to an embodiment, the server is a real-time push notification server. According to an embodiment, the computer program code configured to, when executed on at least one processor, further cause the apparatus to restore the connection data and data structure to the server

memory when server is indicated about a real-time push notification to be delivered to the client device.

According to a fourth aspect of the invention, there is provided an apparatus, comprising means for establishing a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices, means for utilizing a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data and data structures, means for storing the connection content data and data structures to a disk from the server memory when the connection shifts to an idle state and means for removing the connection data and data structure from the server memory.

According to an embodiment, the server is a real-time push notification server. According to an embodiment, the apparatus further comprises means for restoring the connection data and data structure to the server memory when server is indicated about a real-time push notification to be delivered to the client device. According to an embodiment, the server is a video or audio streaming server. According to an embodiment, the connection content data and data structures are stored to the disk after the connection has been in the idle state for a predetermined time. According to an embodiment, the connection content data and data structures are stored to the disk in the idle state if the server has more than a predetermined number of concurrent connections with client devices. According to an embodiment, the connection content data and data structures are stored to the disk in every second or third idle state. According to an embodiment, the connection content data and data structures are stored to the disk for idle state if the server memory capacity is running low.

Description of the Drawings

In the following, various embodiments of the invention will be described in more detail with reference to the appended drawings, in which

Fig. 1 shows a real-time push notification system with a server memory optimizing according to an example embodiment;

Fig. 2 shows a chart of a real-time push notification system with a server memory optimizing according to an example embodiment; and

5 Fig. 3 shows a flow chart of a server memory optimizing method for a real-time push notification system according to an example embodiment.

Description of Example Embodiments

10

Real-time push notifications relating to information preferences expressed in advance for a push notification server are a convenient way of keeping mobile users in touch with fresh and relevant information, for example, in the internet. Server memory is utilized for establishing and maintaining connection of the client with the push notification server for real-time push notifications. The server memory is required for user authentication, for example, through ZLIB compression algorithm and for real-time push notification service, for example through ZLIB and/or Transport Layer Security (TLS) stream compression. Each connection between the client and the push notification server may require 300KB or more server memory. This high server memory utilization may limit the number of active users served by a single push notification server. In order to enable real-time push notifications to be delivered to everyone who wants to receive them – regardless of which mobile, desktop or media device they happen to be using – the push notification server memory utilization should be optimized so that the memory utilization does not so intensely limit the number of active users served by a single push notification server.

15

20

25

30

35

Each time when the connection between a client and a push notification server switches to an idle state i.e. to a state, where no notification is needed to be delivered to the client, the push notification server moves connection information comprising data and data structures, for example, internal data structure required for ZLIB compression or other compression method, state of user/client session, user/client specific information or any other information that is needed to provide service to the user/client, to a disk e.g. a non-volatile, persistent memory and destroys them from the server memory location e.g. RAM, thereby freeing the server memory for other active connections to overcome the memory limit of the push

notification server. For any real-time push notification delivery from the server to the client in future, the server restores the client's connection information i.e. the data and data structures from the disk back to the server memory.

5

Thus, by moving the connection information to the disk for the idle state worth, a single push notification server can serve an increasing number of active client connections on the push notification server. This server memory capacity discharging for the idle state may be called a server memory optimizing method.

10

In the following, several embodiments of the invention will be described in the context of an apparatus, for example, a push notification server. The server is suitable to transmit real-time push notifications to a client i.e. a client device comprising an internet-access i.e. means for connecting the device to the computer networks, such as the world wide web (WWW) by mobile, wired or wireless connection, for example, WAN or WLAN. The push notification server is a system, hardware and software, that may respond to requests across a computer network to provide, or help to provide, a network service. Servers can be run on a dedicated computer, which may be referred to by a term "the server", but many networked computers are capable of hosting servers. In many cases a computer can provide several services and have several servers running. Servers may operate within a client-server architecture; servers are computer programs running to serve the requests of other programs, the clients. Thus, the server performs some task on behalf of clients. The clients either run on the same computer, or they connect through the computer network. The device i.e. a mobile client may be, for example, a, mobile phone, a mobile computer, a mobile collaboration device, a mobile internet device, a smart phone, a tablet computer, a tablet personal computer (PC), a personal digital assistant, a handheld game console, a portable media player, a digital video camera (DVC or digital camcorder), a pager, or a personal navigation device (PND).

15

20

25

30

Fig. 1 shows a real-time push notification system 100 with a server memory optimizing method according to an example embodiment. The system comprises a device i.e. a client device 110, an apparatus i.e. a push notification server 130 comprising a real-time push notification service 140 and a service's application server 150. A Client API (Application

35

Programming Interfaces) which is arranged to handle the connection between user device and the server resides on the client device 110 and allows applications 120 on the client device 110 to receive notifications. For receiving push notifications, for example, real-time push notifications, the client should log in to the notification service 140, for example, by using notification ID. The notification ID uniquely identifies both the end user (client device 110) and the target application 120 and is used by the notification service 140 to properly route notifications. The logging in to the notification service 140 may be done by connecting the client device 110 with the server 130, for example, through ZLIB compression. User authentication through ZLIB compression may require, for example, 300kB server memory 160. The server memory 160 may be, for example, RAM type memory. So, the server memory 160 is utilized for establishing (authentication and logging in) the connection between the client device 110 with server 130, but also for maintaining the connection of the client device 110 with server 130. And when the client device 110 is connected to the notification service 140 of the server 130, it is ready to receive available real-time push notifications from the notification service 140. The real-time push notification to the authenticated and connected client device 110, for example, through ZLIB and/or TLS stream compression requires the above mentioned connection information, but also capacity of the server memory 160. Thus, the notification service 140 stores all the information needed to maintain the connection and to real-time push notification into the server memory 160 in connection with forming the connection. However, because the required server memory capacity of each connection may be quite high, for example, 300kB or more, and the notification service 140 of the server 130 serves many clients, the server 130 may transfer the connection information (data and data structures) to a separate disk 170 e.g. a non-volatile, persistent memory from the server memory 160, when the connection is in an idle state. The idle state is a state, where no notification is needed to be delivered to the client device 110 by the notification service 140. When the connection information is stored in the disk 170, the server 130 destroys the connection information from the server memory 160, thereby freeing the server memory 160 for other active connections. The disk 170 may be an internal memory of the server 130 or it may be an external memory. However, it is also possible that in the idle state the server 130 stores data to the disk 170 and removes data from the server memory 160, but keeps the data structures in the server memory.

A service's application server 150 generates notifications. Generated notifications are sent to the notification service 140. When the notification service 140 receives at least one notification that is specified by the notification ID, the server 130 restores the connection information (data and data structures) relating to that notification ID from the disk 170 to the memory 160. Then the notification service 140 delivers received notifications to the client device 110 relating to that notification ID over the connection established between the client 110 and the notification service 140. Once a notification is received on the client device 110, it will be delivered to the appropriate application 120 through the Notification Client API. Delivery can occur regardless of the current state of the target application 120, whether it is active, in the background, or even if it is not running, in which case it can be woken up and launched. Then the client device 110 may alert the user of the client device 110 by some means available on the device, i.e. sound, pop-up window, by opening the application etc. Application 120 receiving notifications may be, for example, a navigation application that alerts mobile device users to changes in optimal routes and even updates the weather conditions for their destinations or a music store that suggests music play lists based on the user's social network favorites or a news application that sends the latest headlines as soon as they are available.

Fig. 2 shows a chart of a real-time push notification system with a server memory optimizing according to an example embodiment. At the beginning, a user is authenticated normally with ZLIB compression (steps 201-207). In step 201, a mobile client starts to connect to a server. In step 202, the server indicates that ZLIB compression is available. In step 203, the mobile client indicates that it is going to use the compression method. The mobile client may initiate compression method in this step 203. In step 204, the server initiates ZLIB. In step 205, the server indicates that the ZLIB compression is initiated and ready to compress. In step 206, the mobile client initiates the compression method if it is not already done in step 203. All data that is transferred between the mobile client and the server in step 206 and/or further coming steps is compressed by compression method. In addition, in step 206, the mobile client logs in to the server by delivering its identification to the server. Identification includes at least a notification ID and may also include any forms of data that is needed to validate the user

session i.e. the connection between the mobile client and the server. In step 207, the server validates the client identification and indicates that the login has succeeded. This validation process may consist one step or more than one steps, for example, a digest access authentication requires a challenge response schema that includes more than one steps. In step 208, the server checks if there is any stored notification to be delivered to the mobile client. If yes i.e. there is at least one notification for the client, the server sends those existing notifications to the client, steps 209 and 210 (step 210 is optional). After sending at least one notification to the client or if there were any notifications to be sent (i.e. the server is in an idle state) the server stores connection data and possible data structures i.e. the ZLIB compression dictionary and state to a disk from the server memory and removes that stored data from the server memory, in step 211. In step 212, the server receives from a service a notification to be delivered to the client. In step 213, the server restores needed information for the notification delivery from the disk to the server memory. If data structure has been destroyed (in step 211), the server creates a new data structure and restore the data from the disk to the server memory. In step 214, the server sends at least one existing notification to the client. After sending at least one notification to the client, the server shifts to the idle state again. And again in the idle state, step 215, the server stores connection data and possible data structures to the disk from the server memory and removes that data from the server memory. However, it is also possible, for example, if the server has enough memory capacity that the server does not store the connection data and data structures to disk immediately after moving to the idle state. Before storing the connection data and data structures to disk, the server may, for example, wait for a predetermined time or the server may move the data to the disk only in every other or third idle state or when it notices that the server memory capacity is running low or if a number of client connections exceeds a predetermined number of concurrent client connections, or if an estimated next notification time is higher than the predetermined time. The estimated notification time may be estimated by machine learning giving an estimated value for the idle time by analyzing the timestamp of notification delivered from different services subscribed by the mobile client (user) and/or on the basis of other features such as user amount, system load, system usage pattern etc. In addition it is possible that the server does not move/store the connection data and data structures to the disk at all or that it mixes above mentioned methods.

It is also possible that a push notification server restores from a disk only a part of the data relating to a notification ID and/or that a push notification server restores from a disk only a part of the data that is stored to the disk, wherein that restored part is data that is necessary for sending a real-time push notification to a client.

Fig. 3 shows a flow chart of a server memory optimizing method 300 for a real-time system according to an example embodiment. In step 310 a connection is established between a client device and a server, wherein the server is suitable to have a plurality of concurrent connections with plurality of client devices. In step 320 a server memory of the server is utilized for establishing and maintaining connection of the client device with the server. Connection content data and data structures are stored to the server memory. In step 330 the connection shifts to an idle state and the connection content data and data structures are stored to a disk from the server memory. In step 340 the connection data and data structure are removed from the server memory. The server may be, for example, a real-time push notification server, video or audio streaming server or any other suitable server that uses persistent connections.

The various embodiments of the invention can be implemented with the help of computer program code that resides in a memory and causes the relevant apparatuses to carry out the invention. For example, a device may comprise circuitry and electronics for handling, receiving and transmitting data, computer program code in a memory, and a processor that, when running the computer program code, causes the device to carry out the features of an embodiment. Yet further, a network device like a server may comprise circuitry and electronics for handling, receiving and transmitting data, computer program code in a memory, and a processor that, when running the computer program code, causes the network device to carry out the features of an embodiment.

It is obvious that the present invention is not limited solely to the above-presented embodiments, but it can be modified within the scope of the appended claims.

Claims:

1. A method, comprising:

5 establishing a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices;

utilizing a server memory of the server for establishing and maintaining
connection of the client device with the server by connection content data
10 and data structures;

storing the connection content data and data structures to a disk from the
server memory when the connection shifts to an idle state; and

15 removing the connection data and data structure from the server memory.

2. A method according to claim 1, wherein the server is a real-time push
notification server.

20 3. A method according to claim 2, wherein the method further comprises
restoring the connection data and data structure to the server memory when
server is indicated about a real-time push notification to be delivered to the
client device.

25 4. A method according to claim 1, wherein the server is a video or audio
streaming server.

5. A method according to any of the claims 1 to 4, wherein the connection
content data and data structures are stored to the disk after the connection
30 has been in the idle state for a predetermined time.

6. A method according to any of the claims 1 to 4, wherein the connection
content data and data structures are stored to the disk in the idle state if the
server has more than a predetermined number of concurrent connections
35 with client devices.

7. A method according to any of the claims 1 to 4, wherein the connection content data and data structures are stored to the disk in every second or third idle state.

5 8. A method according to any of the claims 1 to 4, wherein the connection content data and data structures are stored to the disk for idle state if the server memory capacity is running low.

10 9. An apparatus comprising at least one processor, memory including computer program code, the memory and the computer program code configured to, with the at least one processor, cause the apparatus to perform at least the following:

15 establish a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices;

utilize a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data and data structures;

20 store the connection content data and data structures to a disk from the server memory when the connection shifts to an idle state; and

remove the connection data and data structure from the server memory.

25 10. An apparatus according to claim 9, wherein the server is a real-time push notification server.

30 11. An apparatus according to claim 10, wherein computer program code is configured to, when executed on at least one processor, further cause the apparatus to:

35 restore the connection data and data structure to the server memory when server is indicated about a real-time push notification to be delivered to the client device.

12. An apparatus according to claim 9, wherein the server is a video or audio streaming server.

13. An apparatus according to any of the claims 9 to 12, wherein the connection content data and data structures are stored to the disk after the connection has been in the idle state for a predetermined time.

5

14. An apparatus according to any of the claims 9 to 12, wherein the connection content data and data structures are stored to the disk in the idle state if the server has more than a predetermined number of concurrent connections with client devices.

10

15. An apparatus according to any of the claims 9 to 12, wherein the connection content data and data structures are stored to the disk in every second or third idle state.

15

16. An apparatus according to any of the claims 9 to 12, wherein the connection content data and data structures are stored to the disk for idle state if the server memory capacity is running low.

20

17. A computer program product embodied on a non-transitory computer readable medium, comprising computer program code configured to, when executed on at least one processor, cause an apparatus to:

establish a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices;

25

utilize a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data and data structures;

30

store the connection content data and data structures to a disk from the server memory when the connection shifts to an idle state; and

remove the connection data and data structure from the server memory.

35

18. A computer program product according to claim 17, wherein the server is a real-time push notification server.

19. A computer program product according to claim 18, wherein computer program code configured to, when executed on at least one processor, further cause the apparatus to:

- 5 restore the connection data and data structure to the server memory when server is indicated about a real-time push notification to be delivered to the client device.

20. An apparatus comprising:

10

means for establishing a connection between a client device and a server suitable to have a plurality of connections with a plurality of client devices;

15

means for utilizing a server memory of the server for establishing and maintaining connection of the client device with the server by connection content data and data structures;

20

means for storing the connection content data and data structures to a disk from the server memory when the connection shifts to an idle state; and

25

means for removing the connection data and data structure from the server memory.

25

21. An apparatus according to claim 20, wherein the server is a real-time push notification server.

30

22. An apparatus according to claim 21, wherein the apparatus further comprises means for restoring the connection data and data structure to the server memory when server is indicated about a real-time push notification to be delivered to the client device.

35

23. An apparatus according to claim 20, wherein the server is a video or audio streaming server.

35

24. An apparatus according to any of the claims 20 to 23, wherein the connection content data and data structures are stored to the disk after the connection has been in the idle state for a predetermined time.

25. An apparatus according to any of the claims 20 to 23, wherein the connection content data and data structures are stored to the disk in the idle state if the server has more than a predetermined number of concurrent connections with client devices.

5

26. An apparatus according to any of the claims 20 to 23, wherein the connection content data and data structures are stored to the disk in every second or third idle state.

10

27. An apparatus according to any of the claims 20 to 23, wherein the connection content data and data structures are stored to the disk for idle state if the server memory capacity is running low.

1 / 3

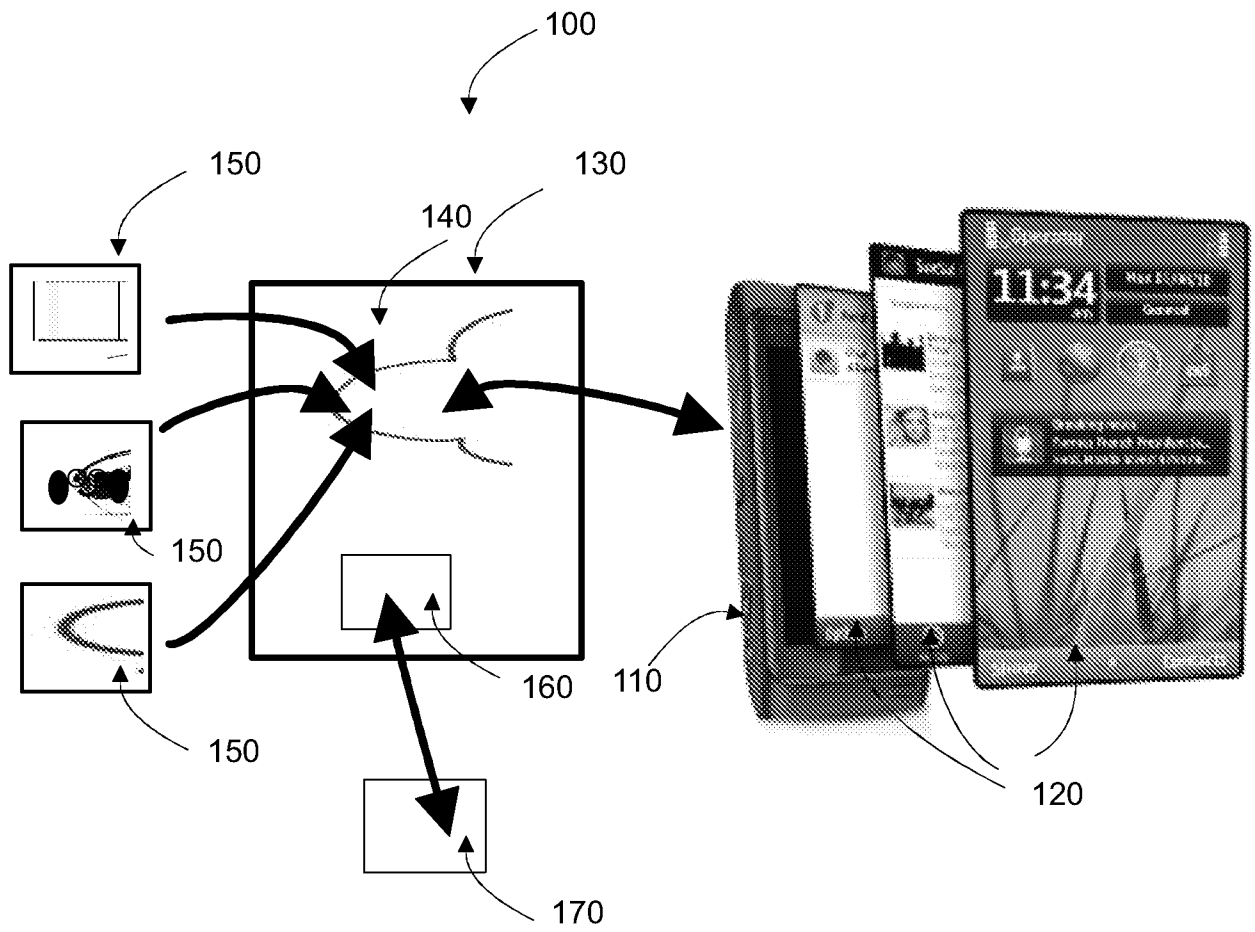


Fig. 1

2 / 3

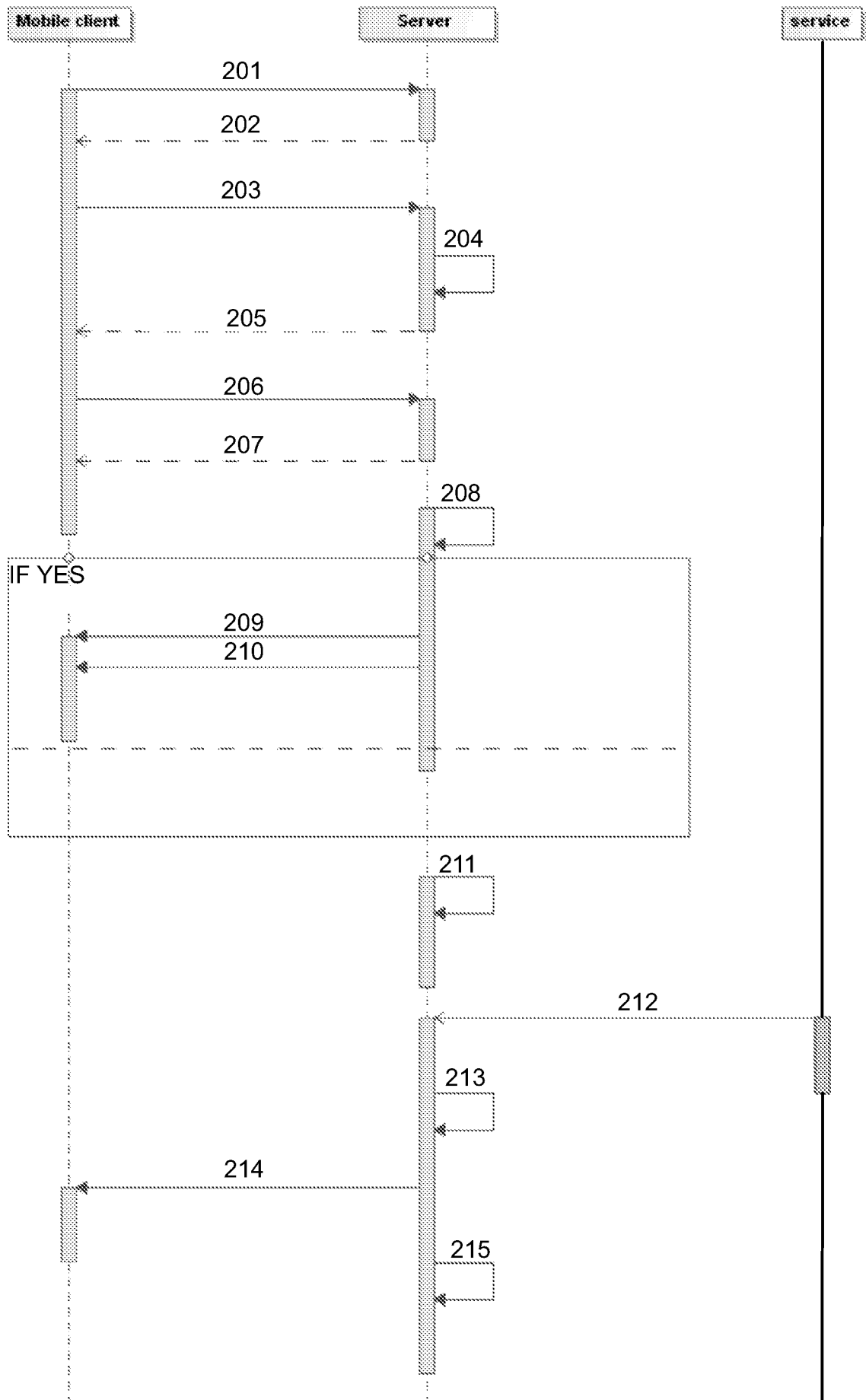


Fig. 2

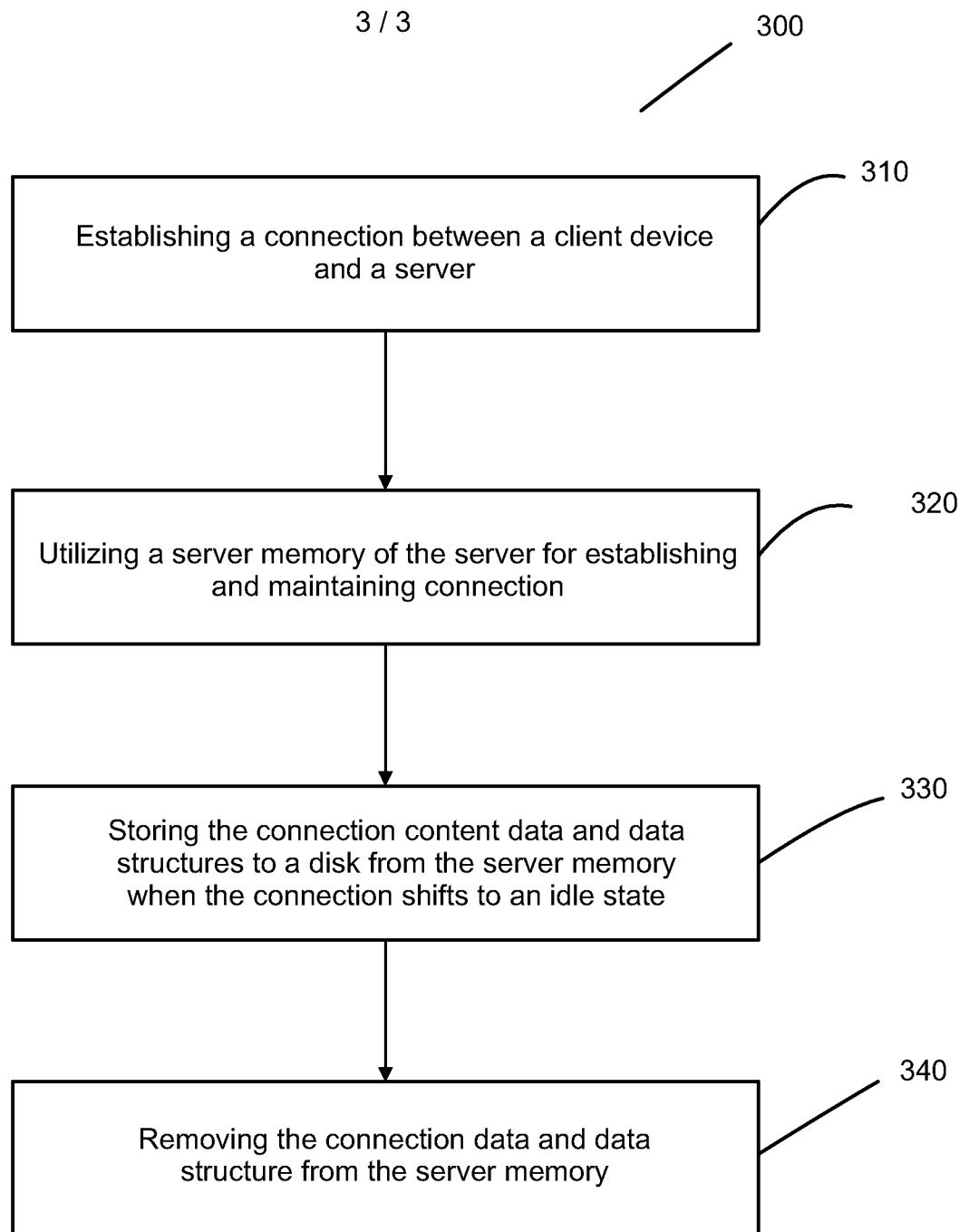


Fig. 3

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2013/050567

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F, G11C, H04L, H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
FI, SE, NO, DK

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

EPO-Internal, WPI, XP3GPP, XPAIP, XPESP, XPESP2, XPETSI, XPI3E, XPIEE, XPIETF, XPIOP, XPIPCOM, XPJPEG, XPOAC, XPRD, XPTK, COMPDX, INSPEC, NPL, Internet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2010083255 A1 (BANE RYAN HOWARD [US] et al.) 01 April 2010 (01.04.2010) Figs. 1-3; paragraphs [0003], [0013]-[0017], [0022], [0031]	1-6, 8-14, 16-25, 27
Y	US 2005138165 A1 (TANG WENTING [US] et al.) 23 June 2005 (23.06.2005) Figs. 1A, 1B; paragraphs [0003], [0005], [0034]-[0045]	1-6, 8-14, 16-25, 27
A	WO 2013053639 A1 (ST ERICSSON SA [CH]) 18 April 2013 (18.04.2013) Figs. 1-6; paragraphs [0008]-[0010], [0021]-[0033]	

☒ 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 application or patent but published on or after the international filing date

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search
05 February 2014 (05.02.2014)

Date of mailing of the international search report
07 February 2014 (07.02.2014)

Name and mailing address of the ISA/FI
Finnish Patent and Registration Office
P.O. Box 1160, FI-00101 HELSINKI, Finland
Facsimile No. +358 9 6939 5328

Authorized officer
Yrjö Raivio
Telephone No. +358 9 6939 500

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2013/050567

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2012018477 A2 (SEVEN NETWORKS INC [US]) 09 February 2012 (09.02.2012) Figs. 1A-2B; paragraphs [0003], [0076]-[0103]	
A	MOUSTAFA, Hassnaa et al. Mobile Multimedia Applications: Delivery Technologies. IT Professional, September/October, 2012, Vol. 14, No. 5, pp. 12-21. the whole document	
A	FOERSTER, Jeff. Keynote 3: Optimizing Video for the Emerging Mobile Cloud. The IEEE ICC 2012 Workshop on Realizing Advanced Video Optimized Wireless Networks [online], June 15, 2012 [retrieved on 2014-02-03]. Retrieved from the Internet: <URL: http://www.profheath.org/wp-content/uploads/2011/09/icc_2012_workshop_keynote_foerster_post.pdf >. the whole document	

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2013/050567

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2010083255 A1	01/04/2010	None	
US 2005138165 A1	23/06/2005	US 7925738 B2	12/04/2011
WO 2013053639 A1	18/04/2013	US 2013094364 A1	18/04/2013
WO 2012018477 A2	09/02/2012	CA 2798523 A1	31/05/2012
		CA 2806527 A1	09/02/2012
		CA 2806529 A1	09/02/2012
		CA 2806548 A1	09/02/2012
		CA 2806549 A1	23/02/2012
		CA 2806550 A1	09/02/2012
		CA 2806557 A1	09/02/2012
		CN 103404193 A	20/11/2013
		EP 2596658 A1	29/05/2013
		EP 2599003 A1	05/06/2013
		EP 2599004 A1	05/06/2013
		EP 2599280 A2	05/06/2013
		EP 2599345 A2	05/06/2013
		EP 2599346 A2	05/06/2013
		EP 2599363 A2	05/06/2013
		EP 2635973 A2	11/09/2013
		EP 2636252 A2	11/09/2013
		EP 2636268 A2	11/09/2013
		EP 2661697 A2	13/11/2013
		FI 20055333 A	22/12/2006
		FI 118451 B	15/11/2007
		FI 20125096 A	02/05/2012
		FI 123227 B	31/12/2012
		FI 20125093 A	23/01/2013
		FI 20125094 A	30/01/2012
		FI 20125095 A	30/01/2012
		FI 20125102 A	31/01/2012
		GB 201301258 D0	06/03/2013
		GB 2495058 A	27/03/2013
		GB 2495066 A	27/03/2013
		GB 201302158 D0	27/03/2013
		GB 2495066 B	18/12/2013
		GB 201302153 D0	27/03/2013
		GB 2495263 A	03/04/2013
		GB 201301235 D0	06/03/2013
		GB 2495455 A	10/04/2013
		GB 2495455 B	13/11/2013
		GB 201302515 D0	27/03/2013
		GB 2495463 A	10/04/2013

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2013/050567

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
		GB 2495463 B	09/10/2013
		GB 201301271 D0	06/03/2013
		GB 2495877 A	24/04/2013
		GB 2495877 B	02/10/2013
		GB 201301219 D0	06/03/2013
		GB 2497012 A	29/05/2013
		GB 2497012 B	30/10/2013
		GB 201307556 D0	12/06/2013
		GB 2499534 A	21/08/2013
		GB 201307573 D0	12/06/2013
		GB 2499741 A	28/08/2013
		GB 201309204 D0	03/07/2013
		GB 2499747 A	28/08/2013
		GB 201309373 D0	10/07/2013
		GB 2499936 A	04/09/2013
		GB 201309234 D0	03/07/2013
		GB 2500327 A	18/09/2013
		GB 201310340 D0	24/07/2013
		GB 2500333 A	18/09/2013
		GB 201310348 D0	24/07/2013
		GB 2500334 A	18/09/2013
		GB 201311949 D0	14/08/2013
		GB 2501416 A	23/10/2013
		GB 201316847 D0	06/11/2013
		GB 2504019 A	15/01/2014
		GB 201318437 D0	04/12/2013
		GB 201318908 D0	11/12/2013
		JP 2013537754 A	03/10/2013
		JP 2013539258 A	17/10/2013
		JP 2013539259 A	17/10/2013
		JP 2013541238 A	07/11/2013
		KR 101227769 B1	30/01/2013
		US 2007019610 A1	25/01/2007
		US 7774007 B2	10/08/2010
		US 8166164 B1	24/04/2012
		US 2012110171 A1	03/05/2012
		US 2012110111 A1	03/05/2012
		US 8190701 B2	29/05/2012
		US 2012110112 A1	03/05/2012
		US 8204953 B2	19/06/2012
		US 2011051610 A1	03/03/2011
		US 8285200 B2	09/10/2012
		US 2012173616 A1	05/07/2012
		US 8291076 B2	16/10/2012
		US 2012110173 A1	03/05/2012
		US 8326985 B2	04/12/2012
		US 2012131184 A1	24/05/2012
		US 8417823 B2	09/04/2013
		US 2012110118 A1	03/05/2012

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2013/050567

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
		US 8484314 B2	09/07/2013
		US 2012158908 A1	21/06/2012
		US 8539040 B2	17/09/2013
		US 2012008536 A1	12/01/2012
		US 2012023190 A1	26/01/2012
		US 2012023226 A1	26/01/2012
		US 2012023236 A1	26/01/2012
		US 2012108225 A1	03/05/2012
		US 2012110109 A1	03/05/2012
		US 2012110110 A1	03/05/2012
		US 2012131095 A1	24/05/2012
		US 2012135726 A1	31/05/2012
		US 2012149352 A1	14/06/2012
		US 2012151044 A1	14/06/2012
		US 2012157170 A1	21/06/2012
		US 2012176968 A1	12/07/2012
		US 2012179801 A1	12/07/2012
		US 2012185597 A1	19/07/2012
		US 2012226767 A1	06/09/2012
		US 2012254417 A1	04/10/2012
		US 2012284356 A1	08/11/2012
		US 2012331087 A1	27/12/2012
		US 2013003634 A1	03/01/2013
		US 2013010693 A1	10/01/2013
		US 2013012180 A1	10/01/2013
		US 2013275563 A1	17/10/2013
		WO 2006136660 A1	28/12/2006
		WO 2012018430 A1	09/02/2012
		WO 2012018431 A1	09/02/2012
		WO 2012018479 A2	09/02/2012
		WO 2012018556 A2	09/02/2012
		WO 2012024030 A2	23/02/2012
		WO 2012060995 A2	10/05/2012
		WO 2012060996 A2	10/05/2012
		WO 2012060997 A2	10/05/2012
		WO 2012061430 A2	10/05/2012
		WO 2012061433 A2	10/05/2012
		WO 2012061437 A1	10/05/2012
		WO 2012071283 A1	31/05/2012
		WO 2012071384 A2	31/05/2012
		WO 2012094675 A2	12/07/2012
		WO 2012161751 A1	29/11/2012
		WO 2013015835 A1	31/01/2013
		WO 2013055413 A1	18/04/2013
.....			

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2013/050567

CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

G06F 12/00 (2006.01)

G06F 9/312 (2006.01)

H04N 21/218 (2011.01)