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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 July 2014 (17.07.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/109630 A1

(51) International Patent Classification:
G06F 9/46 (2006.01) *G06F 21/12* (2013.01)
G06F 9/48 (2006.01)

(21) International Application Number:
PCT/MY2014/000002

(22) International Filing Date:
7 January 2014 (07.01.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI 2013700079 14 January 2013 (14.01.2013) MY

(71) Applicant: **MIMOS BERHAD** [MY/MY]; Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(72) Inventors: **GAI, Chew Kai**; c/o MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **ABD AZIZ, Norazah**; c/o MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(74) Agent: **CHUAH, Jern Ern**; Advanz Fidelis Sdn Bhd, Suite 609, Block D, Phileo Damansara 1, No.9, Jalan 16/11, Petaling Jaya, 46350 (MY).

(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, IR, IS, JP, KE, KG, 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, SA, 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,

[Continued on next page]

(54) Title: A SYSTEM AND METHOD FOR AN APPLICATION EXECUTION

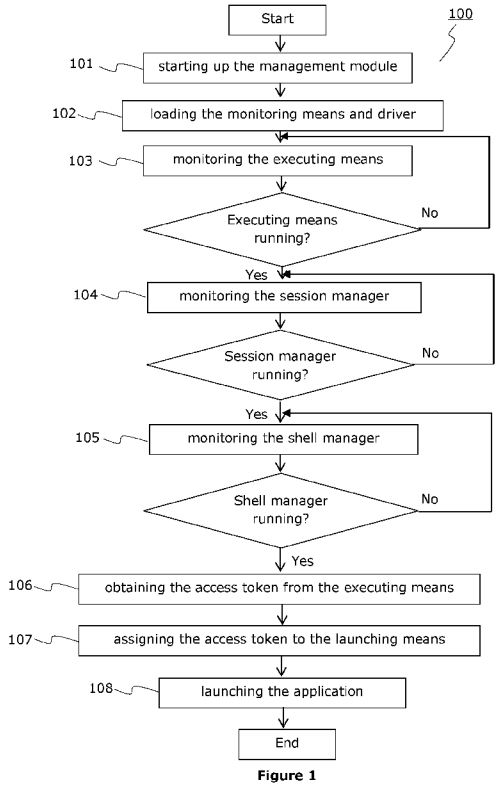

Figure 1

(57) Abstract: The present invention discloses a system for an application execution. The said system comprising a management module with at least an executing means for executing applications in the management module, at least a session manager for saving and restoring applications, at least a shell manager for interpreting and executing commands, and at least a launching means for launching applications, characterized in that the management module further comprises a monitoring means for monitoring the executing means, the session manager and the shell manager, so as to obtain an access token from the executing means and to assign the access token to the launching means, for launching the application without manual intervention.

WO 2014/109630 A1

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

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *of inventorship (Rule 4.17(iv))*

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))*

A SYSTEM AND METHOD FOR AN APPLICATION EXECUTION

TECHNICAL FIELD OF THE INVENTION

This invention is related to a system and method for an application execution, and more particularly for a system and method that executes applications without manual intervention.

BACKGROUND OF THE INVENTION

Generally, a high-privileged software application is a type of application that requires privilege or permission from a user in order to execute or run the software application. In order for the user to run the software application, the user is usually provided with a User Access Control (UAC) prompt or a Credential Login Prompt (CLP) so that the user provides a manual permission to run the software application.

Software applications that are running in the operating system also toggle UAC and CLP prompts. If a user needs to run the software application in the operating system, the user will have to provide, during each usage of the software application, manual permission in order to run the software application. This cannot be considered practical if the user needs to run the software application frequently.

Moreover, some of the high-privileged software application requires the Administrator's permission to run the application. Upon toggling of the UAC and CLP dialogs, the user will have to input the Administrator's login name and password in order to run the application. This would not be practical if there is a huge number of workstation where the Administrator has to serve a huge number of users.

Also, users are presented with UAC and CLP dialogs when they attempt to install new software applications, to access high-privileged software applications, or make changes to the operating system. For skilled addressees, prompting of the dialogs requesting manual interventions

creates unproductive inconveniences, since they are familiar with the type of actions required therein and whether the actions are safe to perform. On the other hand, prompting of the dialogs requesting manual interventions is rather confusing for first time users, since they do not know what actions are safe to perform and they do not know what effects the action would be.

It should also be noted that the process to automatically load software without UAC or CLP prompts via the use of task scheduler, is believed that during this time period, malware may possibly intercept and disable security software applications in the operating system, especially when the high-privileged software application is a security software application.

Prior art EP 1 426 846 B1 discloses a computer system that automatically signs or logs a user to access secured features within a software application without prompting manual intervention. However, the prior art requires the user to continue using the software application without exiting from it, or else the user will be prompted to enter a credential to enable the signed-in session. The prior art also requires the fulfilment of security criteria in order to sign the user in automatically.

Prior art US 7,350,204 B2 discloses a system and method that automatically, transparently and securely controls software execution. The system and method disclosed herein require fulfilment of security levels in order to allow the execution of a software application. If the security levels are not fulfilled, the access to the software application will be restricted and probably denied.

These prior arts require the fulfilment of security criteria in order to execute the software applications. Without the fulfilment of the of security levels, the software applications will not be executed and more particularly, the automated signed-in session would be terminated. It should be noted that it is difficult for the user to sign-in or log in to the software application if credentials or manual permissions are always required to be inputted and security levels are always required to be fulfilled.

Therefore, it is an aim of this present invention to address the aforesaid technical disadvantages by introducing a system and method for an application execution, where software applications are executed without the need of prompting UAC or CLP to the users. The present invention also utilizes a shorter execution time and ensures the secure loading of the executed application. More particularly, the present invention provides a secured and user-friendly system and method to execute high-privileged software applications in the operating system without manual intervention.

SUMMARY OF THE PRESENT INVENTION

The present invention relates to a system for an application execution comprising a management module with at least an executing means for executing applications, at least a session manager for saving and restoring applications, at least a shell manager for interpreting and executing commands, and at least a launching means for launching applications, characterized in that the management module further comprises a monitoring means for monitoring the executing means, the session manager and the shell manager, so as to obtain an access token from the executing means and to assign the access token to the launching means, for launching the application without manual intervention.

Also disclosed in the present invention is a method for an application execution, in accordance with the system as disclosed in the present invention, comprises starting up the management module, loading the monitoring means and a driver of the application in the management module, monitoring the executing means with the monitoring means to determine whether the executing means is running in the management module, monitoring the session manager with the monitoring means to determine whether the session manager is running in the management module, monitoring the shell manager with the monitoring means to determine whether the shell manager is running in the management module, obtaining the access token from the executing means, assigning the access token to

the launching means, and launching the application using the launching means through the driver.

It is an object of the present invention to provide a system and method for an application execution that is capable of executing applications without manual intervention from users.

It is another object of the present invention to provide a system and method for an application execution that is capable of executing applications without prompting the users with User Access Control (UAC) or Credential Login Prompt (CLP).

It is further an object of the present invention to provide a system and method for an application execution that utilizes a shorter execution time period.

It is still further an object of the present invention to provide a system and method for an application execution that ensures the continuation of the executed application.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates the operational flow diagram of the method for an application execution, as described in the present invention.

DETAILED DESCRIPTION OF THE PRESENT INVENTION

The above mentioned and other features and objects of this invention will become more apparent and better understood by reference to the following detailed description. It should be understood that the detailed description made known below is not intended to be exhaustive or limit the invention to the precise form disclosed as the invention may assume various alternative forms. On the contrary, the detailed description covers all the relevant modifications and alterations made to the present invention, unless the claims expressly state otherwise.

The present invention disclosed herein is a system for an application execution. The system comprises a management module with at least an executing means for running applications in the management module, at least a session manager for saving and restoring applications, at least a shell manager for interpreting and executing commands, and at least a launching means for launching applications, characterized in that the management module further comprises a monitoring means for monitoring the executing means, the session manager and the shell manager, so as to obtain an access token from the executing means and to assign the access token to the launching means, for launching the application without manual intervention.

Unlike the prior arts, the present invention introduces a monitoring means into the management module to monitor the executing means, the session manager and the shell manager in the management module, obtaining an access token from the executing means, and assigning the access token to the launching means for launching the application, without manual intervention.

It should be appreciated that the management module is preferably but not limited to an operating system, and that the executing means in the management module is preferably but not limited to a system program. Further, the monitoring means is preferably but not limited to a service program.

The present invention begins with the start-up of the management module, where the management module then loads the monitoring means and a driver in the management module. The monitoring means then monitors the executing means and determines whether the executing means is running in the management module. The session manager is then loaded by the executing means when the executing means is determined to be running in the management module. If the executing means is not running in the management module, the step of monitoring is repeated until the executing means runs in the management module.

It should be appreciated that the running of the executing means, the session manager and the shell manager herein refer to the existence of the executing means, the session manager and the shell manager in the memory of the management module.

Thereafter, the monitoring means monitors the session manager and determines whether the session manager is running in the management module. If the session manager is determined to be running in the management module, the shell manager is loaded by the management module. However, if the session manager is not running in the management module, the step of monitoring the session manager is repeated until the session manager runs in the management module.

Subsequently, the monitoring means monitors the shell manager and determines whether the shell manager is running in the management module. If the shell manager is running in the management module, the monitoring means gains access token from the executing means and assigns the access token to the launching means. Nevertheless, if the shell manager is not running in the management module, the step of monitoring the shell manager is repeated until the shell manager runs in the management module. Upon receiving the access token, the launching means launches the application.

The present invention also discloses a method for an application execution (100) in accordance with the system as disclosed above, which is illustrated in the operational flow diagram in figure 1. Firstly, the present invention starts up the management module (step 101), where the management module loads the monitoring module and the driver of the application in the management module (step 102). Thereafter, the monitoring means monitors the executing means (step 103) to determine whether the executing means is running in the management module. If the executing means is not running

in the management module, the step of monitoring the executing means (step 103) is repeated until the executing means runs in the management module.

Next, the monitoring means monitors the session manager (step 104) to determine whether the session manager is running in the management module. If the session manager is not running in the management module, the monitoring means repeats the step of monitoring the session manager (step 104) until the session manager runs in the management module before proceeding to the subsequent stage.

In the following step, the monitoring means monitors the shell manager (step 105) to determine whether the shell manager is running in the management module. If the shell manager is running in the management module, the monitoring means then obtains the access token from the executing means (step 106) and assigns the access token to the launching means (step 107). However, in the case that the shell manager is not running in the management module, the monitoring means repeats the step of monitoring the shell manager (step 105) until the shell manager runs in the management module before obtaining the access token from the executing means (step 106) and assigning the access token to the launching means (step 107). Upon receiving the access token, the launching means executes the application (step 108) by delegating the access token to the application.

With the use of the present invention, no manual intervention is required in order to execute applications from the users since the monitoring means obtains and assigns the permission to execute the applications based on whether the executing means, session manager and shell manager are running in the management module. By using the present invention, the execution time of the application would be shorten since the monitoring means is only required to determine the existence of the executing means, session manager and shell manager in the management module. The shorten time would ensure that the application is executed under a secured condition.

CLAIM

1. A system for an application execution comprising:

a management module with at least an executing means for executing applications in the management module, at least a session manager for saving and restoring applications, at least a shell manager for interpreting and executing commands, and at least a launching means for launching applications,

characterized in that the management module further comprises a monitoring means for monitoring the executing means, the session manager and the shell manager, so as to obtain an access token from the executing means and to assign the access token to the launching means, for launching the application without manual intervention.

2. The system according to claim 1, wherein the monitoring means monitors the executing means, the session manager and the shell manager to determine whether the executing means, the session manager and the shell manager are running in the management module, prior to the obtainment of the access token from the executing means.

3. The system according to claim 2, wherein the monitoring means obtains the access token from the executing means and assigns the access token to the launching means in the event that the executing means, the session manager and the shell manager are determined to be running in the management module.

4. The system according to claim 2, wherein the monitoring means continues to monitor the executing means when the executing means is not running in the management module.

5. The system according to claim 2, wherein the monitoring means continues to monitor the session manager when the session manager is not running in the management module.

6. The system according to claim 2, wherein the monitoring means continues to monitor the shell manager when the shell manager is not running in the management module.

7. A method (100) for an application execution, in accordance with the system as claimed in claim 1, comprises:

starting up the management module (step 101);

loading the monitoring means and a driver of the application in the management module (step 102);

monitoring the executing means with the monitoring means to determine whether the executing means is running in the management module (step 103);

monitoring the session manager with the monitoring means to determine whether the session manager is running in the management module (step 104);

monitoring the shell manager with the monitoring means to determine whether the shell manager is running in the management module (step 105);

obtaining the access token from the executing means (step 106);

assigning the access token to the launching means (step 107); and

launching the application using the launching means through the driver (step 108).

8. The method according to claim 7, wherein the access token is obtained when the executing means, session manager and shell manager are determined to be running in the management module.

9. The method according to claim 7, wherein the step of monitoring either the executing means (step 103), the session manager (step 104), the shell manager (step 105) or in any combination, is repeated when the executing means is not running in the management module.

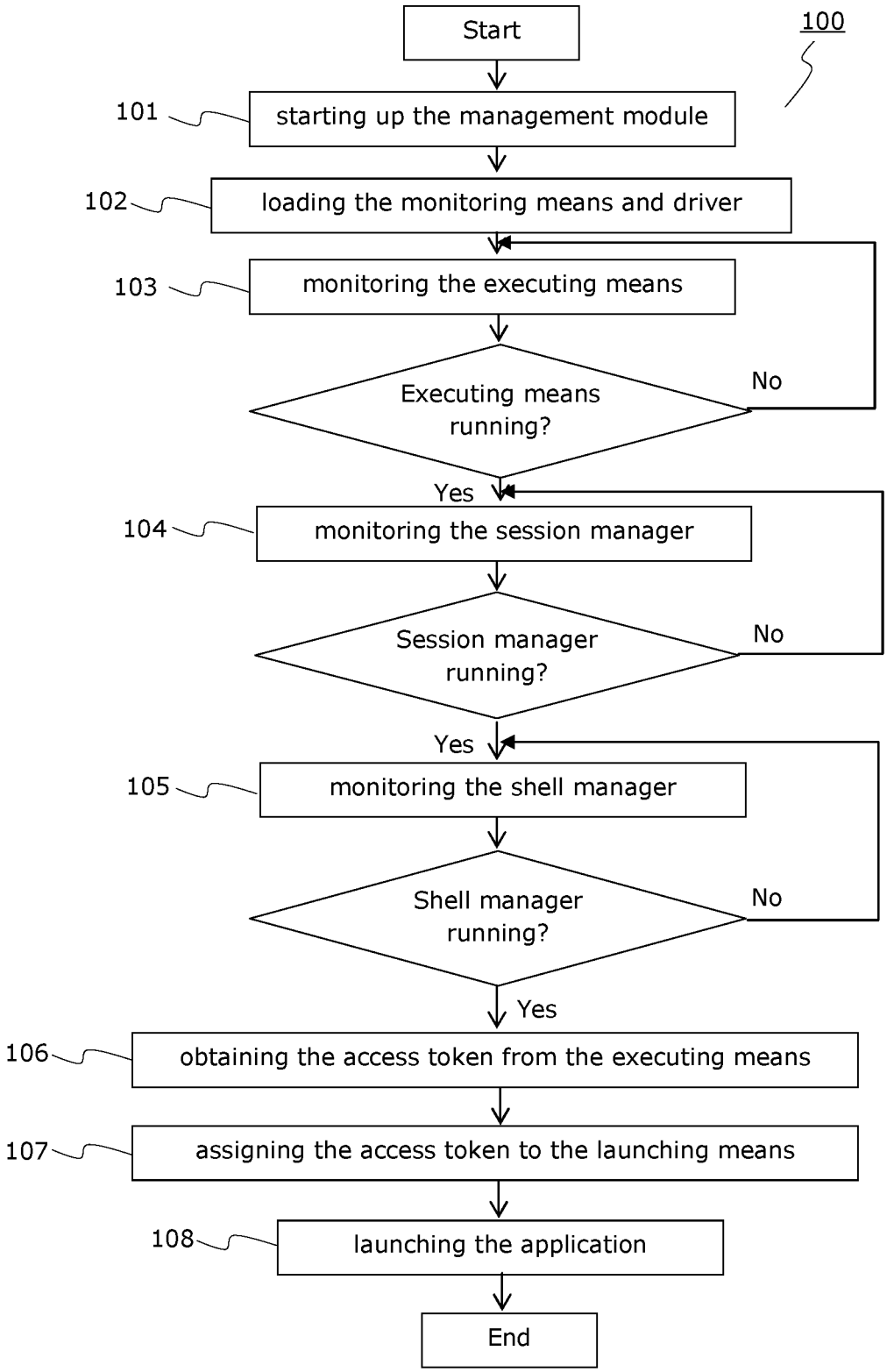
Start
100
101
starting up the management module
102
loading the monitoring means and driver
103
monitoring the executing means
Executing means running?
No
Yes
104
monitoring the session manager
Session manager running?
No
Yes
105
monitoring the shell manager
Shell manager running?
No
Yes
106
obtaining the access token from the executing means
107
assigning the access token to the launching means
108
launching the application
End

Figure 1

INTERNATIONAL SEARCH REPORT

International application No
PCT/MY2014/000002

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/46 G06F9/48 G06F21/12
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)
G06F

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 350 204 B2 (LAMBERT JOHN J [US] ET AL) 25 March 2008 (2008-03-25) cited in the application abstract column 1, line 15 - column 4, line 41 column 8, line 27 - column 17, line 51 figures 1,2,3,4,5A, 5B, 5C	1-9
X	WO 2011/044710 A1 (SAFENET INC [US]; CHENG PETER [CN]) 21 April 2011 (2011-04-21) abstract paragraph [0003] - paragraph [0014] paragraph [0029] - paragraph [0033] claims 1-3	1-9
A	US 2010/325485 A1 (KAMATH SANDEEP [US] ET AL) 23 December 2010 (2010-12-23) the whole document	1-9

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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	Date of mailing of the international search report
19 June 2014	26/06/2014

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	Beltrán-Escavy, José

4

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/MY2014/000002

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2010/031348 A1 (LAM MONICA SIN-LING [US] ET AL) 4 February 2010 (2010-02-04) the whole document	1-9
A	US 2011/251992 A1 (BETHLEHEM ALEXANDER [CA] ET AL) 13 October 2011 (2011-10-13) the whole document	1-9
X	OSMAN UGUS ET AL: "A Smartphone Security Architecture for App Verification and Process Authentication", COMPUTER COMMUNICATIONS AND NETWORKS (ICCCN), 2012 21ST INTERNATIONAL CONFERENCE ON, IEEE, 30 July 2012 (2012-07-30), pages 1-9, XP032229661, DOI: 10.1109/ICCCN.2012.6289217 ISBN: 978-1-4673-1543-2 abstract page 1, left-hand column, paragraph 1 - page 2, left-hand column, paragraph 1 page 4, left-hand column, paragraph 3 - page 6, left-hand column, paragraph 1	1-9
A	PATRICIA ARIAS CABARCOS ET AL: "SuSSo: Seamless and ubiquitous single sign-on for cloud service continuity across devices", IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, IEEE SERVICE CENTER, NEW YORK, NY, US, vol. 58, no. 4, 1 November 2012 (2012-11-01), pages 1425-1433, XP011488763, ISSN: 0098-3063, DOI: 10.1109/TCE.2012.6415016 the whole document	1-9
A	DEJAN KOVACHEV ET AL: "Beyond the client-server architectures: A survey of mobile cloud techniques", COMMUNICATIONS IN CHINA WORKSHOPS (ICCC), 2012 1ST IEEE INTERNATIONAL CONFERENCE ON, IEEE, 15 August 2012 (2012-08-15), pages 20-25, XP032245588, DOI: 10.1109/ICCCW.2012.6316468 ISBN: 978-1-4673-2996-5 the whole document	1-9

4

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/MY2014/000002

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7350204 B2	25-03-2008	NONE	
WO 2011044710 A1	21-04-2011	CN 102576391 A	11-07-2012
		EP 2488982 A1	22-08-2012
		JP 2013507671 A	04-03-2013
		US 2011191593 A1	04-08-2011
		WO 2011044710 A1	21-04-2011
US 2010325485 A1	23-12-2010	NONE	
US 2010031348 A1	04-02-2010	NONE	
US 2011251992 A1	13-10-2011	NONE	