

(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2013208203 B2**

(54) Title
Contextual solicitation in a starter application

(51) International Patent Classification(s)
G06F 9/44 (2006.01) **G06F 21/30** (2013.01)
G06F 15/16 (2006.01)

(21) Application No: **2013208203** (22) Date of Filing: **2013.01.09**

(87) WIPO No: **WO13/106353**

(30) Priority Data

(31)	Number	(32)	Date	(33)	Country
	13/347,700		2012.01.11		US

(43) Publication Date: **2013.07.18**

(44) Accepted Journal Date: **2018.07.05**

(71) Applicant(s)
Microsoft Technology Licensing, LLC

(72) Inventor(s)
Farrell, Terry;Woerkom, Norbert Van;McQuillan, Robert;Mowatt, David;Kasselman, Pieter

(74) Agent / Attorney
Davies Collison Cave Pty Ltd, Level 15 1 Nicholson Street, MELBOURNE, VIC, 3000, AU

(56) Related Art
US 2008/0113809 A1
US 2003/0181242 A1
US 6263499 B1



(51) International Patent Classification:

G06F 9/44 (2006.01) G06F 21/30 (2013.01)
G06F 15/16 (2006.01)

(21) International Application Number:

PCT/US2013/020713

(22) International Filing Date:

9 January 2013 (09.01.2013)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

13/347,700 11 January 2012 (11.01.2012) US

(71) Applicant (for all designated States except US): MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: FARRELL, Terry; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). WOERKOM, Norbert Van; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). MCQUILLAN, Robert; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). MOWATT, David; c/o Microsoft Corporation, LCA - In-

ternational Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). KASSELMAN, Pieter; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(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).

[Continued on next page]

(54) Title: CONTEXTUAL SOLICITATION IN A STARTER APPLICATION

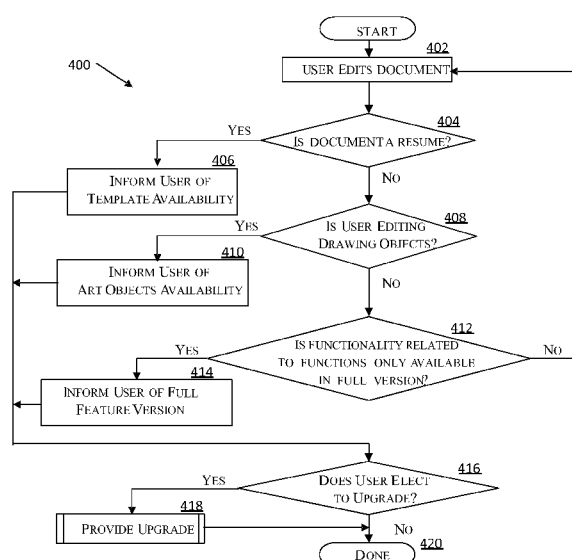


Fig.4A

(57) Abstract: A reduced capability subset of an application referred to as a "starter application" is provided on a computer device prior to purchase and intended for use by a user purchasing the computer device. The starter application may monitor input from the user when the starter application is executing to ascertain opportunities to inform the user of an upgrade and then solicit the user for purchasing the upgrade. The starter application may use a limited access browser to contact a server to assist in the solicitation and complete the transaction. Upon completing the transaction, the upgrade is provided to the computing device. The upgrade may comprise the full capability version of the application, templates, clip art, or other software. The upgrade may entail transmitting the upgrade to the computer device or providing an authorization key to unlock the upgrade that is already stored on the computer device.



**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

2013208203 14 Sep 2017

CONTEXTUAL SOLICITATION IN A STARTER APPLICATION

BACKGROUND

[0001] Manufacturers of consumer-oriented computing devices, such as laptops or desktop computers, may pre-load software on the computing device at the time of manufacture so that the user can immediately use the computing device for various applications after purchasing the computing device. For example, many computers are pre-loaded with some version of word processing software so that a user can immediately use the computer to create letters or other such documents. Other applications, such as spreadsheet processing applications may be included as well. Purchasers of computers that can immediately use the computer may perceive immediate value in the computer.

[0002] The pre-loaded software application may be based on a reduced capability set of an existing software application. Such pre-loaded software is commonly referred to as a “starter” application version of the “full” version of the software application. Thus, the word processing starter application pre-loaded on a computer may have a subset of the features and capabilities available in the full version of the application. Software manufacturers find that providing a starter application of a software application allows users to gain familiarity with the software application and increases the likelihood that the user will later purchase the full version of the software application.

[0003] A user of the starter application may believe that the starter application is sufficient to meet their initial needs. However, as the user becomes more proficient with using the starter application, the user may encounter deficiencies in trying to perform certain functions and may not always know when the additional capabilities in the full version may be useful. Thus, if the user is not made aware in a timely manner of the additional functionality in the full version that is available, then the user may not appreciate that the full version would provide additional benefits to the user. In turn, the user may be less likely to purchase the full version of the application.

[0004] It is with respect to these and other considerations that the disclosure made herein is presented. It is desired to address or ameliorate one or more disadvantages or limitations associated with the prior art, or to at least provide a useful alternative.

2013208203 28 May 2018

- 2 -

SUMMARY

[0005] In accordance with the present invention, there is provided a method executed by at least one processor including:

monitoring input from a user for editing a document received by a starter application executing on the processor, the starter application comprising a subset of functions of a full application;

determining based on the input that the user is requesting invocation of a function for editing the document that is not within the subset of the functions;

ascertaining based on the input from the user an opportunity to solicit the user to receive an upgrade that provides the requested function;

in response to the input performed on the document by the user, soliciting the user by the starter application to purchase the upgrade by prompting the user to purchase the upgrade while the user is working within the document;

completing a transaction to purchase the upgrade while the user is working within the document; and

enabling access to the upgrade by the computing device while the user is working within the document.

[0006] The present invention also provides a system for providing updates to a user comprising:

a memory storing a starter application comprising a reduced capability set of a full version of a document processing application; and

a processor configured to perform the method as defined above.

[0007] The present invention also provides a system comprising:

a processor;

a memory storing instructions that when executed by the processor of the system cause the processor to perform the method as defined above.

[0008] DELETED

[0009] DELETED

[0009A] DELETED

[0009B] DELETED

2013208203 28 May 2018

- 3 -

[0009C] DELETED

[0009D] DELETED

[0009E] DELETED

BRIEF DESCRIPTION OF THE DRAWINGS

[0009A] Some embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, in which:

[0010] FIGURE 1 is an illustration of one context of various computing devices using the capabilities described herein;

[0011] FIGURE 2 is an illustration of an architecture of a starter application providing contextual solicitations and accessing a server for upgrades;

[0012] FIGURE 3 is a user interface of one embodiment of the starter application soliciting a user for an upgrade;

[0013] FIGURES 4A and 4B are process flows for the starter application processing user input to determine when to solicit the user for the upgrade; and

[0014] FIGURE 5 is an illustration of a computer architecture of the computing device executing the starter application for soliciting the user for the upgrade.

DETAILED DESCRIPTION

[0015] Concepts and technologies are described herein for providing contextual solicitations for purchasing a software upgrade. The contextual solicitations may be initiated by a starter version of an application program and the software upgrade may be the full version of the application program. Alternatively, the software upgrade may comprise other related software that operates with the application, such as templates, clip art, or tutorials designed to interact with the starter application or the full application.

[0016] In one embodiment, a method of providing an upgrade includes the operations of monitoring input by the starter application executing on a processor in the computing device, where the input is provided by a user of the starter application. The method further includes ascertaining by the starter application based on the input from the user an opportunity to provide an upgrade to the starter application to the user and soliciting the user to purchase the upgrade to the starter application. The method further includes

2013208203 28 May 2018

- 3A -

completing a transaction by the starter application to purchase the upgrade, and enabling access to the upgrade at the computing device.

[0017] In another embodiment, a computer readable storage medium storing instructions that when executed by a processor of a computing device cause the processor to monitor input by a starter application executing on the processor, where the input is provided by a user of the starter application and the starter application includes a subset of functions of a full application. The instructions further cause the processor to ascertain based on the input from the user an opportunity to solicit the user to receive an upgrade, solicit the user by the starter application to purchase the upgrade by prompting the user to purchase the upgrade, complete a transaction to purchase the upgrade and provide the upgrade to the computing device.

[0017A] In another embodiment a system for providing updates to a user includes a memory storing a starter application that includes a reduced capability set of a full version of a word processing application and a processor, where the processor is configured to receive input from the user editing a word processing document, and inform the user that a requested function is not available in the starter application. The processor is further configured to inform the user that the requested function is available in the full version, solicit the user to purchase the full version, and provide the full version of the word processing application to the user.

[0017B] In the following detailed description, references are made to the accompanying drawings that form a part hereof, and which are shown by way of illustration specific embodiments or examples. Referring now to the drawings, in which like numerals represent like elements throughout the several figures, aspects of configuring a drawing guide will be presented.

[0017C] As described above, many computer manufacturers have found it beneficial to form agreements with software manufacturers to include pre-loaded software on a computing device at the time of sale. Thus, the software manufacturer provides software to the computer manufacturer to pre-load onto the computer. Pre-loading software can facilitate the purchaser's productivity after purchasing the computing device. This allows users to immediately use the computing device to perform various functions after purchasing the device. Software manufacturers frequently provide reduced functionality

2013208203 28 May 2018

- 3B -

applications. Such applications are sometimes referred to as a “starter” version of the software application for this purpose (henceforth referred to herein as a “starter application” or “starter version”). The version of the software application with the full set of capabilities may be referred to as the “full application” or the “full version.”

[0017D] Pre-loading the starter application allows purchasers to immediately begin using and gain familiarity with the software. The starter application can be a word-processing program, a game, spreadsheet program, tutorial program, etc. Typically, the starter application includes a core set of functions that allows the starter application to be useful in its own right, but the more advanced functions or capabilities are removed or disabled.

[0018] The relationship of the starter application and the full application can be viewed as an application with two versions having two levels of functionality – a reduced feature set and a full feature set. Other embodiments of the concepts described herein can apply to a software application having more than two levels of functionality. For example, a software application may be packaged with three functionality levels – a starter level, an advanced level, and a professional level. Other embodiments may have more levels. Further, it is not required that the concepts disclosed herein apply only to upgrading from the current level version of the application, or apply to only when upgrading from a pre-loaded version on the computer device. The concepts herein may apply to upgrading to an enhanced version regardless of how the user obtained the lowest level of the software application. Further, the user may upgrade by “skipping” intermediate versions of the application and upgrading to the most current or advance version. Further, other types of applications may be associated with the upgrade. However, for purposes of illustration, the concepts and technologies herein are mainly described in terms of a starter application pre-loaded on a computing device and wherein the user is upgrading to a full application.

[0019] At any time after using the starter application, the user may subsequently purchase the next higher level, or the full application in order to achieve the maximum productivity and usefulness of the software application. However, the user may not appreciate when the additional capabilities may be useful. In other words, if the user is unable to use certain features in a starter application, the user may be unaware that these features even exist and are available in the full application. Thus, informing the user at an opportune time allows the user to be cognizant when the additional features would be useful. Soliciting the user at the proper time to obtain the full application can significantly increase the likelihood of the user purchasing the upgrade.

[0020] Further, the user can be solicited to obtain not only the full application, but other add-ons that provide value added capabilities to the current version of the application. The add-on can take the form of a template, tutorial, clip art, etc. The term “upgrade” as used herein refers to any productivity enhancement module that can be provided to the user, which includes an upgraded functional level of the application (full application) and/or any add-on, such as a template library that works with the starter application or full application. In some embodiments, the upgrade may be for another type of application, but which integrates with the starter application. For example, the starter application may be a word processing application that allows drawing objects to be edited and inserted into a word processing document. After observing the user editing and inserting a drawing

object (or repeatedly doing so), the user may be then prompted to upgrade. In this example, the word processing application could solicit the user to purchase an upgrade that is a drawing application.

[0021] An upgrade can encompass productivity enhancement modules which provide further information to the user, such as how to use an existing feature. Thus, tutorials in the form of texts, graphical presentations, videos, or other forms describing how to use an existing feature, along with an enhanced feature, may be provided to a user at an opportune time. Upgrades may comprise special offers offering discounts. Upgrades may comprise free or discounted trials where additional capabilities can be tested and used, without a commitment to complete a purchase transaction. In other embodiments, users may be offered to try certain features or identified capabilities for a limited number of times or for a limited duration. After this time, the user may be solicited again to obtain the upgrade.

[0022] As noted, users may not be aware of the existence of the upgrade. Even if they are aware, users may not even be aware that the upgrade is relevant to a particular function or task currently performed by the user. Soliciting the user at the proper time involves achieving a contextual understanding by the starter application of how the user is presently interacting with the application, and what the user is using the application for. This involves identifying when the upgrade would be relevant to the user's current usage of the application and soliciting the user at that moment for the appropriate upgrade.

[0023] One context of how the capabilities disclosed herein can be used is shown in FIGURE 1. FIGURE 1 illustrates a system 100 comprising various computing devices, such as a smart phone 101, laptop computer 102, or a tablet computer 105. These computing devices may execute various applications stored in the computing device, which may have been loaded onto the computing device by the computing device manufacturer. These computing types are illustrative, and are not intended to limit the application of the concepts disclosed herein.

[0024] These computing devices may access, via the Internet 104, various services provided by a cloud computing system 106. The computing devices can include desk-top computers, personal digital assistants, e-book readers, etc. The computing devices can utilize wireless connections 115, 117 to interact with a cloud computing system 106. The wireless connections for the devices are conveyed in one embodiment by the Internet 104 to the cloud computing system 106. The wireless connections can be based on various readily available technologies, including the various cellular based technologies (CDMA,

GSM, 4G, EDGE, etc.) originating from cellular infrastructure 170 provided by a cellular service provider or involving other wireless technologies (e.g., WiMax, WiFi, etc.). The wired technologies can be based on ISDN, cable modems, DSL, and other well-known forms.

5 [0025] The connectivity afforded by the wired 103 and wireless 115, 117 communications through the Internet 104 provides the computing devices with access to a server 108 in the cloud computing system 106. The various applications executing on the computing device may also interact with the server 108. Although the concepts illustrated herein may refer to a single server 108, various servers in the cloud computing system 106
10 may be involved. For example, one server 108 accessed by a device may in turn access another server 113 or a data store 109 to provide service to a user. Thus, a plurality of download servers may be used in certain embodiments.

[0026] The server 108 may store and/or execute various application programs 120 on behalf of a user. These may be executed in a shared or distributed manner across one or
15 more servers and interact with the computing device 101, 102, or 105. The application programs 120 may include a drawing application 121 for creating and editing graphical oriented programs. Applications for editing images, video, etc. may also be considered with the scope of a drawing application. Other forms of authoring applications 125 may also create and edit documents, such as text documents and spread sheet documents.

20 [0027] Other applications executing on the server 108 may include social media applications 120 for group related messaging and communications. Another application program present may be an email/messaging application 135 and may allow for various messaging capabilities, including instant messaging. A communications application 140 may allow for video chat or voice communication. A calendar application 145 may allow
25 for organizing meetings and events. A contact application 150 may allow for organizing contact information for a user. A streaming media application 155 may provide for streaming digital data, such as audio or video files. Each of these applications may store data, which the server 108 may maintain in a data store 109, which in turn may comprise various storage systems.

30 [0028] In one embodiment, the user may be executing an application program in the computing device, such as the laptop 102, which is the starter application. In some embodiments, the starter application may communicate with one of the server computers 108 by initiating and receiving messages sent via the Internet 104.

[0029] A more focused context is shown in FIGURE 2. FIGURE 2 shows a system 200 comprised of the computing device 102 that is able to access the server 108 in the cloud computing system 106. The laptop 102 is loaded with a starter applications package 210, which in this embodiment is a suite of starter application programs. Specifically, the starter applications package 210 suite comprises starter applications 215a and 215b. These starter applications 215a, 215b could be any type of reduced functionality application program, such as a program for word processing, spreadsheets, slide presentation programs, games, tutorials, etc. In the embodiment shown in FIG. 2, starter application 215a is assumed to be a word processing application and starter application 215b is assumed to be a spreadsheet program.

[0030] Each starter application 215a and 215b may respectively incorporate solicitation logic 220a, 220b. This may be accomplished in one embodiment by integrating the solicitation logic into the starter application, or in another embodiment, tightly coupling the solicitation logic with the starter application. The solicitation logic 220a, 220b may be designed for a particular starter application. Thus, the solicitation logic 220a for starter application 215a may be different from the solicitation logic 220b for starter application 215b since the solicitation logic 220a is centered on a word processing context. However, the solicitation logic 220a itself can be upgraded, as will be discussed later. The solicitation logic modules 220a, 220b typically perform similar functions, which is to ascertain an opportune time for soliciting the user for an upgrade. Thus, the solicitation logic may share common logic in some embodiments, such as the ability to identify certain words or identify certain functions that are not available in the starter application 215a, but that are available in the full version of the application. The solicitation logic 220a may interact with the starter application 215a to determine the context at runtime. For example, the solicitation logic 220a may ascertain what type of document is being opened, how frequently the user has interacted with this type of document using the starter application, etc.

[0031] Each starter application 215a, 215b typically accesses a limited access browser 235a, 235b. The limited access browser incorporates a subset of the functionality that may be incorporated in a web browser, but its access is controlled and limited to the starter application. Other applications in the computer 102 may not be able to access this functionality and certain limitations in the functionality of the browser may be defined. For example, the resources in the computer 102 that the server 108 can access may be limited via the browser. Hence, it is referred to herein as a “limited access” browser.

Nothing precludes other applications from incorporating a limited access browser for other purposes, however. In one embodiment, each limited access browser 235a and 235b may be identical in functionality. In other embodiments, a single limited access browser 235a may be shared by all the starter applications 215a, 215b in the starter application package 210. For sake of illustration, the concepts are explained henceforth in terms of a single starter application, namely starter application 215a directed to word processing.

[0032] The starter application 215a may access the limited access browser 235a for interacting with a service identified by a URL or other address identifier in the server 108. The server 108 may also execute applications 120 that may also provide document templates, tips and help, clip art, advertising graphics (static and video), and other upgrades known to those skilled in the art. In another embodiment, the solicitation logic is not integrated with the starter application 215a, but it is executed in server 108. In this embodiment, the starter application 215a is able to trigger to an externally executed solicitation logic and pass context information to the solicitation logic when triggered. Further, the solicitation logic can be upgraded over time by downloading new solicitation logic to the starter application or updating the solicitation logic in the server.

[0033] In one embodiment, the starter application 215a is a word processing program that when executed, provides a reduced capability set of word processing capabilities. The solicitation logic 220a may monitor the activities of the user to ascertain when an opportunity arises for soliciting the user. This may involve communication between the starter application 215a and the solicitation logic 220a as to what actions the user is currently performing on a document. The communication may include reporting each function, keystroke, or history of the last few functions or keystrokes. The communication can also provide contextual information, such as what types of objects are being processed, how long the starter application has been used, what help functions have been invoked, etc.

[0034] For example, turning to FIGURE 3, a display 300 shows a document 315 being edited using the starter application 215a. In FIGURE 3, the user is provided with a user interface 300 in the display 305. The display 305 includes the starter application menu bar 310 displaying various word processing functions, a view of the document 315 in an editing pane, and a solicitation pane 320. In this embodiment, the solicitation pane 320 is an area located to the right of the document 315 that may provide text messages, images and/or video to the user based on the context of what the user is doing.

[0035] The depiction of the user interface 300 in FIGURE 3 does not depict all the possible functions and associated icons that may be provided. The exact text in the document 315 of the document may vary and for illustration purposes represents the beginning of a resumé.

5 [0036] The solicitation logic 220a can parse the structure of document 315 and can ascertain that the document 315 is a resumé of some sort. The solicitation logic 220a in the starter application may derive this from the format and structure of the document, based on for example, the positioning of dates, that a name that is centered on the first page, and/or the file name (e.g., the file may be called “John Smith Resume”). Any
10 number of rules can be used to determine that the document is a resumé.

[0037] These rules can be deterministic or probabilistic. For example, a model can be used to ascertain when a user may be likely to request an upgrade. Collective information on upgrades from various users can be maintained and analyzed to identify patterns that the present user is following to predict the likelihood the user will upgrade in context of
15 the current circumstances. This model can be based in the solicitation logic executing in the user’s computer or executing in a server 108. The model may identify several opportunities and associated probabilities for an upgrade, and select the opportunity with the greatest likelihood of success. For example, the model may predict that after a user has invoked a function in the starter application over a dozen times, then it is appropriate
20 to then solicit the user for a certain type of upgrade related to that function.

[0038] Once the solicitation logic determines that the user is working on a resumé, the solicitation logic can then prompt the user in the solicitation pane 320 for various types of upgrades. For example, if the document is a resumé, the user could be prompted in the solicitation pane 320 to view exemplary resumé templates or receive resumé drafting
25 guidelines. The user may be presented with and select a link 325 to connect to the server 108 for the purpose of viewing templates and/or the guidelines. The user may also be presented with a link to view an on-line tutorial about writing effective resúmes. The user’s response to these solicitations may also be analyzed to provide feedback as to when and how to solicit the user for other types of requests. Further, the user’s response may be
30 used to update a solicitation model, which can be the basis for triggering other solicitations to the user, or to other users. The solicitation model can be executed in the solicitation logic in the user’s computer or in a network based server.

[0039] Returning to FIGURE 2, the starter application 215a is able to interact with the limited access browser 235a. In some embodiments, the solicitation logic 220a can

interact with corresponding solicitation logic in the server 108 to ascertain the type of document being edited (e.g. resumé, letter, memo, etc.). In various embodiments, the solicitation logic 220a may also provide a keystroke history associated with this document, portions of the document, file name, or other information to the server 108. In other

5 embodiments the solicitation logic 220a can provide the type of application, installation date, frequency of use, Internet related cookies, and other usage aspects. The server 108 can respond by providing information as to the type of upgrades that would be applicable for the present document.

[0040] For example, the solicitation logic 220a may note the frequency of use of certain

10 combined functions for certain documents. Specifically, if the user frequently invokes a “help” function prior to invoking that function for a certain type of document object, coupled with long time periods of inactivity, this condition may suggest difficulties the user is encountering in using that particular function for that type of document. This set of conditions may trigger solicitation of a particular upgrade in the form of a tutorial program

15 that focuses on the particular function. However, if the user does not frequently invoke the help function, which suggests the user knows how to use the function, then solicitation of the tutorial function may not be productive.

[0041] In other embodiments, the solicitation logic 220a can determine the type of document without accessing the server 108. The solicitation logic 220a can then inform

20 the user of the type of upgrade that may be applicable and useful to the user, and solicit the user for obtaining the upgrade. The solicitation logic 220a may then use the limited access browser 235a to access the upgrade from the server 108. The server 108 then provides the requested upgrade to the user, which may be a template, guideline, or other form of upgrade.

[0042] The limited access browser 235a is limited in that its access is controlled by the starter application 215a. In some embodiments, the user or other applications cannot directly access the limited access browser 235a. The starter application 215a carefully controls the access to other resources on the computing device by the limited access browser 235a. Further, the capabilities of the limited access browser 235a are limited as

25 well. For example, the limited access browser 235a may be limited from accessing the disk drive of the computer device or executing any files stored thereon.

[0043] As noted before, the server 108 may also execute a form of solicitation logic processing, and interact with the starter application 215a executing in the computing device. The starter application 215a may inform the server 108 as to the contents of the

document, and the solicitation logic in the server 108 may analyze the contents, and in response, provide the appropriate text response in the solicitation pane 320. In either approach, if the user accepts the solicitation offer, a query is launched to the server to obtain the upgrade, where further information is provided on the solicitation pane. The
5 further information may include a sample template, price information, number of templates in the upgrade, etc.

[0044] The starter application may have various serial numbers or authorization information encoded within it that may be reported to the server 108 in conjunction with obtaining the upgrade. If incorrect authorization information is provided by the starter
10 application to the server, the server may deny the upgrade request. In other embodiments, if the authorization information is valid, this may entitle the user to receive a discount or other benefits from the server in conjunction with the upgrade.

[0045] One embodiment of a process flow 400 for providing a context dependent solicitation for a starter application is shown in FIGURE 4A. It should be appreciated that
15 the logical operations described herein with respect to FIGURE 4A and the other FIGURES are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance and other requirements of the computing system.
20 Accordingly, the logical operations described herein are referred to variously as states, operations, structural devices, acts, or modules. These operations, structural devices, acts, and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof. It should also be appreciated that more or less operations may be performed than shown in the FIGURE 4A and described herein. These
25 operations may also be performed in a different order than those described herein. The operations shown in FIGURE 4A are merely illustrative, and additional, or other operations may be defined in other embodiments.

[0046] FIGURE 4A illustrates one form of rule-based logic processing that could occur in the computing device by the solicitation logic when the starter application 215a is
30 executed and the user is editing a document in operation 402. The example provided should not be interpreted as limiting the types of logic processing that can be used to trigger an upgrade solicitation. Other, more complex, types of processing can occur to determine when to trigger a solicitation. The processing could involve a probabilistic model for determining the most appropriate solicitation.

[0047] The document can be a text based document, a spreadsheet, or some other data corresponding to the function of the starter application. The solicitation logic 220a will know the type of starter application being used and may ascertain what type of document that the user is editing. Specifically, in this embodiment, the solicitation logic 220a

5 ascertains whether the document is a resumé in operation 404.

[0048] The solicitation logic 220a may ascertain the document is a resumé by applying rules associated with a resumé (e.g., a name and address in the center of the first page, the existence of the word “resume” in the file name, the presence of dates in a structured manner, etc.). The solicitation logic 220a may further provide a portion of the document

10 to server 108 for further analysis to determine whether the document is a resumé or not.

[0049] If the solicitation logic 220a ascertains in operation 404 that the document is a resumé, then in operation 406 the solicitation logic 220a may initiate a context dependent the solicitation by informing the user of the availability of a template for drafting resumé. The solicitation process may involve informing the user of the type of templates available,

15 potential resumé drafting suggestions, or other resumé related information. This determination of which resumé related information to provide to the user may as complex as applying a sophisticated model based on observed user actions to as simple as providing the most frequently downloaded resumé related information.

[0050] If the solicitation logic 220a in operation 404 determines the document is not a resumé, then in operation 408 the solicitation logic 220a may ascertain whether the user is invoking functionality related to graphical objects, such as editing a drawing object. The user may be inserting an image, editing an image, or otherwise performing actions on drawing objects. This action may trigger the solicitation logic 220a in operation 408 to solicit the user by informing the user of the availability of drawing art objects, including

20 so-called “clip-art” in operation 410.

[0051] If the solicitation logic 220a in operation 408 determines the user is not editing drawing objects, the solicitation logic 220a may then ascertain in operation 412 whether the functionality being requested by the user may be related to functions that are only available in the full version of the application. For example, a spreadsheet program may not allow the full capabilities of manipulating tables, or a word processing program may not allow the full capabilities for editing styles or using templates. As noted earlier, various models and processes may be incorporated in the solicitation logic. Once the solicitation logic 220a in operation 412 determines that the currently invoked functions are

30

related to functions only available in the full version, the solicitation logic 220a could inform the user of the availability of these functions in the full version in operation 414.

[0052] Informing the user of the upgrade could occur by presenting the user with text, video, graphics, sounds, etc. The content of this information could vary according to context, so that solicitation does not appear formalistic to the user. Thus, even if the solicitation in different contexts is focused on the same upgrade, the approach and style of the solicitation could vary in each context.

[0053] Other context dependent solicitations could occur in addition to, or in lieu of, those disclosed above. For example, the solicitation logic could be triggered by detecting an occurrence of a current action or document content, as described above. Alternatively, the solicitation logic may detect a history of actions or related events which collectively indicate an opportunity to solicit an upgrade from the user. Specifically, if the user repeatedly uses certain features in the starter application, then the user may be solicited after a fixed number of times, e.g., on the 5th instance of using the feature. Further, if the user has been presented with the solicitation several times and has not opted to upgrade, the solicitation logic may cease offering solicitations, or may vary the solicitation prompt. This avoids the solicitation logic becoming an annoying occurrence to the user.

[0054] If there are no opportunities identified for soliciting the user, namely if the results are “no” for operations 404, 408, and 412, then the processing loops back to operation 402 where the process continues to monitor the editing performed by the user. It may be that insufficient text has been entered to ascertain the document or to sufficiently identify a context dependent solicitation opportunity. If, however, a solicitation has occurred in either operation 404, 408, or 412, then the process in 416 determines whether the user elects to receive the upgrade. If the determination is yes, then in operation 418 the upgrade is provided to the user. If not, then process completes in operation 420. In other embodiments, the process may loop back to operation 402, where the process repeats. In this case, a different form of solicitation may occur.

[0055] The process for upgrading a user in operation 418 is shown in additional detail in FIGURE 4B. In FIGURE 4B the user in operation 440 may be provided with information, or a preview, of the upgrade. This may be followed in operation 442 with the terms of the upgrade, including the price, method of transaction, system compatibility requirements, etc.

[0056] The user may be provided an opportunity in operation 444 to continue the transaction or abandon the upgrade. If the user continues, then in operation 446 the

upgrade transaction is completed. In some embodiments, the upgrade is downloaded in operation 448. If the user does not continue with the upgrade, then in operation 450 the upgrade is cancelled. The process then completes in operation 452.

[0057] The provision for upgrading a user in operation 418 may occur in different ways.

5 The solicitation logic 220a may connect to a web site via the limited access browser 235a to receive the requested templates, clip-art, etc. Appropriate user interfaces for providing payment may be also included either in the solicitation logic 220a, in another module in the starter application package 102, and interact with the server 108 using the limited access browser 235a. The process of receiving the upgrade may involve actual
10 transmission of the upgrade from a source at another server to the user's computing device. For example, if the solicitation is for graphical objects, a library of graphical objects may be downloaded to the user for selection and insertion into the document.

[0058] Another form of enabling access to an upgrade may be accomplished by transmitting an authorization key or similar code authorizing the user to access capabilities
15 that are normally contained, but locked, in the starter application. For example, in one embodiment, the starter application and the full version of the Application are pre-loaded on the computing device at the time of manufacturer, but the full version is locked. Thus, the user can only use the starter application portion. The process of upgrading involves receiving an authorization key which unlocks the full version, thus allowing the user
20 access to the full capabilities. In this manner, less information must be downloaded to the user to obtain the benefits of the upgrade.

[0059] Further, the upgrade can be accomplished while the user is working within a document of the starter application. Access to the full version may be provided while the document is being edited, so that the user is not required to save and close the application,
25 and then restart the computer or re-open the document. This provides a more convenient approach to the user for instantly accessing the full version.

[0060] Further, the unlocking of access to the full version of the application may also provide the user with access to other applications. This includes applications which the user did not have any corresponding starter applications. For example, if the user is
30 provided with a word processing starter application, the user may upgrade in place to a productivity suite that include the full version of the word processing application and a graphics editing application.

[0061] The process flow in FIGURE 4 represents only a few of the potential rules that the solicitation logic 220a can be incorporated for detecting an opportunity to solicit the

user. It should be appreciated that a number of rules or conditions could be defined for detecting the opportunity to trigger the solicitation. The solicitation logic 220a can be limited to only soliciting the user for the full version of the application, or may also include soliciting the user for libraries of templates, on-line tutorials, etc.

5 **[0062]** In another embodiment, the upgraded application could comprise one of several available versions, with the user selecting which version to upgrade to. For example, as previously noted, the user may be using a starter application, and other functional levels may be available, such as the advanced version and the professional version. In addition, there may be available an advanced version 1.0 and version 2.0 available for upgrade.

10 Similarly, there may be a professional version 2.0 and 3.0. The user may be presented with a number of options of the version to upgrade to.

[0063] Further, the upgrade may not be limited to a single application. One of the options presented to the user may be to upgrade to a suite of related applications. Thus, when upgrading the word processing application, an upgrade to other applications may be included, such as upgrading a graphics presentation program, a spreadsheet program, and other productivity application. In other embodiments, the upgrade may only involve purchasing a different type of application or service capability. For example, the user may be using a starter word processing application that incorporates minimal drawing capabilities. After detecting the user has repeatedly access the drawing capabilities, the solicitation logic prompts the user to upgrade to provide access to a graphical editing program. The user may opt to do so, and retain use of the starter word processing application.

20 **[0064]** As noted previously, the solicitation logic itself can be upgraded and managed by the server 108. The solicitation logic is not necessary static (e.g., limited to the version initially deployed with the starter application). The solicitation logic can be dynamically upgraded, typically under the direction of the server 108, but without the direct intervention or control by the user.

[0065] An initial version of the solicitation logic may be incorporated with a starter application, and this version may incorporate certain algorithms for triggering solicitations. Over the course of time, additional products or applications may be available, which were not available when the solicitation logic was defined. Further, over time, additional algorithms for triggering solicitations may be defined. Without upgrading of the solicitation logic itself, it would not be possible for the solicit logic to be aware of newly defined products. Thus, the solicitation logic may incorporate a version identified

30

number which is indicated to the server 108, which can then perform version management. Specifically, server 108 can download a new version of the solicitation logic as appropriate.

[0066] An embodiment of the computing architecture for the computing device 102 for accomplishing the above operations is shown in FIGURE 5. FIGURE 5 shows an illustrative computing architecture for a computer 500 capable of executing the software components described. The computer architecture shown in FIGURE 5 may illustrate a conventional computer, laptop, tablet, or other type of computer utilized to execute any aspect of the software components presented herein. Other architectures or computers may be used to execute the software components presented herein.

[0067] The computer architecture shown in FIGURE 5 includes a central processing unit 520 ("CPU," and also referred to herein as a "processor"), a system memory 505, including a random access memory 506 ("RAM") and a read-only memory ("ROM") 508, and a system bus 540 that couples the memory to the CPU 520. A basic input/output system containing the basic routines that help to transfer information between elements within the server 500, such as during startup, is stored in the ROM 508. The computer 500 further includes a mass storage device 522 for storing an operating system 528, application programs, and other program modules, as described herein.

[0068] The mass storage device 522 is connected to the CPU 520 through a mass storage controller (not shown), which in turn is connected to the bus 540. The mass storage device 522 and its associated computer-readable media provide non-volatile storage for the computer 500. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it should be appreciated by those skilled in the art that computer-readable media can be any available computer storage media that can be accessed by the computer 500. However, the computer readable media does not encompass transitory signals.

[0069] By way of example, and not limitation, computer-readable media may include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. For example, computer-readable media includes, but is not limited to, RAM, ROM, EPROM, EEPROM, flash memory or other solid state memory technology, CD-ROM, digital versatile disks ("DVD"), HD-DVD, BLU-RAY, or other optical storage, magnetic cassettes, magnetic tape, magnetic disk

storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 500.

[0070] According to various embodiments, the computer 500 may operate in a networked environment using logical connections to remote computers or servers through a network such as the network 553. The computer 500 may connect to the network 553 through a network interface unit 550 connected to the bus 540. It should be appreciated that the network interface unit 550 may also be utilized to connect to other types of networks and remote computer systems.

[0071] The computer 500 may also incorporate a radio interface 514 which can communicate wirelessly with network 553 using an antenna 515. The wireless communication may be based on any of the cellular communication technologies or other technologies, such as WiMax, WiFi, or others.

[0072] The computer 500 may also incorporate a touch-screen display 518 for displaying information and receiving user input by touching portions of the touch-screen. This is typically present on embodiments based on a tablet computer and smart phone, but other embodiments may incorporate a touch-screen 518.

[0073] The computer 500 may also include an input/output controller 504 for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus (not shown in FIGURE 5). Similarly, an input/output controller may provide output to a display screen, a printer, or other type of output device (also not shown in FIGURE 5). The input/output controller may also provide an interface to an audio device, such as speakers, and/or an interface to a video source, such as a camera, or cable set top box, antenna, or other video signal service provider.

[0074] As mentioned briefly above, a number of program modules and data files may be stored in the mass storage device 522 and RAM 506 of the computer 500, including an operating system 528 suitable for controlling the operation of a networked desktop, laptop, tablet, or server computer. The mass storage device 522 and RAM 506 may also store one or more program modules or data files. In particular, the mass storage device 522 and the RAM 506 may store the solicitation logic 220a. The same storage device 522 and the RAM 506 may store the limited access browser 235a. The mass storage device 522 and the RAM 506 may also store other types of applications and data, including the starter application module 215 and various upgrades 526. The upgrades 526 are examples of upgrades which are locked in the starter application 215 and can be accessed only once an authorization key is obtained through the upgrade process.

[0075] It should be appreciated that the software components described herein may, when loaded into the CPU 520 and executed, transform the CPU 520 and the overall computer 500 from a general-purpose computing system into a special-purpose computing system customized to facilitate the functionality presented herein. The CPU 520 may be constructed from any number of transistors or other discrete circuit elements, which may individually or collectively assume any number of states. More specifically, the CPU 520 may operate as a finite-state machine, in response to executable instructions contained within the software modules disclosed herein. These computer-executable instructions may transform the CPU 520 by specifying how the CPU 520 transitions between states, thereby transforming the transistors or other discrete hardware elements constituting the CPU 520.

[0076] Encoding the software modules presented herein may also transform the physical structure of the computer-readable media presented herein. The specific transformation of physical structure may depend on various factors, in different implementations of this description. Examples of such factors may include, but are not limited to the technology used to implement the computer-readable media, whether the computer-readable media is characterized as primary or secondary storage, and the like. For example, if the computer-readable media is implemented as semiconductor-based memory, the software disclosed herein may be encoded on the computer-readable media by transforming the physical state of the semiconductor memory. For example, the software may transform the state of transistors, capacitors, or other discrete circuit elements constituting the semiconductor memory. The software may also transform the physical state of such components in order to store data thereupon.

[0077] As another example, the computer-readable media disclosed herein may be implemented using magnetic or optical technology. In such implementations, the software presented herein may transform the physical state of magnetic or optical media, when the software is encoded therein. These transformations may include altering the magnetic characteristics of particular locations within given magnetic media. These transformations may also include altering the physical features or characteristics of particular locations within given optical media, to change the optical characteristics of those locations. Other transformations of physical media are possible without departing from the scope and spirit of the present description, with the foregoing examples provided only to facilitate this discussion.

2013208203 14 Sep 2017

[0078] In light of the above, it should be appreciated that many types of physical transformations take place in the computer 500 in order to store and execute the software components presented herein. It also should be appreciated that the computer 500 may comprise other types of computing devices, including hand-held computers, embedded computer systems, personal digital assistants, and other types of computing devices known to those skilled in the art. It is also contemplated that the computer 500 may not include all of the components shown in FIGURE 5, may include other components that are not explicitly shown in FIGURE 5, or may utilize an architecture completely different than that shown in FIGURE 5. For example, some devices may utilize a main processor in conjunction with a graphics display processor, or a digital signal processor. In another example, a device may have an interface for a keyboard, whereas other embodiments will incorporate a touch screen.

[0079] Based on the foregoing, it should be appreciated that systems and methods have been disclosed for providing context dependent solicitation of an upgrade in a starter application. It should also be appreciated that the subject matter described above is provided by way of illustration only and should not be construed as limiting. Various modifications and changes may be made to the subject matter described herein without following the example embodiments and applications illustrated and described, and without departing from the true spirit and scope of the present invention, which is set forth in the following claims.

[0080] Throughout this specification and claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0081] The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

2013208203 28 May 2018

- 20 -

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A method executed by at least one processor including:
 - monitoring input from a user for editing a document received by a starter application executing on the processor, the starter application comprising a subset of functions of a full application;
 - determining based on the input that the user is requesting invocation of a function for editing the document that is not within the subset of the functions;
 - ascertaining based on the input from the user an opportunity to solicit the user to receive an upgrade that provides the requested function;
 - in response to the input performed on the document by the user, soliciting the user by the starter application to purchase the upgrade by prompting the user to purchase the upgrade while the user is working within the document;
 - completing a transaction to purchase the upgrade while the user is working within the document; and
 - enabling access to the upgrade by the computing device while the user is working within the document.
2. The method of claim 1, further comprising:
 - informing a server that the user has purchased the upgrade;
 - receiving an authorization key at the processor from the server; and
 - in response to processing the authorization key, providing the user with access to the full application.
3. The method of claim 1 or 2, wherein the starter application is a word processing application comprising the subset of the functions of a full version of the word processing application, and further comprising:
 - providing at least a portion of the document edited using the starter application to a limited access browser module;
 - directing the limited access browser module to provide the at least portion of the document to a server; and

2013208203 28 May 2018

- 21 -

receiving an indication from the server of the upgrade of a type of the document.

4. The method of claim 1 or 2, wherein the starter application is a word processing application comprising the subset of the functions of a full word processing application, and further comprising:

informing the user that the function is not within the subset of the functions in a solicitation pane adjacent to the document being edited by the starter application; and

informing the user that the function is within the functions of the full version of the word processing application.

5. The method of claim 4, further comprising:

soliciting the user to purchase the full version of the word processing application;

transacting a purchase transaction for the full version of the word processing application by the starter application; and

transmitting the upgrade comprising the full version of the word processing application to a computing device comprising the processor.

6. The method of claim 4, further comprising:

soliciting the user to purchase the full version of the word processing application;

transacting a purchase transaction for the full version of the word processing application by the starter application; and

enabling access to the upgrade by providing an authorization key to access the full version of the word processing application by the processor.

7. The method of any one of claims 1 to 6, further comprising ascertaining based on a plurality of requests for the requested function in the input from the user the opportunity to solicit the user to receive the upgrade that provides the requested function.

8. A system for providing updates to a user comprising:

a memory storing a starter application comprising a reduced capability set of a full version of a document processing application; and

2013208203 28 May 2018

- 22 -

a processor configured to perform the method of any one of claims 1 to 7.

9. The system of claim 8, further comprising:

an input/output interface configured to access a server over an Internet,

wherein the processor is further configured to access a limited access browser module to provide the full version of the document processing application to the user.

10. The system of claim 9, wherein the memory further stores the full version of the document processing application in a locked form, and

wherein the processor is further configured to receive an authorization key from the server using the limited access browser module to unlock and provide access to the full version.

11. The system of any one of claims 8 to 10, wherein the processor is further configured to:

process a transaction to purchase the full version of the document processing application.

12. The system of any one of claims 8 to 11, wherein the processor is further configured to:

transmit at least a portion of the document to the server; and receive an indication from the server as to a type of the document.

13. A system comprising:

a processor;

a memory storing instructions that when executed by the processor of the system cause the processor to perform the method of any one of claims 1 to 7.

14. At least one computer-readable storage medium storing instructions that when executed by a processor of a computing device cause the processor to perform the method of any one of claims 1 to 7.

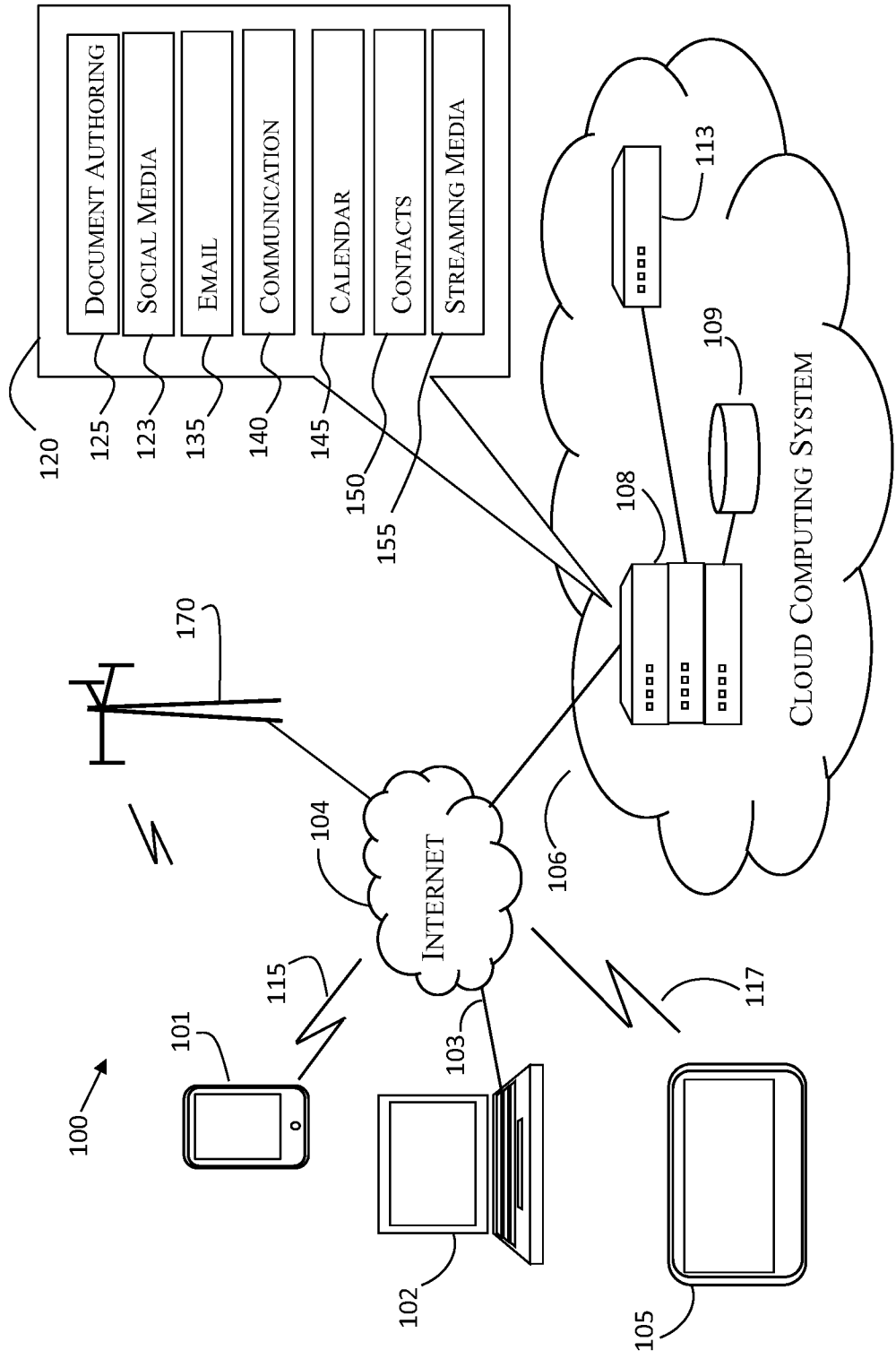


Fig.1

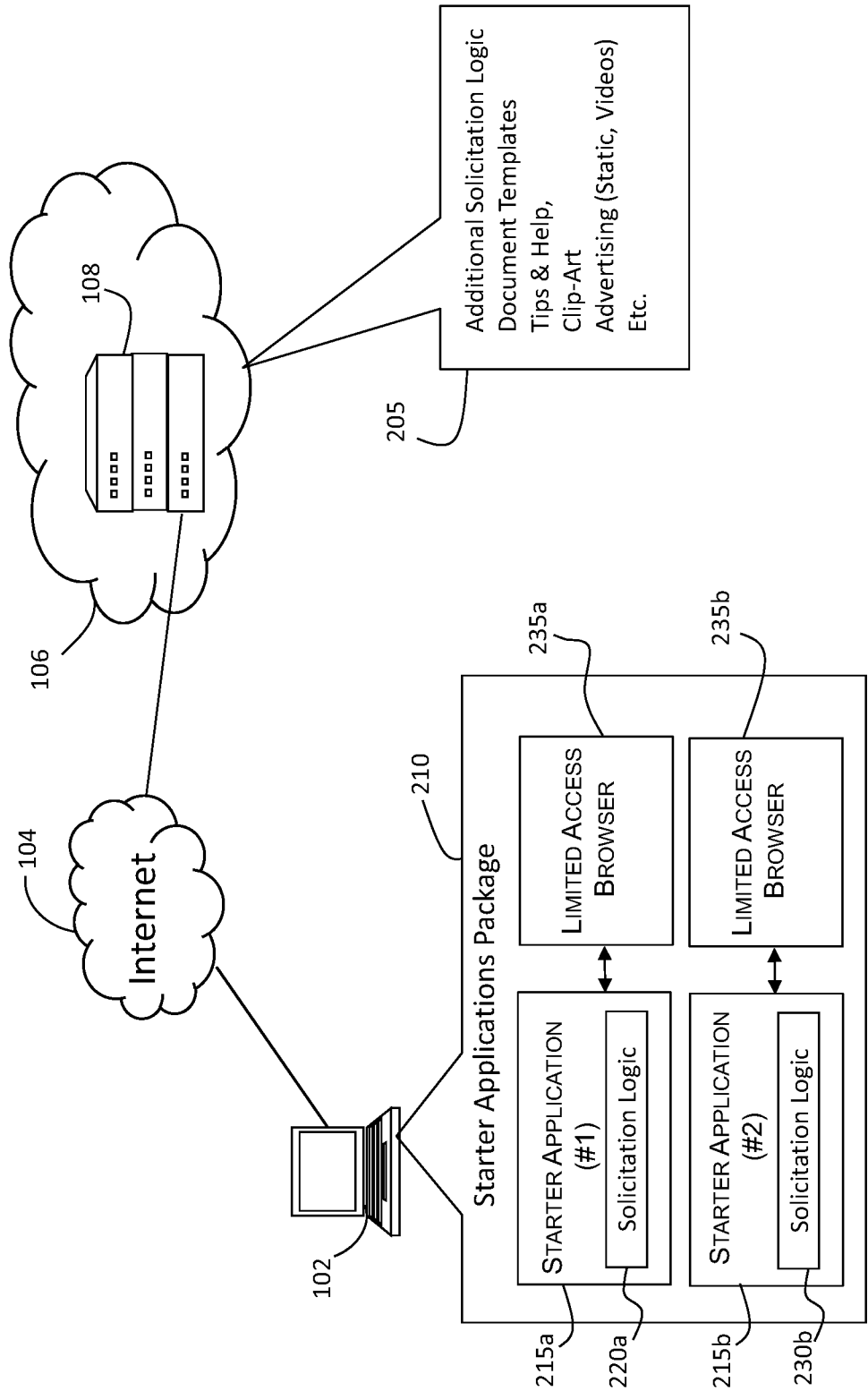


Fig.2

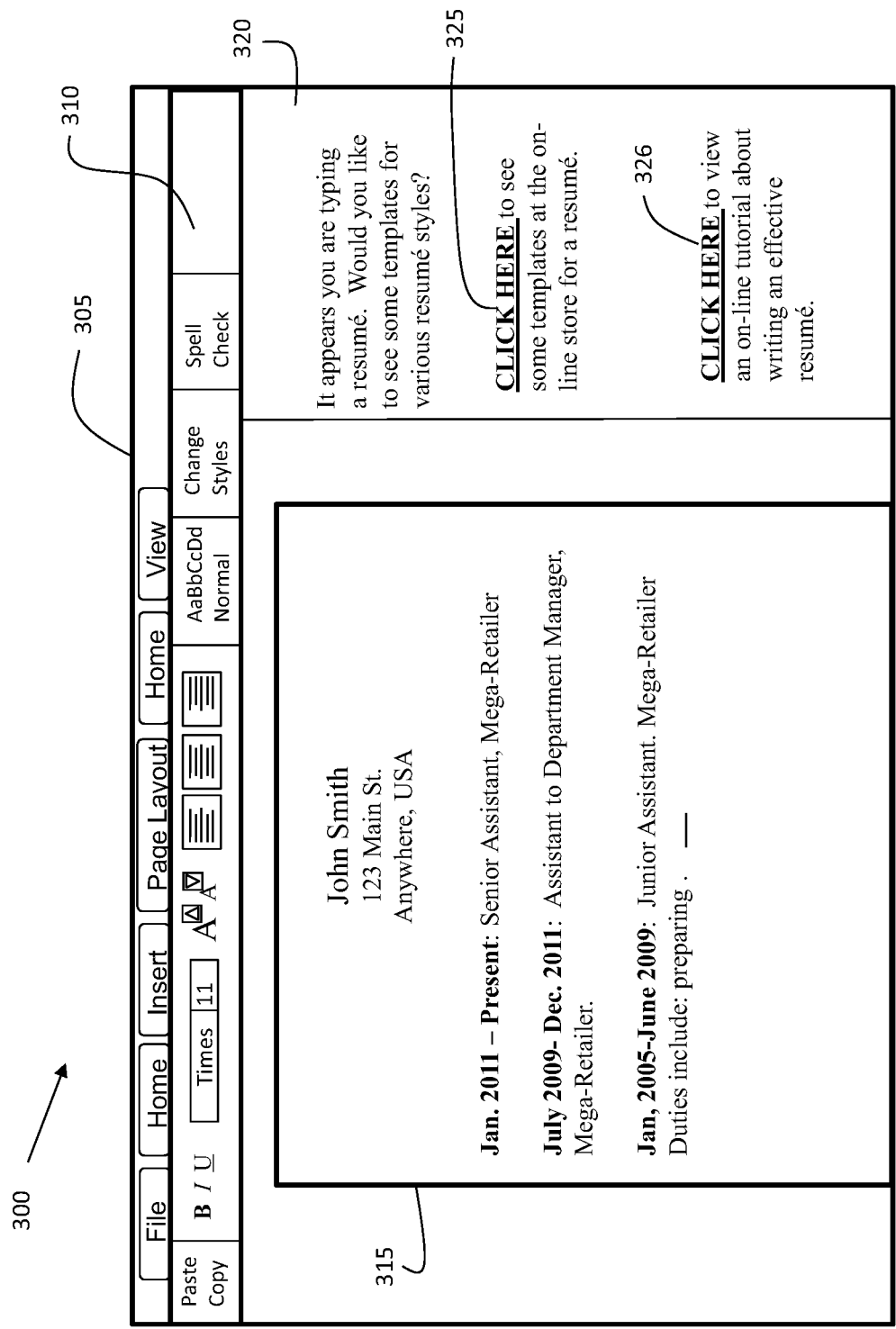
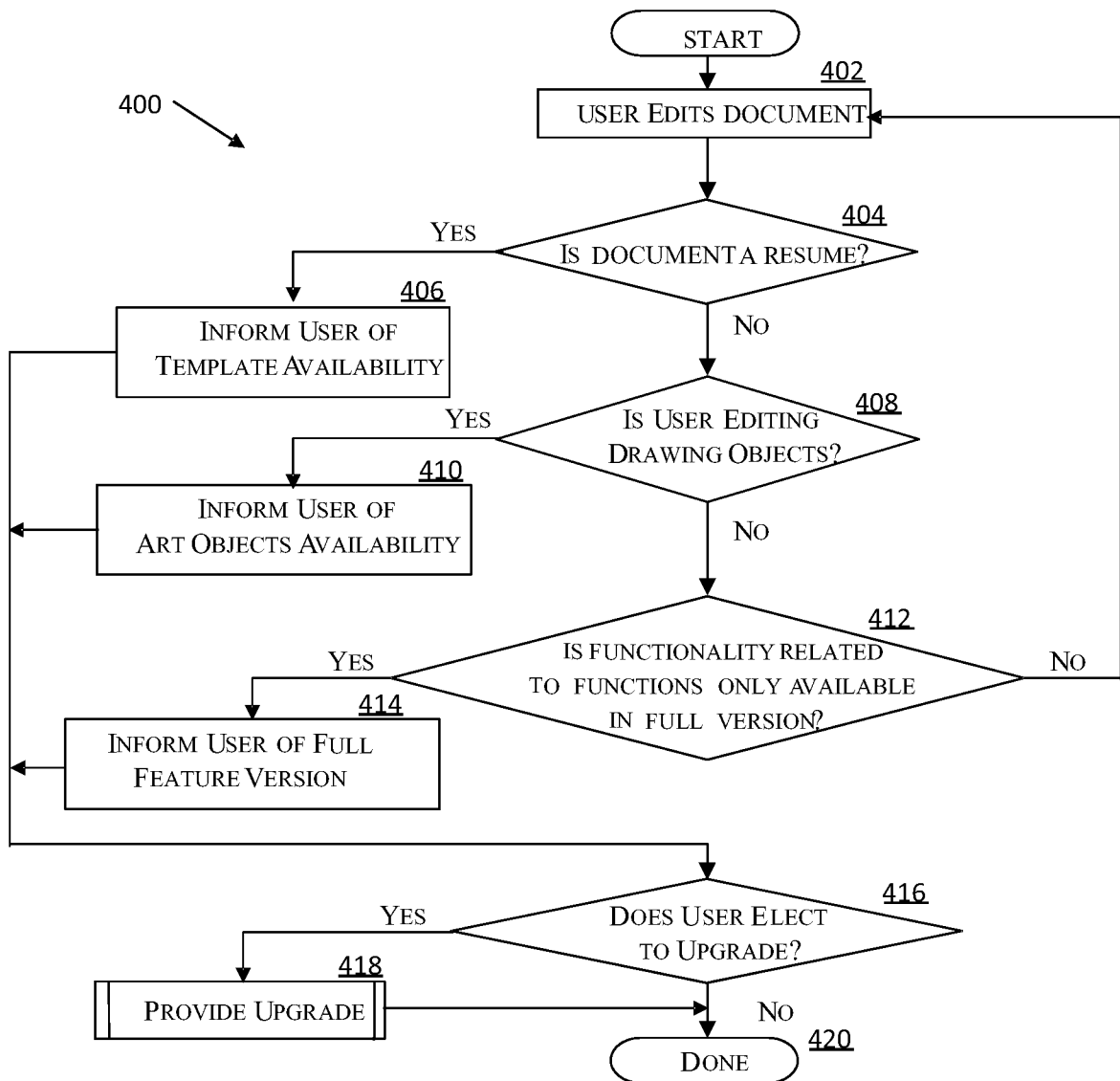
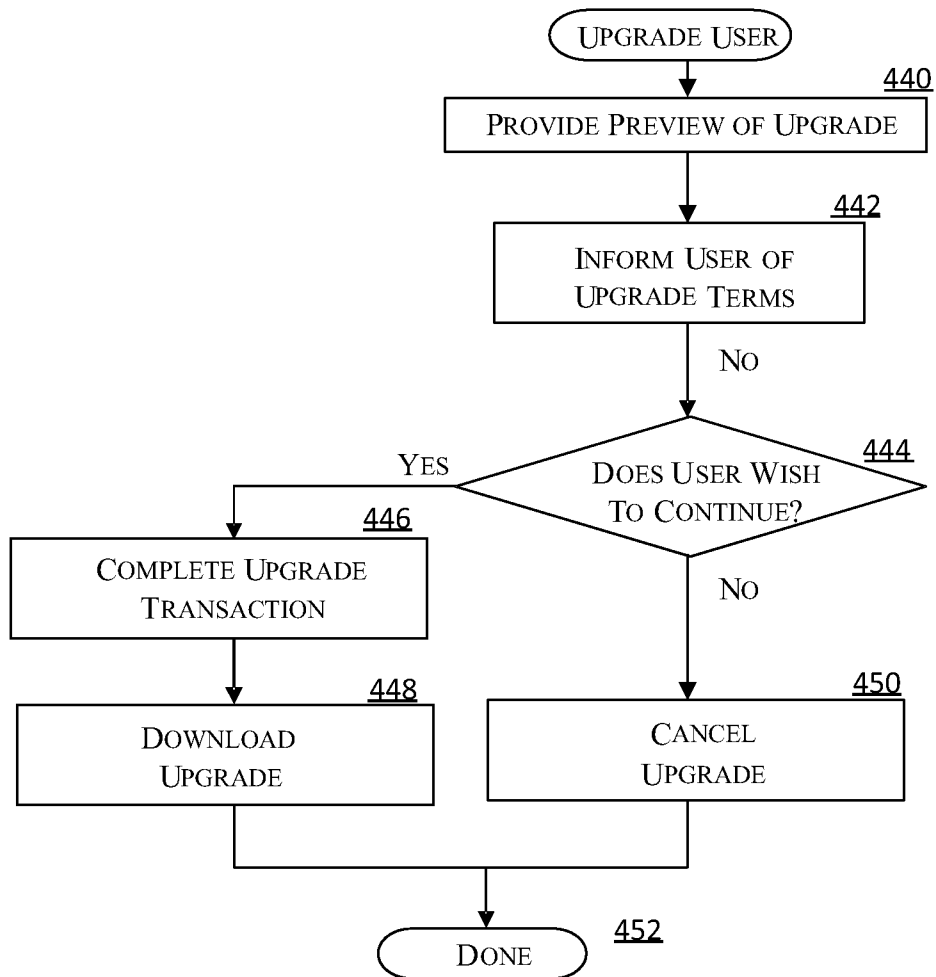


Fig.3

**Fig.4A**

**Fig.4B**

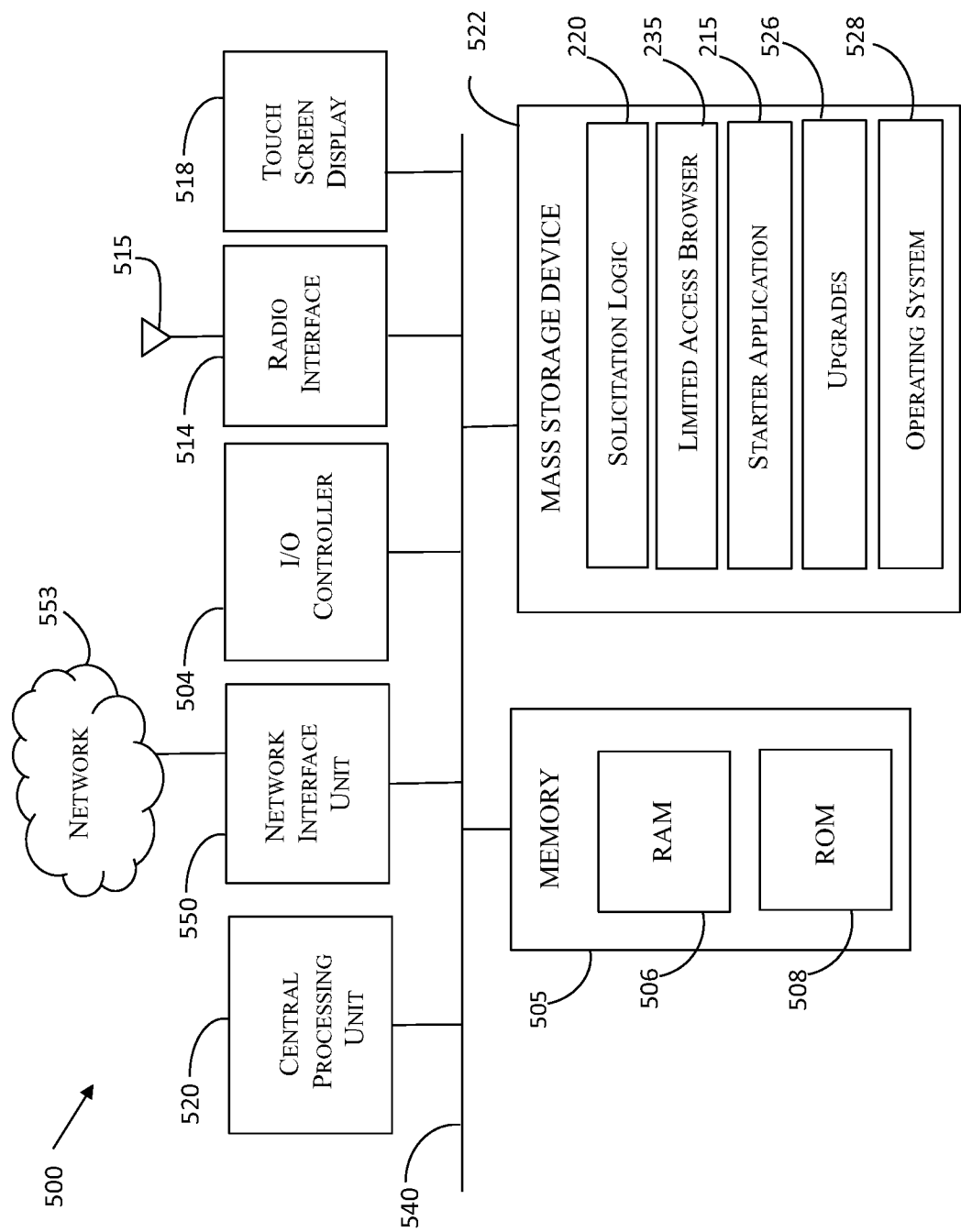


Fig. 5