



(86) Date de dépôt PCT/PCT Filing Date: 2015/03/27

(87) Date publication PCT/PCT Publication Date: 2015/10/08

(45) Date de délivrance/Issue Date: 2022/05/31

(85) Entrée phase nationale/National Entry: 2016/09/19

(86) N° demande PCT/PCT Application No.: US 2015/022894

(87) N° publication PCT/PCT Publication No.: 2015/153319

(30) Priorités/Priorities: 2014/04/02 (US61/974,377);
2014/08/06 (US14/453,445)

(51) Cl.Int./Int.Cl. *G06F 3/0481* (2022.01),
G06F 3/0484 (2022.01), *G06F 9/451* (2018.01)

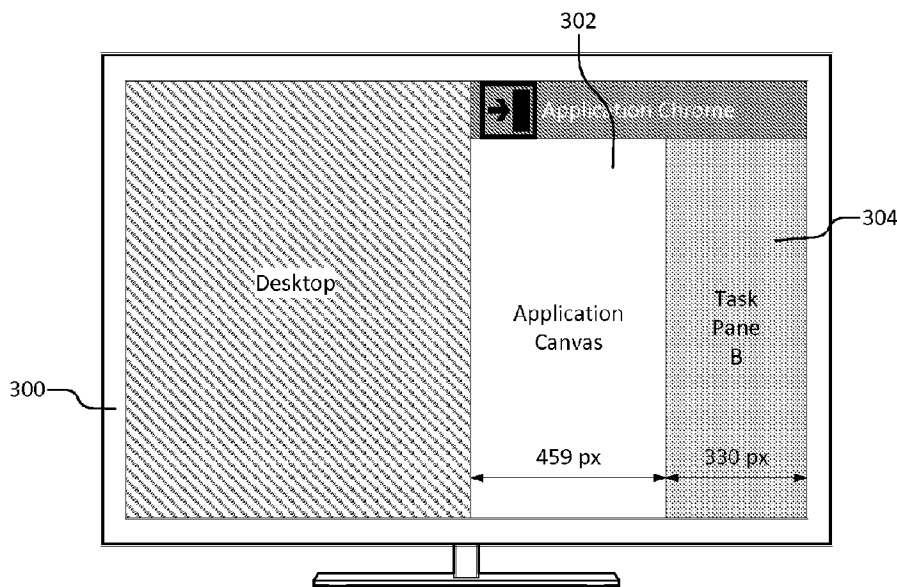
(72) Inventeurs/Inventors:
KAUFTHAL, JONATHAN S., US;
SPOMER, TRAVIS, US;
SLOAN, RYAN, US

(73) Propriétaire/Owner:
MICROSOFT TECHNOLOGY LICENSING, LLC, US

(74) Agent: SMART & BIGGAR LLP

(54) Titre : GESTIONNAIRE ADAPTATIF DE PANNEAUX D'INTERFACE UTILISATEUR

(54) Title: ADAPTIVE USER INTERFACE PANE MANAGER



(57) **Abrégé/Abstract:**

Embodiments of a system for automatically managing panes in relation to the application canvas in a user interface. The system includes a pane manager that works in conjunction with applications to provide access to panes while keeping a minimum amount of user content visible. The system allows an application to define a protected canvas size. When a pane is needed, the system determines if sufficient room for the pane exists or can be created by dismissing another pane. If so, the pane is added. If not, the pane may be displayed in a temporary manner. The system also responds to changes in the user interface size. If the user interface size decreases to a point where sufficient room for existing panes is no longer available, one or more of the existing panes are dismissed.

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

(19) World Intellectual Property
Organization
International Bureau

(43) International Publication Date
8 October 2015 (08.10.2015)



(10) International Publication Number
WO 2015/153319 A1

- (51) International Patent Classification:
G06F 9/44 (2006.01) *G06F 3/0482* (2013.01)
- (21) International Application Number:
PCT/US2015/022894
- (22) International Filing Date:
27 March 2015 (27.03.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
61/974,377 2 April 2014 (02.04.2014) US
14/453,445 6 August 2014 (06.08.2014) US
- (71) Applicant: MICROSOFT TECHNOLOGY LICENSING, LLC [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) Inventors: KAUFTHAL, Jonathan S.; c/o Microsoft Technology Licensing, LLC, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). SPOMER, Travis; c/o Microsoft Technology Licensing, LLC, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). SLOAN, Ryan; c/o Microsoft Technology Licensing, LLC, LCA - International Patents (8/1172), 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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, 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, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,

[Continued on next page]

(54) Title: ADAPTIVE USER INTERFACE PANE MANAGER

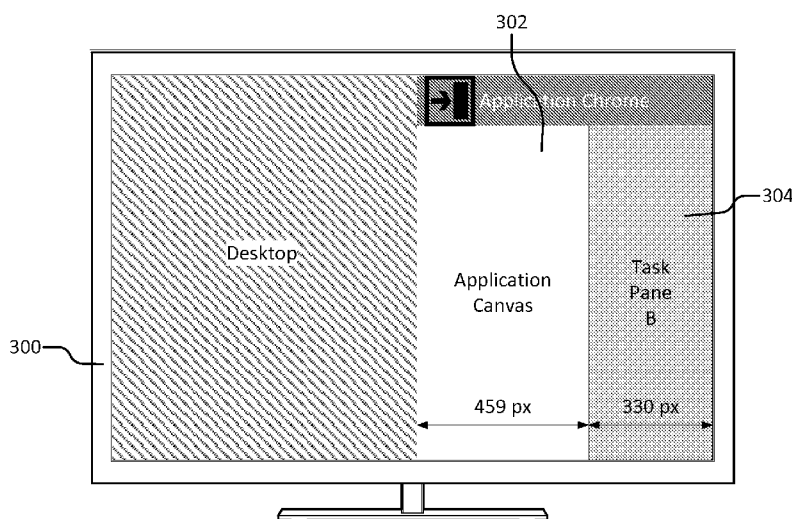


Fig. 3D

(57) Abstract: Embodiments of a system for automatically managing panes in relation to the application canvas in a user interface. The system includes a pane manager that works in conjunction with applications to provide access to panes while keeping a minimum amount of user content visible. The system allows an application to define a protected canvas size. When a pane is needed, the system determines if sufficient room for the pane exists or can be created by dismissing another pane. If so, the pane is added. If not, the pane may be displayed in a temporary manner. The system also responds to changes in the user interface size. If the user interface size decreases to a point where sufficient room for existing panes is no longer available, one or more of the existing panes are dismissed.

WO 2015/153319 A1

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, KM, ML, MR, NE, SN, TD, TG).

— *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))*

Declarations under Rule 4.17:

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

ADAPTIVE USER INTERFACE PANE MANAGER

BACKGROUND

- [0001]** Many user interfaces employ panes to expose application functionality and information to users. Panes are often dedicated to a specific feature or function of the application. When a user performs a task utilizing a feature or function supported by a pane, the pane is added to the user interface. In the user interface, the panes must share space with the application canvas, which holds the primary application content. While useful, panes reduce the amount of space available for the application canvas.
- [0002]** Without restrictions, panes can dominate the user interface, leaving little to no room for the application canvas. One conventional approach to pane management is to have no management and leaving users with the responsibility of pane management. Panes are simply opened and remain until manually closed by the user. The user may also selectively resize the open panes; however, low resolution or small form factor devices and touch-based interfaces are often incapable of providing the precise input response needed to keep the panes at usable sizes and maximum screen usage. Even when the precision is available, the effort of managing the panes detracts from the user experience.
- [0003]** Another conventional approach to pane management has been to place hard-coded restrictions on the number of panes. Hard-coded restrictions are not appropriate in all situations and do not take factors such as display configuration into account. When ample display area is available, the hard-coded number may unnecessarily limit a user from having access to the full number of panes that the user desires. On the other hand, when limited display area is available, the practical limit on the number of panes may be lower than the hard-coded number.
- [0004]** It is with respect to these and other considerations that the present disclosure has been made. Although relatively specific problems have been discussed, it should be understood that the embodiments disclosed herein should not be limited to solving the specific problems identified in the background.

BRIEF SUMMARY

- [0005]** This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description section. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0006] Aspects of a system for automatically managing panes in relation to the application canvas in a user interface include a computing device having at least a processing device, a memory, and a display. The system determines when an event that may affect the canvas size occurs, such as changing the display configuration, resizing the application window, or reorienting the display, which in turn may affect pane management, or the opening or closing of a pane. When such an event occurs, the system collects information, which may include, but is not limited to, the protected canvas size and the current application user interface size.

[0007] When opening of a pane that is not currently open is requested, the system determines if the new pane may be added while preserving the protected canvas size. If the new pane will fit in the available space, the system opens the new pane as a persistent pane. If the new pane will not fit, the system determines if any persistent panes are currently open. If no persistent panes are open, the system opens the new pane as a transient pane rather than a persistent pane. If more than one persistent pane is open, the system closes the lowest priority persistent pane. If only one persistent pane is open and the open pane is non-critical, the system replaces the persistent pane with the new pane. If the only open pane is a critical pane, the system may temporarily close (i.e., suspend) the critical pane. Once the space is made available for the new pane by closing an open non-critical pane or suspending a critical pane, the system opens the new pane as a persistent pane.

[0008] When closing of a currently open pane is requested or the UI size increased, the system attempts to restore a pane that was automatically closed if sufficient space is available for the restored pane. If the UI size is decreased to point where the canvas size is less than the protected canvas size, the system closes panes with the lowest priority until the protected canvas size is reached or all panes have been closed. After managing the panes, the system resizes the canvas and reflows and/or zooms the content as appropriate.

[0008a] According to one aspect of the present invention, there is provided a computer-implemented method of automatically managing panes in relation to an application canvas in a user interface having a width, the method comprising: receiving an indication from an application of a minimum width for the application canvas; receiving a request to add a selected persistent pane in the user interface of the application, wherein a persistent pane comprises a pane that is displayed along with the canvas without overlapping the canvas;

determining that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas; based on the determination that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas, determining if any persistent panes are currently displayed in the user interface; based on a determination that

5 two or more persistent panes are currently displayed in the user interface: closing a sufficient number of the two or more persistent panes, in order of least priority, that are currently displayed, so that the selected persistent pane is able to be displayed along with the canvas without overlapping the canvas while maintaining the minimum width of the canvas; and displaying the selected persistent pane as an opened persistent pane; based on a determination

10 no persistent panes are currently displayed in the user interface: opening a transient pane, wherein a transient pane comprises a pane that exists above the canvas enabling overlapping the canvas; and displaying the selected persistent pane as the transient pane; based on a determination that only one persistent pane is currently displayed in the user interface: determining if the currently displayed persistent pane is a critical pane, wherein a criticality of

15 the persistent pane is previously designated by the application; based on the determination that the currently displayed persistent pane is a non-critical pane, closing the critical pane and determining if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas; and based on the determination that the currently displayed persistent pane is a critical pane,

20 temporarily closing or suspending display of the critical pane and determining if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas.

[0008b] According to another aspect of the present invention, there is provided a computer-implemented system for automatically managing application panes in relation to an

25 application canvas in a user interface, the computer-implemented system comprising: a computing device comprising a processor, a minimum canvas size memory, and an output device for presenting a visual output to a user, the visual output comprising the user interface of an application, wherein the computing device is caused to: receive an indication from the application of a minimum width for the application canvas; receive a request to add a selected

30 persistent pane in the user interface of the application, wherein a persistent pane comprises a pane that is displayed along with the canvas without overlapping the canvas; determine that

the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas; based on the determination that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas, determine if any persistent panes are currently displayed in the user interface; based on a determination that two or more persistent panes are currently displayed in the user interface: close a sufficient number of the two or more persistent panes, in order of least priority, that are currently displayed so that the selected persistent pane is able to be displayed along with the canvas without overlapping the canvas while maintaining the minimum width of the canvas; and display the selected persistent pane as an opened persistent pane; based on a determination no persistent panes are currently displayed in the user interface: open a transient pane, wherein a transient pane comprises a pane that exists above the canvas enabling overlapping the canvas; and display the selected persistent pane as the transient pane; based on a determination that only one persistent pane is currently displayed in the user interface: determine if the currently displayed persistent pane is a critical pane, wherein a criticality of the persistent pane is previously designated by the application; based on the determination that the currently displayed persistent pane is a non-critical pane, close the critical pane and determine if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas; and based on the determination that the currently displayed persistent pane is a critical pane, temporarily close or suspend display of the critical pane and determine if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas.

[0008c] According to still another aspect of the present invention, there is provided a computer readable storage medium, implemented in hardware, containing computer executable instructions which, when executed by a computer, perform a method of automatically managing panes in relation to an application canvas in a user interface of an application, the method comprising: receiving an indication from the application of a minimum width for the application canvas; receiving a request to add a selected persistent pane in the user interface of the application, wherein a persistent pane comprises a pane that is displayed along with the canvas without overlapping the canvas; determining that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas; based on the determination that the selected persistent pane cannot be displayed while

maintaining the minimum width of the canvas, determining if any persistent panes are currently displayed in the user interface; based on a determination that two or more persistent panes are currently displayed in the user interface: closing a sufficient number of the two or more persistent panes, in order of least priority, that are currently displayed so that the
5 selected persistent pane is able to be displayed along with the canvas without overlapping the canvas while maintaining the minimum width of the canvas; and displaying the selected persistent pane as an opened persistent pane; based on a determination no persistent panes are currently displayed in the user interface: opening a transient pane, wherein a transient pane comprises a pane that exists above the canvas enabling overlapping the canvas; and displaying
10 the selected persistent pane as the transient pane; based on a determination that only one persistent pane is currently displayed in the user interface: determining if the currently displayed persistent pane is a critical pane, wherein a criticality of the persistent pane is previously designated by the application: based on the determination that the currently displayed persistent pane is a non-critical pane, closing the critical pane and determining if the
15 selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas; and based on the determination that the currently displayed persistent pane is a critical pane, temporarily closing or suspending display of the critical pane and determining if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the
20 minimum width of the canvas.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Further features, aspects, and advantages of the present disclosure will become better understood by reference to the following figures, wherein elements are not to scale so as to more clearly show the details and wherein like reference numbers indicate like elements
25 throughout the several views:

Figure 1 is a block diagram illustrating a system for automatically managing panes in relation to the application canvas in a user interface of an application;

Figures 2A and 2B form a high level flowchart of one embodiment of a method of automatically managing panes in relation to the application canvas in a user interface;

Figure 2C illustrates a high-level flowchart detailing of one embodiment of the open transient pane operation;

Figure 2D illustrates a high-level flowchart detailing of one embodiment of the critical pane suspension check operation;

5 Figures 3A to 3E illustrate one embodiment of a pane management sequence for an application running on a higher resolution monitor in a landscape orientation;

Figures 3F to 3I illustrate one embodiment of a pane management sequence involving a critical pane for an application running on a higher resolution monitor in a landscape orientation;

10 Figures 4A to 4G illustrate one embodiment of a pane management sequence for an application running on a lower resolution tablet;

Figure 5 is a block diagram illustrating one embodiment of the physical components of a computing device with which embodiments of the present disclosure may be practiced;

15 Figures 6A and 6B are simplified block diagrams of a mobile computing device with which embodiments of the present disclosure may be practiced; and

Figure 7 is a simplified block diagram of a distributed computing system in which embodiments of the present disclosure may be practiced.

DETAILED DESCRIPTION

[0010] Various embodiments are described more fully below with reference to the accompanying drawings, which form a part hereof, and which show specific exemplary
20 embodiments. However, embodiments may be implemented in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the embodiments to those skilled in the art. Embodiments may be
25 practiced as methods, systems, or devices. Accordingly, embodiments may take the form of a hardware implementation, an entirely software implementation or an implementation combining software and hardware aspects. The following detailed description is, therefore, not to be taken in a limiting sense.

[0011] Embodiments of a system for automatically managing panes in relation to
30 the application canvas in a user interface are described herein and illustrated in the accompanying figures. The system includes a pane manager that works in conjunction with applications to provide access to panes while keeping at least a minimum amount of user content visible. The system allows an application to define a protected canvas size. When a pane is needed, the system determines if sufficient room for the pane exists or can be created

by dismissing another pane. If so, the pane is added. If not, the pane may be displayed in a temporary manner. The system also responds to changes in the user interface size. If the user interface size decreases to a point where sufficient room for existing panes is no longer available, one or more of the existing panes are dismissed.

5 **[0012]** Embodiments of the disclosure are described in terms of a traditional user interface layout merely to provide a standard frame of reference; however, the disclosure is not limited to traditional user interface layouts. As such, usage of a particular orientation (e.g., horizontal, vertical, landscape, portrait, etc.), direction (e.g., left, right, up down), position (e.g., top, bottom, side, above, below, front, back, etc.), and dimension (e.g., height, 10 width, etc.) should not be construed as limiting to that particular orientation, direction, position, or dimension.

[0013] Figure 1 is a block diagram illustrating a system for automatically managing panes in the user interface of an application. The system **100** includes a pane manager **102** to manage the display of panes in a user interface **104** of an application **106** based on the 15 display properties of the device being used and the current user interface properties (e.g., canvas size, number of panes open, etc.). The pane manager **102**, user interface **104**, an application **106** are executed on a computing device **108**. The computing device may be one of various types of computing devices (e.g., a tablet computing device, a desktop computer, a mobile communication device, a laptop computer, a laptop/tablet hybrid computing 20 device, a large screen multi-touch display, a gaming device, a smart television, or other types of computing devices) for executing applications for performing a variety of tasks.

[0014] A user **110** may utilize the application on a computing device for a variety of tasks, which may include, for example, to write, calculate, draw, organize, prepare presentations, send and receive electronic mail, take and organize notes, make music, and 25 the like. Applications may include thick client applications, which may be stored locally on the computing device, or may include thin client applications (i.e., web applications) that may reside on a remote server and accessible over a network, such as the Internet or an intranet. A thin client application may be hosted in a browser-controlled environment or coded in a browser-supported language and reliant on a common web browser to render the 30 application executable on a computing device. The computing device may be configured to receive content for presentation on a display **120** (which may comprise a touch screen display).

[0015] The application may be configured to enable a user **110** to use a pointing device (e.g., a mouse, pen/stylus, etc.) and/or to utilize sensors **112** (e.g., touch sensor,

accelerometer, hover, facial recognition, voice recognition, light sensor, proximity sensor, gyroscope, tilt sensor, GPS, etc.) on the computing device **110** to interact with content **114** via a number of input modes. The content may be displayed on the application canvas **116**. The user interface may include one or more selectively displayable panes **118** that contain
5 a plurality of selectable functionality controls and elements. The pane manager automatically controls the display of the panes based on a configuration **122** provided by the application to preserve a minimum amount of space on the display for the application canvas.

[0016] Figures 2A and 2B form a high level flowchart of one embodiment of a
10 method of automatically managing panes in relation to the application canvas in a user interface. The method **200** begins when an event that may affect the canvas size occurs, such as a change in the display configuration (e.g., resolution or orientation) of the device (e.g., caused by changing the display settings or reorienting the display) or altering the user interface (e.g., resizing the application window or opening/closing a pane), which in turn
15 may affect pane management. When such an event occurs, a current size check operation **202** collects information, which may include, but is not limited to, the protected canvas size and the current application user interface size (i.e., window). The protected canvas size specifies an application-specific minimum canvas size to ensure that a meaningful amount of user content remains visible. In other words, the size of the canvas cannot be reduced
20 below the protected canvas size. In various embodiments, the protected canvas size may be specified in a single dimension (e.g., width or height). For example, a protected canvas width may be specified to protect against encroachment by side panes or a protected canvas width height may be specified to protect against encroachment by top/bottom panes. In other embodiments, the protected canvas size may be specified in two dimensions (e.g., width and
25 height).

[0017] The method continues with a response that depends, at least in part, upon the type of event that occurs. If opening of a pane that is not currently open is requested (i.e., a new pane), a fit check operation **204** determines if the new pane may be added while preserving the protected canvas size. The fit check operation may take into account various
30 properties of the requested pane, such as, but not limited to, alignment, entry behavior, title, chrome, critical, height, dock state, and width. Entry behavior specifies whether a pane is intended to be persistent or transient. Persistent (i.e., sticky) panes are intended to coexist with the canvas and may be docked in the user interface. A persistent pane remains until manually dismissed by the user (e.g., using the close widget) or programmatically closed by

the application. Transient panes exist above the canvas. A transient (i.e., light dismiss or popover) pane remains until the user moves the focus to an area outside of the pane. Parameters such as entry behavior, priority, and width may directly influence treatment of the pane by the pane manager.

5 **[0018]** If the new pane will fit in the available space, an open persistent pane operation **216** opens the new pane as a persistent pane. The position of the new pane may be inward of the last opened pane or, if no other panes are open, from one side of the application user interface, or to the outside of the open panes. For example, a pane may be opened to the left of the existing panes. A canvas reconfiguration operation **218** resizes the
10 canvas and reflows and/or zooms the content as appropriate.

[0019] If the new pane will not fit (i.e., the new canvas size would be less than the protected canvas size), a pane inventory operation **206** determines if any persistent panes are currently open. If no persistent panes are open, an open transient pane operation **208** opens the pane as a popover pane rather than a persistent pane. If more than one persistent
15 pane is open, a close persistent pane operation **214** closes the persistent pane with the lowest priority. The selection of the pane with lowest priority may be based on criteria such as, but not limited to, primacy (i.e., the oldest open pane is closed), recency (i.e., the last pane to be opened is closed), and non-criticality (i.e., a non-critical pane is closed before closing a critical pane).

20 **[0020]** If only one persistent pane is open, a criticality determination **210** is made. If the open pane is non-critical, the method proceeds to the close persistent pane operation **214**. If the only open pane is critical, a critical pane suspension operation **212** may temporarily close the critical pane. In various embodiments, the critical pane suspension operation may flag that a critical pane has been closed. The critical pane may actually be
25 closed or may simply be hidden from view.

[0021] After panes have been closed or suspended through the critical pane suspension operation **212** or the close persistent pane operation **214**, operation may return to the fit check operation **204** to re-evaluate the available space. If sufficient space is available, the method continues with the open persistent pane operation **216** and the canvas
30 reconfiguration operation **218**, if necessary. If not, operations **206** to **214** may be repeated or explored based on the current pane state.

[0022] Returning to the event type decision, if a close pane event occurs, a close pane operation **219** closes the pane freeing up space in the UI area. A critical pane suspension check operation **220** attempts to restore a critical pane that has been previously

suspended, provided that sufficient space for the critical pane is available. The method continues with the canvas reconfiguration operation **218**, if necessary.

[0023] Returning to the event type decision, if a UI size change event occurs, the method continues (Figure 2B) with a UI size change direction determination **222**. A UI size change may occur independently of opening or closing a pane. A UI size change event may be caused by directly resizing the UI area (i.e., window) or a component of the UI (e.g., the application canvas or panes) or by reconfiguration of the display (e.g., changing the resolution or orientation of the display). If the UI size has increased, meaning more space is potentially available for the canvas and panes, the critical pane suspension check operation **220** attempts to restore a critical pane that has been previously suspended, provided that sufficient space for the critical pane is available. If the UI size has decreased, an ensure protected canvas size operation **224** determines whether the resulting canvas size is less than the protected canvas size and closes panes with the lowest priority until the protected canvas size is reached or all panes have been closed. Instead of closing the low-priority panes, the ensure protected canvas size operation **224** may selectively convert one or more of the low priority persistent panes into transient panes to create space. Conversion of currently opened persistent panes to transient panes may be limited to only those panes that would have been originally opened as transient panes if sufficient space had not been available when that pane was opened. The method continues with the canvas reconfiguration operation **218** (Figure 2A), if necessary.

[0024] Figure 2C illustrates a high-level flowchart detailing of one embodiment of the open transient pane operation. The open transient operation **208** begins with an open popover pane operation **230** that opens the new pane as a transient pane covering a portion of the canvas. An optional focusing operation **232** may provide any special handling of the canvas, document, and/or user interface when a transient pane is displayed. Depending upon the application, the focusing operation may manipulate the canvas or document to keep the selected object in view in the portion of the canvas not covered by the transient pane. For example, in a word processing or a spreadsheet application, the focusing operation may scroll the canvas to keep the selected object in view. Similarly, in a presentation application, the focusing operation may scale the document to keep the selected object in view. In the case of a more modular application, such as a combination email, calendaring, and task application, the focusing operation may optionally change the shape and/or configuration the application user interface to make the best use of the remaining available space. An automatic dismissal operation **234** dismisses the transient pane once the transient pane loses

focus. An optional canvas restore operation **236** may undo any special handling performed by the pan operation. For example, the canvas restore operation may scroll the canvas back to its original position.

[0025] Figure 2D illustrates a high-level flowchart detailing of one embodiment of the critical pane suspension check operation. The critical pane suspension check operation **220** begins with suspended critical pane identification operation **240** that determines if a panel designated as critical by the application is not currently open. If a critical pane is closed, a fit check determination **242** determines if there is sufficient space to open the critical pane. If so, a restore critical pane operation **244** loads or unhides the critical pane, as appropriate. Although described in terms of critical panes, the critical pane suspension check operation may be used to reopen other panes (e.g., non-critical panes that have been automatically closed).

[0026] Figures 3A to 3E illustrate one embodiment of a pane management sequence for an application running on a higher resolution monitor in a landscape orientation. The application defines the protected canvas size as being 400 pixels wide. In landscape orientation, the monitor **300** provides a display width of 1600 pixels. In Figure 3A, no panes are displayed and the application canvas **302** occupies the entire width of the display.

[0027] Figure 3B shows the user interface after a first pane **304** with a width of 330 pixels has been requested. The resulting width of the canvas is 1270 pixels so the pane manager displays the first pane and resizes/reflows the canvas accordingly.

[0028] Figure 3C shows the user interface after a second pane **306** with a width of 330 pixels has been requested. Even after adding a second pane, the resulting width of the canvas is 940 pixels, which is above the protected canvas size of 400 pixels. Accordingly, the pane manager displays the second pane to the left of the first pane and resizes/reflows the canvas accordingly.

[0029] Figure 3D shows the result of resizing the application user interface to a split screen layout which reduces the portion of the display available to the application user interface to 789 pixels. In the new configuration, the canvas size would be limited to 129 pixels if both panes were displayed. To maintain the minimum protected canvas size, the pane manager dismisses the first pane, which is the oldest pane, and retains the second pane. The canvas is resized/reflowed accordingly.

[0030] Figure 3E shows the result of resizing the application user interface to a full screen layout which restores the portion of the display available to the application user interface to the full 1600 pixels. In the illustrated embodiment, existing panes are retained,

but previously closed panes are not restored (i.e., reopened). In various embodiments, the system may attempt to restore closed panes if space is available. For example, the system may maintain a pane history and attempt to automatically reopen the most recently closed panes as space becomes available. In some embodiments, only automatically closed panes and critical panes are automatically restored. In some embodiments, only critical panes are automatically restored. The canvas is resized/reflowed accordingly.

[0031] Figures 3F to 3I illustrate one embodiment of a pane management sequence for an application running on a higher resolution monitor in a landscape orientation involving a critical pane. Figure 3F shows the application user interface with the application canvas **302** bounded by a critical pane **308** and two non-critical panes **304**, **306**. The application may define a pane as a critical pane. A critical pane may be a pane that is fundamental to the application user interface. A critical pane need not be present at all times, but should usually be available to the user. The pane manager does not automatically dismiss a critical pane unless the current user interface size only allows for a single pane to be open at a time. The pane manager will automatically restore the critical pane when the user interface size increases sufficiently or the other pane is closed. In various embodiments, the pane manager may automatically restore a critical pane even if it was manually closed. In some embodiments, the application may only designate one pane as a critical pane.

[0032] Figure 3G shows the result of resizing the application user interface to a split screen which reduces the portion of the display available to the application user interface to 789 pixels. In the new configuration, there is only sufficient room for one pane to the canvas size above the protected canvas size. Accordingly, the pane manager dismisses both non-critical panes and retains the critical pane. The canvas is resized/reflowed accordingly.

[0033] Figure 3H shows the user interface in the split screen layout after a third pane **310** with a width of 330 pixels has been requested. Because only there is only room for one pane, critical pane is temporarily suspended in favor of the third task pane. The canvas is resized/reflowed accordingly.

[0034] Figure 3I shows the result of resizing the application user interface to a full screen layout which restores the portion of the display available to the application user interface to the full 1600 pixels. Existing panes are retained. In addition, because sufficient space is now available, the critical pane is restored. The canvas is resized/reflowed accordingly.

[0035] Figures 4A to 4C illustrate one embodiment of a pane management sequence for an application running on a lower resolution tablet. The application defines the protected

canvas size as being 400 pixels wide. In landscape orientation, the tablet provides a display width of 1024 pixels. In Figure 4A, no panes are displayed and the application canvas **402** occupies the entire width of the display.

[0036] Figure 4B shows the user interface after a first pane **404** with a width of 330 pixels has been requested. The resulting width of the canvas is 694 pixels so the pane manager displays the first pane and resizes/reflows the canvas accordingly.

[0037] Figure 4C shows the user interface after a second pane **406** with a width of 330 pixels has been requested. Adding a second pane would reduce the width of the canvas to 364 pixels, which is below the protected canvas size of 400 pixels. The display cannot accommodate both the first pane and second pane at the same time. Accordingly, the first pane is dismissed and the second pane is added. In effect, the second pane replaces the first pane.

[0038] Figure 4D shows the application user interface in a split screen layout, which reduces the portion of the display available to the application user interface to 501 pixels. In this configuration, the canvas size would be limited to 171 pixels if a pane was displayed. Accordingly, the existing pane is dismissed. The canvas is resized/reflowed accordingly.

[0039] Figure 4E shows the handling of panes when the display size and the protected application size do not allow for persistent panes. Rather than to exclude all panes for being displayed, when a pane is requested, the pane manager displays the pane as a transient pane **408**. The transient pane covers at least a portion of the canvas. In the illustrated embodiment, the transient pane covers a portion of the application chrome. In other embodiments, the transient pane may not cover the application chrome. Transient panes may also appear as full screen panes or near-immersive panes, which cover a substantial portion of the total user interface area or the total display area. The application canvas is not resized/reflowed. When transient pane loses focus (i.e., the user clicks outside of the transient pane), the transient pane is dismissed, leaving the application canvas fully visible once more.

[0040] Figures 4F and 4G illustrate one embodiment of a pane management sequence with the tablet rotated to a portrait orientation. In portrait orientation, the tablet provides a display width of 768 pixels. In Figure 4F, no panes are displayed and the application canvas occupies the entire width of the display.

[0041] Figure 4G shows the user interface after the first pane **404** with a width of 330 pixels has been requested. The resulting width of the canvas is 438 pixels so the pane manager displays the first pane and resizes/reflows the canvas accordingly.

[0042] The subject matter of this application may be practiced in a variety of embodiments as systems, devices, and other articles of manufacture or as methods. Embodiments may be implemented as hardware, software, computer readable media, or a combination thereof. The embodiments and functionalities described herein may operate via a multitude of computing systems including, without limitation, desktop computer systems, wired and wireless computing systems, mobile computing systems (e.g., mobile telephones, netbooks, tablet or slate type computers, notebook computers, and laptop computers), handheld devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, and mainframe computers.

[0043] User interfaces and information of various types may be displayed via on-board computing device displays or via remote display units associated with one or more computing devices. For example, user interfaces and information of various types may be displayed and interacted with on a wall surface onto which user interfaces and information of various types are projected. Interaction with the multitude of computing systems with which embodiments of the disclosure may be practiced include, keystroke entry, touch screen entry, voice or other audio entry, gesture entry where an associated computing device is equipped with detection (e.g., camera) functionality for capturing and interpreting user gestures for controlling the functionality of the computing device, and the like.

[0044] Figures 5 through 7 and the associated descriptions provide a discussion of a variety of operating environments in which embodiments of the disclosure may be practiced. However, the devices and systems illustrated and discussed are for purposes of example and illustration and are not limiting of a vast number of computing device configurations that may be utilized for practicing embodiments of the disclosure described above.

[0045] Figure 5 is a block diagram illustrating physical components (i.e., hardware) of a computing device **500** with which embodiments of the disclosure may be practiced. The computing device components described below may be suitable for embodying computing devices including, but not limited to, a personal computer, a tablet computer, a surface computer, and a smart phone, or any other computing device discussed herein. In a basic configuration, the computing device **500** may include at least one processing unit **502** and a system memory **504**. Depending on the configuration and type of computing device, the system memory **504** may comprise, but is not limited to, volatile storage (e.g., random access memory), non-volatile storage (e.g., read-only memory), flash memory, or any combination of such memories. The system memory **504** may include an operating system **505** and one or more program modules **506** suitable for running software applications **520**

such as the pane manager **102**. For example, the operating system **505** may be suitable for controlling the operation of the computing device **500**. Furthermore, embodiments of the disclosure may be practiced in conjunction with a graphics library, other operating systems, or any other application program and is not limited to any particular application or system.

5 This basic configuration is illustrated by those components within a dashed line **508**. The computing device **500** may have additional features or functionality. For example, the computing device **500** may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated by a removable storage device **509** and a non-removable storage device
10 **510**.

[0046] As stated above, a number of program modules and data files may be stored in the system memory **504**. While executing on the processing unit **502**, the software applications **520** may perform processes including, but not limited to, one or more of the stages of the method **200**. Other program modules that may be used in accordance with
15 embodiments of the present disclosure may include electronic mail and contacts applications, word processing applications, spreadsheet applications, database applications, slide presentation applications, or computer-aided drawing application programs, etc.

[0047] Furthermore, embodiments of the disclosure may be practiced in an electrical circuit comprising discrete electronic elements, packaged or integrated electronic chips
20 containing logic gates, a circuit utilizing a microprocessor, or on a single chip containing electronic elements or microprocessors. For example, embodiments of the disclosure may be practiced via a system-on-a-chip (SOC) where each or many of the illustrated components may be integrated onto a single integrated circuit. Such an SOC device may include one or more processing units, graphics units, communications units, system
25 virtualization units and various application functionality all of which are integrated (or “burned”) onto the chip substrate as a single integrated circuit. When operating via an SOC, the functionality described herein with respect to the software applications **520** may be operated via application-specific logic integrated with other components of the computing device **500** on the single integrated circuit (chip). Embodiments of the disclosure may also
30 be practiced using other technologies capable of performing logical operations such as, for example, AND, OR, and NOT, including but not limited to mechanical, optical, fluidic, and quantum technologies. In addition, embodiments of the disclosure may be practiced within a general purpose computer or in any other circuits or systems.

[0048] The computing device **500** may also have one or more input device(s) **512** such as a keyboard, a mouse, a pen, a sound input device, a touch input device, etc. The output device(s) **514** such as a display, a projector, speakers, a printer, etc. may also be included. The aforementioned devices are examples and others may be used. The computing
5 device **500** may include one or more communication connections **516** allowing communications with other computing devices **518**. Examples of suitable communication connections **516** include, but are not limited to, RF transmitter, receiver, and/or transceiver circuitry; universal serial bus (USB), parallel, and/or serial ports.

[0049] The term computer readable media as used herein may include computer
10 storage media. Computer storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, or program modules. The system memory **504**, the removable storage device **509**, and the non-removable storage device **510** are all examples of computer storage media (i.e., memory storage). Computer
15 storage media may include random access memory (RAM), read only memory (ROM), electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, compact disc read only memory (CD-ROM), digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other article of manufacture which can be used to
20 store information and which can be accessed by the computing device **500**. Any such computer storage media may be part of the computing device **500**.

[0050] Figures 6A and 6B illustrate a mobile computing device **600** with which embodiments of the disclosure may be practiced. Examples of suitable mobile computing devices include, but are not limited to, a mobile telephone, a smart phone, a tablet computer,
25 a surface computer, and a laptop computer. In a basic configuration, the mobile computing device **600** is a handheld computer having both input elements and output elements. The mobile computing device **600** typically includes a display **605** and one or more input buttons **610** that allow the user to enter information into the mobile computing device **600**. The display **605** of the mobile computing device **600** may also function as an input device (e.g.,
30 a touch screen display). If included, an optional side input element **615** allows further user input. The side input element **615** may be a rotary switch, a button, or any other type of manual input element. In alternative embodiments, mobile computing device **600** may incorporate more or less input elements. For example, the display **605** may not be a touch screen in some embodiments. In yet another alternative embodiment, the mobile computing

device **600** is a portable phone system, such as a cellular phone. The mobile computing device **600** may also include an optional keypad **635**. Optional keypad **635** may be a physical keypad or a “soft” keypad generated on the touch screen display. In various embodiments, the output elements include the display **605** for showing a graphical user interface, a visual
5 indicator **620** (e.g., a light emitting diode), and/or an audio transducer **625** (e.g., a speaker). In some embodiments, the mobile computing device **600** incorporates a vibration transducer for providing the user with tactile feedback. In yet another embodiment, the mobile computing device **600** incorporates input and/or output ports, such as an audio input (e.g., a microphone jack), an audio output (e.g., a headphone jack), and a video output (e.g., a HDMI
10 port) for sending signals to or receiving signals from an external device.

[0051] Figure 6B is a block diagram illustrating the architecture of one embodiment of a mobile computing device. That is, the mobile computing device **600** can incorporate a system (i.e., an architecture) **602** to implement some embodiments. In one embodiment, the system **602** is implemented as a smart phone capable of running one or more applications
15 (e.g., browsers, e-mail clients, notes, contact managers, messaging clients, games, and media clients/players). In some embodiments, the system **602** is integrated as a computing device, such as an integrated personal digital assistant (PDA) and wireless phone.

[0052] One or more application programs **665** may be loaded into the memory **662** and run on or in association with the operating system **664**. Examples of the application
20 programs include phone dialer programs, e-mail programs, personal information management (PIM) programs, word processing programs, spreadsheet programs, Internet browser programs, messaging programs, and so forth. The system **602** also includes a non-volatile storage area **668** within the memory **662**. The non-volatile storage area **668** may be used to store persistent information that should not be lost if the system **602** is powered
25 down. The application programs **665** may use and store information in the non-volatile storage area **668**, such as e-mail or other messages used by an e-mail application, and the like. A synchronization application (not shown) also resides on the system **602** and is programmed to interact with a corresponding synchronization application resident on a host computer to keep the information stored in the non-volatile storage area **668** synchronized
30 with corresponding information stored at the host computer. As should be appreciated, other applications may be loaded into the memory **662** and run on the mobile computing device **600**, including software applications **520** described herein.

[0053] The system **602** has a power supply **670**, which may be implemented as one or more batteries. The power supply **670** might further include an external power source,

such as an AC adapter or a powered docking cradle that supplements or recharges the batteries.

[0054] The system **602** may also include a radio **672** that performs the function of transmitting and receiving radio frequency communications. The radio **672** facilitates wireless connectivity between the system **602** and the outside world via a communications carrier or service provider. Transmissions to and from the radio **672** are conducted under control of the operating system **664**. In other words, communications received by the radio **672** may be disseminated to the application programs **665** via the operating system **664**, and vice versa.

[0055] The visual indicator **620** may be used to provide visual notifications, and/or an audio interface **674** may be used for producing audible notifications via the audio transducer **625**. In the illustrated embodiment, the visual indicator **620** is a light emitting diode (LED) and the audio transducer **625** is a speaker. These devices may be directly coupled to the power supply **670** so that when activated, they remain on for a duration dictated by the notification mechanism even though the processor **660** and other components might shut down for conserving battery power. The LED may be programmed to remain on indefinitely until the user takes action to indicate the powered-on status of the device. The audio interface **674** is used to provide audible signals to and receive audible signals from the user. For example, in addition to being coupled to the audio transducer **625**, the audio interface **674** may also be coupled to a microphone to receive audible input, such as to facilitate a telephone conversation. In accordance with embodiments of the present disclosure, the microphone may also serve as an audio sensor to facilitate control of notifications, as will be described below. The system **602** may further include a video interface **676** that enables an operation of an on-board camera **630** to record still images, video stream, and the like.

[0056] A mobile computing device **600** implementing the system **602** may have additional features or functionality. For example, the mobile computing device **600** may also include additional data storage devices (removable and/or non-removable) such as, magnetic disks, optical disks, or tape. Such additional storage is illustrated by the non-volatile storage area **668**.

[0057] Data/information generated or captured by the mobile computing device **600** and stored via the system **602** may be stored locally on the mobile computing device **600**, as described above, or the data may be stored on any number of storage media that may be accessed by the device via the radio **672** or via a wired connection between the mobile

computing device **600** and a separate computing device associated with the mobile computing device **600**, for example, a server computer in a distributed computing network, such as the Internet. As should be appreciated such data/information may be accessed via the mobile computing device **600** via the radio **672** or via a distributed computing network.

- 5 Similarly, such data/information may be readily transferred between computing devices for storage and use according to well-known data/information transfer and storage means, including electronic mail and collaborative data/information sharing systems.

[0058] Figure 7 illustrates one embodiment of the architecture of a system for providing multi-voice font interpolation functionality to one or more client devices, as described above. Content developed, interacted with, or edited in association with the software applications **520** may be stored in different communication channels or other storage types. For example, various documents may be stored using a directory service **722**, a web portal **724**, a mailbox service **726**, an instant messaging store **728**, or a social networking site **730**. The software applications **520** may use any of these types of systems or the like for enabling data utilization, as described herein. A server **720** may provide the software applications **520** to clients. As one example, the server **720** may be a web server providing the software applications **520** over the web. The server **720** may provide the software applications **520** over the web to clients through a network **715**. By way of example, the client computing device may be implemented as the computing device **500** and embodied in a personal computer **702a**, a tablet computer **702b**, and/or a mobile computing device (e.g., a smart phone) **702c**. Any of these embodiments of the client device may obtain content from the store **716**.

10
15
20

[0059] The description and illustration of one or more embodiments provided in this application are intended to provide a complete thorough and complete disclosure the full scope of the subject matter to those skilled in the art and not intended to limit or restrict the scope of the disclosure as claimed in any way. The embodiments, examples, and details provided in this application are considered sufficient to convey possession and enable those skilled in the art to practice the best mode of claimed disclosure. Descriptions of structures, resources, operations, and acts considered well-known to those skilled in the art may be brief or omitted to avoid obscuring lesser known or unique aspects of the subject matter of this application. The claimed disclosure should not be construed as being limited to any embodiment, example, or detail provided in this application unless expressly stated herein. Regardless of whether shown or described collectively or separately, the various features (both structural and methodological) are intended to be selectively included or omitted to

25
30

produce an embodiment with a particular set of features. Further, any or all of the functions and acts shown or described may be performed in any order or concurrently. Having been provided with the description and illustration of the present application, one skilled in the art may envision variations, modifications, and alternate embodiments of the broader
5 aspects of the general inventive concept embodied in this application that do not depart from the broader scope of the claimed disclosure.

CLAIMS:

1. A computer-implemented method of automatically managing panes in relation to an application canvas in a user interface having a width, the method comprising:

receiving an indication from an application of a minimum width for the application canvas;

receiving a request to add a selected persistent pane in the user interface of the application, wherein a persistent pane comprises a pane that is displayed along with the canvas without overlapping the canvas;

determining that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas;

based on the determination that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas, determining if any persistent panes are currently displayed in the user interface;

based on a determination that two or more persistent panes are currently displayed in the user interface:

closing a sufficient number of the two or more persistent panes, in order of least priority, that are currently displayed, so that the selected persistent pane is able to be displayed along with the canvas without overlapping the canvas while maintaining the minimum width of the canvas; and

displaying the selected persistent pane as an opened persistent pane;

based on a determination no persistent panes are currently displayed in the user interface:

opening a transient pane, wherein a transient pane comprises a pane that exists above the canvas enabling overlapping the canvas; and

displaying the selected persistent pane as the transient pane;

based on a determination that only one persistent pane is currently displayed in the user interface:

determining if the currently displayed persistent pane is a critical pane, wherein a criticality of the persistent pane is previously designated by the application;

based on the determination that the currently displayed persistent pane is a non-critical pane, closing the critical pane and determining if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas; and

based on the determination that the currently displayed persistent pane is a critical pane, temporarily closing or suspending display of the critical pane and determining if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas.

2. The method of claim 1, wherein the priority is determined based on one or more of primacy of the persistent pane, recency of the persistent pane and criticality of the persistent pane.
3. The method of claim 1, further comprising dismissing the transient pane when the transient pane loses focus.
4. The method of claim 1, wherein displaying the selected persistent pane as the opened persistent pane includes reducing the width the of the canvas to its minimum width.
5. A computer-implemented system for automatically managing application panes in relation to an application canvas in a user interface, the computer-implemented system comprising:

a computing device comprising a processor, a minimum canvas size memory, and an output device for presenting a visual output to a user, the visual output comprising the user interface of an application, wherein the computing device is caused to:

receive an indication from the application of a minimum width for the application canvas;

receive a request to add a selected persistent pane in the user interface of the application, wherein a persistent pane comprises a pane that is displayed along with the canvas without overlapping the canvas;

determine that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas;

based on the determination that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas, determine if any persistent panes are currently displayed in the user interface;

based on a determination that two or more persistent panes are currently displayed in the user interface:

close a sufficient number of the two or more persistent panes, in order of least priority, that are currently displayed so that the selected persistent pane is able to be displayed along with the canvas without overlapping the canvas while maintaining the minimum width of the canvas; and

display the selected persistent pane as an opened persistent pane;

based on a determination no persistent panes are currently displayed in the user interface:

open a transient pane, wherein a transient pane comprises a pane that exists above the canvas enabling overlapping the canvas; and

display the selected persistent pane as the transient pane;

based on a determination that only one persistent pane is currently displayed in the user interface:

determine if the currently displayed persistent pane is a critical pane, wherein a criticality of the persistent pane is previously designated by the application;

based on the determination that the currently displayed persistent pane is a non-critical pane, close the critical pane and determine if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas; and

based on the determination that the currently displayed persistent pane is a critical pane, temporarily close or suspend display of the critical pane and determine if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas.

6. The system of claim 5, wherein the priority is determined based on one or more of primacy of the persistent pane, recency of the persistent pane and criticality of the persistent pane.

7. The system of claim 5, wherein the computing device is further caused to dismiss the transient pane when the transient pane loses focus.

8. The system of claim 5, wherein display of the selected persistent pane as the opened persistent pane includes the reduction of the width the of the canvas to its minimum width.

9. A computer readable storage medium, implemented in hardware, containing computer executable instructions which, when executed by a computer, perform a method of automatically managing panes in relation to an application canvas in a user interface of an application, the method comprising:

receiving an indication from the application of a minimum width for the application canvas;

receiving a request to add a selected persistent pane in the user interface of the application, wherein a persistent pane comprises a pane that is displayed along with the canvas without overlapping the canvas;

determining that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas;

based on the determination that the selected persistent pane cannot be displayed while maintaining the minimum width of the canvas, determining if any persistent panes are currently displayed in the user interface;

based on a determination that two or more persistent panes are currently displayed in the user interface:

closing a sufficient number of the two or more persistent panes, in order of least priority, that are currently displayed so that the selected persistent pane is able to be displayed along with the canvas without overlapping the canvas while maintaining the minimum width of the canvas; and

displaying the selected persistent pane as an opened persistent pane;

based on a determination no persistent panes are currently displayed in the user interface:

opening a transient pane, wherein a transient pane comprises a pane that exists above the canvas enabling overlapping the canvas; and

displaying the selected persistent pane as the transient pane;

based on a determination that only one persistent pane is currently displayed in the user interface:

determining if the currently displayed persistent pane is a critical pane, wherein a criticality of the persistent pane is previously designated by the application:

based on the determination that the currently displayed persistent pane is a non-critical pane, closing the critical pane and determining if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas; and

based on the determination that the currently displayed persistent pane is a critical pane, temporarily closing or suspending display of the critical pane and determining if the selected persistent pane can now be displayed along with the canvas, without overlapping the canvas, while maintaining the minimum width of the canvas.

10. The computer-readable storage medium of claim 9, wherein the priority is determined based on one or more of primacy of the persistent pane, recency of the persistent pane and criticality of the persistent pane.

11. The computer-readable storage medium of claim 9, the method further comprising dismissing the transient pane when the transient pane loses focus.

12. The computer-readable storage medium of claim 9, wherein displaying the selected persistent pane as the opened persistent pane includes reducing the width the of the canvas to its minimum width.

1/15

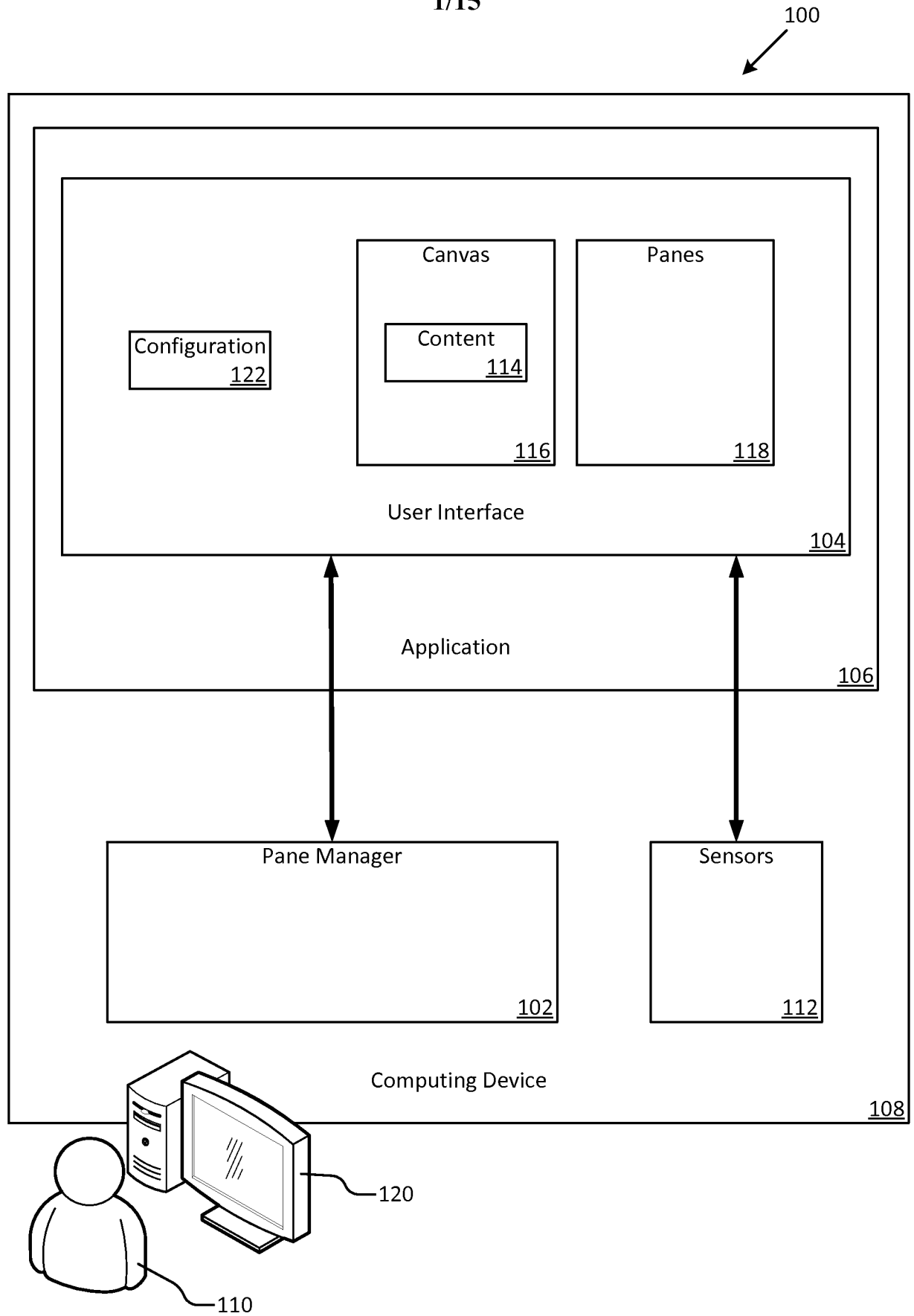


Fig. 1

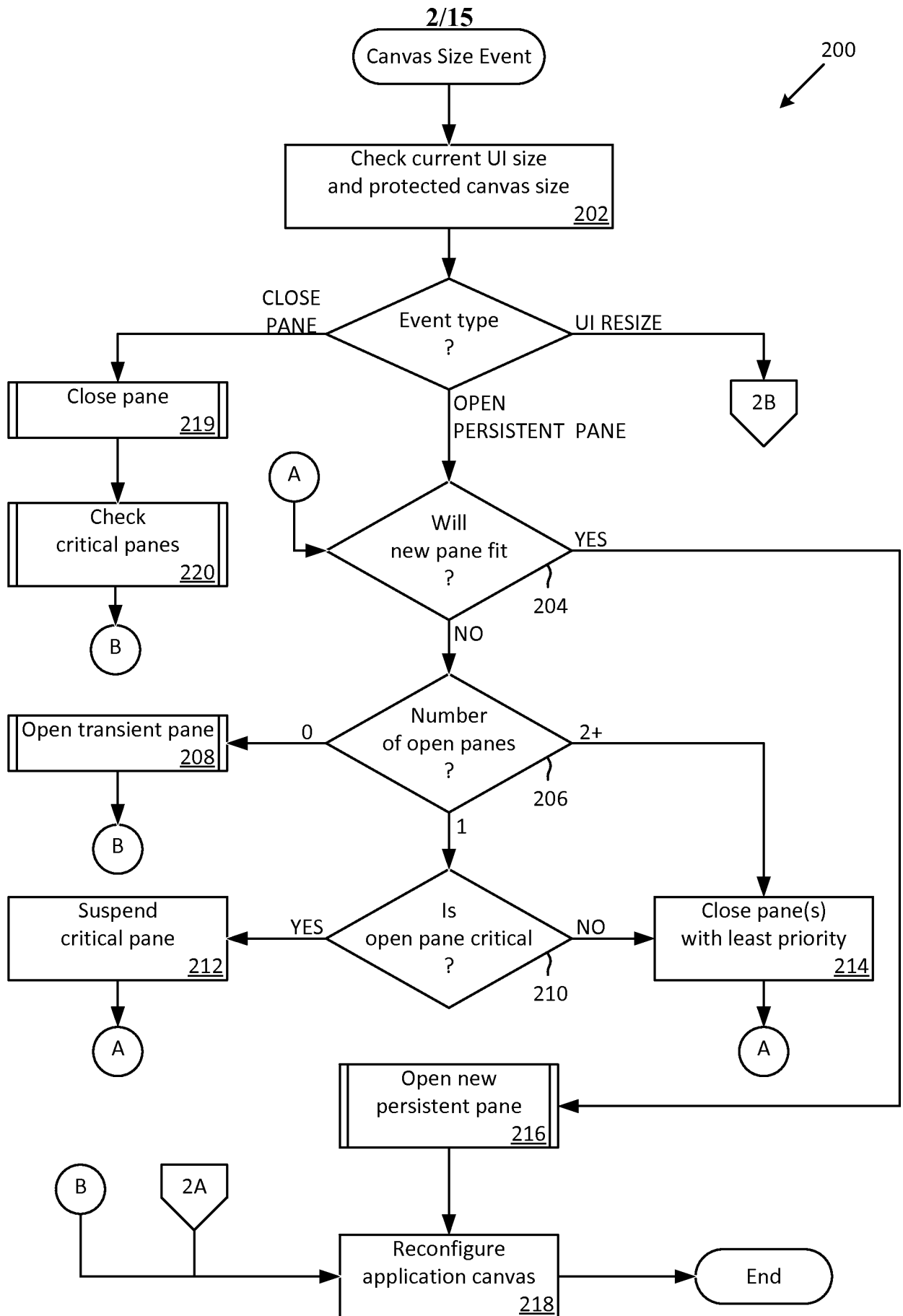


Fig. 2A

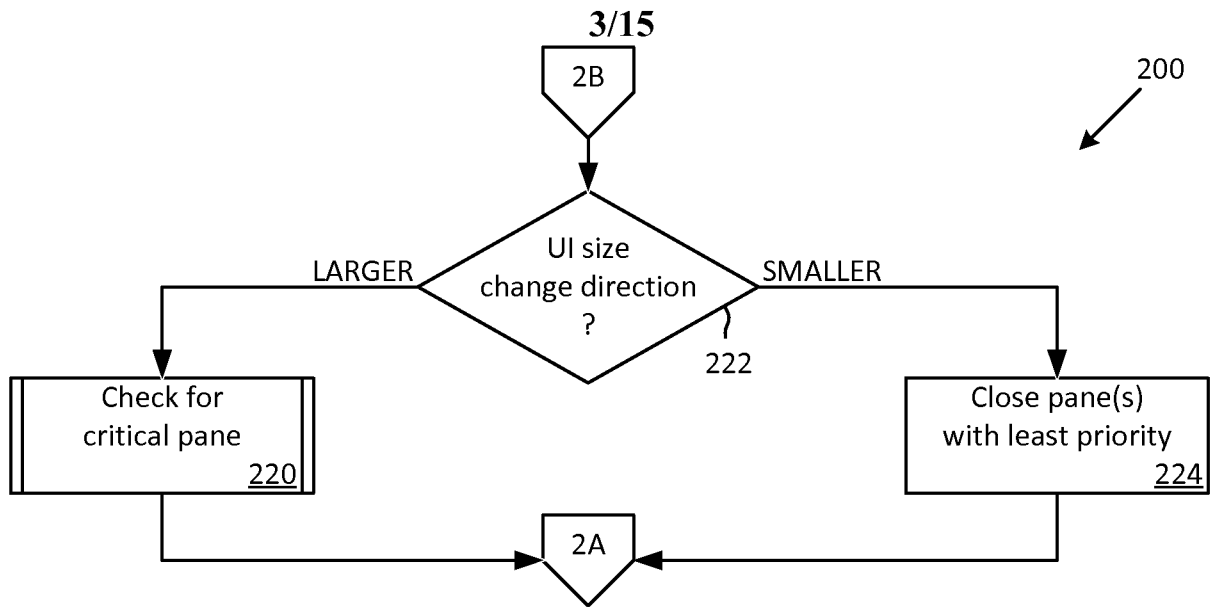


Fig. 2B

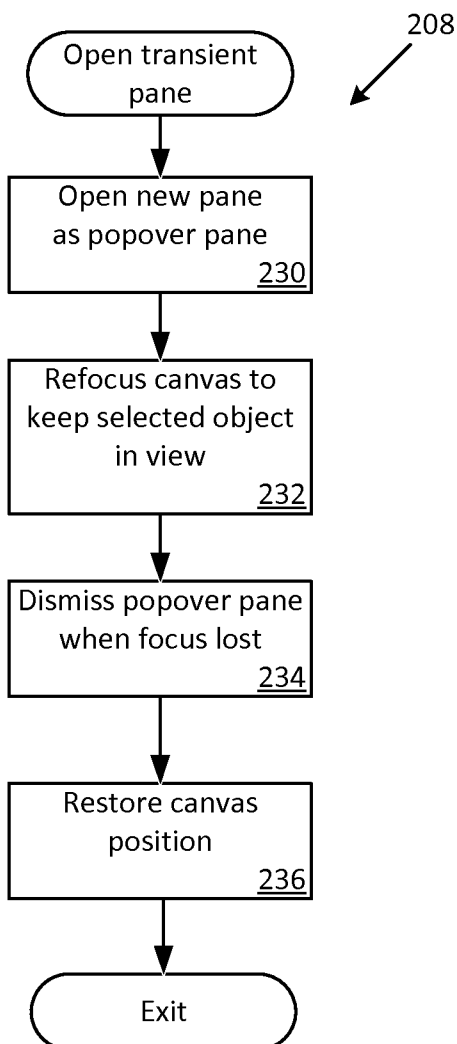


Fig. 2C

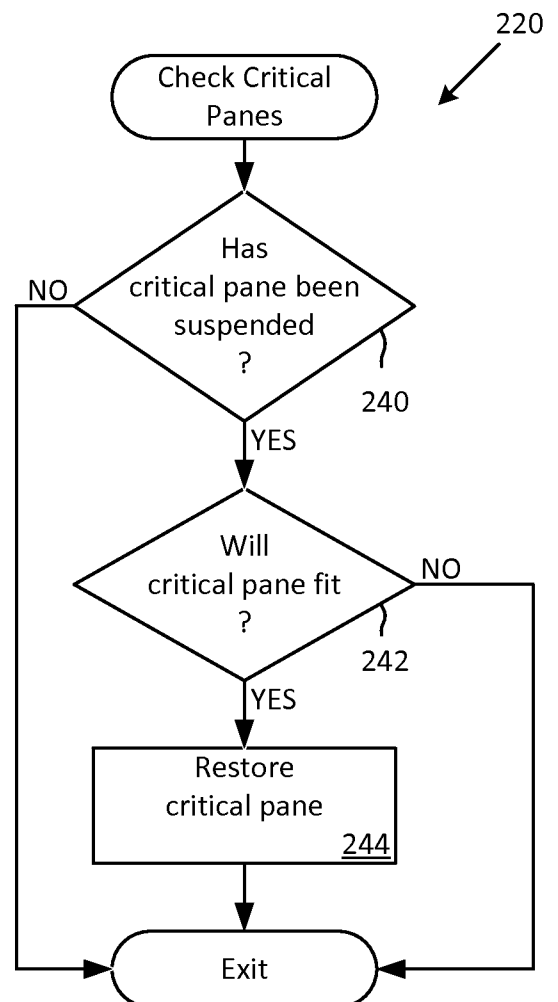


Fig. 2D

4/15

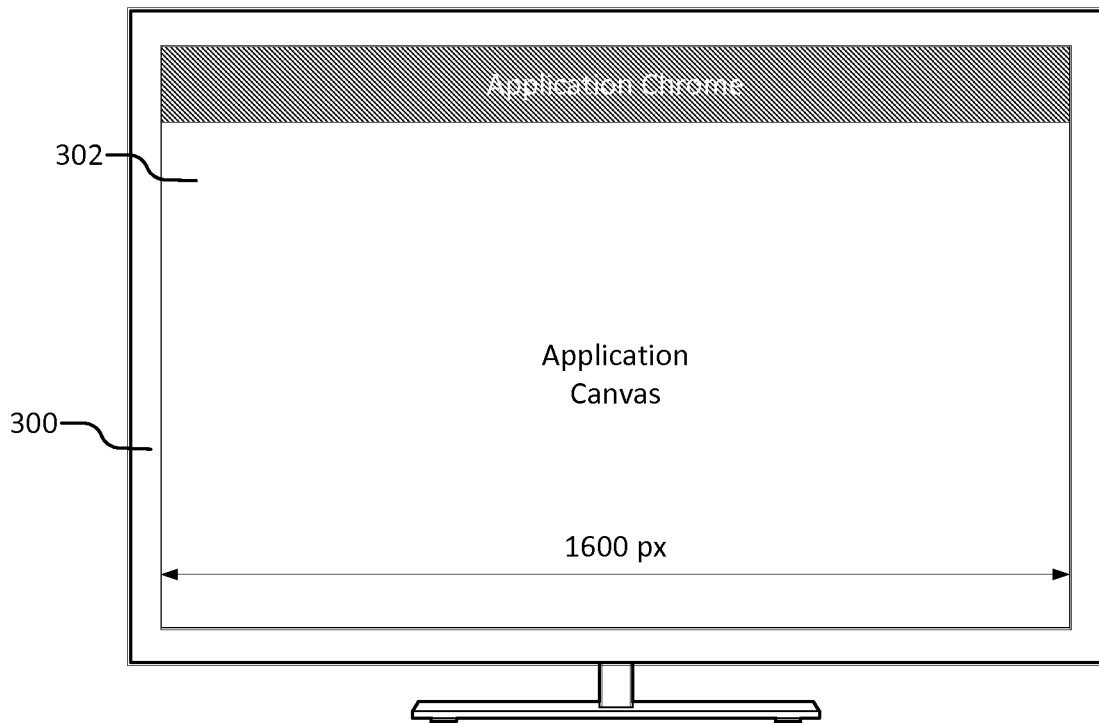


Fig. 3A

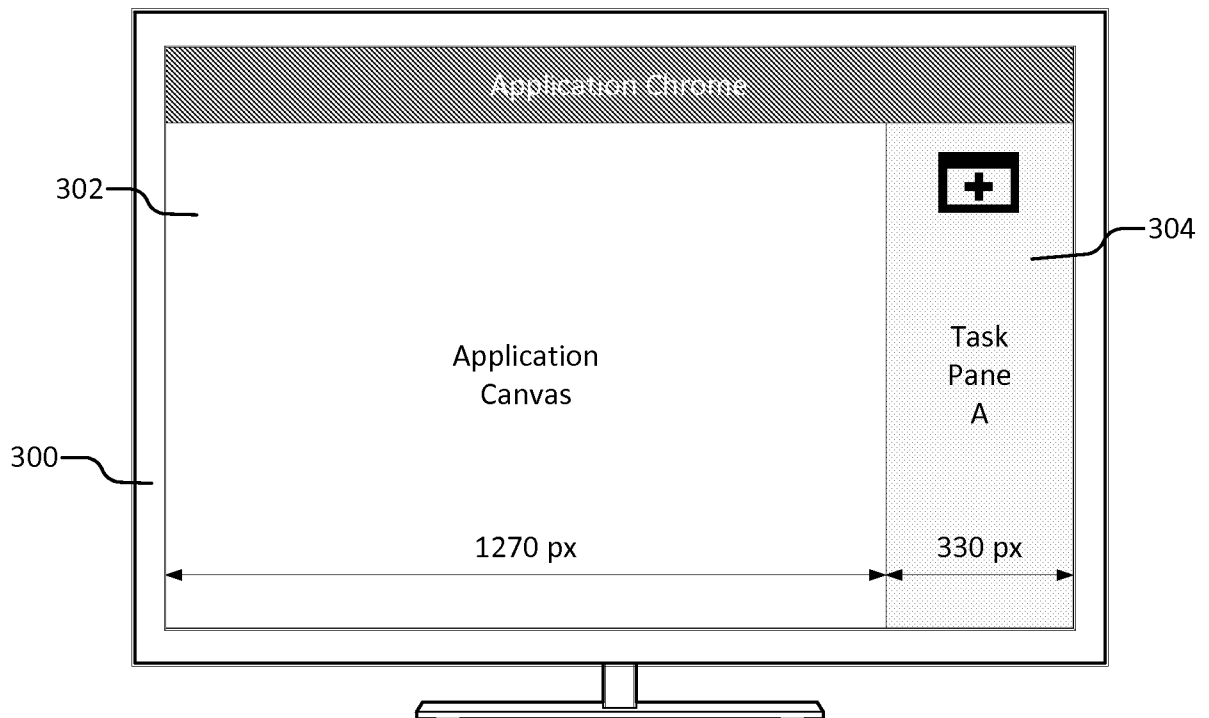


Fig. 3B

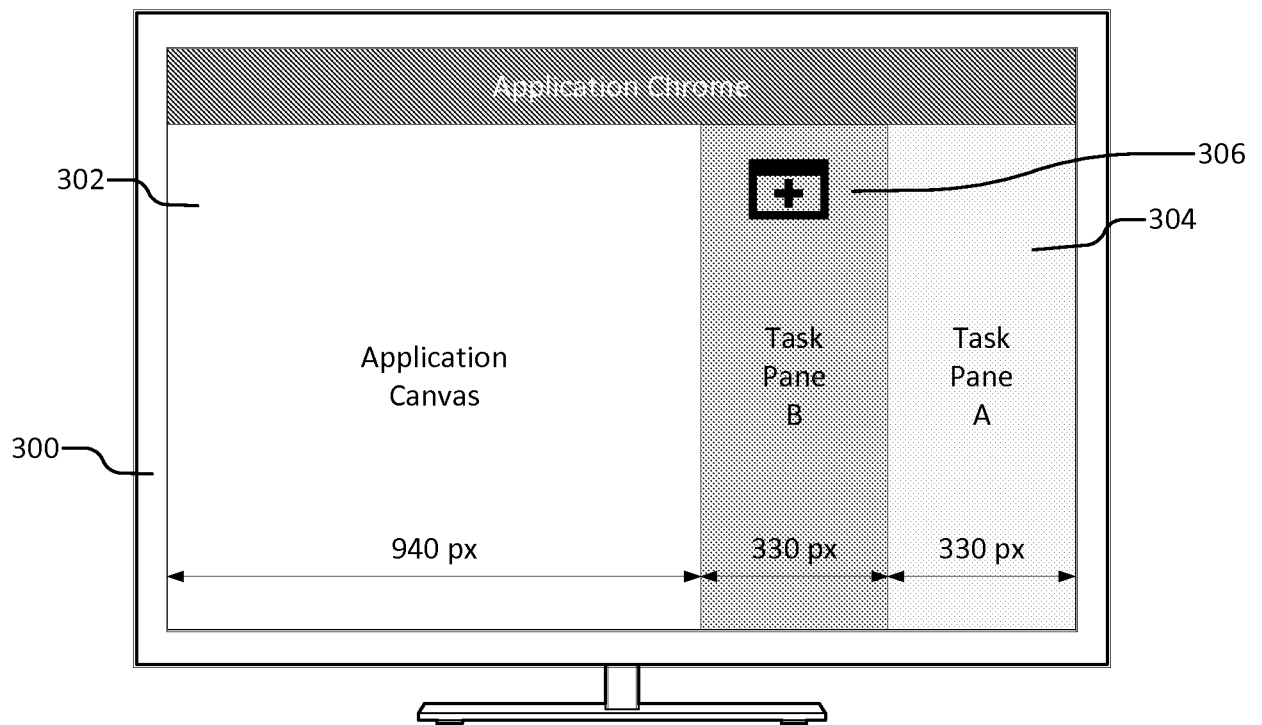


Fig. 3C

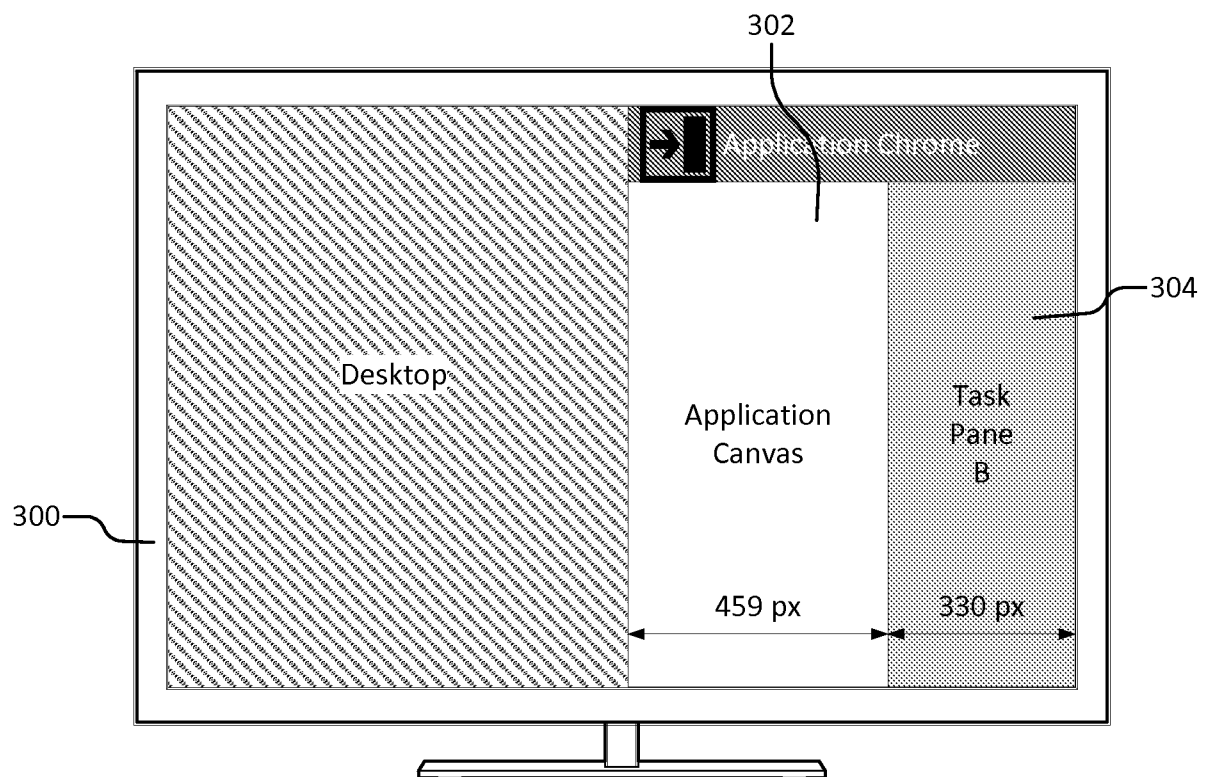


Fig. 3D

6/15

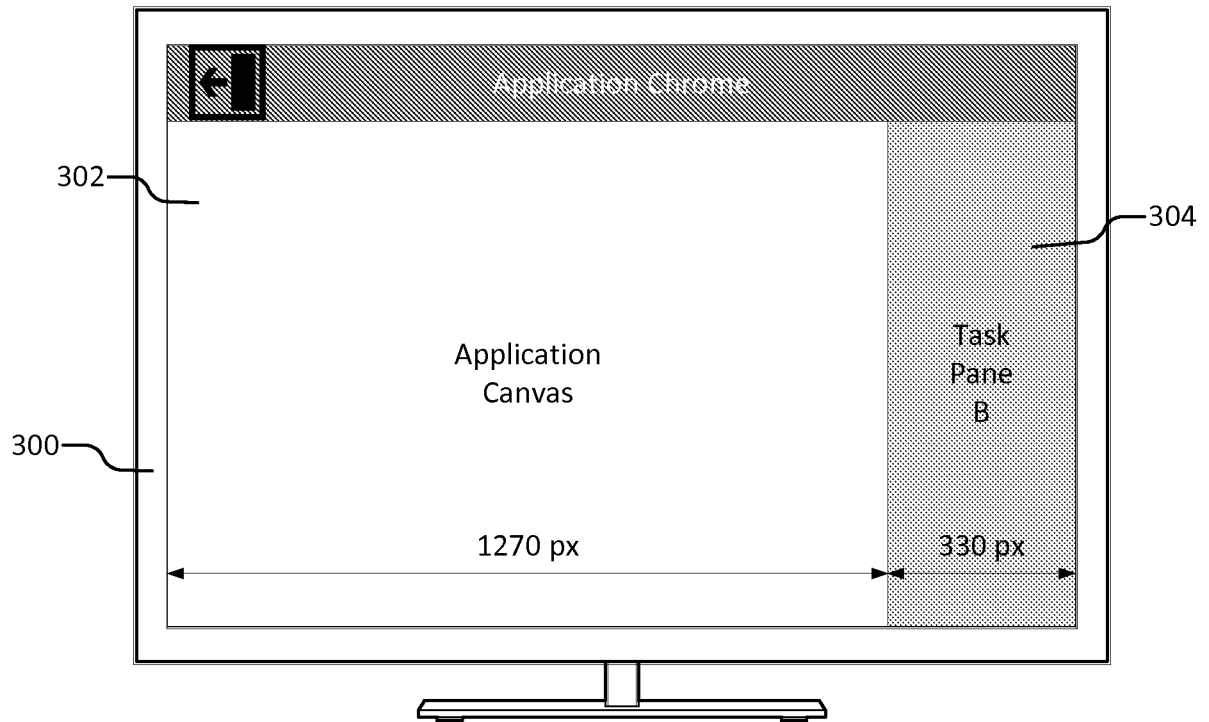


Fig. 3E

7/15

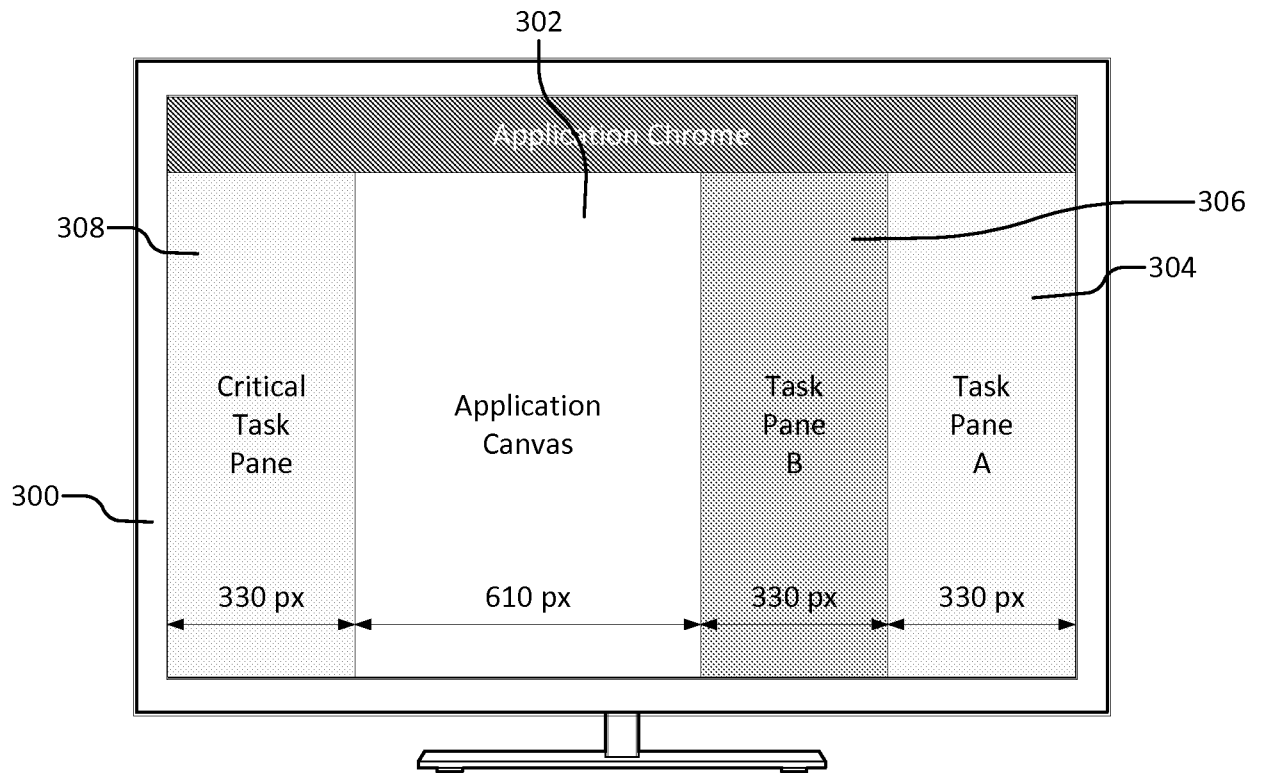


Fig. 3F

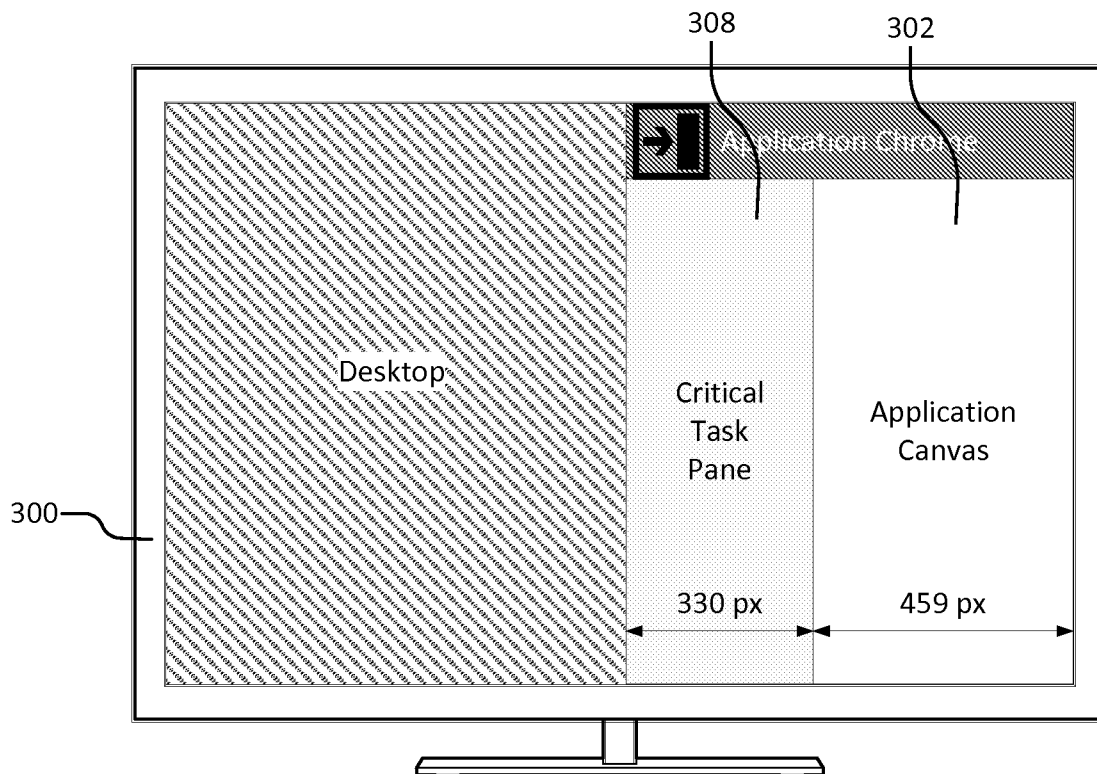


Fig. 3G

8/15

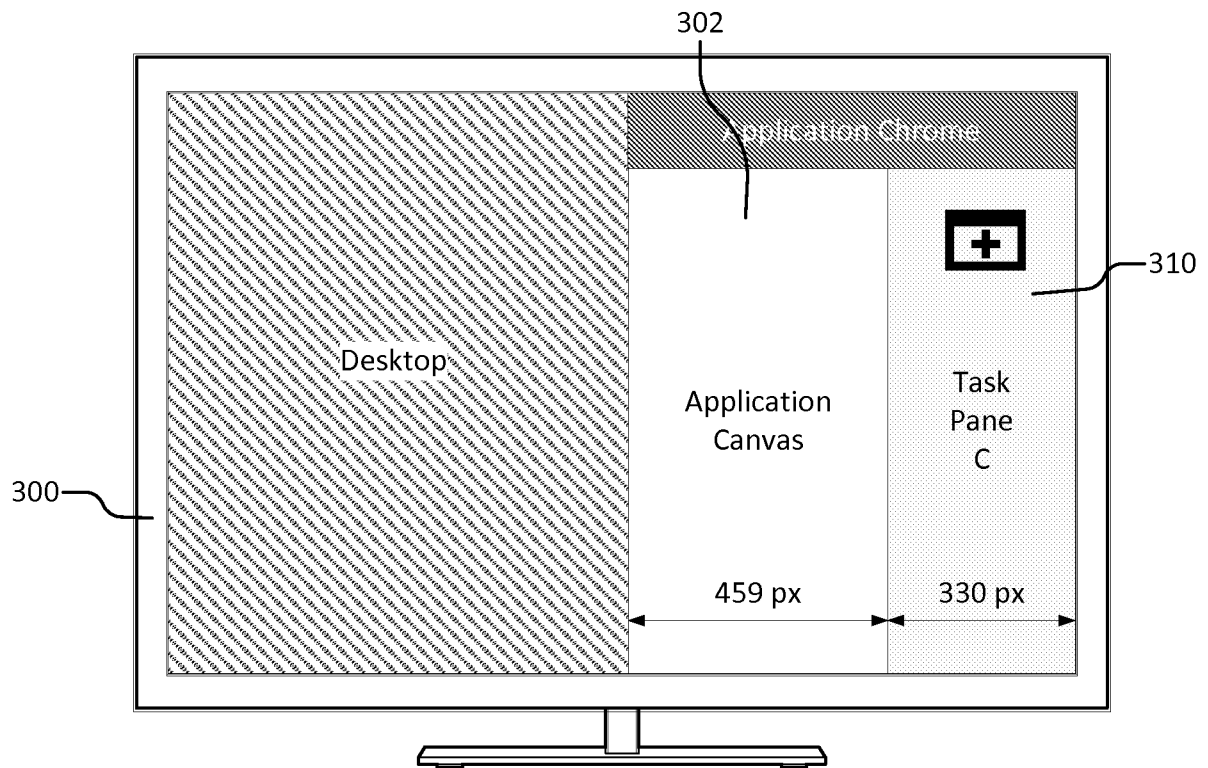


Fig. 3H

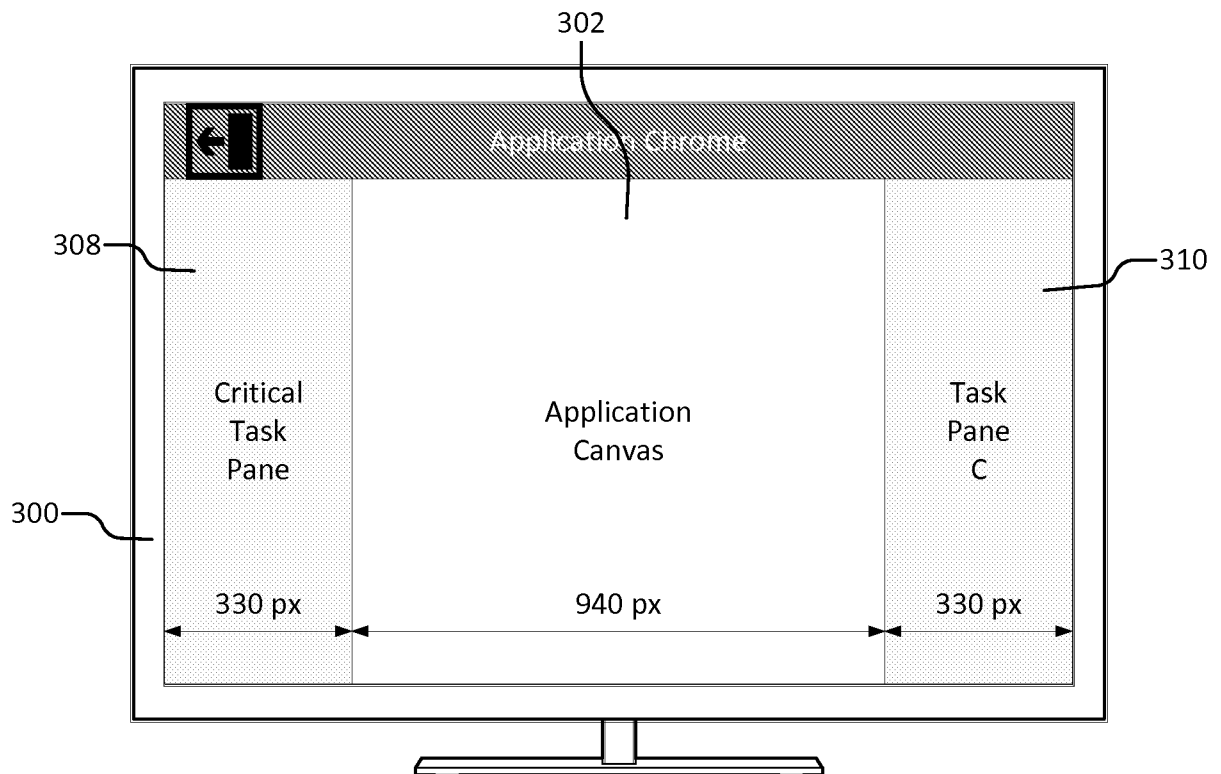


Fig. 3I

9/15

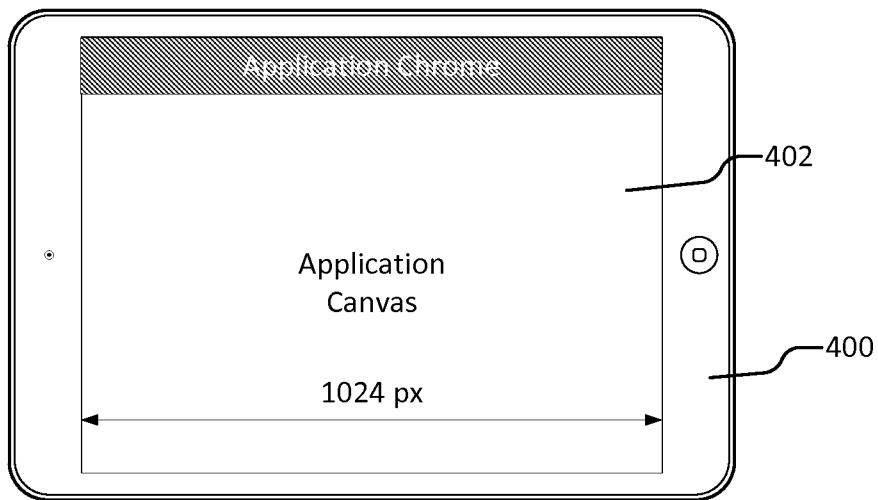


Fig. 4A

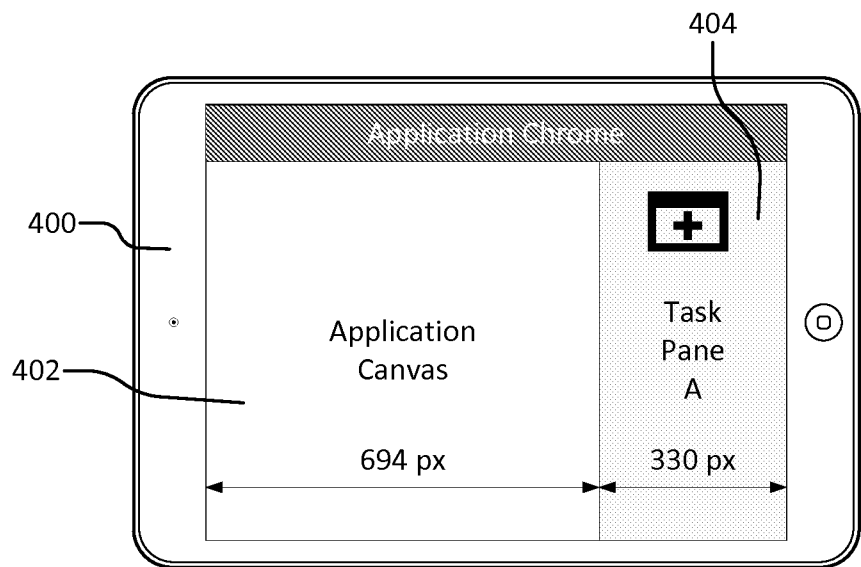


Fig. 4B

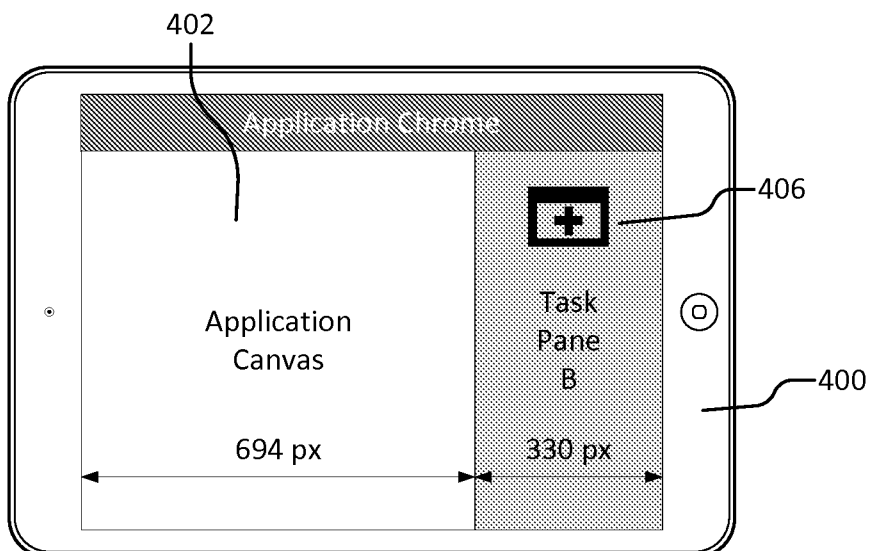


Fig. 4C

10/15

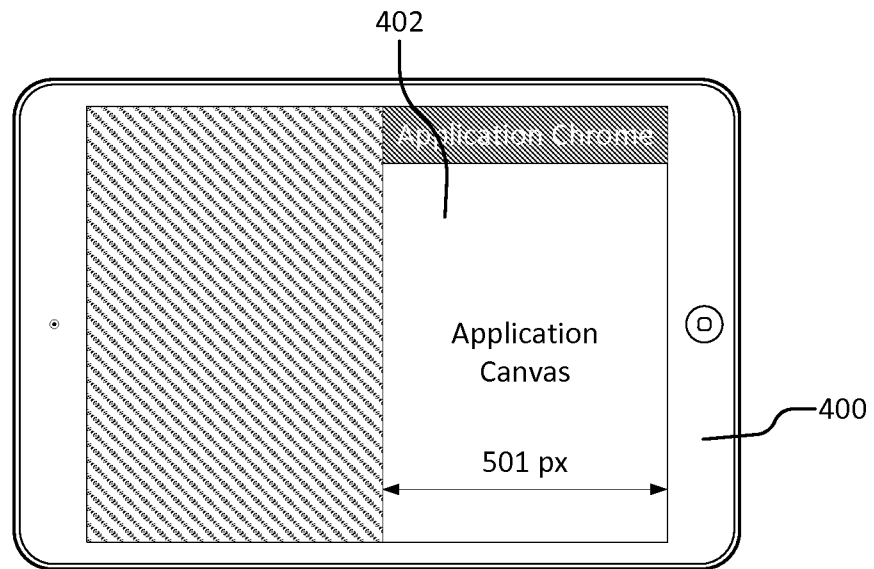


Fig. 4D

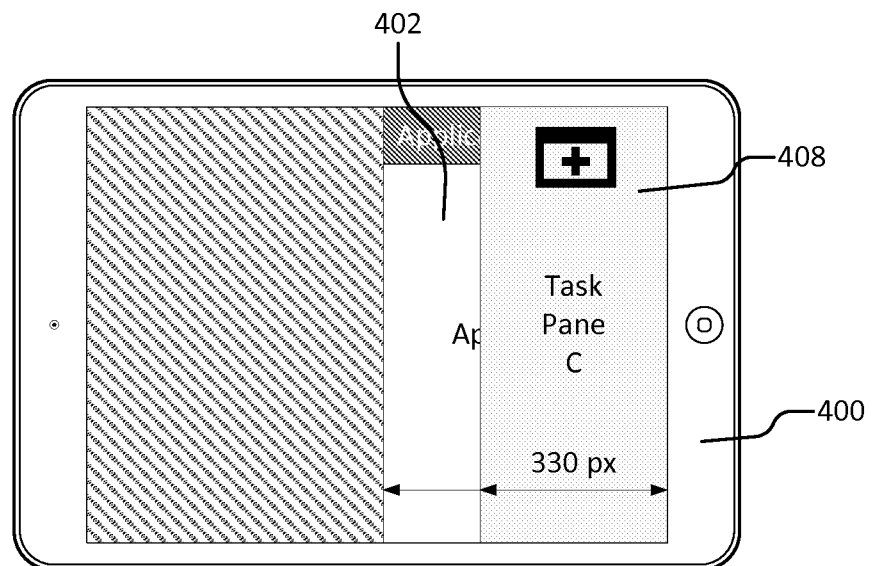


Fig. 4E

11/15

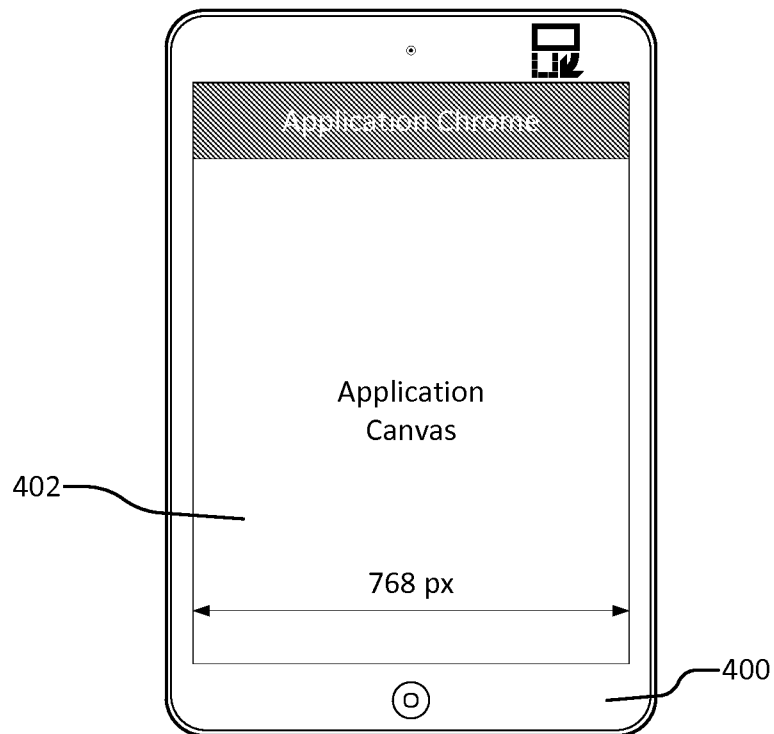


Fig. 4F

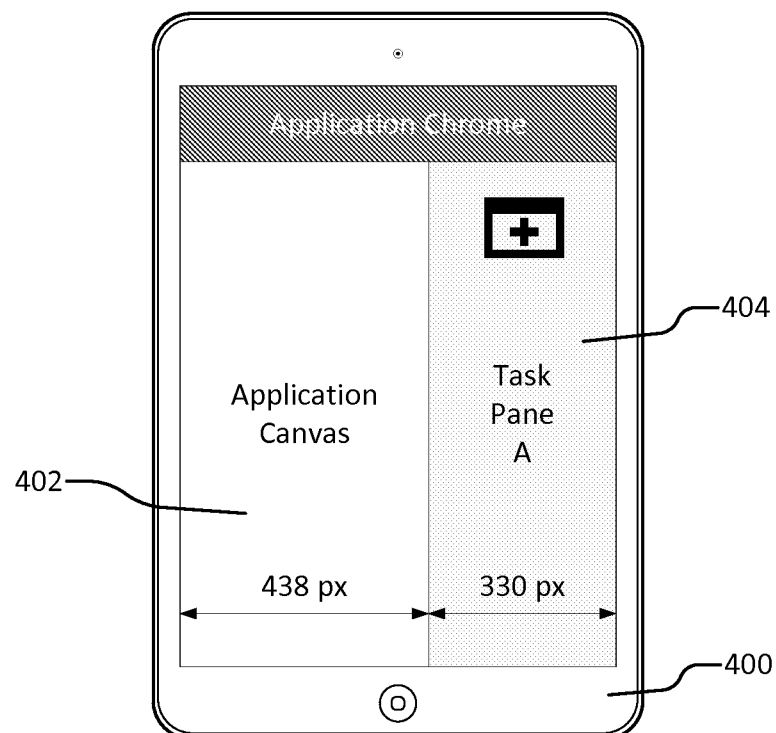


Fig. 4G

12/15

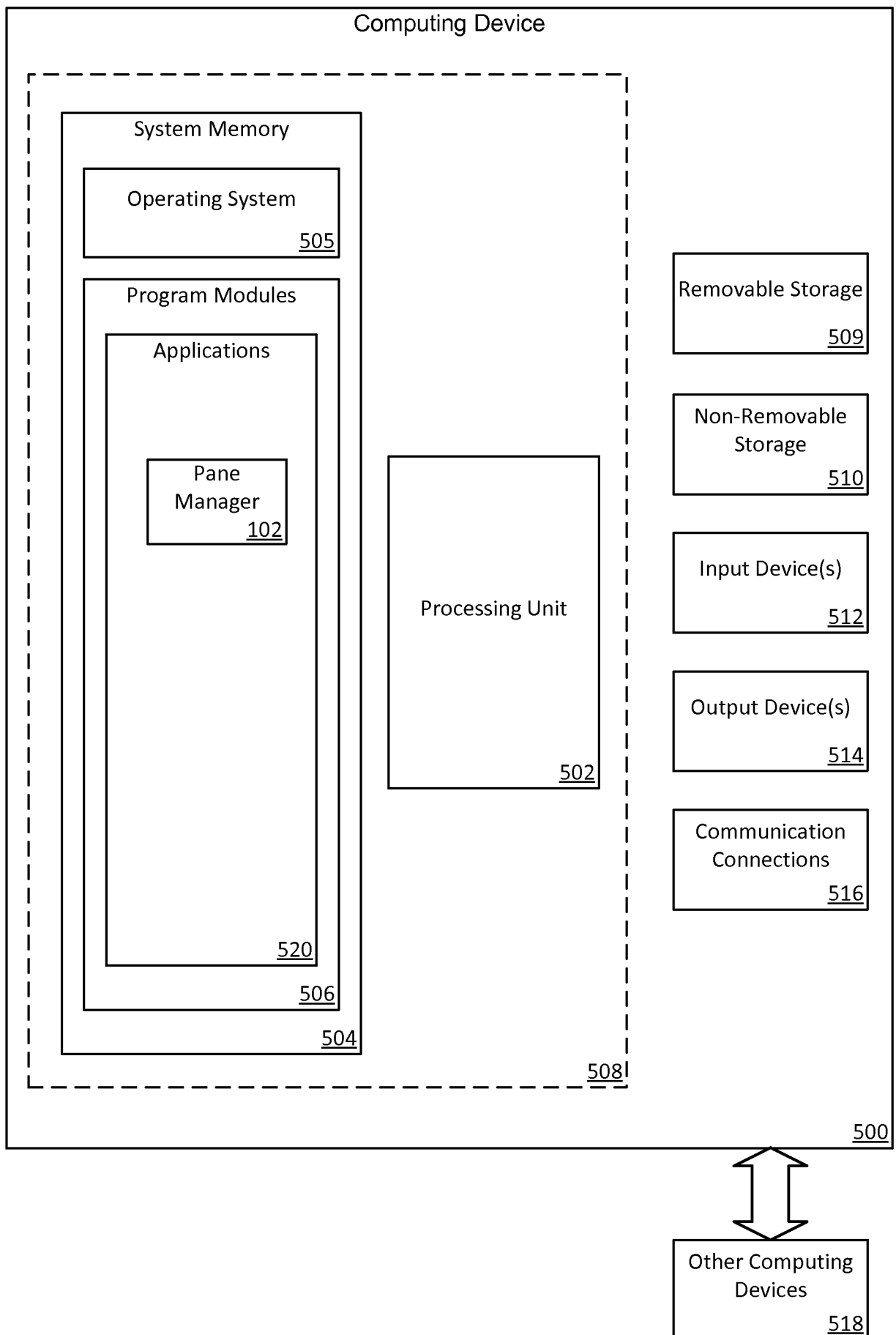


Fig. 5

13/15

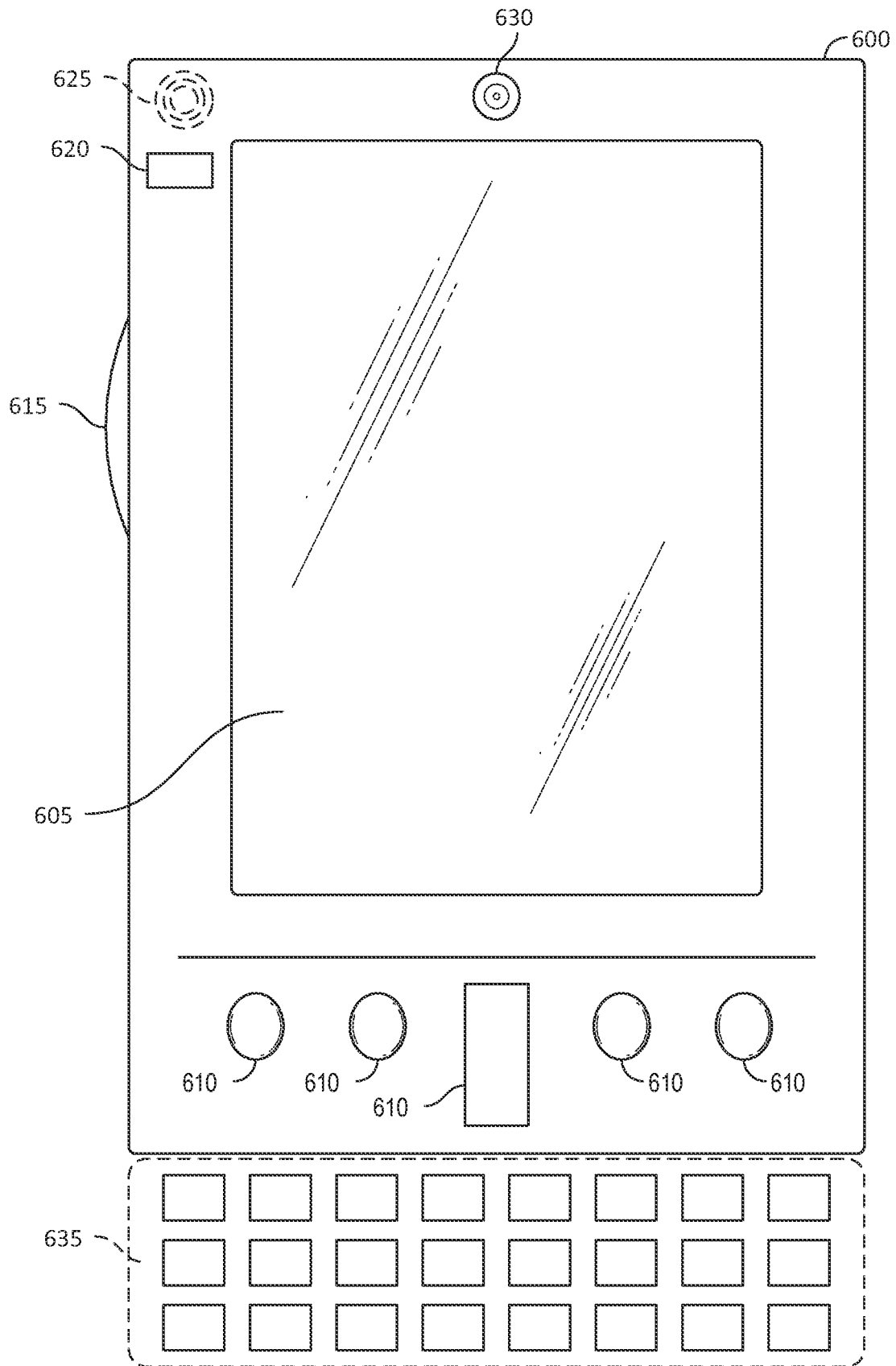


Fig. 6A

14/15

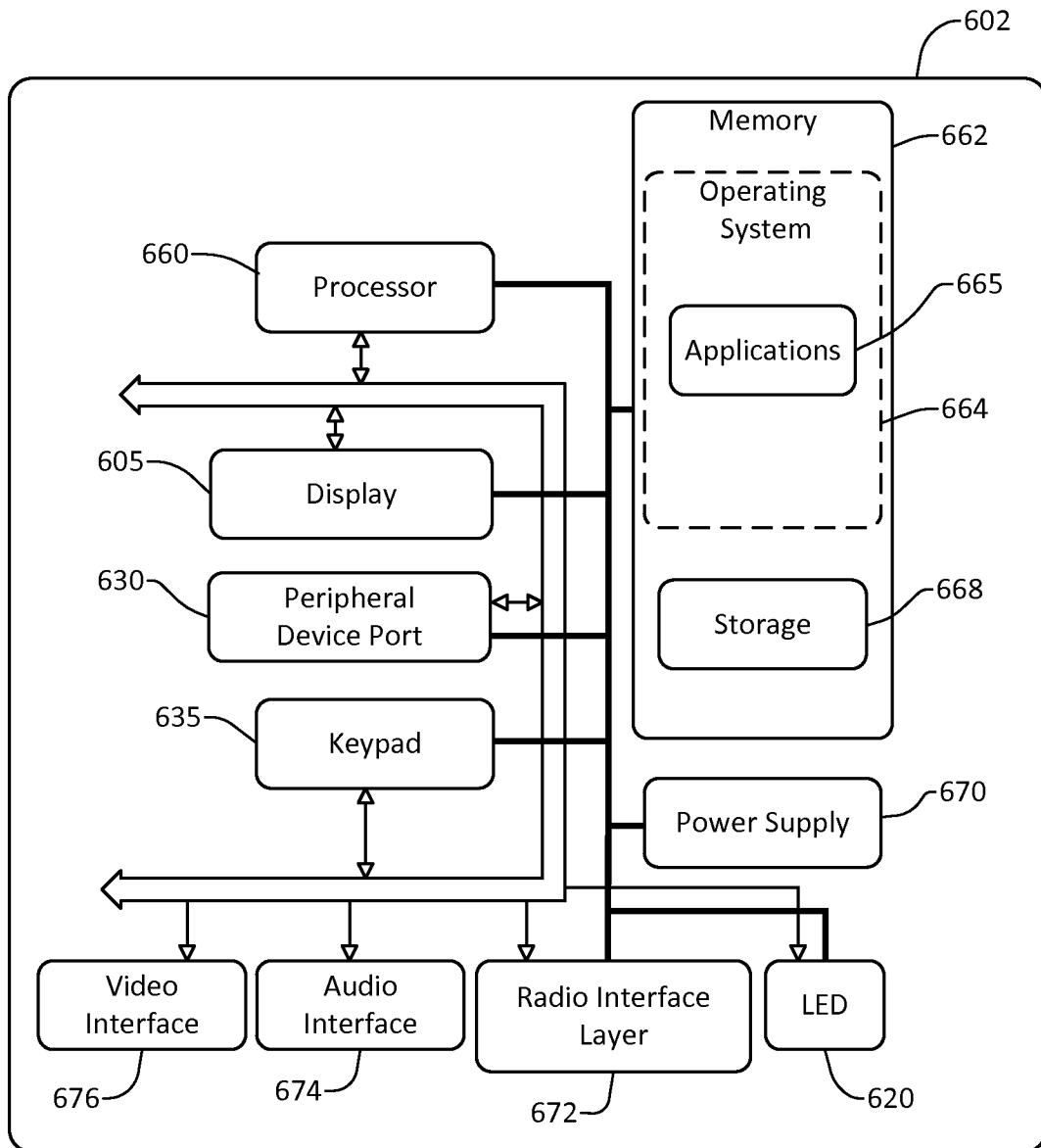


Fig. 6B

15/15

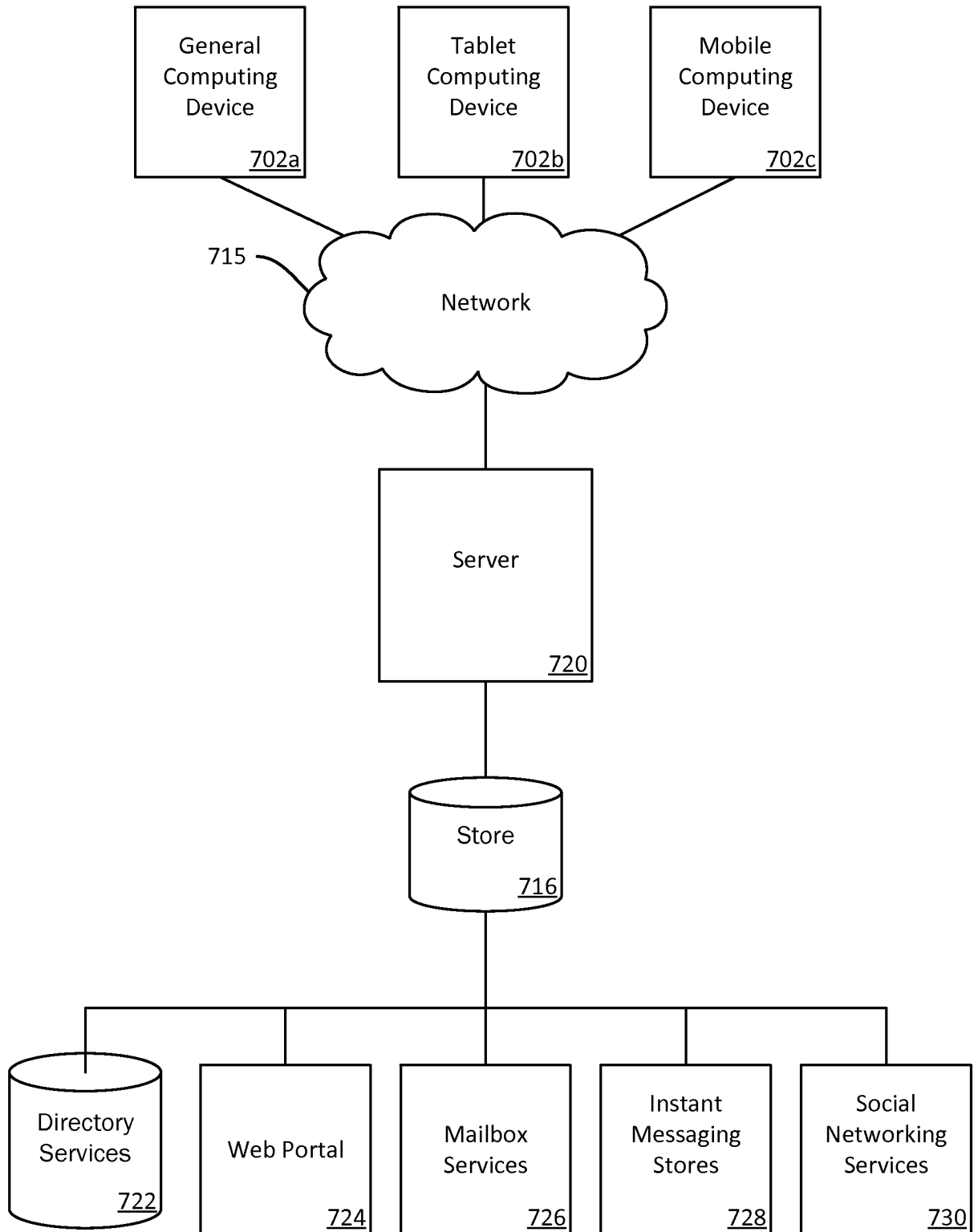


Fig. 7

