

(12) STANDARD PATENT APPLICATION (11) Application No. AU 2016230001 A9
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Virtual assistant continuity

(51) International Patent Classification(s)
G06F 9/48 (2006.01)

(21) Application No: **2016230001** (22) Date of Filing: **2016.03.04**

(87) WIPO No: **WO16/144807**

(30) Priority Data

(31) Number	(32) Date	(33) Country
14/841,455	2015.08.31	US
62/129,928	2015.03.08	US

(43) Publication Date: **2016.09.15**

(48) Corrigenda Journal Date: **2018.06.14**

(71) Applicant(s)
Apple Inc.

(72) Inventor(s)
Sumner, Michael R.;Dibiase, Evan S.;Phipps, Benjamin S.

(74) Agent / Attorney
FPA Patent Attorneys Pty Ltd, ANZ Tower 161 Castlereagh Street, Sydney, NSW, 2000, AU

(56) Related Art
US 2013/0325967 A1



- (51) International Patent Classification:
G06F 9/48 (2006.01)
- (21) International Application Number:
PCT/US2016/021012
- (22) International Filing Date:
4 March 2016 (04.03.2016)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
62/129,928 8 March 2015 (08.03.2015) US
14/841,455 31 August 2015 (31.08.2015) US
- (71) Applicant: APPLE INC. [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).
- (72) Inventors: SUMNER, Michael, R.; c/o Apple Inc., 1 Infinite Loop, MS 39-SIR1, Cupertino, CA 95014 (US). DI-BIASE, Evan, S.; c/o Apple Inc., 1 Infinite Loop, MS 36-PAT, Cupertino, CA 95014 (US). PHIPPS, Benjamin, S.; c/o Apple Inc., 1 Infinite Loop, Mail Stop 39-SIR1, Cupertino, CA 95014 (US).

- (74) Agents: EIDE, Christopher, B. et al.; Morrison & Foerster LLP, 755 Page Mill Road, Palo Alto, CA 94304-1018 (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, 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).

[Continued on next page]

(54) Title: VIRTUAL ASSISTANT CONTINUITY

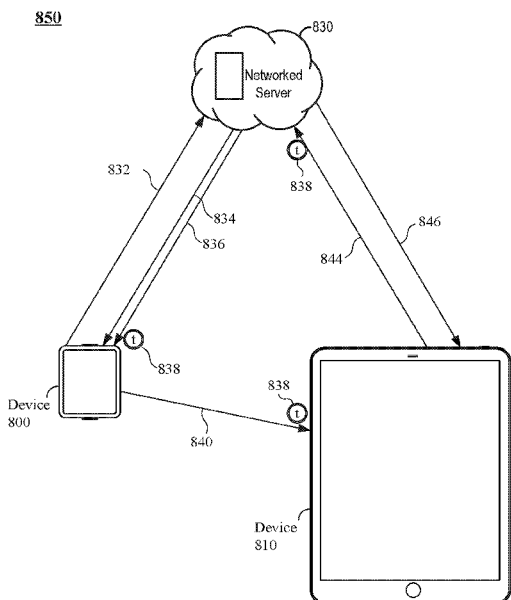


FIG. 8F

(57) Abstract: At a first electronic device with a display and a microphone, receiving audio input via the microphone, wherein the audio input comprises a request; sending data representing the request to a service; receiving a token from the service, wherein the token permits look-up, from the service, of at least one of: the request, and result responsive to the request; and sending the token to a second electronic device external to the first electronic device.

WO 2016/144807 A1

Published:

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

VIRTUAL ASSISTANT CONTINUITY**CROSS-REFERENCE TO RELATED APPLICATIONS**

[0001] This application claims the benefit of priority of U.S. Provisional Patent Application Serial No. 62/129,928, entitled “VIRTUAL ASSISTANT CONTINUITY,” filed March 8, 2015 and U.S. Patent Application Serial No. 14/841,455, entitled “VIRTUAL ASSISTANT CONTINUITY,” filed August 31, 2015. The content of this applications are hereby incorporated by reference in its entirety.

[0002] This application relates to the following co-pending provisional applications: U.S. Patent Application Serial No. 62/005,781, “ACTIVITY CONTINUATION BETWEEN ELECTRONIC DEVICES,” filed May 30, 2014; U.S. Patent Application Serial No. 62/005,793, “COMPANION APPLICATION FOR ACTIVITY COOPERATION,” filed May 30, 2014; U.S. Patent Application Serial No. 62/005,751, “PREDEFINED WIRELESS PAIRING,” filed May 30, 2014; U.S. Patent Application Serial No. 62/005,755, “OPERATING-MODE TRANSITIONS BASED ON ADVERTISING INFORMATION,” filed May 30, 2014; U.S. Patent Application Serial No. 62/006,043, “CONTINUITY,” filed May 3, 2014; U.S. Patent Application Serial No. 62/035,348, “CONTINUITY,” filed August 8, 2014; and U.S. Patent Application Serial No. 12/987,982, titled “INTELLIGENT AUTOMATED ASSISTANT,” filed January 10, 2011. The content of these applications is hereby incorporated by reference.

BACKGROUND**1. Field**

[0003] The present disclosure relates generally to computer user interfaces, and more specifically to techniques for permitting seamless user transitions from use of one device to another.

2. Description of Related Art

[0004] Modern electronic devices can support various software functions. For example, cellular phones, tablet computers, and laptop computers can each support intelligent automated assistants (or virtual assistants) that allow users interaction with the device or services using natural language in spoken and/or text forms. Given the availability of these

services across different devices, a user who owns multiple devices may elect to use one device over another simply based on the device's availability at a moment. For example, while on-the-go, the user may start from a virtual assistant that is available on the closest, most accessible device. But, the chosen device's suitability for providing a response can vary depending on the type of information that is to be provided. When a user inquiry that is made on a highly-accessible (but small) device produces a lengthy output (e.g., such as a complete showing of movies show times), the user may prefer to view the output on a larger computer so that the information can be more easily accessed. Using conventional techniques, the user may activate, for example, a tablet computer and repeat the request.

BRIEF SUMMARY

[0005] It would be helpful if the user could transition from use of one device to another in a seamless and cognitively continuous manner, without needing to repeat a request that has already been made to one of the devices, for example.

[0006] In accordance with some embodiments, a method comprises: at a first electronic device with a display and a microphone: receiving audio input via the microphone, wherein the audio input comprises a request; sending data representing the request to a service; receiving a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and sending the token to a second electronic device external to the first electronic device.

[0007] In accordance with some embodiments, a method comprises: at a first electronic device: receiving input comprising a request; sending data representing the request to a service; receiving a uniform resource locator (URL) from the service identifying a resource; and sending at least part of the URL to a second electronic device external to the first electronic device.

[0008] In accordance with some embodiments, A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first electronic device with a display, a microphone, cause the first electronic device to: receive audio input via the microphone, wherein the audio input comprises a request; send data representing the request to a service; receive a token from the service, wherein the token permits lookup, from the service, of at least one of: the

request, and result responsive to the request; and send the token to a second electronic device external to the first electronic device.

[0009] In accordance with some embodiments, a non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first electronic device with a display, cause the first electronic device to: receive input comprising a request; send data representing the request to a service; receive a uniform resource locator (URL) from the service identifying a resource; and send at least part of the URL to a second electronic device external to the first electronic device.

[0010] In accordance with some embodiments, an electronic device comprises: a display; one or more processors; a memory; a microphone; and one or more programs, wherein the one or more programs are stored in memory and configured to be executed by the one or more processors, the one or more programs including instructions for: receiving audio input via the microphone, wherein the audio input comprises a request; sending data representing the request to a service; receiving a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and sending the token to a second electronic device external to the electronic device.

[0011] In accordance with some embodiments, an electronic device comprises: a display; one or more processors; a memory; and one or more programs, wherein the one or more programs are stored in memory and configured to be executed by the one or more processors, the one or more programs including instructions for: receiving input comprising a request; sending data representing the request to a service; receiving a uniform resource locator (URL) from the service identifying a resource; and sending at least part of the URL to a second electronic device external to the electronic device.

[0012] In accordance with some embodiments, an electronic device comprises: means for receiving audio input via the microphone, wherein the audio input comprises a request; means for sending data representing the request to a service; means for receiving a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and means for sending the token to a second electronic device external to the electronic device.

[0013] In accordance with some embodiments, an electronic device comprises: means for receiving input comprising a request; means for sending data representing the request to a service; means for receiving a uniform resource locator (URL) from the service identifying a resource; and means for sending at least part of the URL to a second electronic device external to the electronic device.

[0014] In accordance with some embodiments, an electronic device comprises: a display; a microphone; and a processing unit coupled to the display, the microphone, and the processing unit configured to: receive audio input via the microphone, wherein the audio input comprises a request; send data representing the request to a service; receive a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and send the token to a second electronic device external to the electronic device.

[0015] In accordance with some embodiments, an electronic device comprises: a display; and a processing unit coupled to the display, the processing unit configured to: receive input comprising a request; send data representing the request to a service; receive a uniform resource locator (URL) from the service identifying a resource; and send at least part of the URL to a second electronic device external to the electronic device.

BRIEF DESCRIPTION OF THE FIGURES

[0016] For a better understanding of the various described embodiments, reference should be made to the Description of Embodiments below, in conjunction with the following drawings in which like reference numerals refer to corresponding parts throughout the figures.

[0017] **FIG. 1** illustrates a system and environment for implementing a digital assistant in accordance with some embodiments.

[0018] **FIG. 2A** is a block diagram illustrating a portable multifunction device implementing the client-side portion of a digital assistant in accordance with some embodiments.

[0019] **FIG. 2B** is a block diagram illustrating exemplary components for event handling in accordance with some embodiments.

[0020] **FIG. 3** illustrates a portable multifunction device implementing the client-side portion of a digital assistant in accordance with some embodiments.

[0021] **FIG. 4** is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments.

[0022] **FIG. 5A** illustrates an exemplary user interface for a menu of applications on a portable multifunction device in accordance with some embodiments.

[0023] **FIG. 5B** illustrates an exemplary user interface for a multifunction device with a touch-sensitive surface that is separate from the display in accordance with some embodiments.

[0024] **FIG. 6A** illustrates a personal electronic device in accordance with some embodiments.

[0025] **FIG. 6B** is a block diagram illustrating a personal electronic device in accordance with some embodiments.

[0026] **FIG. 7A** illustrates a digital assistant system or a server portion thereof in accordance with some embodiments.

[0027] **FIG. 7B** illustrates the functions of the digital assistant shown in FIG. 7A in accordance with some embodiments.

[0028] **FIG. 7C** illustrates a portion of an ontology in accordance with some embodiments.

[0029] **FIGS. 8A-8F** illustrate exemplary user interfaces for transitioning between electronic devices in accordance with some embodiments.

[0030] **FIGS. 9A-9F** illustrate exemplary user interfaces for transitioning between electronic devices in accordance with some embodiments.

[0031] **FIGS. 10A and 10B** are flow diagrams illustrating a process for transitioning between electronic devices in accordance with some embodiments.

[0032] **FIGS. 11A and 11B** are flow diagrams illustrating a process for transitioning between electronic devices in accordance with some embodiments.

[0033] FIG. 12 is a functional block diagram of an electronic device in accordance with some embodiments.

[0034] FIG. 13 is a functional block diagram of an electronic device in accordance with some embodiments.

DESCRIPTION OF EMBODIMENTS

[0035] In the following description of the disclosure and embodiments, reference is made to the accompanying drawings in which it is shown by way of illustration specific embodiments that can be practiced. It is to be understood that other embodiments and examples can be practiced and changes can be made without departing from the scope of the disclosure.

[0036] It is desirable for electronic devices to permit user transitions from the use of one device to another while maintaining an overall sense of continuity. For example, if a user is using one application on a first device, it would be helpful for the second device—to which the user is transitioning—to automatically launch the same application, and optionally to show the same content, so that the user may continue without loss of progress. Such techniques can reduce processor and battery power otherwise wasted on redundant user inputs. Such techniques also can reduce the cognitive burden on a user who switches between uses of multiple computing devices, thereby enhancing productivity.

[0037] Below, FIGS. 1, 2A, 7A-7B, 12, and 13 describe exemplary computing devices that permit efficient transitioning between devices. Figures 8A-8F and 9A-9F illustrate exemplary techniques and user interfaces for transitioning between computing devices. The user interfaces in the figures are also used to illustrate the processes described below, including the processes in FIGS. 10A-10B and 11A-11B.

[0038] Although the following description uses terms “first,” “second,” etc. to describe various elements, these elements should not be limited by the terms. These terms are only used to distinguish one element from another. For example, a first input could be termed a second input, and, similarly, a second input could be termed a first input, without departing from the scope of the various described examples. The first input and the second input can both be outputs and, in some cases, can be separate and different inputs.

[0039] The terminology used in the description of the various described examples herein is for the purpose of describing particular examples only and is not intended to be limiting. As used in the description of the various described examples and the appended claims, the singular forms “a”, “an,” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “includes,” “including,” “comprises,” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0040] The term “if” may be construed to mean “when” or “upon” or “in response to determining” or “in response to detecting,” depending on the context. Similarly, the phrase “if it is determined” or “if [a stated condition or event] is detected” may be construed to mean “upon determining” or “in response to determining” or “upon detecting [the stated condition or event]” or “in response to detecting [the stated condition or event],” depending on the context.

1. System and Environment

[0041] FIG. 1 illustrates a block diagram of system 100 according to various examples. In some examples, system 100 can implement a digital assistant. The terms “digital assistant,” “virtual assistant,” “intelligent automated assistant,” or “automatic digital assistant” can refer to any information processing system that interprets natural language input in spoken and/or textual form to infer user intent, and performs actions based on the inferred user intent. For example, to act on an inferred user intent, the system can perform one or more of the following: identifying a task flow with steps and parameters designed to accomplish the inferred user intent, inputting specific requirements from the inferred user intent into the task flow; executing the task flow by invoking programs, methods, services, APIs, or the like; and generating output responses to the user in an audible (e.g., speech) and/or visual form.

[0042] Specifically, a digital assistant can be capable of accepting a user request at least partially in the form of a natural language command, request, statement, narrative, and/or

inquiry. Typically, the user request can seek either an informational answer or performance of a task by the digital assistant. A satisfactory response to the user request can be a provision of the requested informational answer, a performance of the requested task, or a combination of the two. For example, a user can ask the digital assistant a question, such as “Where am I right now?” Based on the user’s current location, the digital assistant can answer, “You are in Central Park near the west gate.” The user can also request the performance of a task, for example, “Please invite my friends to my girlfriend’s birthday party next week.” In response, the digital assistant can acknowledge the request by saying “Yes, right away,” and then send a suitable calendar invite on behalf of the user to each of the user’s friends listed in the user’s electronic address book. During performance of a requested task, the digital assistant can sometimes interact with the user in a continuous dialogue involving multiple exchanges of information over an extended period of time. There are numerous other ways of interacting with a digital assistant to request information or performance of various tasks. In addition to providing verbal responses and taking programmed actions, the digital assistant can also provide responses in other visual or audio forms, e.g., as text, alerts, music, videos, animations, etc.

[0043] As shown in FIG. 1, in some examples, a digital assistant can be implemented according to a client-server model. The digital assistant can include client-side portion 102 (hereafter “DA client 102”) executed on user device 104 and server-side portion 106 (hereafter “DA server 106”) executed on server system 108. DA client 102 can communicate with DA server 106 through one or more networks 110. DA client 102 can provide client-side functionalities such as user-facing input and output processing and communication with DA server 106. DA server 106 can provide server-side functionalities for any number of DA clients 102 each residing on a respective user device 104.

[0044] In some examples, DA server 106 can include client-facing I/O interface 112, one or more processing modules 114, data and models 116, and I/O interface to external services 118. The client-facing I/O interface 112 can facilitate the client-facing input and output processing for DA server 106. One or more processing modules 114 can utilize data and models 116 to process speech input and determine the user’s intent based on natural language input. Further, one or more processing modules 114 perform task execution based on inferred user intent. In some examples, DA server 106 can communicate with external

services 120 through network(s) 110 for task completion or information acquisition. I/O interface to external services 118 can facilitate such communications.

[0045] User device 104 can be any suitable electronic device. For example, user devices can be a portable multifunctional device (e.g., device 200, described below with reference to FIG. 2A), a multifunctional device (e.g., device 400, described below with reference to FIG. 4), or a personal electronic device (e.g., device 600, described below with reference to FIG. 6A-B.) A portable multifunctional device can be, for example, a mobile telephone that also contains other functions, such as PDA and/or music player functions. Specific examples of portable multifunction devices can include the iPhone®, iPod Touch®, and iPad® devices from Apple Inc. of Cupertino, California. Other examples of portable multifunction devices can include, without limitation, laptop or tablet computers. Further, in some examples, user device 104 can be a non-portable multifunctional device. In particular, user device 104 can be a desktop computer, a game console, a television, or a television set-top box. In some examples, user device 104 can include a touch-sensitive surface (e.g., touch screen displays and/or touchpads). Further, user device 104 can optionally include one or more other physical user-interface devices, such as a physical keyboard, a mouse, and/or a joystick. Various examples of electronic devices, such as multifunctional devices, are described below in greater detail.

[0046] Examples of communication network(s) 110 can include local area networks (LAN) and wide area networks (WAN), e.g., the Internet. Communication network(s) 110 can be implemented using any known network protocol, including various wired or wireless protocols, such as, for example, Ethernet, Universal Serial Bus (USB), FIREWIRE, Global System for Mobile Communications (GSM), Enhanced Data GSM Environment (EDGE), code division multiple access (CDMA), time division multiple access (TDMA), Bluetooth, Wi-Fi, voice over Internet Protocol (VoIP), Wi-MAX, or any other suitable communication protocol.

[0047] Server system 108 can be implemented on one or more standalone data processing apparatus or a distributed network of computers. In some examples, server system 108 can also employ various virtual devices and/or services of third-party service providers (e.g., third-party cloud service providers) to provide the underlying computing resources and/or infrastructure resources of server system 108.

[0048] In some examples, user device 104 can communicate with DA server 106 via second user device 122. Second user device 122 can be similar or identical to user device 104. For example, second user device 122 can be similar to any one of devices 200, 400, or 600, described below with reference to FIGS. 2A, 4, and 6A-B, respectively. User device 104 can be configured to communicatively couple to second user device 122 via a direct communication connection, such as Bluetooth, NFC, BTLE, or the like, or via a wired or wireless network, such as a local Wi-Fi network. In some examples, second user device 122 can be configured to act as a proxy between user device 104 and DA server 106. For example, DA client 102 of user device 104 can be configured to transmit information (e.g., a user request received at user device 104) to DA server 106 via second user device 122. DA server 106 can process the information and return relevant data (e.g., data content responsive to the user request) to user device 104 via second user device 122.

[0049] In some examples, user device 104 can be configured to communicate abbreviated requests for data to second user device 122 to reduce the amount of information transmitted from user device 104. Second user device 122 can be configured to determine supplemental information to add to the abbreviated request to generate a complete request to transmit to DA server 106. This system architecture can advantageously allow user device 104 having limited communication capabilities and/or limited battery power (e.g., a watch or a similar compact electronic device) to access services provided by DA server 106 by using second user device 122, having greater communication capabilities and/or battery power (e.g., a mobile phone, laptop computer, tablet computer, or the like), as a proxy to DA server 106. While only two user devices 104 and 122 are shown in FIG. 1, it should be appreciated that system 100 can include any number and type of user devices configured in this proxy configuration to communicate with DA server system 106.

[0050] Although the digital assistant shown in FIG. 1 can include both a client-side portion (e.g., DA client 102) and a server-side portion (e.g., DA server 106), in some examples, the functions of a digital assistant can be implemented as a standalone application installed on a user device. In addition, the divisions of functionalities between the client and server portions of the digital assistant can vary in different implementations. For instance, in some examples, the DA client can be a thin-client that provides only user-facing input and output processing functions, and delegate all other functionalities of the digital assistant to a backend server.

2. Electronic Devices

[0051] Attention is now directed toward embodiments of electronic devices for implementing the client-side portion of a digital assistant. FIG. 2A is a block diagram illustrating portable multifunction device 200 with touch-sensitive display system 212 in accordance with some embodiments. Touch-sensitive display 212 is sometimes called a “touch screen” for convenience and is sometimes known as or called a “touch-sensitive display system.” Device 200 includes memory 202 (which optionally includes one or more computer-readable storage mediums), memory controller 222, one or more processing units (CPUs) 220, peripherals interface 218, RF circuitry 208, audio circuitry 210, speaker 211, microphone 213, input/output (I/O) subsystem 206, other input control devices 216, and external port 224. Device 200 optionally includes one or more optical sensors 264. Device 200 optionally includes one or more contact intensity sensors 265 for detecting intensity of contacts on device 200 (e.g., a touch-sensitive surface such as touch-sensitive display system 212 of device 200). Device 200 optionally includes one or more tactile output generators 267 for generating tactile outputs on device 200 (e.g., generating tactile outputs on a touch-sensitive surface such as touch-sensitive display system 212 of device 200 or touchpad 455 of device 400). These components optionally communicate over one or more communication buses or signal lines 203.

[0051] As used in the specification and claims, the term “intensity” of a contact on a touch-sensitive surface refers to the force or pressure (force per unit area) of a contact (e.g., a finger contact) on the touch-sensitive surface, or to a substitute (proxy) for the force or pressure of a contact on the touch-sensitive surface. The intensity of a contact has a range of values that includes at least four distinct values and more typically includes hundreds of distinct values (e.g., at least 256). Intensity of a contact is, optionally, determined (or measured) using various approaches and various sensors or combinations of sensors. For example, one or more force sensors underneath or adjacent to the touch-sensitive surface are, optionally, used to measure force at various points on the touch-sensitive surface. In some implementations, force measurements from multiple force sensors are combined (e.g., a weighted average) to determine an estimated force of a contact. Similarly, a pressure-sensitive tip of a stylus is, optionally, used to determine a pressure of the stylus on the touch-sensitive surface. Alternatively, the size of the contact area detected on the touch-sensitive surface and/or changes thereto, the capacitance of the touch-sensitive surface proximate to

the contact and/or changes thereto, and/or the resistance of the touch-sensitive surface proximate to the contact and/or changes thereto are, optionally, used as a substitute for the force or pressure of the contact on the touch-sensitive surface. In some implementations, the substitute measurements for contact force or pressure are used directly to determine whether an intensity threshold has been exceeded (e.g., the intensity threshold is described in units corresponding to the substitute measurements). In some implementations, the substitute measurements for contact force or pressure are converted to an estimated force or pressure, and the estimated force or pressure is used to determine whether an intensity threshold has been exceeded (e.g., the intensity threshold is a pressure threshold measured in units of pressure). Using the intensity of a contact as an attribute of a user input allows for user access to additional device functionality that may otherwise not be accessible by the user on a reduced-size device with limited real estate for displaying affordances (e.g., on a touch-sensitive display) and/or receiving user input (e.g., via a touch-sensitive display, a touch-sensitive surface, or a physical/mechanical control such as a knob or a button).

[0052] As used in the specification and claims, the term “tactile output” refers to physical displacement of a device relative to a previous position of the device, physical displacement of a component (e.g., a touch-sensitive surface) of a device relative to another component (e.g., housing) of the device, or displacement of the component relative to a center of mass of the device that will be detected by a user with the user’s sense of touch. For example, in situations where the device or the component of the device is in contact with a surface of a user that is sensitive to touch (e.g., a finger, palm, or other part of a user’s hand), the tactile output generated by the physical displacement will be interpreted by the user as a tactile sensation corresponding to a perceived change in physical characteristics of the device or the component of the device. For example, movement of a touch-sensitive surface (e.g., a touch-sensitive display or trackpad) is, optionally, interpreted by the user as a “down click” or “up click” of a physical actuator button. In some cases, a user will feel a tactile sensation such as an “down click” or “up click” even when there is no movement of a physical actuator button associated with the touch-sensitive surface that is physically pressed (e.g., displaced) by the user’s movements. As another example, movement of the touch-sensitive surface is, optionally, interpreted or sensed by the user as “roughness” of the touch-sensitive surface, even when there is no change in smoothness of the touch-sensitive surface. While such interpretations of touch by a user will be subject to the individualized sensory perceptions of the user, there are many sensory perceptions of touch that are common to a large majority of

users. Thus, when a tactile output is described as corresponding to a particular sensory perception of a user (e.g., an “up click,” a “down click,” “roughness”), unless otherwise stated, the generated tactile output corresponds to physical displacement of the device or a component thereof that will generate the described sensory perception for a typical (or average) user.

[0053] It should be appreciated that device 200 is only one example of a portable multifunction device, and that device 200 optionally has more or fewer components than shown, optionally combines two or more components, or optionally has a different configuration or arrangement of the components. The various components shown in FIG. 2A are implemented in hardware, software, or a combination of both hardware and software, including one or more signal processing and/or application-specific integrated circuits.

[0054] Memory 202 may include one or more computer-readable storage mediums. The computer-readable storage mediums may be tangible and non-transitory. Memory 202 may include high-speed random access memory and may also include non-volatile memory, such as one or more magnetic disk storage devices, flash memory devices, or other non-volatile solid-state memory devices. Memory controller 222 may control access to memory 202 by other components of device 200.

[0055] In some examples, a non-transitory computer-readable storage medium of memory 202 can be used to store instructions (e.g., for performing aspects of methods 1000 and 1100, described below) for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In other examples, the instructions (e.g., for performing aspects of methods 1000 and 1100, described below) can be stored on a non-transitory computer-readable storage medium (not shown) of the server system 108 or can be divided between the non-transitory computer-readable storage medium of memory 202 and the non-transitory computer-readable storage medium of server system 108. In the context of this document, a “non-transitory computer-readable storage medium” can be any medium that can contain or store the program for use by or in connection with the instruction execution system, apparatus, or device.

[0056] Peripherals interface 218 can be used to couple input and output peripherals of the device to CPU 220 and memory 202. The one or more processors 220 run or execute various software programs and/or sets of instructions stored in memory 202 to perform various functions for device 200 and to process data. In some embodiments, peripherals interface 218, CPU 220, and memory controller 222 may be implemented on a single chip, such as chip 204. In some other embodiments, they may be implemented on separate chips.

[0057] RF (radio frequency) circuitry 208 receives and sends RF signals, also called electromagnetic signals. RF circuitry 208 converts electrical signals to/from electromagnetic signals and communicates with communications networks and other communications devices via the electromagnetic signals. RF circuitry 208 optionally includes well-known circuitry for performing these functions, including but not limited to an antenna system, an RF transceiver, one or more amplifiers, a tuner, one or more oscillators, a digital signal processor, a CODEC chipset, a subscriber identity module (SIM) card, memory, and so forth. RF circuitry 208 optionally communicates with networks, such as the Internet, also referred to as the World Wide Web (WWW), an intranet and/or a wireless network, such as a cellular telephone network, a wireless local area network (LAN) and/or a metropolitan area network (MAN), and other devices by wireless communication. The RF circuitry 208 optionally includes well-known circuitry for detecting near field communication (NFC) fields, such as by a short-range communication radio. The wireless communication optionally uses any of a plurality of communications standards, protocols, and technologies, including but not limited to Global System for Mobile Communications (GSM), Enhanced Data GSM Environment (EDGE), high-speed downlink packet access (HSDPA), high-speed uplink packet access (HSUPA), Evolution, Data-Only (EV-DO), HSPA, HSPA+, Dual-Cell HSPA (DC-HSPDA), long term evolution (LTE), near field communication (NFC), wideband code division multiple access (W-CDMA), code division multiple access (CDMA), time division multiple access (TDMA), Bluetooth, Bluetooth Low Energy (BTLE), Wireless Fidelity (Wi-Fi) (e.g., IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n, and/or IEEE 802.11ac), voice over Internet Protocol (VoIP), Wi-MAX, a protocol for e mail (e.g., Internet message access protocol (IMAP) and/or post office protocol (POP)), instant messaging (e.g., extensible messaging and presence protocol (XMPP), Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE), Instant Messaging and Presence Service (IMPS)), and/or Short Message Service (SMS), or any other suitable communication

protocol, including communication protocols not yet developed as of the filing date of this document.

[0058] Audio circuitry 210, speaker 211, and microphone 213 provide an audio interface between a user and device 200. Audio circuitry 210 receives audio data from peripherals interface 218, converts the audio data to an electrical signal, and transmits the electrical signal to speaker 211. Speaker 211 converts the electrical signal to human-audible sound waves. Audio circuitry 210 also receives electrical signals converted by microphone 213 from sound waves. Audio circuitry 210 converts the electrical signal to audio data and transmits the audio data to peripherals interface 218 for processing. Audio data may be retrieved from and/or transmitted to memory 202 and/or RF circuitry 208 by peripherals interface 218. In some embodiments, audio circuitry 210 also includes a headset jack (e.g., 312, FIG. 3). The headset jack provides an interface between audio circuitry 210 and removable audio input/output peripherals, such as output-only headphones or a headset with both output (e.g., a headphone for one or both ears) and input (e.g., a microphone).

[0059] I/O subsystem 206 couples input/output peripherals on device 200, such as touch screen 212 and other input control devices 216, to peripherals interface 218. I/O subsystem 206 optionally includes display controller 256, optical sensor controller 258, intensity sensor controller 259, haptic feedback controller 261, and one or more input controllers 260 for other input or control devices. The one or more input controllers 260 receive/send electrical signals from/to other input control devices 216. The other input control devices 216 optionally include physical buttons (e.g., push buttons, rocker buttons, etc.), dials, slider switches, joysticks, click wheels, and so forth. In some alternate embodiments, input controller(s) 260 are, optionally, coupled to any (or none) of the following: a keyboard, an infrared port, a USB port, and a pointer device such as a mouse. The one or more buttons (e.g., 308, FIG. 3) optionally include an up/down button for volume control of speaker 211 and/or microphone 213. The one or more buttons optionally include a push button (e.g., 306, FIG. 3).

[0060] A quick press of the push button may disengage a lock of touch screen 212 or begin a process that uses gestures on the touch screen to unlock the device, as described in U.S. Patent Application 11/322,549, "Unlocking a Device by Performing Gestures on an Unlock Image," filed December 23, 2005, U.S. Pat. No. 7,657,849, which is hereby incorporated by reference in its entirety. A longer press of the push button (e.g., 306) may

turn power to device 200 on or off. The user may be able to customize a functionality of one or more of the buttons. Touch screen 212 is used to implement virtual or soft buttons and one or more soft keyboards.

[0061] Touch-sensitive display 212 provides an input interface and an output interface between the device and a user. Display controller 256 receives and/or sends electrical signals from/to touch screen 212. Touch screen 212 displays visual output to the user. The visual output may include graphics, text, icons, video, and any combination thereof (collectively termed “graphics”). In some embodiments, some or all of the visual output may correspond to user-interface objects.

[0062] Touch screen 212 has a touch-sensitive surface, sensor, or set of sensors that accepts input from the user based on haptic and/or tactile contact. Touch screen 212 and display controller 256 (along with any associated modules and/or sets of instructions in memory 202) detect contact (and any movement or breaking of the contact) on touch screen 212 and convert the detected contact into interaction with user-interface objects (e.g., one or more soft keys, icons, web pages, or images) that are displayed on touch screen 212. In an exemplary embodiment, a point of contact between touch screen 212 and the user corresponds to a finger of the user.

[0063] Touch screen 212 may use LCD (liquid crystal display) technology, LPD (light emitting polymer display) technology, or LED (light emitting diode) technology, although other display technologies may be used in other embodiments. Touch screen 212 and display controller 256 may detect contact and any movement or breaking thereof using any of a plurality of touch sensing technologies now known or later developed, including but not limited to capacitive, resistive, infrared, and surface acoustic wave technologies, as well as other proximity sensor arrays or other elements for determining one or more points of contact with touch screen 212. In an exemplary embodiment, projected mutual capacitance sensing technology is used, such as that found in the iPhone® and iPod Touch® from Apple Inc. of Cupertino, California.

[0064] A touch-sensitive display in some embodiments of touch screen 212 may be analogous to the multi-touch sensitive touchpads described in the following U.S. Patents: 6,323,846 (Westerman et al.), 6,570,557 (Westerman et al.), and/or 6,677,932 (Westerman), and/or U.S. Patent Publication 2002/0015024A1, each of which is hereby incorporated by

reference in its entirety. However, touch screen 212 displays visual output from device 200, whereas touch-sensitive touchpads do not provide visual output.

[0065] A touch-sensitive display in some embodiments of touch screen 212 may be as described in the following applications: (1) U.S. Patent Application No. 11/381,313, “Multipoint Touch Surface Controller,” filed May 2, 2006; (2) U.S. Patent Application No. 10/840,862, “Multipoint Touchscreen,” filed May 6, 2004; (3) U.S. Patent Application No. 10/903,964, “Gestures For Touch Sensitive Input Devices,” filed July 30, 2004; (4) U.S. Patent Application No. 11/048,264, “Gestures For Touch Sensitive Input Devices,” filed January 31, 2005; (5) U.S. Patent Application No. 11/038,590, “Mode-Based Graphical User Interfaces For Touch Sensitive Input Devices,” filed January 18, 2005; (6) U.S. Patent Application No. 11/228,758, “Virtual Input Device Placement On A Touch Screen User Interface,” filed September 16, 2005; (7) U.S. Patent Application No. 11/228,700, “Operation Of A Computer With A Touch Screen Interface,” filed September 16, 2005; (8) U.S. Patent Application No. 11/228,737, “Activating Virtual Keys Of A Touch-Screen Virtual Keyboard,” filed September 16, 2005; and (9) U.S. Patent Application No. 11/367,749, “Multi-Functional Hand-Held Device,” filed March 3, 2006. All of these applications are incorporated by reference herein in their entirety.

[0066] Touch screen 212 may have a video resolution in excess of 100 dpi. In some embodiments, the touch screen has a video resolution of approximately 160 dpi. The user may make contact with touch screen 212 using any suitable object or appendage, such as a stylus, a finger, and so forth. In some embodiments, the user interface is designed to work primarily with finger-based contacts and gestures, which can be less precise than stylus-based input due to the larger area of contact of a finger on the touch screen. In some embodiments, the device translates the rough finger-based input into a precise pointer/cursor position or command for performing the actions desired by the user.

[0067] In some embodiments, in addition to the touch screen, device 200 may include a touchpad (not shown) for activating or deactivating particular functions. In some embodiments, the touchpad is a touch-sensitive area of the device that, unlike the touch screen, does not display visual output. The touchpad may be a touch-sensitive surface that is separate from touch screen 212 or an extension of the touch-sensitive surface formed by the touch screen.

[0068] Device 200 also includes power system 262 for powering the various components. Power system 262 may include a power management system, one or more power sources (e.g., battery, alternating current (AC)), a recharging system, a power failure detection circuit, a power converter or inverter, a power status indicator (e.g., a light-emitting diode (LED)) and any other components associated with the generation, management and distribution of power in portable devices.

[0069] Device 200 may also include one or more optical sensors 264. FIG. 2A shows an optical sensor coupled to optical sensor controller 258 in I/O subsystem 206. Optical sensor 264 may include charge-coupled device (CCD) or complementary metal-oxide semiconductor (CMOS) phototransistors. Optical sensor 264 receives light from the environment, projected through one or more lenses, and converts the light to data representing an image. In conjunction with imaging module 243 (also called a camera module), optical sensor 264 may capture still images or video. In some embodiments, an optical sensor is located on the back of device 200, opposite touch screen display 212 on the front of the device so that the touch screen display may be used as a viewfinder for still and/or video image acquisition. In some embodiments, an optical sensor is located on the front of the device so that the user's image may be obtained for video conferencing while the user views the other video conference participants on the touch screen display. In some embodiments, the position of optical sensor 264 can be changed by the user (e.g., by rotating the lens and the sensor in the device housing) so that a single optical sensor 264 may be used along with the touch screen display for both video conferencing and still and/or video image acquisition.

[0070] Device 200 optionally also includes one or more contact intensity sensors 265. FIG. 2A shows a contact intensity sensor coupled to intensity sensor controller 259 in I/O subsystem 206. Contact intensity sensor 265 optionally includes one or more piezoresistive strain gauges, capacitive force sensors, electric force sensors, piezoelectric force sensors, optical force sensors, capacitive touch-sensitive surfaces, or other intensity sensors (e.g., sensors used to measure the force (or pressure) of a contact on a touch-sensitive surface). Contact intensity sensor 265 receives contact intensity information (e.g., pressure information or a proxy for pressure information) from the environment. In some embodiments, at least one contact intensity sensor is collocated with, or proximate to, a touch-sensitive surface (e.g., touch-sensitive display system 212). In some embodiments, at least one contact

intensity sensor is located on the back of device 200, opposite touch screen display 212, which is located on the front of device 200.

[0071] Device 200 may also include one or more proximity sensors 266. FIG. 2A shows proximity sensor 266 coupled to peripherals interface 218. Alternately, proximity sensor 266 may be coupled to input controller 260 in I/O subsystem 206. Proximity sensor 266 may perform as described in U.S. Patent Application Nos. 11/241,839, “Proximity Detector In Handheld Device”; 11/240,788, “Proximity Detector In Handheld Device”; 11/620,702, “Using Ambient Light Sensor To Augment Proximity Sensor Output”; 11/586,862, “Automated Response To And Sensing Of User Activity In Portable Devices”; and 11/638,251, “Methods And Systems For Automatic Configuration Of Peripherals,” which are hereby incorporated by reference in their entirety. In some embodiments, the proximity sensor turns off and disables touch screen 212 when the multifunction device is placed near the user’s ear (e.g., when the user is making a phone call).

[0072] Device 200 optionally also includes one or more tactile output generators 267. FIG. 2A shows a tactile output generator coupled to haptic feedback controller 261 in I/O subsystem 206. Tactile output generator 267 optionally includes one or more electroacoustic devices such as speakers or other audio components and/or electromechanical devices that convert energy into linear motion such as a motor, solenoid, electroactive polymer, piezoelectric actuator, electrostatic actuator, or other tactile output generating component (e.g., a component that converts electrical signals into tactile outputs on the device). Contact intensity sensor 265 receives tactile feedback generation instructions from haptic feedback module 233 and generates tactile outputs on device 200 that are capable of being sensed by a user of device 200. In some embodiments, at least one tactile output generator is collocated with, or proximate to, a touch-sensitive surface (e.g., touch-sensitive display system 212) and, optionally, generates a tactile output by moving the touch-sensitive surface vertically (e.g., in/out of a surface of device 200) or laterally (e.g., back and forth in the same plane as a surface of device 200). In some embodiments, at least one tactile output generator sensor is located on the back of device 200, opposite touch screen display 212, which is located on the front of device 200.

[0073] Device 200 may also include one or more accelerometers 268. FIG. 2A shows accelerometer 268 coupled to peripherals interface 218. Alternately, accelerometer 268 may be coupled to an input controller 260 in I/O subsystem 206. Accelerometer 268 may perform

as described in U.S. Patent Publication No. 20050190059, “Acceleration-based Theft Detection System for Portable Electronic Devices,” and U.S. Patent Publication No. 20060017692, “Methods And Apparatuses For Operating A Portable Device Based On An Accelerometer,” both of which are incorporated by reference herein in their entirety. In some embodiments, information is displayed on the touch screen display in a portrait view or a landscape view based on an analysis of data received from the one or more accelerometers. Device 200 optionally includes, in addition to accelerometer(s) 268, a magnetometer (not shown) and a GPS (or GLONASS or other global navigation system) receiver (not shown) for obtaining information concerning the location and orientation (e.g., portrait or landscape) of device 200.

[0074] In some embodiments, the software components stored in memory 202 include operating system 226, communication module (or set of instructions) 228, contact/motion module (or set of instructions) 230, graphics module (or set of instructions) 232, text input module (or set of instructions) 234, Global Positioning System (GPS) module (or set of instructions) 235, Digital Assistant Client Module 229, and applications (or sets of instructions) 236. Further, memory 202 can store data and models, such as user data and models 231. Furthermore, in some embodiments, memory 202 (FIG. 2A) or 470 (FIG. 4) stores device/global internal state 257, as shown in FIGS. 2A and 4. Device/global internal state 257 includes one or more of: active application state, indicating which applications, if any, are currently active; display state, indicating what applications, views or other information occupy various regions of touch screen display 212; sensor state, including information obtained from the device’s various sensors and input control devices 216; and location information concerning the device’s location and/or attitude.

[0075] Operating system 226 (e.g., Darwin, RTXC, LINUX, UNIX, OS X, iOS, WINDOWS, or an embedded operating system such as VxWorks) includes various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communication between various hardware and software components.

[0076] Communication module 228 facilitates communication with other devices over one or more external ports 224 and also includes various software components for handling data received by RF circuitry 208 and/or external port 224. External port 224 (e.g., Universal Serial Bus (USB), FIREWIRE, etc.) is adapted for coupling directly to other devices or

indirectly over a network (e.g., the Internet, wireless LAN, etc.). In some embodiments, the external port is a multi-pin (e.g., 30-pin) connector that is the same as, or similar to and/or compatible with, the 30-pin connector used on iPod® (trademark of Apple Inc.) devices.

[0077] Contact/motion module 230 optionally detects contact with touch screen 212 (in conjunction with display controller 256) and other touch-sensitive devices (e.g., a touchpad or physical click wheel). Contact/motion module 230 includes various software components for performing various operations related to detection of contact, such as determining if contact has occurred (e.g., detecting a finger-down event), determining an intensity of the contact (e.g., the force or pressure of the contact or a substitute for the force or pressure of the contact), determining if there is movement of the contact and tracking the movement across the touch-sensitive surface (e.g., detecting one or more finger-dragging events), and determining if the contact has ceased (e.g., detecting a finger-up event or a break in contact). Contact/motion module 230 receives contact data from the touch-sensitive surface. Determining movement of the point of contact, which is represented by a series of contact data, optionally includes determining speed (magnitude), velocity (magnitude and direction), and/or an acceleration (a change in magnitude and/or direction) of the point of contact. These operations are, optionally, applied to single contacts (e.g., one finger contacts) or to multiple simultaneous contacts (e.g., “multitouch”/multiple finger contacts). In some embodiments, contact/motion module 230 and display controller 256 detect contact on a touchpad.

[0078] In some embodiments, contact/motion module 230 uses a set of one or more intensity thresholds to determine whether an operation has been performed by a user (e.g., to determine whether a user has “clicked” on an icon). In some embodiments, at least a subset of the intensity thresholds are determined in accordance with software parameters (e.g., the intensity thresholds are not determined by the activation thresholds of particular physical actuators and can be adjusted without changing the physical hardware of device 200). For example, a mouse “click” threshold of a trackpad or touch screen display can be set to any of a large range of predefined threshold values without changing the trackpad or touch screen display hardware. Additionally, in some implementations, a user of the device is provided with software settings for adjusting one or more of the set of intensity thresholds (e.g., by adjusting individual intensity thresholds and/or by adjusting a plurality of intensity thresholds at once with a system-level click “intensity” parameter).

[0079] Contact/motion module 230 optionally detects a gesture input by a user. Different gestures on the touch-sensitive surface have different contact patterns (e.g., different motions, timings, and/or intensities of detected contacts). Thus, a gesture is, optionally, detected by detecting a particular contact pattern. For example, detecting a finger tap gesture includes detecting a finger-down event followed by detecting a finger-up (liftoff) event at the same position (or substantially the same position) as the finger-down event (e.g., at the position of an icon). As another example, detecting a finger swipe gesture on the touch-sensitive surface includes detecting a finger-down event followed by detecting one or more finger-dragging events, and subsequently followed by detecting a finger-up (liftoff) event.

[0080] Graphics module 232 includes various known software components for rendering and displaying graphics on touch screen 212 or other display, including components for changing the visual impact (e.g., brightness, transparency, saturation, contrast, or other visual property) of graphics that are displayed. As used herein, the term “graphics” includes any object that can be displayed to a user, including, without limitation, text, web pages, icons (such as user-interface objects including soft keys), digital images, videos, animations, and the like.

[0081] In some embodiments, graphics module 232 stores data representing graphics to be used. Each graphic is, optionally, assigned a corresponding code. Graphics module 232 receives, from applications etc., one or more codes specifying graphics to be displayed along with, if necessary, coordinate data and other graphic property data, and then generates screen image data to output to display controller 256.

[0082] Haptic feedback module 233 includes various software components for generating instructions used by tactile output generator(s) 267 to produce tactile outputs at one or more locations on device 200 in response to user interactions with device 200.

[0083] Text input module 234, which may be a component of graphics module 232, provides soft keyboards for entering text in various applications (e.g., contacts 237, e mail 240, IM 241, browser 247, and any other application that needs text input).

[0084] GPS module 235 determines the location of the device and provides this information for use in various applications (e.g., to telephone 238 for use in location-based dialing; to camera 243 as picture/video metadata; and to applications that provide location-

based services such as weather widgets, local yellow page widgets, and map/navigation widgets).

[0085] Digital assistant client module 229 can include various client-side digital assistant instructions to provide the client-side functionalities of the digital assistant. For example, digital assistant client module 229 can be capable of accepting voice input (e.g., speech input), text input, touch input, and/or gestural input through various user interfaces (e.g., microphone 213, accelerometer(s) 268, touch-sensitive display system 212, optical sensor(s) 229, other input control devices 216, etc.) of portable multifunction device 200. Digital assistant client module 229 can also be capable of providing output in audio (e.g., speech output), visual, and/or tactile forms through various output interfaces (e.g., speaker 211, touch-sensitive display system 212, tactile output generator(s) 267, etc.) of portable multifunction device 200. For example, output can be provided as voice, sound, alerts, text messages, menus, graphics, videos, animations, vibrations, and/or combinations of two or more of the above. During operation, digital assistant client module 229 can communicate with DA server 106 using RF circuitry 208.

[0086] User data and models 231 can include various data associated with the user (e.g., user-specific vocabulary data, user preference data, user-specified name pronunciations, data from the user's electronic address book, to-do lists, shopping lists, etc.) to provide the client-side functionalities of the digital assistant. Further, user data and models 231 can include various models (e.g., speech recognition models, statistical language models, natural language processing models, ontology, task flow models, service models, etc.) for processing user input and determining user intent.

[0087] In some examples, digital assistant client module 229 can utilize the various sensors, subsystems, and peripheral devices of portable multifunction device 200 to gather additional information from the surrounding environment of the portable multifunction device 200 to establish a context associated with a user, the current user interaction, and/or the current user input. In some examples, digital assistant client module 229 can provide the contextual information or a subset thereof with the user input to DA server 106 to help infer the user's intent. In some examples, the digital assistant can also use the contextual information to determine how to prepare and deliver outputs to the user. Contextual information can be referred to as context data.

[0088] In some examples, the contextual information that accompanies the user input can include sensor information, e.g., lighting, ambient noise, ambient temperature, images or videos of the surrounding environment, etc. In some examples, the contextual information can also include the physical state of the device, e.g., device orientation, device location, device temperature, power level, speed, acceleration, motion patterns, cellular signals strength, etc. In some examples, information related to the software state of DA server 106, e.g., running processes, installed programs, past and present network activities, background services, error logs, resources usage, etc., and of portable multifunction device 200 can be provided to DA server 106 as contextual information associated with a user input.

[0089] In some examples, the digital assistant client module 229 can selectively provide information (e.g., user data 231) stored on the portable multifunction device 200 in response to requests from DA server 106. In some examples, digital assistant client module 229 can also elicit additional input from the user via a natural language dialogue or other user interfaces upon request by DA server 106. Digital assistant client module 229 can pass the additional input to DA server 106 to help DA server 106 in intent deduction and/or fulfillment of the user's intent expressed in the user request.

[0090] A more detailed description of a digital assistant is described below with reference to FIGS. 7A-C. It should be recognized that digital assistant client module 229 can include any number of the sub-modules of digital assistant module 726 described below.

[0091] Applications 236 may include the following modules (or sets of instructions), or a subset or superset thereof:

- Contacts module 237 (sometimes called an address book or contact list);
- Telephone module 238;
- Video conference module 239;
- E-mail client module 240;
- Instant messaging (IM) module 241;
- Workout support module 242;

- Camera module 243 for still and/or video images;
- Image management module 244;
- Video player module;
- Music player module;
- Browser module 247;
- Calendar module 248;
- Widget modules 249, which may include one or more of: weather widget 249-1, stocks widget 249-2, calculator widget 249-3, alarm clock widget 249-4, dictionary widget 249-5, and other widgets obtained by the user, as well as user-created widgets 249-6;
- Widget creator module 250 for making user-created widgets 249-6;
- Search module 251;
- Video and music player module 252, which merges video player module and music player module;
- Notes module 253;
- Map module 254; and/or
- Online video module 255.

[0092] Examples of other applications 236 that may be stored in memory 202 include other word processing applications, other image editing applications, drawing applications, presentation applications, JAVA-enabled applications, encryption, digital rights management, voice recognition, and voice replication.

[0093] In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, contacts module 237 may be used to manage an address book or contact list (e.g., stored in application internal state 292 of contacts module 237 in memory 202 or memory 470), including: adding name(s) to the

address book; deleting name(s) from the address book; associating telephone number(s), e-mail address(es), physical address(es) or other information with a name; associating an image with a name; categorizing and sorting names; providing telephone numbers or e-mail addresses to initiate and/or facilitate communications by telephone 238, video conference module 239, e-mail 240, or IM 241; and so forth.

[0094] In conjunction with RF circuitry 208, audio circuitry 210, speaker 211, microphone 213, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, telephone module 238 may be used to enter a sequence of characters corresponding to a telephone number, access one or more telephone numbers in contacts module 237, modify a telephone number that has been entered, dial a respective telephone number, conduct a conversation, and disconnect or hang up when the conversation is completed. As noted above, the wireless communication may use any of a plurality of communications standards, protocols, and technologies.

[0095] In conjunction with RF circuitry 208, audio circuitry 210, speaker 211, microphone 213, touch screen 212, display controller 256, optical sensor 264, optical sensor controller 258, contact/motion module 230, graphics module 232, text input module 234, contacts module 237, and telephone module 238, video conference module 239 includes executable instructions to initiate, conduct, and terminate a video conference between a user and one or more other participants in accordance with user instructions.

[0096] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, e-mail client module 240 includes executable instructions to create, send, receive, and manage e-mail in response to user instructions. In conjunction with image management module 244, e-mail client module 240 makes it very easy to create and send e-mails with still or video images taken with camera module 243.

[0097] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, the instant messaging module 241 includes executable instructions to enter a sequence of characters corresponding to an instant message, to modify previously entered characters, to transmit a respective instant message (for example, using a Short Message Service (SMS) or Multimedia Message Service (MMS) protocol for telephony-based instant messages or using

XMPP, SIMPLE, or IMPS for Internet-based instant messages), to receive instant messages, and to view received instant messages. In some embodiments, transmitted and/or received instant messages may include graphics, photos, audio files, video files and/or other attachments as are supported in an MMS and/or an Enhanced Messaging Service (EMS). As used herein, “instant messaging” refers to both telephony-based messages (e.g., messages sent using SMS or MMS) and Internet-based messages (e.g., messages sent using XMPP, SIMPLE, or IMPS).

[0098] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, GPS module 235, map module 254, and music player module, workout support module 242 includes executable instructions to create workouts (e.g., with time, distance, and/or calorie burning goals); communicate with workout sensors (sports devices); receive workout sensor data; calibrate sensors used to monitor a workout; select and play music for a workout; and display, store, and transmit workout data.

[0099] In conjunction with touch screen 212, display controller 256, optical sensor(s) 264, optical sensor controller 258, contact/motion module 230, graphics module 232, and image management module 244, camera module 243 includes executable instructions to capture still images or video (including a video stream) and store them into memory 202, modify characteristics of a still image or video, or delete a still image or video from memory 202.

[0100] In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, and camera module 243, image management module 244 includes executable instructions to arrange, modify (e.g., edit), or otherwise manipulate, label, delete, present (e.g., in a digital slide show or album), and store still and/or video images.

[0101] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, browser module 247 includes executable instructions to browse the Internet in accordance with user instructions, including searching, linking to, receiving, and displaying web pages or portions thereof, as well as attachments and other files linked to web pages.

[0102] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, e-mail client module 240, and browser module 247, calendar module 248 includes executable instructions to create, display, modify, and store calendars and data associated with calendars (e.g., calendar entries, to-do lists, etc.) in accordance with user instructions.

[0103] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, and browser module 247, widget modules 249 are mini-applications that may be downloaded and used by a user (e.g., weather widget 249-1, stocks widget 249-2, calculator widget 249-3, alarm clock widget 249-4, and dictionary widget 249-5) or created by the user (e.g., user-created widget 249-6). In some embodiments, a widget includes an HTML (Hypertext Markup Language) file, a CSS (Cascading Style Sheets) file, and a JavaScript file. In some embodiments, a widget includes an XML (Extensible Markup Language) file and a JavaScript file (e.g., Yahoo! Widgets).

[0104] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, and browser module 247, the widget creator module 250 may be used by a user to create widgets (e.g., turning a user-specified portion of a web page into a widget).

[0105] In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, search module 251 includes executable instructions to search for text, music, sound, image, video, and/or other files in memory 202 that match one or more search criteria (e.g., one or more user-specified search terms) in accordance with user instructions.

[0106] In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, audio circuitry 210, speaker 211, RF circuitry 208, and browser module 247, video and music player module 252 includes executable instructions that allow the user to download and play back recorded music and other sound files stored in one or more file formats, such as MP3 or AAC files, and executable instructions to display, present, or otherwise play back videos (e.g., on touch screen 212 or on an external, connected display via external port 224). In some embodiments, device 200 optionally includes the functionality of an MP3 player, such as an iPod (trademark of Apple Inc.).

[0107] In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, and text input module 234, notes module 253 includes executable instructions to create and manage notes, to-do lists, and the like in accordance with user instructions.

[0108] In conjunction with RF circuitry 208, touch screen 212, display controller 256, contact/motion module 230, graphics module 232, text input module 234, GPS module 235, and browser module 247, map module 254 may be used to receive, display, modify, and store maps and data associated with maps (e.g., driving directions, data on stores and other points of interest at or near a particular location, and other location-based data) in accordance with user instructions.

[0109] In conjunction with touch screen 212, display controller 256, contact/motion module 230, graphics module 232, audio circuitry 210, speaker 211, RF circuitry 208, text input module 234, e-mail client module 240, and browser module 247, online video module 255 includes instructions that allow the user to access, browse, receive (e.g., by streaming and/or download), play back (e.g., on the touch screen or on an external, connected display via external port 224), send an e-mail with a link to a particular online video, and otherwise manage online videos in one or more file formats, such as H.264. In some embodiments, instant messaging module 241, rather than e-mail client module 240, is used to send a link to a particular online video. Additional description of the online video application can be found in U.S. Provisional Patent Application No. 60/936,562, "Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos," filed June 20, 2007, and U.S. Patent Application No. 11/968,067, "Portable Multifunction Device, Method, and Graphical User Interface for Playing Online Videos," filed December 31, 2007, the contents of which are hereby incorporated by reference in their entirety.

[0110] Each of the above-identified modules and applications corresponds to a set of executable instructions for performing one or more functions described above and the methods described in this application (e.g., the computer-implemented methods and other information processing methods described herein). These modules (e.g., sets of instructions) need not be implemented as separate software programs, procedures, or modules, and thus various subsets of these modules may be combined or otherwise rearranged in various embodiments. For example, video player module may be combined with music player module into a single module (e.g., video and music player module 252, FIG. 2A). In some

embodiments, memory 202 may store a subset of the modules and data structures identified above. Furthermore, memory 202 may store additional modules and data structures not described above.

[0111] In some embodiments, device 200 is a device where operation of a predefined set of functions on the device is performed exclusively through a touch screen and/or a touchpad. By using a touch screen and/or a touchpad as the primary input control device for operation of device 200, the number of physical input control devices (such as push buttons, dials, and the like) on device 200 may be reduced.

[0112] The predefined set of functions that are performed exclusively through a touch screen and/or a touchpad optionally include navigation between user interfaces. In some embodiments, the touchpad, when touched by the user, navigates device 200 to a main, home, or root menu from any user interface that is displayed on device 200. In such embodiments, a “menu button” is implemented using a touchpad. In some other embodiments, the menu button is a physical push button or other physical input control device instead of a touchpad.

[0113] FIG. 2B is a block diagram illustrating exemplary components for event handling in accordance with some embodiments. In some embodiments, memory 202 (FIG. 2A) or 470 (FIG. 4) includes event sorter 270 (e.g., in operating system 226) and a respective application 236-1 (e.g., any of the aforementioned applications 237-251, 255, 480-490).

[0114] Event sorter 270 receives event information and determines the application 236-1 and application view 291 of application 236-1 to which to deliver the event information. Event sorter 270 includes event monitor 271 and event dispatcher module 274. In some embodiments, application 236-1 includes application internal state 292, which indicates the current application view(s) displayed on touch-sensitive display 212 when the application is active or executing. In some embodiments, device/global internal state 257 is used by event sorter 270 to determine which application(s) is (are) currently active, and application internal state 292 is used by event sorter 270 to determine application views 291 to which to deliver event information.

[0115] In some embodiments, application internal state 292 includes additional information, such as one or more of: resume information to be used when application 236-1 resumes execution, user interface state information that indicates information being displayed or that is ready for display by application 236-1, a state queue for enabling the user to go

back to a prior state or view of application 236-1, and a redo/undo queue of previous actions taken by the user.

[0116] Event monitor 271 receives event information from peripherals interface 218. Event information includes information about a sub-event (e.g., a user touch on touch-sensitive display 212, as part of a multi-touch gesture). Peripherals interface 218 transmits information it receives from I/O subsystem 206 or a sensor, such as proximity sensor 266, accelerometer(s) 268, and/or microphone 213 (through audio circuitry 210). Information that peripherals interface 218 receives from I/O subsystem 206 includes information from touch-sensitive display 212 or a touch-sensitive surface.

[0117] In some embodiments, event monitor 271 sends requests to the peripherals interface 218 at predetermined intervals. In response, peripherals interface 218 transmits event information. In other embodiments, peripherals interface 218 transmits event information only when there is a significant event (e.g., receiving an input above a predetermined noise threshold and/or for more than a predetermined duration).

[0118] In some embodiments, event sorter 270 also includes a hit view determination module 272 and/or an active event recognizer determination module 273.

[0119] Hit view determination module 272 provides software procedures for determining where a sub-event has taken place within one or more views when touch-sensitive display 212 displays more than one view. Views are made up of controls and other elements that a user can see on the display.

[0120] Another aspect of the user interface associated with an application is a set of views, sometimes herein called application views or user interface windows, in which information is displayed and touch-based gestures occur. The application views (of a respective application) in which a touch is detected may correspond to programmatic levels within a programmatic or view hierarchy of the application. For example, the lowest level view in which a touch is detected may be called the hit view, and the set of events that are recognized as proper inputs may be determined based, at least in part, on the hit view of the initial touch that begins a touch-based gesture.

[0121] Hit view determination module 272 receives information related to sub events of a touch-based gesture. When an application has multiple views organized in a hierarchy, hit

view determination module 272 identifies a hit view as the lowest view in the hierarchy which should handle the sub-event. In most circumstances, the hit view is the lowest level view in which an initiating sub-event occurs (e.g., the first sub-event in the sequence of sub-events that form an event or potential event). Once the hit view is identified by the hit view determination module 272, the hit view typically receives all sub-events related to the same touch or input source for which it was identified as the hit view.

[0122] Active event recognizer determination module 273 determines which view or views within a view hierarchy should receive a particular sequence of sub-events. In some embodiments, active event recognizer determination module 273 determines that only the hit view should receive a particular sequence of sub-events. In other embodiments, active event recognizer determination module 273 determines that all views that include the physical location of a sub-event are actively involved views, and therefore determines that all actively involved views should receive a particular sequence of sub-events. In other embodiments, even if touch sub-events were entirely confined to the area associated with one particular view, views higher in the hierarchy would still remain as actively involved views.

[0123] Event dispatcher module 274 dispatches the event information to an event recognizer (e.g., event recognizer 280). In embodiments including active event recognizer determination module 273, event dispatcher module 274 delivers the event information to an event recognizer determined by active event recognizer determination module 273. In some embodiments, event dispatcher module 274 stores in an event queue the event information, which is retrieved by a respective event receiver 282.

[0124] In some embodiments, operating system 226 includes event sorter 270. Alternatively, application 236-1 includes event sorter 270. In yet other embodiments, event sorter 270 is a stand-alone module, or a part of another module stored in memory 202, such as contact/motion module 230.

[0125] In some embodiments, application 236-1 includes a plurality of event handlers 290 and one or more application views 291, each of which includes instructions for handling touch events that occur within a respective view of the application's user interface. Each application view 291 of the application 236-1 includes one or more event recognizers 280. Typically, a respective application view 291 includes a plurality of event recognizers 280. In other embodiments, one or more of event recognizers 280 are part of a separate module, such

as a user interface kit (not shown) or a higher level object from which application 236-1 inherits methods and other properties. In some embodiments, a respective event handler 290 includes one or more of: data updater 276, object updater 277, GUI updater 278, and/or event data 279 received from event sorter 270. Event handler 290 may utilize or call data updater 276, object updater 277, or GUI updater 278 to update the application internal state 292. Alternatively, one or more of the application views 291 include one or more respective event handlers 290. Also, in some embodiments, one or more of data updater 276, object updater 277, and GUI updater 278 are included in a respective application view 291.

[0126] A respective event recognizer 280 receives event information (e.g., event data 279) from event sorter 270 and identifies an event from the event information. Event recognizer 280 includes event receiver 282 and event comparator 284. In some embodiments, event recognizer 280 also includes at least a subset of: metadata 283, and event delivery instructions 288 (which may include sub-event delivery instructions).

[0127] Event receiver 282 receives event information from event sorter 270. The event information includes information about a sub-event, for example, a touch or a touch movement. Depending on the sub-event, the event information also includes additional information, such as location of the sub-event. When the sub-event concerns motion of a touch, the event information may also include speed and direction of the sub-event. In some embodiments, events include rotation of the device from one orientation to another (e.g., from a portrait orientation to a landscape orientation, or vice versa), and the event information includes corresponding information about the current orientation (also called device attitude) of the device.

[0128] Event comparator 284 compares the event information to predefined event or sub-event definitions and, based on the comparison, determines an event or sub event, or determines or updates the state of an event or sub-event. In some embodiments, event comparator 284 includes event definitions 286. Event definitions 286 contain definitions of events (e.g., predefined sequences of sub-events), for example, event 1 (287-1), event 2 (287-2), and others. In some embodiments, sub-events in an event (287) include, for example, touch begin, touch end, touch movement, touch cancellation, and multiple touching. In one example, the definition for event 1 (287-1) is a double tap on a displayed object. The double tap, for example, comprises a first touch (touch begin) on the displayed object for a predetermined phase, a first liftoff (touch end) for a predetermined phase, a second touch

(touch begin) on the displayed object for a predetermined phase, and a second liftoff (touch end) for a predetermined phase. In another example, the definition for event 2 (287-2) is a dragging on a displayed object. The dragging, for example, comprises a touch (or contact) on the displayed object for a predetermined phase, a movement of the touch across touch-sensitive display 212, and liftoff of the touch (touch end). In some embodiments, the event also includes information for one or more associated event handlers 290.

[0129] In some embodiments, event definition 287 includes a definition of an event for a respective user-interface object. In some embodiments, event comparator 284 performs a hit test to determine which user-interface object is associated with a sub-event. For example, in an application view in which three user-interface objects are displayed on touch-sensitive display 212, when a touch is detected on touch-sensitive display 212, event comparator 284 performs a hit test to determine which of the three user-interface objects is associated with the touch (sub-event). If each displayed object is associated with a respective event handler 290, the event comparator uses the result of the hit test to determine which event handler 290 should be activated. For example, event comparator 284 selects an event handler associated with the sub-event and the object triggering the hit test.

[0130] In some embodiments, the definition for a respective event (287) also includes delayed actions that delay delivery of the event information until after it has been determined whether the sequence of sub-events does or does not correspond to the event recognizer's event type.

[0131] When a respective event recognizer 280 determines that the series of sub-events do not match any of the events in event definitions 286, the respective event recognizer 280 enters an event impossible, event failed, or event ended state, after which it disregards subsequent sub-events of the touch-based gesture. In this situation, other event recognizers, if any, that remain active for the hit view continue to track and process sub-events of an ongoing touch-based gesture.

[0132] In some embodiments, a respective event recognizer 280 includes metadata 283 with configurable properties, flags, and/or lists that indicate how the event delivery system should perform sub-event delivery to actively involved event recognizers. In some embodiments, metadata 283 includes configurable properties, flags, and/or lists that indicate how event recognizers may interact, or are enabled to interact, with one another. In some

embodiments, metadata 283 includes configurable properties, flags, and/or lists that indicate whether sub-events are delivered to varying levels in the view or programmatic hierarchy.

[0133] In some embodiments, a respective event recognizer 280 activates event handler 290 associated with an event when one or more particular sub-events of an event are recognized. In some embodiments, a respective event recognizer 280 delivers event information associated with the event to event handler 290. Activating an event handler 290 is distinct from sending (and deferred sending) sub-events to a respective hit view. In some embodiments, event recognizer 280 throws a flag associated with the recognized event, and event handler 290 associated with the flag catches the flag and performs a predefined process.

[0134] In some embodiments, event delivery instructions 288 include sub-event delivery instructions that deliver event information about a sub-event without activating an event handler. Instead, the sub-event delivery instructions deliver event information to event handlers associated with the series of sub-events or to actively involved views. Event handlers associated with the series of sub-events or with actively involved views receive the event information and perform a predetermined process.

[0135] In some embodiments, data updater 276 creates and updates data used in application 236-1. For example, data updater 276 updates the telephone number used in contacts module 237, or stores a video file used in video player module. In some embodiments, object updater 277 creates and updates objects used in application 236-1. For example, object updater 277 creates a new user-interface object or updates the position of a user-interface object. GUI updater 278 updates the GUI. For example, GUI updater 278 prepares display information and sends it to graphics module 232 for display on a touch-sensitive display.

[0136] In some embodiments, event handler(s) 290 includes or has access to data updater 276, object updater 277, and GUI updater 278. In some embodiments, data updater 276, object updater 277, and GUI updater 278 are included in a single module of a respective application 236-1 or application view 291. In other embodiments, they are included in two or more software modules.

[0137] It shall be understood that the foregoing discussion regarding event handling of user touches on touch-sensitive displays also applies to other forms of user inputs to operate multifunction devices 200 with input devices, not all of which are initiated on touch screens.

For example, mouse movement and mouse button presses, optionally coordinated with single or multiple keyboard presses or holds; contact movements such as taps, drags, scrolls, etc. on touchpads; pen stylus inputs; movement of the device; oral instructions; detected eye movements; biometric inputs; and/or any combination thereof are optionally utilized as inputs corresponding to sub-events which define an event to be recognized.

[0138] FIG. 3 illustrates a portable multifunction device 200 having a touch screen 212 in accordance with some embodiments. The touch screen optionally displays one or more graphics within user interface (UI) 300. In this embodiment, as well as others described below, a user is enabled to select one or more of the graphics by making a gesture on the graphics, for example, with one or more fingers 302 (not drawn to scale in the figure) or one or more styluses 303 (not drawn to scale in the figure). In some embodiments, selection of one or more graphics occurs when the user breaks contact with the one or more graphics. In some embodiments, the gesture optionally includes one or more taps, one or more swipes (from left to right, right to left, upward and/or downward), and/or a rolling of a finger (from right to left, left to right, upward and/or downward) that has made contact with device 200. In some implementations or circumstances, inadvertent contact with a graphic does not select the graphic. For example, a swipe gesture that sweeps over an application icon optionally does not select the corresponding application when the gesture corresponding to selection is a tap.

[0139] Device 200 may also include one or more physical buttons, such as “home” or menu button 304. As described previously, menu button 304 may be used to navigate to any application 236 in a set of applications that may be executed on device 200. Alternatively, in some embodiments, the menu button is implemented as a soft key in a GUI displayed on touch screen 212.

[0140] In one embodiment, device 200 includes touch screen 212, menu button 304, push button 306 for powering the device on/off and locking the device, volume adjustment button(s) 308, subscriber identity module (SIM) card slot 310, headset jack 312, and docking/charging external port 224. Push button 306 is, optionally, used to turn the power on/off on the device by depressing the button and holding the button in the depressed state for a predefined time interval; to lock the device by depressing the button and releasing the button before the predefined time interval has elapsed; and/or to unlock the device or initiate an unlock process. In an alternative embodiment, device 200 also accepts verbal input for

activation or deactivation of some functions through microphone 213. Device 200 also, optionally, includes one or more contact intensity sensors 265 for detecting intensity of contacts on touch screen 212 and/or one or more tactile output generators 267 for generating tactile outputs for a user of device 200.

[0141] FIG. 4 is a block diagram of an exemplary multifunction device with a display and a touch-sensitive surface in accordance with some embodiments. Device 400 need not be portable. In some embodiments, device 400 is a laptop computer, a desktop computer, a tablet computer, a multimedia player device, a navigation device, an educational device (such as a child's learning toy), a gaming system, or a control device (e.g., a home or industrial controller). Device 400 typically includes one or more processing units (CPUs) 410, one or more network or other communications interfaces 460, memory 470, and one or more communication buses 420 for interconnecting these components. Communication buses 420 optionally include circuitry (sometimes called a chipset) that interconnects and controls communications between system components. Device 400 includes input/output (I/O) interface 430 comprising display 440, which is typically a touch screen display. I/O interface 430 also optionally includes a keyboard and/or mouse (or other pointing device) 450 and touchpad 455, tactile output generator 457 for generating tactile outputs on device 400 (e.g., similar to tactile output generator(s) 267 described above with reference to FIG. 2A), sensors 459 (e.g., optical, acceleration, proximity, touch-sensitive, and/or contact intensity sensors similar to contact intensity sensor(s) 265 described above with reference to FIG. 2A). Memory 470 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM, or other random access solid state memory devices; and optionally includes non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory 470 optionally includes one or more storage devices remotely located from CPU(s) 410. In some embodiments, memory 470 stores programs, modules, and data structures analogous to the programs, modules, and data structures stored in memory 202 of portable multifunction device 200 (FIG. 2A), or a subset thereof. Furthermore, memory 470 optionally stores additional programs, modules, and data structures not present in memory 202 of portable multifunction device 200. For example, memory 470 of device 400 optionally stores drawing module 480, presentation module 482, word processing module 484, website creation module 486, disk authoring module 488, and/or spreadsheet module 490, while memory 202 of portable multifunction device 200 (FIG. 2A) optionally does not store these modules.

[0142] Each of the above-identified elements in FIG. 4 may be stored in one or more of the previously mentioned memory devices. Each of the above-identified modules corresponds to a set of instructions for performing a function described above. The above-identified modules or programs (e.g., sets of instructions) need not be implemented as separate software programs, procedures, or modules, and thus various subsets of these modules may be combined or otherwise rearranged in various embodiments. In some embodiments, memory 470 may store a subset of the modules and data structures identified above. Furthermore, memory 470 may store additional modules and data structures not described above.

[0143] Attention is now directed towards embodiments of user interfaces that may be implemented on, for example, portable multifunction device 200.

[0144] FIG. 5A illustrates an exemplary user interface for a menu of applications on portable multifunction device 200 in accordance with some embodiments. Similar user interfaces may be implemented on device 400. In some embodiments, user interface 500 includes the following elements, or a subset or superset thereof:

[0145] Signal strength indicator(s) 502 for wireless communication(s), such as cellular and Wi-Fi signals;

- Time 504;
- Bluetooth indicator 505;
- Battery status indicator 506;
- Tray 508 with icons for frequently used applications, such as:
 - Icon 516 for telephone module 238, labeled “Phone,” which optionally includes an indicator 514 of the number of missed calls or voicemail messages;
 - Icon 518 for e-mail client module 240, labeled “Mail,” which optionally includes an indicator 510 of the number of unread e-mails;
 - Icon 520 for browser module 247, labeled “Browser;” and

- Icon 522 for video and music player module 252, also referred to as iPod (trademark of Apple Inc.) module 252, labeled “iPod;” and
- Icons for other applications, such as:
 - Icon 524 for IM module 241, labeled “Messages;”
 - Icon 526 for calendar module 248, labeled “Calendar;”
 - Icon 528 for image management module 244, labeled “Photos;”
 - Icon 530 for camera module 243, labeled “Camera;”
 - Icon 532 for online video module 255, labeled “Online Video;”
 - Icon 534 for stocks widget 249-2, labeled “Stocks;”
 - Icon 536 for map module 254, labeled “Maps;”
 - Icon 538 for weather widget 249-1, labeled “Weather;”
 - Icon 540 for alarm clock widget 249-4, labeled “Clock;”
 - Icon 542 for workout support module 242, labeled “Workout Support;”
 - Icon 544 for notes module 253, labeled “Notes;” and
 - Icon 546 for a settings application or module, labeled “Settings,” which provides access to settings for device 200 and its various applications 236.

[0146] It should be noted that the icon labels illustrated in FIG. 5A are merely exemplary. For example, icon 522 for video and music player module 252 may optionally be labeled “Music” or “Music Player.” Other labels are, optionally, used for various application icons. In some embodiments, a label for a respective application icon includes a name of an application corresponding to the respective application icon. In some embodiments, a label for a particular application icon is distinct from a name of an application corresponding to the particular application icon.

[0147] FIG. 5B illustrates an exemplary user interface on a device (e.g., device 400, FIG. 4) with a touch-sensitive surface 551 (e.g., a tablet or touchpad 455, FIG. 4) that is separate

from the display 550 (e.g., touch screen display 212). Device 400 also, optionally, includes one or more contact intensity sensors (e.g., one or more of sensors 457) for detecting intensity of contacts on touch-sensitive surface 551 and/or one or more tactile output generators 459 for generating tactile outputs for a user of device 400.

[0148] Although some of the examples which follow will be given with reference to inputs on touch screen display 212 (where the touch-sensitive surface and the display are combined), in some embodiments, the device detects inputs on a touch-sensitive surface that is separate from the display, as shown in FIG. 5B. In some embodiments, the touch-sensitive surface (e.g., 551 in FIG. 5B) has a primary axis (e.g., 552 in FIG. 5B) that corresponds to a primary axis (e.g., 553 in FIG. 5B) on the display (e.g., 550). In accordance with these embodiments, the device detects contacts (e.g., 560 and 562 in FIG. 5B) with the touch-sensitive surface 551 at locations that correspond to respective locations on the display (e.g., in FIG. 5B, 560 corresponds to 568 and 562 corresponds to 570). In this way, user inputs (e.g., contacts 560 and 562, and movements thereof) detected by the device on the touch-sensitive surface (e.g., 551 in FIG. 5B) are used by the device to manipulate the user interface on the display (e.g., 550 in FIG. 5B) of the multifunction device when the touch-sensitive surface is separate from the display. It should be understood that similar methods are, optionally, used for other user interfaces described herein.

[0149] Additionally, while the following examples are given primarily with reference to finger inputs (e.g., finger contacts, finger tap gestures, finger swipe gestures), it should be understood that, in some embodiments, one or more of the finger inputs are replaced with input from another input device (e.g., a mouse-based input or stylus input). For example, a swipe gesture is, optionally, replaced with a mouse click (e.g., instead of a contact) followed by movement of the cursor along the path of the swipe (e.g., instead of movement of the contact). As another example, a tap gesture is, optionally, replaced with a mouse click while the cursor is located over the location of the tap gesture (e.g., instead of detection of the contact followed by ceasing to detect the contact). Similarly, when multiple user inputs are simultaneously detected, it should be understood that multiple computer mice are, optionally, used simultaneously, or a mouse and finger contacts are, optionally, used simultaneously.

[0150] FIG. 6A illustrates exemplary personal electronic device 600. Device 600 includes body 602. In some embodiments, device 600 can include some or all of the features described with respect to devices 200 and 400 (e.g., FIGS. 2A-4B). In some embodiments,

device 600 has touch-sensitive display screen 604, hereafter touch screen 604. Alternatively, or in addition to touch screen 604, device 600 has a display and a touch-sensitive surface. As with devices 200 and 400, in some embodiments, touch screen 604 (or the touch-sensitive surface) may have one or more intensity sensors for detecting intensity of contacts (e.g., touches) being applied. The one or more intensity sensors of touch screen 604 (or the touch-sensitive surface) can provide output data that represents the intensity of touches. The user interface of device 600 can respond to touches based on their intensity, meaning that touches of different intensities can invoke different user interface operations on device 600.

[0151] Techniques for detecting and processing touch intensity may be found, for example, in related applications: International Patent Application Serial No. PCT/US2013/040061, titled “Device, Method, and Graphical User Interface for Displaying User Interface Objects Corresponding to an Application,” filed May 8, 2013, and International Patent Application Serial No. PCT/US2013/069483, titled “Device, Method, and Graphical User Interface for Transitioning Between Touch Input to Display Output Relationships,” filed November 11, 2013, each of which is hereby incorporated by reference in their entirety.

[0152] In some embodiments, device 600 has one or more input mechanisms 606 and 608. Input mechanisms 606 and 608, if included, can be physical. Examples of physical input mechanisms include push buttons and rotatable mechanisms. In some embodiments, device 600 has one or more attachment mechanisms. Such attachment mechanisms, if included, can permit attachment of device 600 with, for example, hats, eyewear, earrings, necklaces, shirts, jackets, bracelets, watch straps, chains, trousers, belts, shoes, purses, backpacks, and so forth. These attachment mechanisms may permit device 600 to be worn by a user.

[0153] FIG. 6B depicts exemplary personal electronic device 600. In some embodiments, device 600 can include some or all of the components described with respect to FIGS. 2A, 2B, and 4. Device 600 has bus 612 that operatively couples I/O section 614 with one or more computer processors 616 and memory 618. I/O section 614 can be connected to display 604, which can have touch-sensitive component 622 and, optionally, touch-intensity sensitive component 624. In addition, I/O section 614 can be connected with communication unit 630 for receiving application and operating system data, using Wi-Fi, Bluetooth, near field communication (NFC), cellular, and/or other wireless communication techniques. Device

600 can include input mechanisms 606 and/or 608. Input mechanism 606 may be a rotatable input device or a depressible and rotatable input device, for example. Input mechanism 608 may be a button, in some examples.

[0154] Input mechanism 608 may be a microphone, in some examples. Personal electronic device 600 can include various sensors, such as GPS sensor 632, accelerometer 634, directional sensor 640 (e.g., compass), gyroscope 636, motion sensor 638, and/or a combination thereof, all of which can be operatively connected to I/O section 614.

[0155] Memory 618 of personal electronic device 600 can be a non-transitory computer-readable storage medium, for storing computer-executable instructions, which, when executed by one or more computer processors 616, for example, can cause the computer processors to perform the techniques described below, including methods 1000 and 1100 (FIGS. 10A-B and 11A-B). The computer-executable instructions can also be stored and/or transported within any non-transitory computer-readable storage medium for use by or in connection with an instruction execution system, apparatus, or device, such as a computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. For purposes of this document, a “non-transitory computer-readable storage medium” can be any medium that can tangibly contain or store computer-executable instructions for use by or in connection with the instruction execution system, apparatus, or device. The non-transitory computer-readable storage medium can include, but is not limited to, magnetic, optical, and/or semiconductor storages. Examples of such storage include magnetic disks, optical discs based on CD, DVD, or Blu-ray technologies, as well as persistent solid-state memory such as flash, solid-state drives, and the like. Personal electronic device 600 is not limited to the components and configuration of FIG. 6B, but can include other or additional components in multiple configurations.

[0156] As used here, the term “affordance” refers to a user-interactive graphical user interface object that may be displayed on the display screen of devices 200, 400, and/or 600 (FIGS. 2, 4, and 6). For example, an image (e.g., icon), a button, and text (e.g., hyperlink) may each constitute an affordance.

[0157] As used herein, the term “focus selector” refers to an input element that indicates a current part of a user interface with which a user is interacting. In some implementations

that include a cursor or other location marker, the cursor acts as a “focus selector” so that when an input (e.g., a press input) is detected on a touch-sensitive surface (e.g., touchpad 455 in FIG. 4 or touch-sensitive surface 551 in FIG. 5B) while the cursor is over a particular user interface element (e.g., a button, window, slider or other user interface element), the particular user interface element is adjusted in accordance with the detected input. In some implementations that include a touch screen display (e.g., touch-sensitive display system 212 in FIG. 2A or touch screen 212 in FIG. 5A) that enables direct interaction with user interface elements on the touch screen display, a detected contact on the touch screen acts as a “focus selector” so that when an input (e.g., a press input by the contact) is detected on the touch screen display at a location of a particular user interface element (e.g., a button, window, slider, or other user interface element), the particular user interface element is adjusted in accordance with the detected input. In some implementations, focus is moved from one region of a user interface to another region of the user interface without corresponding movement of a cursor or movement of a contact on a touch screen display (e.g., by using a tab key or arrow keys to move focus from one button to another button); in these implementations, the focus selector moves in accordance with movement of focus between different regions of the user interface. Without regard to the specific form taken by the focus selector, the focus selector is generally the user interface element (or contact on a touch screen display) that is controlled by the user so as to communicate the user’s intended interaction with the user interface (e.g., by indicating, to the device, the element of the user interface with which the user is intending to interact). For example, the location of a focus selector (e.g., a cursor, a contact, or a selection box) over a respective button while a press input is detected on the touch-sensitive surface (e.g., a touchpad or touch screen) will indicate that the user is intending to activate the respective button (as opposed to other user interface elements shown on a display of the device).

[0158] As used in the specification and claims, the term “characteristic intensity” of a contact refers to a characteristic of the contact based on one or more intensities of the contact. In some embodiments, the characteristic intensity is based on multiple intensity samples. The characteristic intensity is, optionally, based on a predefined number of intensity samples, or a set of intensity samples collected during a predetermined time period (e.g., 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10 seconds) relative to a predefined event (e.g., after detecting the contact, prior to detecting liftoff of the contact, before or after detecting a start of movement of the contact, prior to detecting an end of the contact, before or after detecting an increase in intensity of the

contact, and/or before or after detecting a decrease in intensity of the contact). A characteristic intensity of a contact is, optionally based on one or more of: a maximum value of the intensities of the contact, a mean value of the intensities of the contact, an average value of the intensities of the contact, a top 10 percentile value of the intensities of the contact, a value at the half maximum of the intensities of the contact, a value at the 90 percent maximum of the intensities of the contact, or the like. In some embodiments, the duration of the contact is used in determining the characteristic intensity (e.g., when the characteristic intensity is an average of the intensity of the contact over time). In some embodiments, the characteristic intensity is compared to a set of one or more intensity thresholds to determine whether an operation has been performed by a user. For example, the set of one or more intensity thresholds may include a first intensity threshold and a second intensity threshold. In this example, a contact with a characteristic intensity that does not exceed the first threshold results in a first operation, a contact with a characteristic intensity that exceeds the first intensity threshold and does not exceed the second intensity threshold results in a second operation, and a contact with a characteristic intensity that exceeds the second threshold results in a third operation. In some embodiments, a comparison between the characteristic intensity and one or more thresholds is used to determine whether or not to perform one or more operations (e.g., whether to perform a respective operation or forgo performing the respective operation) rather than being used to determine whether to perform a first operation or a second operation.

[0159] In some embodiments, a portion of a gesture is identified for purposes of determining a characteristic intensity. For example, a touch-sensitive surface may receive a continuous swipe contact transitioning from a start location and reaching an end location, at which point the intensity of the contact increases. In this example, the characteristic intensity of the contact at the end location may be based on only a portion of the continuous swipe contact, and not the entire swipe contact (e.g., only the portion of the swipe contact at the end location). In some embodiments, a smoothing algorithm may be applied to the intensities of the swipe contact prior to determining the characteristic intensity of the contact. For example, the smoothing algorithm optionally includes one or more of: an unweighted sliding-average smoothing algorithm, a triangular smoothing algorithm, a median filter smoothing algorithm, and/or an exponential smoothing algorithm. In some circumstances, these smoothing algorithms eliminate narrow spikes or dips in the intensities of the swipe contact for purposes of determining a characteristic intensity.

[0160] The intensity of a contact on the touch-sensitive surface may be characterized relative to one or more intensity thresholds, such as a contact-detection intensity threshold, a light press intensity threshold, a deep press intensity threshold, and/or one or more other intensity thresholds. In some embodiments, the light press intensity threshold corresponds to an intensity at which the device will perform operations typically associated with clicking a button of a physical mouse or a trackpad. In some embodiments, the deep press intensity threshold corresponds to an intensity at which the device will perform operations that are different from operations typically associated with clicking a button of a physical mouse or a trackpad. In some embodiments, when a contact is detected with a characteristic intensity below the light press intensity threshold (e.g., and above a nominal contact-detection intensity threshold below which the contact is no longer detected), the device will move a focus selector in accordance with movement of the contact on the touch-sensitive surface without performing an operation associated with the light press intensity threshold or the deep press intensity threshold. Generally, unless otherwise stated, these intensity thresholds are consistent between different sets of user interface figures.

[0161] An increase of characteristic intensity of the contact from an intensity below the light press intensity threshold to an intensity between the light press intensity threshold and the deep press intensity threshold is sometimes referred to as a “light press” input. An increase of characteristic intensity of the contact from an intensity below the deep press intensity threshold to an intensity above the deep press intensity threshold is sometimes referred to as a “deep press” input. An increase of characteristic intensity of the contact from an intensity below the contact-detection intensity threshold to an intensity between the contact-detection intensity threshold and the light press intensity threshold is sometimes referred to as detecting the contact on the touch-surface. A decrease of characteristic intensity of the contact from an intensity above the contact-detection intensity threshold to an intensity below the contact-detection intensity threshold is sometimes referred to as detecting liftoff of the contact from the touch-surface. In some embodiments, the contact-detection intensity threshold is zero. In some embodiments, the contact-detection intensity threshold is greater than zero.

[0162] In some embodiments described herein, one or more operations are performed in response to detecting a gesture that includes a respective press input or in response to detecting the respective press input performed with a respective contact (or a plurality of

contacts), where the respective press input is detected based at least in part on detecting an increase in intensity of the contact (or plurality of contacts) above a press-input intensity threshold. In some embodiments, the respective operation is performed in response to detecting the increase in intensity of the respective contact above the press-input intensity threshold (e.g., a “down stroke” of the respective press input). In some embodiments, the press input includes an increase in intensity of the respective contact above the press-input intensity threshold and a subsequent decrease in intensity of the contact below the press-input intensity threshold, and the respective operation is performed in response to detecting the subsequent decrease in intensity of the respective contact below the press-input threshold (e.g., an “up stroke” of the respective press input).

[0163] In some embodiments, the device employs intensity hysteresis to avoid accidental inputs sometimes termed “jitter,” where the device defines or selects a hysteresis intensity threshold with a predefined relationship to the press-input intensity threshold (e.g., the hysteresis intensity threshold is X intensity units lower than the press-input intensity threshold or the hysteresis intensity threshold is 75%, 90%, or some reasonable proportion of the press-input intensity threshold). Thus, in some embodiments, the press input includes an increase in intensity of the respective contact above the press-input intensity threshold and a subsequent decrease in intensity of the contact below the hysteresis intensity threshold that corresponds to the press-input intensity threshold, and the respective operation is performed in response to detecting the subsequent decrease in intensity of the respective contact below the hysteresis intensity threshold (e.g., an “up stroke” of the respective press input). Similarly, in some embodiments, the press input is detected only when the device detects an increase in intensity of the contact from an intensity at or below the hysteresis intensity threshold to an intensity at or above the press-input intensity threshold and, optionally, a subsequent decrease in intensity of the contact to an intensity at or below the hysteresis intensity, and the respective operation is performed in response to detecting the press input (e.g., the increase in intensity of the contact or the decrease in intensity of the contact, depending on the circumstances).

[0164] For ease of explanation, the descriptions of operations performed in response to a press input associated with a press-input intensity threshold or in response to a gesture including the press input are, optionally, triggered in response to detecting either: an increase in intensity of a contact above the press-input intensity threshold, an increase in intensity of a

contact from an intensity below the hysteresis intensity threshold to an intensity above the press-input intensity threshold, a decrease in intensity of the contact below the press-input intensity threshold, and/or a decrease in intensity of the contact below the hysteresis intensity threshold corresponding to the press-input intensity threshold. Additionally, in examples where an operation is described as being performed in response to detecting a decrease in intensity of a contact below the press-input intensity threshold, the operation is, optionally, performed in response to detecting a decrease in intensity of the contact below a hysteresis intensity threshold corresponding to, and lower than, the press-input intensity threshold.

3. Digital Assistant System

[0165] FIG. 7A illustrates a block diagram of digital assistant system 700 in accordance with various examples. In some examples, digital assistant system 700 can be implemented on a standalone computer system. In some examples, digital assistant system 700 can be distributed across multiple computers. In some examples, some of the modules and functions of the digital assistant can be divided into a server portion and a client portion, where the client portion resides on one or more user devices (e.g., devices 104, 122, 200, 400, or 600) and communicates with the server portion (e.g., server system 108) through one or more networks, e.g., as shown in FIG. 1. In some examples, digital assistant system 700 can be an implementation of server system 108 (and/or DA server 106) shown in FIG. 1. It should be noted that digital assistant system 700 is only one example of a digital assistant system, and that digital assistant system 700 can have more or fewer components than shown, may combine two or more components, or may have a different configuration or arrangement of the components. The various components shown in FIG. 7A can be implemented in hardware, software instructions for execution by one or more processors, firmware, including one or more signal processing and/or application specific integrated circuits, or a combination thereof.

[0166] Digital assistant system 700 can include memory 702, one or more processors 704, input/output (I/O) interface 706, and network communications interface 708. These components can communicate with one another over one or more communication buses or signal lines 710.

[0167] In some examples, memory 702 can include a non-transitory computer-readable medium, such as high-speed random access memory and/or a non-volatile computer-readable

storage medium (e.g., one or more magnetic disk storage devices, flash memory devices, or other non-volatile solid-state memory devices).

[0168] In some examples, I/O interface 706 can couple input/output devices 716 of digital assistant system 700, such as displays, keyboards, touch screens, and microphones, to user interface module 722. I/O interface 706, in conjunction with user interface module 722, can receive user inputs (e.g., voice input, keyboard inputs, touch inputs, etc.) and processes them accordingly. In some examples, e.g., when the digital assistant is implemented on a standalone user device, digital assistant system 700 can include any of the components and I/O communication interfaces described with respect to devices 200, 400, or 600 in FIGs. 2A, 4, 6A-B, respectively. In some examples, digital assistant system 700 can represent the server portion of a digital assistant implementation, and can interact with the user through a client-side portion residing on a user device (e.g., devices 104, 122, 200, 400, or 600).

[0169] In some examples, the network communications interface 708 can include wired communication port(s) 712 and/or wireless transmission and reception circuitry 714. The wired communication port(s) can receive and send communication signals via one or more wired interfaces, e.g., Ethernet, Universal Serial Bus (USB), FIREWIRE, etc. The wireless circuitry 714 can receive and send RF signals and/or optical signals from/to communications networks and other communications devices. The wireless communications can use any of a plurality of communications standards, protocols, and technologies, such as GSM, EDGE, CDMA, TDMA, Bluetooth, Wi-Fi, VoIP, Wi-MAX, or any other suitable communication protocol. Network communications interface 708 can enable communication between digital assistant system 700 with networks, such as the Internet, an intranet, and/or a wireless network, such as a cellular telephone network, a wireless local area network (LAN), and/or a metropolitan area network (MAN), and other devices.

[0170] In some examples, memory 702, or the computer-readable storage media of memory 702, can store programs, modules, instructions, and data structures including all or a subset of: operating system 718, communications module 720, user interface module 722, one or more applications 724, and digital assistant module 726. In particular, memory 702, or the computer-readable storage media of memory 702, can store instructions for performing methods 1000 and 1100, described below. One or more processors 704 can execute these programs, modules, and instructions, and reads/writes from/to the data structures.

[0171] Operating system 718 (e.g., Darwin, RTXC, LINUX, UNIX, iOS, OS X, WINDOWS, or an embedded operating system such as VxWorks) can include various software components and/or drivers for controlling and managing general system tasks (e.g., memory management, storage device control, power management, etc.) and facilitates communications between various hardware, firmware, and software components.

[0172] Communications module 720 can facilitate communications between digital assistant system 700 with other devices over network communications interface 708. For example, communications module 720 can communicate with RF circuitry 208 of electronic devices such as devices 200, 400, and 600 shown in FIG. 2A, 4, 6A-B, respectively. Communications module 720 can also include various components for handling data received by wireless circuitry 714 and/or wired communications port 712.

[0173] User interface module 722 can receive commands and/or inputs from a user via I/O interface 706 (e.g., from a keyboard, touch screen, pointing device, controller, and/or microphone), and generate user interface objects on a display. User interface module 722 can also prepare and deliver outputs (e.g., speech, sound, animation, text, icons, vibrations, haptic feedback, light, etc.) to the user via the I/O interface 706 (e.g., through displays, audio channels, speakers, touch-pads, etc.).

[0174] Applications 724 can include programs and/or modules that are configured to be executed by one or more processors 704. For example, if the digital assistant system is implemented on a standalone user device, applications 724 can include user applications, such as games, a calendar application, a navigation application, or an email application. If digital assistant system 700 is implemented on a server, applications 724 can include resource management applications, diagnostic applications, or scheduling applications, for example.

[0175] Memory 702 can also store digital assistant module 726 (or the server portion of a digital assistant). In some examples, digital assistant module 726 can include the following sub-modules, or a subset or superset thereof: input/output processing module 728, speech-to-text (STT) processing module 730, natural language processing module 732, dialogue flow processing module 734, task flow processing module 736, service processing module 738, and speech synthesis module 740. Each of these modules can have access to one or more of the following systems or data and models of the digital assistant module 726, or a subset or

superset thereof: ontology 760, vocabulary index 744, user data 748, task flow models 754, service models 756, and ASR systems.

[0176] In some examples, using the processing modules, data, and models implemented in digital assistant module 726, the digital assistant can perform at least some of the following: converting speech input into text; identifying a user's intent expressed in a natural language input received from the user; actively eliciting and obtaining information needed to fully infer the user's intent (e.g., by disambiguating words, phrases, intentions, etc.); determining the task flow for fulfilling the inferred intent; and executing the task flow to fulfill the inferred intent.

[0177] In some examples, as shown in FIG. 7B, I/O processing module 728 can interact with the user through I/O devices 716 in FIG. 7A or with a user device (e.g., devices 104, 200, 400, or 600) through network communications interface 708 in FIG. 7A to obtain user input (e.g., a speech input) and to provide responses (e.g., as speech outputs) to the user input. I/O processing module 728 can optionally obtain contextual information associated with the user input from the user device, along with or shortly after the receipt of the user input. The contextual information can include user-specific data, vocabulary, and/or preferences relevant to the user input. In some examples, the contextual information also includes software and hardware states of the user device at the time the user request is received, and/or information related to the surrounding environment of the user at the time that the user request was received. In some examples, I/O processing module 728 can also send follow-up questions to, and receive answers from, the user regarding the user request. When a user request is received by I/O processing module 728 and the user request can include speech input, I/O processing module 728 can forward the speech input to STT processing module 730 (or speech recognizer) for speech-to-text conversions.

[0178] STT processing module 730 can include one or more ASR systems. The one or more ASR systems can process the speech input that is received through I/O processing module 728 to produce a recognition result. Each ASR system can include a front-end speech pre-processor. The front-end speech pre-processor can extract representative features from the speech input. For example, the front-end speech pre-processor can perform a Fourier transform on the speech input to extract spectral features that characterize the speech input as a sequence of representative multi-dimensional vectors. Further, each ASR system can include one or more speech recognition models (e.g., acoustic models and/or language

models) and can implement one or more speech recognition engines. Examples of speech recognition models can include Hidden Markov Models, Gaussian-Mixture Models, Deep Neural Network Models, n-gram language models, and other statistical models. Examples of speech recognition engines can include the dynamic time warping based engines and weighted finite-state transducers (WFST) based engines. The one or more speech recognition models and the one or more speech recognition engines can be used to process the extracted representative features of the front-end speech pre-processor to produce intermediate recognitions results (e.g., phonemes, phonemic strings, and sub-words), and ultimately, text recognition results (e.g., words, word strings, or sequence of tokens). In some examples, the speech input can be processed at least partially by a third-party service or on the user's device (e.g., device 104, 200, 400, or 600) to produce the recognition result. Once STT processing module 730 produces recognition results containing a text string (e.g., words, or sequence of words, or sequence of tokens), the recognition result can be passed to natural language processing module 732 for intent deduction.

[0179] More details on the speech-to-text processing are described in U.S. Utility Application Serial No. 13/236,942 for "Consolidating Speech Recognition Results," filed on September 20, 2011, the entire disclosure of which is incorporated herein by reference.

[0180] In some examples, STT processing module 730 can include and/or access a vocabulary of recognizable words via phonetic alphabet conversion module 731. Each vocabulary word can be associated with one or more candidate pronunciations of the word represented in a speech recognition phonetic alphabet. In particular, the vocabulary of recognizable words can include a word that is associated with a plurality of candidate pronunciations. For example, the vocabulary may include the word "tomato" that is associated with the candidate pronunciations of /tə'meɪrəʊ/ and /tə'matəʊ/. Further, vocabulary words can be associated with custom candidate pronunciations that are based on previous speech inputs from the user. Such custom candidate pronunciations can be stored in STT processing module 730 and can be associated with a particular user via the user's profile on the device. In some examples, the candidate pronunciations for words can be determined based on the spelling of the word and one or more linguistic and/or phonetic rules. In some examples, the candidate pronunciations can be manually generated, e.g., based on known canonical pronunciations.

[0181] In some examples, the candidate pronunciations can be ranked based on the commonness of the candidate pronunciation. For example, the candidate pronunciation /tə'meɪrou/ can be ranked higher than /tə'matou/, because the former is a more commonly used pronunciation (e.g., among all users, for users in a particular geographical region, or for any other appropriate subset of users). In some examples, candidate pronunciations can be ranked based on whether the candidate pronunciation is a custom candidate pronunciation associated with the user. For example, custom candidate pronunciations can be ranked higher than canonical candidate pronunciations. This can be useful for recognizing proper nouns having a unique pronunciation that deviates from canonical pronunciation. In some examples, candidate pronunciations can be associated with one or more speech characteristics, such as geographic origin, nationality, or ethnicity. For example, the candidate pronunciation /tə'meɪrou/ can be associated with the United States, whereas the candidate pronunciation /tə'matou/ can be associated with Great Britain. Further, the rank of the candidate pronunciation can be based on one or more characteristics (e.g., geographic origin, nationality, ethnicity, etc.) of the user stored in the user's profile on the device. For example, it can be determined from the user's profile that the user is associated with the United States. Based on the user being associated with the United States, the candidate pronunciation /tə'meɪrou/ (associated with the United States) can be ranked higher than the candidate pronunciation /tə'matou/ (associated with Great Britain). In some examples, one of the ranked candidate pronunciations can be selected as a predicted pronunciation (e.g., the most likely pronunciation).

[0182] When a speech input is received, STT processing module 730 can be used to determine the phonemes corresponding to the speech input (e.g., using an acoustic model), and then attempt to determine words that match the phonemes (e.g., using a language model). For example, if STT processing module 730 can first identify the sequence of phonemes /tə'meɪrou/ corresponding to a portion of the speech input, it can then determine, based on vocabulary index 744, that this sequence corresponds to the word "tomato."

[0183] In some examples, STT processing module 730 can use approximate matching techniques to determine words in an utterance. Thus, for example, the STT processing module 730 can determine that the sequence of phonemes /tə'meɪrou/ corresponds to the word "tomato," even if that particular sequence of phonemes is not one of the candidate sequence of phonemes for that word.

[0184] Natural language processing module 732 (“natural language processor”) of the digital assistant can take the sequence of words or tokens (“token sequence”) generated by STT processing module 730, and attempt to associate the token sequence with one or more “actionable intents” recognized by the digital assistant. An “actionable intent” can represent a task that can be performed by the digital assistant, and can have an associated task flow implemented in task flow models 754. The associated task flow can be a series of programmed actions and steps that the digital assistant takes in order to perform the task. The scope of a digital assistant’s capabilities can be dependent on the number and variety of task flows that have been implemented and stored in task flow models 754, or in other words, on the number and variety of “actionable intents” that the digital assistant recognizes. The effectiveness of the digital assistant, however, can also be dependent on the assistant’s ability to infer the correct “actionable intent(s)” from the user request expressed in natural language.

[0185] In some examples, in addition to the sequence of words or tokens obtained from STT processing module 730, natural language processing module 732 can also receive contextual information associated with the user request, e.g., from I/O processing module 728. The natural language processing module 732 can optionally use the contextual information to clarify, supplement, and/or further define the information contained in the token sequence received from STT processing module 730. The contextual information can include, for example, user preferences, hardware, and/or software states of the user device, sensor information collected before, during, or shortly after the user request, prior interactions (e.g., dialogue) between the digital assistant and the user, and the like. As described herein, contextual information can be dynamic, and can change with time, location, content of the dialogue, and other factors.

[0186] In some examples, the natural language processing can be based on, e.g., ontology 760. Ontology 760 can be a hierarchical structure containing many nodes, each node representing either an “actionable intent” or a “property” relevant to one or more of the “actionable intents” or other “properties.” As noted above, an “actionable intent” can represent a task that the digital assistant is capable of performing, i.e., it is “actionable” or can be acted on. A “property” can represent a parameter associated with an actionable intent or a sub-aspect of another property. A linkage between an actionable intent node and a property node in ontology 760 can define how a parameter represented by the property node pertains to the task represented by the actionable intent node.

[0187] In some examples, ontology 760 can be made up of actionable intent nodes and property nodes. Within ontology 760, each actionable intent node can be linked to one or more property nodes either directly or through one or more intermediate property nodes. Similarly, each property node can be linked to one or more actionable intent nodes either directly or through one or more intermediate property nodes. For example, as shown in FIG. 7C, ontology 760 can include a “restaurant reservation” node (i.e., an actionable intent node). Property nodes “restaurant,” “date/time” (for the reservation), and “party size” can each be directly linked to the actionable intent node (i.e., the “restaurant reservation” node).

[0188] In addition, property nodes “cuisine,” “price range,” “phone number,” and “location” can be sub-nodes of the property node “restaurant,” and can each be linked to the “restaurant reservation” node (i.e., the actionable intent node) through the intermediate property node “restaurant.” For another example, as shown in FIG. 7C, ontology 760 can also include a “set reminder” node (i.e., another actionable intent node). Property nodes “date/time” (for setting the reminder) and “subject” (for the reminder) can each be linked to the “set reminder” node. Since the property “date/time” can be relevant to both the task of making a restaurant reservation and the task of setting a reminder, the property node “date/time” can be linked to both the “restaurant reservation” node and the “set reminder” node in ontology 760.

[0189] An actionable intent node, along with its linked concept nodes, can be described as a “domain.” In the present discussion, each domain can be associated with a respective actionable intent, and refers to the group of nodes (and the relationships there between) associated with the particular actionable intent. For example, ontology 760 shown in FIG. 7C can include an example of restaurant reservation domain 762 and an example of reminder domain 764 within ontology 760. The restaurant reservation domain includes the actionable intent node “restaurant reservation,” property nodes “restaurant,” “date/time,” and “party size,” and sub-property nodes “cuisine,” “price range,” “phone number,” and “location.” Reminder domain 764 can include the actionable intent node “set reminder,” and property nodes “subject” and “date/time.” In some examples, ontology 760 can be made up of many domains. Each domain can share one or more property nodes with one or more other domains. For example, the “date/time” property node can be associated with many different domains (e.g., a scheduling domain, a travel reservation domain, a movie ticket domain, etc.), in addition to restaurant reservation domain 762 and reminder domain 764.

[0190] While FIG. 7C illustrates two example domains within ontology 760, other domains can include, for example, “find a movie,” “initiate a phone call,” “find directions,” “schedule a meeting,” “send a message,” and “provide an answer to a question,” “read a list,” “providing navigation instructions,” “provide instructions for a task” and so on. A “send a message” domain can be associated with a “send a message” actionable intent node, and may further include property nodes such as “recipient(s),” “message type,” and “message body.” The property node “recipient” can be further defined, for example, by the sub-property nodes such as “recipient name” and “message address.”

[0191] In some examples, ontology 760 can include all the domains (and hence actionable intents) that the digital assistant is capable of understanding and acting upon. In some examples, ontology 760 can be modified, such as by adding or removing entire domains or nodes, or by modifying relationships between the nodes within the ontology 760.

[0192] In some examples, nodes associated with multiple related actionable intents can be clustered under a “super domain” in ontology 760. For example, a “travel” super-domain can include a cluster of property nodes and actionable intent nodes related to travel. The actionable intent nodes related to travel can include “airline reservation,” “hotel reservation,” “car rental,” “get directions,” “find points of interest,” and so on. The actionable intent nodes under the same super domain (e.g., the “travel” super domain) can have many property nodes in common. For example, the actionable intent nodes for “airline reservation,” “hotel reservation,” “car rental,” “get directions,” and “find points of interest” can share one or more of the property nodes “start location,” “destination,” “departure date/time,” “arrival date/time,” and “party size.”

[0193] In some examples, each node in ontology 760 can be associated with a set of words and/or phrases that are relevant to the property or actionable intent represented by the node. The respective set of words and/or phrases associated with each node can be the so-called “vocabulary” associated with the node. The respective set of words and/or phrases associated with each node can be stored in vocabulary index 744 in association with the property or actionable intent represented by the node. For example, returning to FIG. 7B, the vocabulary associated with the node for the property of “restaurant” can include words such as “food,” “drinks,” “cuisine,” “hungry,” “eat,” “pizza,” “fast food,” “meal,” and so on. For another example, the vocabulary associated with the node for the actionable intent of “initiate a phone call” can include words and phrases such as “call,” “phone,” “dial,” “ring,” “call this

number,” “make a call to,” and so on. The vocabulary index 744 can optionally include words and phrases in different languages.

[0194] Natural language processing module 732 can receive the token sequence (e.g., a text string) from STT processing module 730, and determine what nodes are implicated by the words in the token sequence. In some examples, if a word or phrase in the token sequence is found to be associated with one or more nodes in ontology 760 (via vocabulary index 744), the word or phrase can “trigger” or “activate” those nodes. Based on the quantity and/or relative importance of the activated nodes, natural language processing module 732 can select one of the actionable intents as the task that the user intended the digital assistant to perform. In some examples, the domain that has the most “triggered” nodes can be selected. In some examples, the domain having the highest confidence value (e.g., based on the relative importance of its various triggered nodes) can be selected. In some examples, the domain can be selected based on a combination of the number and the importance of the triggered nodes. In some examples, additional factors are considered in selecting the node as well, such as whether the digital assistant has previously correctly interpreted a similar request from a user.

[0195] User data 748 can include user-specific information, such as user-specific vocabulary, user preferences, user address, user’s default and secondary languages, user’s contact list, and other short-term or long-term information for each user. In some examples, natural language processing module 732 can use the user-specific information to supplement the information contained in the user input to further define the user intent. For example, for a user request “invite my friends to my birthday party,” natural language processing module 732 can be able to access user data 748 to determine who the “friends” are and when and where the “birthday party” would be held, rather than requiring the user to provide such information explicitly in his/her request.

[0196] Other details of searching an ontology based on a token string is described in U.S. Utility Application Serial No. 12/341,743 for “Method and Apparatus for Searching Using An Active Ontology,” filed December 22, 2008, the entire disclosure of which is incorporated herein by reference.

[0197] In some examples, once natural language processing module 732 identifies an actionable intent (or domain) based on the user request, natural language processing module

732 can generate a structured query to represent the identified actionable intent. In some examples, the structured query can include parameters for one or more nodes within the domain for the actionable intent, and at least some of the parameters are populated with the specific information and requirements specified in the user request. For example, the user may say “Make me a dinner reservation at a sushi place at 7.” In this case, natural language processing module 732 can be able to correctly identify the actionable intent to be “restaurant reservation” based on the user input. According to the ontology, a structured query for a “restaurant reservation” domain may include parameters such as {Cuisine}, {Time}, {Date}, {Party Size}, and the like. In some examples, based on the speech input and the text derived from the speech input using STT processing module 730, natural language processing module 732 can generate a partial structured query for the restaurant reservation domain, where the partial structured query includes the parameters {Cuisine = “Sushi”} and {Time = “7pm”}. However, in this example, the user’s utterance contains insufficient information to complete the structured query associated with the domain. Therefore, other necessary parameters such as {Party Size} and {Date} may not be specified in the structured query based on the information currently available. In some examples, natural language processing module 732 can populate some parameters of the structured query with received contextual information. For example, in some examples, if the user requested a sushi restaurant “near me,” natural language processing module 732 can populate a {location} parameter in the structured query with GPS coordinates from the user device.

[0198] In some examples, natural language processing module 732 can pass the generated structured query (including any completed parameters) to task flow processing module 736 (“task flow processor”). Task flow processing module 736 can be configured to receive the structured query from natural language processing module 732, complete the structured query, if necessary, and perform the actions required to “complete” the user’s ultimate request. In some examples, the various procedures necessary to complete these tasks can be provided in task flow models 754. In some examples, task flow models 754 can include procedures for obtaining additional information from the user and task flows for performing actions associated with the actionable intent.

[0199] As described above, in order to complete a structured query, task flow processing module 736 may need to initiate additional dialogue with the user in order to obtain additional information, and/or disambiguate potentially ambiguous utterances. When such

interactions are necessary, task flow processing module 736 can invoke dialogue flow processing module 734 to engage in a dialogue with the user. In some examples, dialogue flow processing module 734 can determine how (and/or when) to ask the user for the additional information and receives and processes the user responses. The questions can be provided to and answers can be received from the users through I/O processing module 728. In some examples, dialogue flow processing module 734 can present dialogue output to the user via audio and/or visual output, and receives input from the user via spoken or physical (e.g., clicking) responses. Continuing with the example above, when task flow processing module 736 invokes dialogue flow processing module 734 to determine the “party size” and “date” information for the structured query associated with the domain “restaurant reservation,” dialogue flow processing module 734 can generate questions such as “For how many people?” and “On which day?” to pass to the user. Once answers are received from the user, dialogue flow processing module 734 can then populate the structured query with the missing information, or pass the information to task flow processing module 736 to complete the missing information from the structured query.

[0200] Once task flow processing module 736 has completed the structured query for an actionable intent, task flow processing module 736 can proceed to perform the ultimate task associated with the actionable intent. Accordingly, task flow processing module 736 can execute the steps and instructions in the task flow model according to the specific parameters contained in the structured query. For example, the task flow model for the actionable intent of “restaurant reservation” can include steps and instructions for contacting a restaurant and actually requesting a reservation for a particular party size at a particular time. For example, using a structured query such as: {restaurant reservation, restaurant = ABC Café, date = 3/12/2012, time = 7pm, party size = 5}, task flow processing module 736 can perform the steps of: (1) logging onto a server of the ABC Café or a restaurant reservation system such as OPENTABLE®, (2) entering the date, time, and party size information in a form on the website, (3) submitting the form, and (4) making a calendar entry for the reservation in the user’s calendar.

[0201] In some examples, task flow processing module 736 can employ the assistance of service processing module 738 (“service processing module”) to complete a task requested in the user input or to provide an informational answer requested in the user input. For example, service processing module 738 can act on behalf of task flow processing module

736 to make a phone call, set a calendar entry, invoke a map search, invoke or interact with other user applications installed on the user device, and invoke or interact with third-party services (e.g., a restaurant reservation portal, a social networking website, a banking portal, etc.). In some examples, the protocols and application programming interfaces (API) required by each service can be specified by a respective service model among service models 756. Service processing module 738 can access the appropriate service model for a service and generate requests for the service in accordance with the protocols and APIs required by the service according to the service model.

[0202] For example, if a restaurant has enabled an online reservation service, the restaurant can submit a service model specifying the necessary parameters for making a reservation and the APIs for communicating the values of the necessary parameter to the online reservation service. When requested by task flow processing module 736, service processing module 738 can establish a network connection with the online reservation service using the web address stored in the service model, and send the necessary parameters of the reservation (e.g., time, date, party size) to the online reservation interface in a format according to the API of the online reservation service.

[0203] In some examples, natural language processing module 732, dialogue flow processing module 734, and task flow processing module 736 can be used collectively and iteratively to infer and define the user's intent, obtain information to further clarify and refine the user intent, and finally generate a response (i.e., an output to the user, or the completion of a task) to fulfill the user's intent. The generated response can be a dialogue response to the speech input that at least partially fulfills the user's intent. Further, in some examples, the generated response can be output as a speech output. In these examples, the generated response can be sent to speech synthesis module 740 (e.g., speech synthesizer) where it can be processed to synthesize the dialogue response in speech form. In yet other examples, the generated response can be data content relevant to satisfying a user request in the speech input.

[0204] Speech synthesis module 740 can be configured to synthesize speech outputs for presentation to the user. Speech synthesis module 740 synthesizes speech outputs based on text provided by the digital assistant. For example, the generated dialogue response can be in the form of a text string. Speech synthesis module 740 can convert the text string to an audible speech output. Speech synthesis module 740 can use any appropriate speech

synthesis technique in order to generate speech outputs from text, including, but not limited, to concatenative synthesis, unit selection synthesis, diphone synthesis, domain-specific synthesis, formant synthesis, articulatory synthesis, hidden Markov model (HMM) based synthesis, and sinewave synthesis. In some examples, speech synthesis module 740 can be configured to synthesize individual words based on phonemic strings corresponding to the words. For example, a phonemic string can be associated with a word in the generated dialogue response. The phonemic string can be stored in metadata associated with the word. Speech synthesis model 740 can be configured to directly process the phonemic string in the metadata to synthesize the word in speech form.

[0205] In some examples, instead of (or in addition to) using speech synthesis module 740, speech synthesis can be performed on a remote device (e.g., the server system 108), and the synthesized speech can be sent to the user device for output to the user. For example, this can occur in some implementations where outputs for a digital assistant are generated at a server system. And because server systems generally have more processing power or resources than a user device, it can be possible to obtain higher quality speech outputs than would be practical with client-side synthesis.

[0206] Additional details on digital assistants can be found in the U.S. Utility Application No. 12/987,982, entitled “Intelligent Automated Assistant,” filed January 10, 2011, and U.S. Utility Application No. 13/251,088, entitled “Generating and Processing Task Items That Represent Tasks to Perform,” filed September 30, 2011, the entire disclosures of which are incorporated herein by reference.

4. Exemplary techniques for transitioning between computing devices.

[0207] Attention is now directed towards embodiments of user interfaces (“UI”) and associated processes that may be implemented on a multifunction device with a display and a touch-sensitive surface, such as devices 104, 122, 200, 400, 600, 1200, and 1300 (FIGS. 1, 2, 4, 6, 12, and 13), to improve a user’s experience in transitioning between electronic devices. For brevity, the functionalities described herein may be referred to as “continuity” functionalities.

[0208] FIGS. 8A-8F demonstrate an exemplary transition between the use of two electronic devices in accordance with some embodiments. These figures are also used to illustrate the processes described below, including the processes in FIGS. 10A and 10B.

[0209] FIG. 8A illustrates exemplary electronic device 800 worn by user 802. Device 800 is one of devices 104, 122, 200, 400, 600, or 1200 (FIGS. 1, 2, 4, 6 or 12) in some embodiments. Device 800 is wearable and has touch screen 804 and a microphone for receiving audio input. Device 800 uses its microphone to detect a user's spoken (e.g., audio) input. Through the use of a virtual assistant service, device 800 also provides output responsive to the user's input.

[0210] As seen in FIG. 8A, user 802 provides input 806 requesting movie show times. Optionally, device 800 acknowledges the incoming input by displaying, on touch screen 804, user interface ("UI") screen 808. In addition, device 800 queries an intelligent assistant service to process the request. In some embodiments, device 800 queries an external digital assistant server such as server 106 (FIG. 1). In some embodiments, the service is provided by a local digital assistant process such as digital assistant 102 (FIG. 1).

[0211] Turning to FIG. 8B, with the help of the intelligent assistant service, device 800 obtains and displays a set of results, which includes movies information including show times, responsive to the user's instructions 806. But, touch screen 804 on wearable device 800 is relatively small and is able to display only a fraction of movies information that has been obtained by the device. In the illustrated example, device 800 is able to display essentially just one image thumbnail representing a movie, and does not have sufficient room to present movie show times in addition to the movie thumbnail in a viewable manner.

[0212] In this example, the available movies information is better presented on a larger display screen. FIG. 8C depicts electronic device 810, which is a tablet computer having a larger display, in proximity with device 800. Wearable device 800 and tablet device 810 are within the wireless communications range of one or more protocols such as Bluetooth, Bluetooth Low Energy (BTLE), and/or WiFi. Because the two devices are in-range, device 800 can detect tablet 810 wirelessly. While in-range, device 800 transmits information related to the user's request for movies information (e.g., data associated with spoken instructions 806 in FIG. 8A) to tablet 810.

[0213] After receiving this information, tablet 810 displays affordance 812. Affordance 812 is referred to as a continuity affordance. Continuity affordance 812 includes a thumbnail image indicating to the user that tablet 810 can provide information responsive to the user's previous instructions, which were given to wearable device 800. In the illustrated example,

affordance 812 is provided on the user interface lock screen 814 of tablet 810, thereby permitting a user to unlock and access the requested information (e.g., movie show times) conveniently upon physically retrieving tablet 810. Also, in the illustrated example, user 802 activates continuity affordance 812 by swiping upwards (as indicated by arrow 816), though other methods of activating the affordance are contemplated, such as tapping, touching and holding on the affordance, pressing with a force above a predetermined threshold, etc.

[0214] As seen in FIGS. 8D and 8E, when tablet 810 detects a user activation of continuity affordance 812, the tablet unlocks and provides user interface screen 820 presenting movies information such as show times. Due to its larger screen size, tablet 810 is able to display significantly more information than device 800.

[0215] The placement of and inputs for activating continuity affordance 812 are not limited to the examples provided above. For instance, affordance 812 may reside on a home screen of the device instead of the user interface lock screen. Also, affordance 812 may be activated by a tap or click instead of a swipe. Related applications U.S. Patent Application Serial No. 62/006,043, "CONTINUITY," filed May 3, 2014 and U.S. Patent Application Serial No. 62/035,348, "CONTINUITY," filed August 8, 2014 describe additional user interfaces that provide access to continuity features. The content of these applications are incorporated by reference; for brevity, they are not repeated here.

[0216] FIG. 8F depicts exemplary transmissions of information between various devices that support the continuity techniques described with reference to FIGS. 8A-8D. Exemplary continuity system 850 includes a first electronic device (e.g., wearable device 800), a second electronic device (e.g., tablet device 810), and virtual assistant service 830 (e.g., provided by server system 108 of FIG. 1).

[0217] Upon receiving an input such as spoken input 806 (FIG. 8A), device 800 sends data representing at least a part of the instructions to service 830, as indicated by transmission arrow 832. Service 830 aides device 800 in processing the user's input and in providing output (e.g., movie show times) responsive to the user's request.

[0218] In turn, device 800 receives two sets of information from service 830. First, device 800 receives result(s) responsive to the user's input, as represented by transmission arrow 834. For example, transmission 834 includes movies information. Second, device 800 receives token 838 as represented by transmission arrow 836.

[0219] In some embodiments, token 838 is generated by service 830 and is associated with the user's input (e.g., "Hey Siri give me movie show times"). Token 838 permits subsequent lookup, from service 830, of the user's input. Token 838 also permits subsequent lookup of the results provided by service 830 to the user's input (e.g., movie show times). In some embodiments, service 830 stores the association between token 838, user input 806, and output 820 in a database to permit later identification of the relationship.

[0220] After receiving token 838, device 800 transmits the token to tablet 810, as represented by transmission arrow 840. Upon receiving token 838, tablet 810 displays an indication that continuity functionality is available (e.g., continuity affordance 812 in FIG. 8C). When a user invokes continuity functionality on tablet 810, tablet 810 transmits token 838 to service 830 (as indicated by transmission arrow 844) in order to obtain relevant results—based on the user's previous input to device 800—for display. In the illustrated example, transmission arrow 846 indicates movies show times being provided by service 830 to device 846.

[0221] Through these techniques, user 802 is able to transition from the viewing of movies information on wearable device 800 to tablet 810 without a loss of progress. During the transition, user 802 did not need to repeat his spoken instructions (e.g., instructions 806). Further, during the transition, user 802 did not need to manually search for movie show times. Rather, the output of the virtual assistant of device 800 is transitioned onto tablet 810 through the use of token 838 and the supporting features provided by server 830.

[0222] This technique of token transmission between source device 800, target device 810, and networked server 830 reduces the amount of information that is transmitted between devices 800 and 810 in order to support continuity functionality, particularly with respect to those involving natural language requests. As can be appreciated, the transmission of speech datagrams is more data intensive than the transmission of tokens. The reduction of data transmission, particularly wireless transmissions, reduces overall power consumption by the device, and beneficially extends the runtime of portable devices such as devices 800 and 810.

[0223] It is noted that although the exemplary illustrations of FIGS. 8A-8F depict device 800 as a wearable device and device 810 as a tablet device, the described techniques are not limited to such devices. In some examples, device 800 is a phone, portable music device,

tablet, laptop, or desktop computing device. In some examples, device 810 is a wearable device, portable music device, a tablet, a laptop, or desktop computing device.

[0224] FIGS. 9A-9F demonstrate an exemplary transition between the use of two electronic devices in accordance with some embodiments. These figures are also used to illustrate the processes described below, including the processes in FIGS. 11.

[0225] FIG. 9A illustrates electronic device 900 worn by user 902. Device 900 is one of devices 104, 122, 200, 400, 600, and 1300 (FIGS. 1, 2, 4, 6, or 13) in some embodiments. Device 900 is a wearable and has touch screen 904 and, optionally, an input mechanism for receiving input. Device 900 uses touch screen 904 and/or additional input mechanisms, such as a microphone and/or virtual keyboard, to detect a user's input. Through the use of a virtual assistant service, device 900 provides output responsive to the user's input.

[0226] As seen in FIG. 9A, user 902 provides spoken input 906 requesting product information of a manufacturer. Responsive to the input, device 900 displays, on touch screen 904, user interface ("UI") screen 908 acknowledging the input. In addition, device 900 queries an intelligent assistant service to process the request. In some embodiments, device 800 queries an external digital assistant server such as server 106 (FIG. 1). In some embodiments, the service is provided by a local digital assistant process such as digital assistant 102 (FIG. 1).

[0227] Turning to FIG. 9B, in some embodiments, device 900 displays web-based results 908 on touch screen 904, responsive to the user's instructions. In some embodiments, results 908 contain web-based results—such as a web page—because the intelligent assistant service has ascertained that user input 906 is best responded to with a web search. In some embodiments, results 908 contain web-based results because the intelligent assistant service has ascertained that user input 906 includes a request for web search.

[0228] Because touch screen 904 of device 900 has a relatively small display size, it is able to display only a fraction of results 908, which is a product manufacturer's web page. The web page may be better presented on a larger display screen. FIG. 9C depicts electronic device 910 that is within the wireless communication range of device 900. Device 910 is a tablet computer in the illustrated example that has a larger display screen. While wearable device 900 and tablet 910 are within communications range of one another, wearable device 900 sends information, usable by tablet 910 to provide results 908, to tablet 910. In some

embodiments, this information includes the URL to the product manufacturer's web site. In response to receiving this information, tablet 910 displays continuity affordance 912, indicating to the user that continuity features are available with respect to the user's input 906 (FIG. 9A).

[0229] In some embodiments (not illustrated), device 900 refrains from displaying web-based results 908, and instead displays a message informing the user that continuity features can be invoked on a nearby device to process the request (e.g., 906) and/or to view the resulting information.

[0230] Turning to FIGS. 9D and 9E, when tablet 910 detects user activation of continuity affordance 912 (e.g., as indicated by arrow 916), tablet 910 unlocks and displays web browser user interface screen 920, which presents product information responsive to instructions 906 (FIG. 9A). Specifically, web browser UI screen 920 presents information from URL 922, which was also presented on device 900. However, due to its larger screen size, tablet 910 is able to display significantly more information than device 900.

[0231] FIG. 9E depicts exemplary transmissions of information between various devices that support the continuity techniques described with reference to FIGS. 9A-9D. Exemplary continuity system 950 includes a first electronic device (e.g., wearable device 900), a second electronic device (e.g., tablet device 910), and virtual assistant service 930, (e.g., provided by server system 108 of FIG. 1).

[0232] Upon receiving spoken instructions 906 (FIG. 9A), device 900 sends data representing at least a part of the instructions to service 930, as indicated by transmission arrow 932. Service 930 aides device 900 in processing the user's input and in providing output (e.g., web site) responsive to the user's request. In turn, device 900 receives from service 930 web site URL 938 identified by service 930 as being responsive to the user's instructions, as indicated by transmission arrow 936.

[0233] After receiving URL 938, device 900 transmits the URL to tablet 910, as represented by transmission arrow 940. Upon receiving URL 936, tablet 910 displays an indication that continuity functionality is available (e.g., continuity affordance 912 in FIG. 9C). When the continuity functionality is invoked on tablet 910, a web browser application of the device retrieves information identified by URL 938 and displays the retrieved

information. For example, the retrieved information includes product information as previously requested by the user via device 900.

[0234] Through these techniques, user 902 is able to transition from the browsing of web results on wearable device 900 to tablet 910 without a loss of progress. During the transition, user 902 did not need to repeat his spoken instructions (e.g., instructions 906). Further, during the transition, user 802 did not need to manually search for the requested product information using a web browser. Rather, the output of the virtual assistant of device 900 is transitioned automatically onto tablet 910. As can be appreciated, the transmission of speech datagrams is more data intensive than the transmission of URL information. The reduction of data transmission, particularly wireless transmissions, reduces overall power consumption by the device, and beneficially extends the runtime of portable devices such as devices 900 and 910.

5. Exemplary processes for transitioning between computing devices.

[0235] FIGS. 10A and 10B is a flow diagram illustrating method 1000 of triggering a virtual assistant on an electronic device in accordance with some embodiments. Method 1000 is performed at one of devices 104, 122, 200, 400, 600, 800, and 1200 (FIGS. 1, 2, 8 or 12) in some embodiments. In some embodiments, method 1000 is performed at a device having a touch screen. In some embodiments, method 1000 is performed at a device with a separate display screen and a touch-sensitive surface. Operations in method 1000 are, optionally, combined and/or the order of some operations is, optionally, changed.

[0236] At block 1002, the device receives audio input via its microphone. The received audio input includes a request (e.g., a request for movie show times as seen in FIG. 8A). The request is a request for information, or a request for performance of a task, for example. At block 1004, the device sends data representing at least a part of the request to a service, such as service 830 (FIG. 8F). At block 1006, the device receives results (e.g., movie show times as seen in FIG. 8B) that are responsive to the request. At block 1006, the device also receives a token from the service. In some embodiments, the token was generated by service 830 (FIG. 8F) and has been associated with the user's request in a data store of the service. The token permits subsequent lookup, from the service, of at least one of the user's request and results responsive to the request. At block 1008, the device sends the token to an external electronic device, such as tablet computer 810 (FIG. 8C), for example. The token may be

used by the external electronic for purposes of continuity. At block 1010, the device at least partially displays the results received from the service responsive to the user's instructions.

[0237] Optionally, at block 1004, the device sends data representing a device context to the virtual assistant service. As described above, contextual information can include, for example, user preferences, hardware, and/or software states of the user device, sensor information collected before, during, or shortly after the user request, prior interactions (e.g., dialogue) between the digital assistant and the user, and the like. Exemplary prior interaction history includes active applications, request history, previous selections from results, and so forth. Exemplary sensor context includes location, time, movement, and so forth. Exemplary user preference information includes name, email addresses, physical addresses, phone numbers, account numbers, and so forth. The virtual assistant service can use context information to help provide relevant results.

[0238] In addition, the virtual assistant service can associate context information with a token (together with the user's instruction), such that the virtual assistant service can later re-run a user's earlier instruction based on context information collected at the time of the instruction. For example, when a user device later provides the token to the virtual assistant, the virtual assistant can identify the user instruction and the context in which the user instruction should be operationalized. In addition, the service can permit lookup of context information via an associated token. In this way, the virtual assistant service can provide consistent outputs to different devices over time, based on the same user instruction. Examples of context information include the currently running application(s), the application currently in focus, an indication of a recently received message, a recent user request, the location of the device, etc. As will be understood, context information can be used to disambiguate user intent – for example, while viewing a webpage a user may utter to the virtual assistant “send this to my mom.” The virtual assistant, using the current context (e.g., the webpage in focus), can disambiguate “this” to mean the current webpage and use this to send the correct information. As another example, a user may have recently received a message (e.g., SMS or email) from “Bob” – the may utter to the virtual assistant “respond to him that I'll be 5 minutes late.” The virtual assistant, using the context (e.g., a message was recently received) can use this to disambiguate “him” to mean “Bob” and take the appropriate action. One of ordinary skill will understand the previous examples to be exemplary and non-limiting uses of context information to disambiguate user intent and will also appreciate

the efficiency obtained by maintaining context information and passing/transferring it to a second device to continue a virtual assistant session such that user requests can continue to be handled seamlessly.

[0239] In some embodiments, at block 1006, the device receives a generic response from the virtual assistant instead of a token. While the generic response is responsive to the user instructions provided to the virtual assistant at block 1002, the response is generic in that its contents are not specific to the user's instructions. The generic message, for example, indicates that the virtual assistant is unable to identify information responsive to the user's request. As another example, the generic message indicates that the virtual assistant is unable to operationalize the user's request into a task, as would be the case if a user requests a task that cannot be performed by the device.

[0240] Optionally, at block 1008, the device determines whether a virtual assistant session is actively executing on the device prior to sending the token to the external device. The device sends the token to the external device if the virtual assistant is active. The device does not send the token to the external device if the virtual assistant is inactive. For example, a virtual assistant session is considered active if it is in the foreground of the user interface of the device. As another example, a virtual assistant session is considered active if the microphone of the device is actively sampling and processing audio input. As another example, a virtual assistant is considered active if the backlight of the device is on, meaning, for instance, that the assistant runs whenever the user is interacting with the device's display.

[0241] In some embodiments, at block 1008, the device sends the token to the external device by broadcasting the token on an electronic communication channel, such as Bluetooth, Bluetooth Low Energy, WiFi, so forth. In some embodiments, at block 1008, the device sends the token to the external device as part of a transmission addressed to the external electronic device, such as through the use of network packets addressed to the external electronic device.

[0242] FIGS. 11A and 11B is a flow diagram illustrating method 1100 of triggering a virtual assistant on an electronic device in accordance with some embodiments. Method 1100 is performed at one of devices 104, 122, 200, 400, 600, 900, and 1300 (FIGS. 1, 2, 4, 6, 9 or 13) in some embodiments. In some embodiments, method 1100 is performed at a device having a touch screen. In some embodiments, method 1100 is performed at a device with a

separate display screen and a touch-sensitive surface. Operations in method 1100 are, optionally, combined and/or the order of some operations is, optionally, changed.

[0243] At block 1102, the device receives input, from the user, containing a request. In some embodiments, the device receives audio input containing natural language instructions (e.g., instruction 906 in FIG. 9A) from a microphone of the device. In some embodiments, the device receives textual input provided by the user via a keyboard, such as a virtual keyboard of the device.

[0244] At block 1104, the device sends data representing at least a part of the request to a service, such as service 930 (FIG. 9F). At block 1106, the device receives a uniform resource locator (URL) from the service, such as URL 938. The URL identifies a resource, such as a web site, a networked location, or another computing resource. In some embodiments, the URL was identified by service 930 (FIG. 9F) as being relevant to the user's instructions sent at block 1102. At block 1108, the device sends the URL to an external electronic device, such as tablet computer 910 (FIG. 9C), for example. Optionally, at block 1110, the device obtains and at least partially displays information located using the URL. Optionally, at block 1110, the device displays an indication indicating that information responsive to the user's request is available for display at the external electronic device.

[0245] Optionally, at block 1104, the device determines whether a web browsing application is running actively (e.g., in the foreground of the device's user interface) while the user instructions were received. This context information is optionally provided to the intelligent assistant service. This indication is used to determine whether the service should return a URL (e.g., URL 938 in FIG. 9F) instead of a token (e.g., token 838 in FIG. 8F), as web-based searches typically produce results that can be referenced using a URL.

[0246] In some embodiments, in addition to processing context information received from the device, the service determines whether a request sent by the device at block 1104 represents user instructions to perform a web search. In some embodiments, if the user instructions included direction to perform a web search, the service returns a URL (e.g., URL 938) instead of a token (e.g., token 8 in FIG. 8F) to the device.

[0247] In some embodiments, at block 1106, the device receives a generic response from the virtual assistant instead of a URL. While the generic response is responsive to the user instructions provided to the virtual assistant at block 1102, the response is generic in that its

contents are not specific to the user's instructions. The generic message, for example, indicates that the virtual assistant is unable to identify information responsive to the user's request. As another example, the generic message indicates that the virtual assistant is unable to operationalize the user's request into a task, as would be the case if a user requests for a task that cannot be performed by the device.

[0248] Optionally, at block 1108, the device determines whether a virtual assistant session is actively executing on the device prior to sending the token to the external device. The device sends the token to the external device if the virtual assistant is active. The device does not send the URL to the external device if the virtual assistant is inactive. For example, a virtual assistant session is considered active if it is in the foreground of the user interface of the device. As another example, a virtual assistant session is considered active if the microphone of the device is actively sampling and processing audio input. As another example, a virtual assistant is considered active if the backlight of the device is on, meaning, for instance, that the assistant runs whenever the user is interacting with the device's display.

[0249] In some embodiments, at block 1108, the device sends the token to the external device by broadcasting the token on an electronic communication channel, such as Bluetooth, Bluetooth Low Energy, WiFi, so forth. In some embodiments, at block 1108, the device sends the token to the external device as part of a transmission addressed to the external electronic device, such as through the use of network packets addressed to the external electronic device.

[0250] In accordance with some embodiments, FIG. 12 shows a functional block diagram of an electronic device 1200 configured in accordance with the principles of the various described embodiments, including those described with reference to FIGS. 8 and 10. The functional blocks of the device are, optionally, implemented by hardware, software, or a combination of hardware and software to carry out the principles of the various described embodiments. It is understood by persons of skill in the art that the functional blocks described in FIG. 12 are, optionally, combined or separated into sub-blocks to implement the principles of the various described embodiments. Therefore, the description herein optionally supports any possible combination or separation or further definition of the functional blocks described herein.

[0251] As shown in FIG. 12, an electronic device 1200 includes a display unit 1202 configured to display a graphical user interface, optional touch-sensitive surface unit 1204 to receive contacts, a microphone unit 1206 to detect audio input, communication unit 1208 to provide send and receive information using one or more data transmission mechanisms, and processing 1210 coupled to the above-described units. In some embodiments, processing unit 1210 includes display enabling unit 1212, audio receiving unit 1214, virtual assistant unit 1216, and token management unit 1218.

[0252] In some embodiments, processing unit 1210 is configured to: receive (e.g., with audio receiving unit 1214) audio input via a microphone (e.g., with microphone unit 1206), where the audio input comprises a request; send (e.g., using communication unit 1208) data representing the request to a service, such as virtual assistant service 830 described above with reference to FIG. 8F; receive (e.g., using communication unit 1208 and token management unit 1218) a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and send (e.g., using communication unit 1208 and token management unit 1218) the token to a second electronic device external to the device 1200.

[0253] In some embodiments, processing unit 1210 is configured to: receive (e.g., using communication unit 1208), from the service, a result responsive to the request; and display (e.g., with display enabling unit 1212) at least part of the result on display unit 1202.

[0254] In some embodiments, processing unit 1210 is configured to: determine (e.g., with virtual assistant unit 1216) whether a virtual assistant session is actively running; and in accordance with a determination that a virtual assistant session is actively running device 1200, send (e.g., with token management unit 1218 and communication unit 1208) the token to the second electronic device, and in accordance with a determination that no virtual assistant session is actively running on the electronic device, foregoing sending the token to the second electronic device.

[0255] In some embodiments, processing unit 1210 is configured to: in accordance with sending the token to the second electronic device, broadcasting (e.g., with communication unit 1208) the token on an electronic communication channel. In some embodiments, processing unit 1210 is configured to: in accordance with sending the token to the second electronic device, sending (e.g., with communication unit 1208) the token as part of a

transmission addressed to the second electronic device on an electronic communication channel.

[0256] In some embodiments, processing unit 1210 is configured to consider a virtual assistant session is active on device 1200 when the device 1200 is sampling (e.g., with audio receiving unit 1214 and microphone unit 1206) audio input.

[0257] In some embodiments, processing unit 1210 is configured to consider a virtual assistant session is active on device 1210 when a backlight of the display of the device (e.g., display unit 1202) is on.

[0258] In some embodiments, processing unit 1210 is configured to: in accordance with receiving (e.g., with audio receiving unit 1214 and microphone unit 1206) the request, send (e.g., with virtual assistant unit 1216) to the service context information related to a context of device 1200. Note that the token received from the service permits subsequent lookup of the context information from the service.

[0259] In some embodiments, an external device in possession of the token is configured to: send, to the service, the token; and receive, from the service, at least one of: the request, and result responsive to the request. Such an external device, in some embodiments, uses a token management unit corresponding and communication unit similar to units 1218 and 1208 of device 1200 to obtain results in this way. In some embodiments, the external device is configured to display the received results using a display enabling unit and a display unit similar to units 1212 and 1202.

[0260] In some embodiments, processing unit 1210 is configured to consider the audio input received as a first audio input comprising a first request, and: receive (e.g., with audio receiving unit 1214) a second audio input via the microphone (e.g., microphone unit 1206), the second audio input comprising a second request; send (e.g., with communication unit 1208) data representing the second request to the service; and receive (e.g., with communication unit 1208), from the service, a message corresponding to the second request, but no token corresponding to the second request. For example, the message is a generic message indicating a lack of available information identified by the service responsive to the second request. For example, the second request comprises a request for a task unsupported by device 1200.

[0261] The operations described above with respect to FIG. 10 are, optionally, implemented by components depicted in FIGS. 1, 2, or 12. For example, audio receiving operation 1002 is optionally implemented by audio circuitry 210, and token management operations at blocks 1006 and 1008 are optionally implemented by processor(s) 220. It would be clear to a person of ordinary skill in the art how other processes can be implemented based on the components depicted in FIGS. 1, 2, or 12.

[0262] It is understood by persons of skill in the art that the functional blocks described in FIG. 12 are, optionally, combined or separated into sub-blocks to implement the principles of the various described embodiments. Therefore, the description herein optionally supports any possible combination or separation or further definition of the functional blocks described herein. For example, units 1202-1208 can have associated “controller” units that are operatively coupled with the respective unit and processing unit 1210 to enable operation. These controller units are not separately illustrated in FIG. 12 but are understood to be within the grasp of one of ordinary skill in the art who is designing a device having units 1202-1208 such as device 1200. The description herein thus optionally supports combination, separation, and/or further definition of the functional blocks described herein.

[0263] In accordance with some embodiments, FIG. 13 shows a functional block diagram of an electronic device 1300 configured in accordance with the principles of the various described embodiments, including those described with reference to FIGS. 9 and 11. The functional blocks of the device are, optionally, implemented by hardware, software, or a combination of hardware and software to carry out the principles of the various described embodiments. It is understood by persons of skill in the art that the functional blocks described in FIG. 13 are, optionally, combined or separated into sub-blocks to implement the principles of the various described embodiments. Therefore, the description herein optionally supports any possible combination or separation or further definition of the functional blocks described herein.

[0264] As shown in FIG. 13, an electronic device 1300 includes a display unit 1302 configured to display a graphical user interface, optional touch-sensitive surface unit 1304 to receive contacts, at least one of microphone unit 1306 and input device unit 1308 to detect user input, communication unit 1310 to provide send and receive information using one or more data transmission mechanisms, and processing 1312 coupled to the above-described units. In some embodiments, processing unit 1312 includes display enabling unit 1314, input

receiving unit 1316, virtual assistant unit 1318, web browser unit 1320, and URL management unit 1322.

[0265] In some embodiments, processing unit 1312 is configured to: receive (e.g., with input detecting unit 1116) input comprising a request; send (e.g., using communication unit 1310) data representing the request to a service; receive (e.g., using communication unit 1310) a uniform resource locator (URL) from the service identifying a resource; and send (e.g., using communication unit 1310) at least part of the URL (e.g., managed by URL management unit 1322) to a second electronic device that is external to device 1300.

[0266] In some embodiments, processing unit 1312 is configured to: enable display (e.g., with display enabling unit 1314), on the display (e.g., display unit 1302) of the electronic device, an indication indicating the availability of information responsive to the request on the second electronic device.

[0267] In some embodiments, processing unit 1312 is configured to: obtain (e.g., using communication unit 1310) a web page via at least part of the URL; and display (e.g., with display enabling unit 1314) at least a subset of the web page on a display (e.g., display unit 1302) of device 1300.

[0268] In some embodiments, processing unit 1312 is configured to: in accordance with a determination that the input comprising the request was received while a web browser is actively running on device 1300 (e.g., with web browsing unit 1320), sending an indication (e.g., with communication unit 1310), from device 1300 to the service, that the input comprising the request was received while a web browser is actively running on the electronic device (e.g., with web browsing unit 1320); and in accordance with a determination that the input comprising the request was not received while the web browser is actively running on the electronic device (e.g., with web browsing unit 1320), foregoing sending the indication from the electronic device to the service.

[0269] In some embodiments, the request includes instructions from the user to perform a web search. In some embodiments, processing unit 1312 is configured to: in accordance with receiving input comprising a request (e.g., with input detection unit 1316), receive audio input from a microphone (e.g., microphone unit 1306) of the electronic device. In some embodiments, the request is part of a textual input received via input device unit 1308.

[0270] In some embodiments, processing unit 1312 is configured to: determine (e.g., with virtual assistant unit 1318) whether a virtual assistant session is actively running on device 1300; and in accordance with a determination that the virtual assistant session is actively running on device 1300 (e.g., with virtual assistant unit 1318), send (e.g., with communication unit 1310) at least part of the URL (e.g., maintained by URL management unit 1322) to the second electronic device, and in accordance with a determination that no virtual assistant session is actively running on the electronic device (e.g., with virtual assistant unit 1318), foregoing sending the URL to the second electronic device.

[0271] In some embodiments, processing unit 1312 is configured to: in accordance with sending (e.g., with communication unit 1310) the at least part of the URL to the second electronic device, broadcasting (e.g., with communication unit 1310) the at least part of the URL on an electronic communication channel. In some embodiments, processing unit 1312 is configured to: in accordance with sending (e.g., with communication unit 1310) the at least part of the URL to the second electronic device, sending (e.g., with communication unit 1310) the at least part of the URL (e.g., maintained by URL management unit 1322) as part of a transmission addressed to the second electronic device on an electronic communication channel.

[0272] In some embodiments, processing unit 1312 is configured to consider: a virtual assistant session is active on device 1300 when the electronic device is sampling audio input (e.g., with input detection unit 1316 and microphone 1306). In some embodiments, processing unit 1312 is configured to consider a virtual assistant session is active on device 1300 when a backlight of display unit 1302 is on.

[0273] In some embodiments, processing unit 1312 is configured to consider the audio input received as a first audio input comprising a first request, and: receive (e.g., with input detecting unit 1314) a second audio input via the microphone (e.g., microphone unit 1206), the second audio input comprising a second request; send (e.g., with communication unit 1310) data representing the second request to the service; and receive (e.g., with communication unit 1310), from the service, a URL corresponding to the second request, but no token corresponding to the second request. For example, the message is a generic message indicating a lack of available information identified by the service responsive to the second request. For example, the second request comprises a request for a task unsupported by device 1300.

[0274] The operations described above with respect to FIG. 11, optionally, implemented by components depicted in FIGS. 1, 2, or 13. For example, audio receiving operation 1002 is optionally implemented by audio circuitry 210, and token management operations at blocks 1106 and 1108 are optionally implemented by processor(s) 220. It would be clear to a person of ordinary skill in the art how other processes can be implemented based on the components depicted in FIGS. 1, 2, or 13.

[0275] It is understood by persons of skill in the art that the functional blocks described in FIG. 13 are, optionally, combined or separated into sub-blocks to implement the principles of the various described embodiments. Therefore, the description herein optionally supports any possible combination or separation or further definition of the functional blocks described herein. For example, units 1302-1310 can have associated “controller” units that are operatively coupled with the respective unit and processing unit 1312 to enable operation. These controller units are not separately illustrated in FIG. 13 but are understood to be within the grasp of one of ordinary skill in the art who is designing a device having units 1302-1310 such as device 1300. The description herein thus optionally supports combination, separation, and/or further definition of the functional blocks described herein.

[0276] Although the disclosure and examples have been fully described with reference to the accompanying figures, it is to be noted that various changes and modifications will become apparent to those skilled in the art. Such changes and modifications are to be understood as being included within the scope of the disclosure and examples as defined by the appended claims.

CLAIMS

What is claimed is:

1. A method, comprising:

at a first electronic device with a display and a microphone:

receiving audio input via the microphone, wherein the audio input comprises a request;

sending data representing the request to a service;

receiving a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and

sending the token to a second electronic device external to the first electronic device.

2. The method of claim 1, further comprising:

at the first electronic device:

receiving, from the service, a result responsive to the request; and

displaying at least part of the result on the display of the first electronic device.

3. The method of any one of claims 1 and 2, further comprising:

determining whether a virtual assistant session is actively running on the first electronic device; and

in accordance with a determination that a virtual assistant session is actively running on the first electronic device, sending the token to the second electronic device, and

in accordance with a determination that no virtual assistant session is actively running on the first electronic device, foregoing sending the token to the second electronic device.

4. The method of any one of claims 1-3, further comprising:

in accordance with sending the token to the second electronic device, broadcasting the token on an electronic communication channel.

5. The method of any one of claims 1-3, further comprising:

in accordance with sending the token to the second electronic device, sending the token as part of a transmission addressed to the second electronic device on an electronic communication channel.

6. The method of any one of claims 3-5, wherein a virtual assistant session is active on the first electronic device when the first electronic device is sampling audio input.

7. The method of any one of claims 3-5, wherein a virtual assistant session is active on the first electronic device when a backlight of the display of the first electronic device is on.

8. The method of any one of claims 1-7, further comprising:

in accordance with receiving the request at the first electronic device:

sending, from the first electronic device to the service, context information related to a context of the first electronic device,

wherein the token received from the service permits lookup, by the second electronic device, of the context information, from the service.

9. The method of any one of claims 1-8, further comprising:

at the second electronic device:

sending, to the service, the token; and

receiving, from the service, at least one of: the request, and result responsive to the request.

10. The method of claim 9, further comprising:

at the second electronic device:

receiving, from the service, result responsive to the request; and

displaying at least a part of the received result.

11. The method of any one of claims 1-10, wherein the audio input received at the first electronic device is a first audio input comprising a first request, the method further comprising:

at the first electronic device, receiving a second audio input via the microphone, the second audio input comprising a second request;

sending data representing the second request to the service; and

receiving, from the service, a message corresponding to the second request, but no token corresponding to the second request.

12. The method of claim 11, wherein the message is a generic message.

13. The method of claim 11, wherein the message corresponding to the second request indicates a lack of available information identified by the service responsive to the second request.

14. The method of claim 11, wherein the second request comprises a request for a task unsupported by the first electronic device.

15. A method, comprising:

at a first electronic device:

receiving input comprising a request;

sending data representing the request to a service;

receiving a uniform resource locator (URL) from the service identifying a resource; and

sending at least part of the URL to a second electronic device external to the first electronic device.

16. The method of claim 15, further comprising:

displaying, on the display of the electronic device, an indication indicating the availability of information responsive to the request on the second electronic device.

17. The method of claim 15, further comprising:

obtaining a web page via at least part of the URL; and

displaying at least a subset of the web page on a display of the first electronic device.

18. The method of any one of claims 15-17, further comprising:

in accordance with a determination that the input comprising the request was received while a web browser is actively running on the first electronic device, sending an indication, from the first electronic device to the service, that the input comprising the request was received while a web browser is actively running on the first electronic device; and

in accordance with a determination that the input comprising the request was not received while the web browser is actively running on the first electronic device, foregoing sending the indication from the first electronic device to the service.

19. The method of any one of claims 15-18, wherein the request includes instructions from the user to perform a web search.

20. The method of any one of claims 15-19, further comprising:

in accordance with receiving input comprising a request, receiving audio input from a microphone of the first electronic device.

21. The method of any one of claims 15-20, further comprising:

determining whether a virtual assistant session is actively running on the first electronic device; and

in accordance with a determination that the virtual assistant session is actively running on the first electronic device, sending at least part of the URL to the second electronic device, and

in accordance with a determination that no virtual assistant session is actively running on the first electronic device, foregoing sending the URL to the second electronic device.

22. The method of any one of claims 15-21, further comprising:

in accordance with sending the at least part of the URL to the second electronic device, broadcasting the at least part of the URL on an electronic communication channel.

23. The method of any one of claims 15-22, further comprising:

in accordance with sending the at least part of the URL to the second electronic device, sending the at least part of the URL as part of a transmission addressed to the second electronic device on an electronic communication channel.

24. The method of any one of claims 20-23, wherein a virtual assistant session is active on the first electronic device when the first electronic device is sampling audio input.

25. The method of any one of claims 20-24, wherein a virtual assistant session is active on the first electronic device when a backlight of the display of the first electronic device is on.

26. The method of any one of claims 15-25, wherein the input received at the first electronic device is a first input comprising a first request, the method further comprising:

at the first electronic device, receiving a second input, the second input comprising a second request;

sending data representing the second request to the service; and

receiving, from the service, a message corresponding to the second request, but no URL corresponding to the second request.

27. The method of claim 26, wherein the message is a generic message.

28. The method of claim 27, wherein the message corresponding to the second request indicates a lack of available information identified by the service responsive to the second request.

29. The method of claim 27, wherein the second request comprises a request for a task unsupported by the first electronic device.

30. A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first electronic device with a display, a microphone, cause the first electronic device to:

receive audio input via the microphone, wherein the audio input comprises a request;

send data representing the request to a service;

receive a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and

send the token to a second electronic device external to the first electronic device.

31. A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first electronic device with a display, a microphone, and an input device, cause the first electronic device to:

perform the methods of any one of claims 1-14.

32. A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first electronic device with a display, cause the first electronic device to:

receive input comprising a request;

send data representing the request to a service;

receive a uniform resource locator (URL) from the service identifying a resource; and

send at least part of the URL to a second electronic device external to the first electronic device.

33. A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first electronic device with a display, cause the first electronic device to:

perform the methods of any one of claims 15-29.

34. An electronic device comprising:

a display;

one or more processors;

a memory;

a microphone; and

one or more programs, wherein the one or more programs are stored in memory and configured to be executed by the one or more processors, the one or more programs including instructions for:

receiving audio input via the microphone, wherein the audio input comprises a request;

sending data representing the request to a service;

receiving a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and

sending the token to a second electronic device external to the electronic device.

35. An electronic device comprising:

a display;

one or more processors;

a memory;

a microphone; and

one or more programs, wherein the one or more programs are stored in memory and configured to be executed by the one or more processors, the one or more programs including instructions for perform the methods of any one of claims 1-14.

36. An electronic device comprising:

a display;

one or more processors;

a memory; and

one or more programs, wherein the one or more programs are stored in memory and configured to be executed by the one or more processors, the one or more programs including instructions for:

receiving input comprising a request;

sending data representing the request to a service;

receiving a uniform resource locator (URL) from the service identifying a resource; and

sending at least part of the URL to a second electronic device external to the electronic device.

37. An electronic device comprising:

a display;

one or more processors;

a memory; and

one or more programs, wherein the one or more programs are stored in memory and configured to be executed by the one or more processors, the one or more programs including instructions for perform the methods of any one of claims 15-29.

38. An electronic device comprising:

means for receiving audio input via the microphone, wherein the audio input comprises a request;

means for sending data representing the request to a service;

means for receiving a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and

means for sending the token to a second electronic device external to the electronic device.

39. An electronic device comprising:

means for performing the methods of any one of claims 1-14.

40. An electronic device comprising:

means for receiving input comprising a request;

means for sending data representing the request to a service;

means for receiving a uniform resource locator (URL) from the service identifying a resource; and

means for sending at least part of the URL to a second electronic device external to the electronic device.

41. An electronic device comprising:

means for performing the methods of any one of claims 15-29.

42. An electronic device comprising:

a display; a microphone; and

a processing unit coupled to the display, the microphone, and the processing unit configured to:

receive audio input via the microphone, wherein the audio input comprises a request;

send data representing the request to a service;

receive a token from the service, wherein the token permits lookup, from the service, of at least one of: the request, and result responsive to the request; and

send the token to a second electronic device external to the electronic device.

43. The electronic device of claim 42, wherein the processing unit is further configured to:

receive, from the service, a result responsive to the request; and

enable a display at least part of the result on the display of the electronic device.

44. The electronic device of any one of claims 42 and 43, wherein the processing unit is further configured to:

determine whether a virtual assistant session is actively running on the electronic device; and

in accordance with a determination that a virtual assistant session is actively running on the electronic device, send the token to the second electronic device, and

in accordance with a determination that no virtual assistant session is actively running on the electronic device, forego sending the token to the second electronic device.

45. The electronic device of any one of claims 42-44, wherein the processing unit is further configured to:

in accordance with sending the token to the second electronic device, broadcast the token on an electronic communication channel.

46. The electronic device of any one of claims 42-44, wherein the processing unit is further configured to:

in accordance with sending the token to the second electronic device, send the token as part of a transmission addressed to the second electronic device on an electronic communication channel.

47. The electronic device of any one of claims 44-46, wherein a virtual assistant session is active on the electronic device when the electronic device is sampling audio input.

48. The electronic device of any one of claims 44-46, wherein a virtual assistant session is active on the electronic device when a backlight of the display of the electronic device is on.

49. The electronic device of any one of claims 42-48, wherein the processing unit is further configured to:

in accordance with receiving the request at the electronic device:

sending, from the electronic device to the service, context information related to a context of the electronic device,

wherein the token received from the service permits lookup, by the second electronic device, of the context information, from the service.

50. The electronic device of any one of claims 42-49, wherein the second electronic device is configured to:

send, to the service, the token; and

receive, from the service, at least one of: the request, and result responsive to the request.

51. The electronic device of claim 50, wherein the second electronic device is further configured to:

receive, from the service, result responsive to the request; and

enable display of at least a part of the received result.

52. The electronic device of any one of claims 42-51, wherein the audio input received at the electronic device is a first audio input comprising a first request, wherein the processing unit is further configured to:

receive a second audio input via the microphone, the second audio input comprising a second request;

send data representing the second request to the service; and

receive, from the service, a message corresponding to the second request, but no token corresponding to the second request.

53. The electronic device of claim 52, wherein the message is a generic message.

54. The electronic device of claim 52, wherein the message corresponding to the second request indicates a lack of available information identified by the service responsive to the second request.

55. The electronic device of claim 52, wherein the second request comprises a request for a task unsupported by the electronic device.

56. An electronic device comprising:

a display; and

a processing unit coupled to the display, the processing unit configured to:

receive input comprising a request;

send data representing the request to a service;

receive a uniform resource locator (URL) from the service identifying a resource; and

send at least part of the URL to a second electronic device external to the electronic device.

57. The electronic device of claim 56, wherein the processing unit is further configured to:

enable display, on the display of the electronic device, an indication indicating the availability of information responsive to the request on the second electronic device.

58. The electronic device of claim 57, wherein the processing unit is further configured to:

obtain a web page via at least part of the URL; and

enable display of at least a subset of the web page on a display of the first electronic device.

59. The electronic device of any one of claims 57 and 58, wherein the processing unit is further configured to:

in accordance with a determination that the input comprising the request was received while a web browser is actively running on the electronic device, send an indication, from the electronic device to the service, that the input comprising the request was received while a web browser is actively running on the electronic device; and

in accordance with a determination that the input comprising the request was not received while the web browser is actively running on the electronic device, foregoing sending the indication from the electronic device to the service.

60. The electronic device of any one of claims 57 and 58, wherein the request includes instructions from the user to perform a web search.

61. The electronic device of any one of claims 57-60, wherein the processing unit is further configured to:

in accordance with receiving input comprising a request, receive audio input from a microphone of the electronic device.

62. The electronic device of any one of claims 57-61, wherein the processing unit is further configured to:

determine whether a virtual assistant session is actively running on the first electronic device; and

in accordance with a determination that the virtual assistant session is actively running on the electronic device, sending at least part of the URL to the second electronic device, and

in accordance with a determination that no virtual assistant session is actively running on the electronic device, foregoing sending the URL to the second electronic device.

63. The electronic device of any one of claims 56-62, wherein the processing unit is further configured to:

in accordance with sending the at least part of the URL to the second electronic device, broadcasting the at least part of the URL on an electronic communication channel.

64. The electronic device of any one of claims 56-63, wherein the processing unit is further configured to:

in accordance with sending the at least part of the URL to the second electronic device, sending the at least part of the URL as part of a transmission addressed to the second electronic device on an electronic communication channel.

65. The electronic device of any one of claims 62-64, wherein a virtual assistant session is active on the electronic device when the first electronic device is sampling audio input.

66. The electronic device of any one of claims 62-65, wherein a virtual assistant session is active on the electronic device when a backlight of the display of the first electronic device is on.

67. The electronic device of any one of claims 56-66, wherein the input received at the electronic device is a first input comprising a first request, wherein the processing unit is further configured to:

receive a second input, the second input comprising a second request;

send data representing the second request to the service; and

receive, from the service, a message corresponding to the second request, but no URL corresponding to the second request.

68. The electronic device of claim 67, wherein the message is a generic message.

69. The electronic device of claim 68, wherein the message corresponding to the second request indicates a lack of available information identified by the service responsive to the second request.

70. The electronic device of claim 68, wherein the second request comprises a request for a task unsupported by the first electronic device.

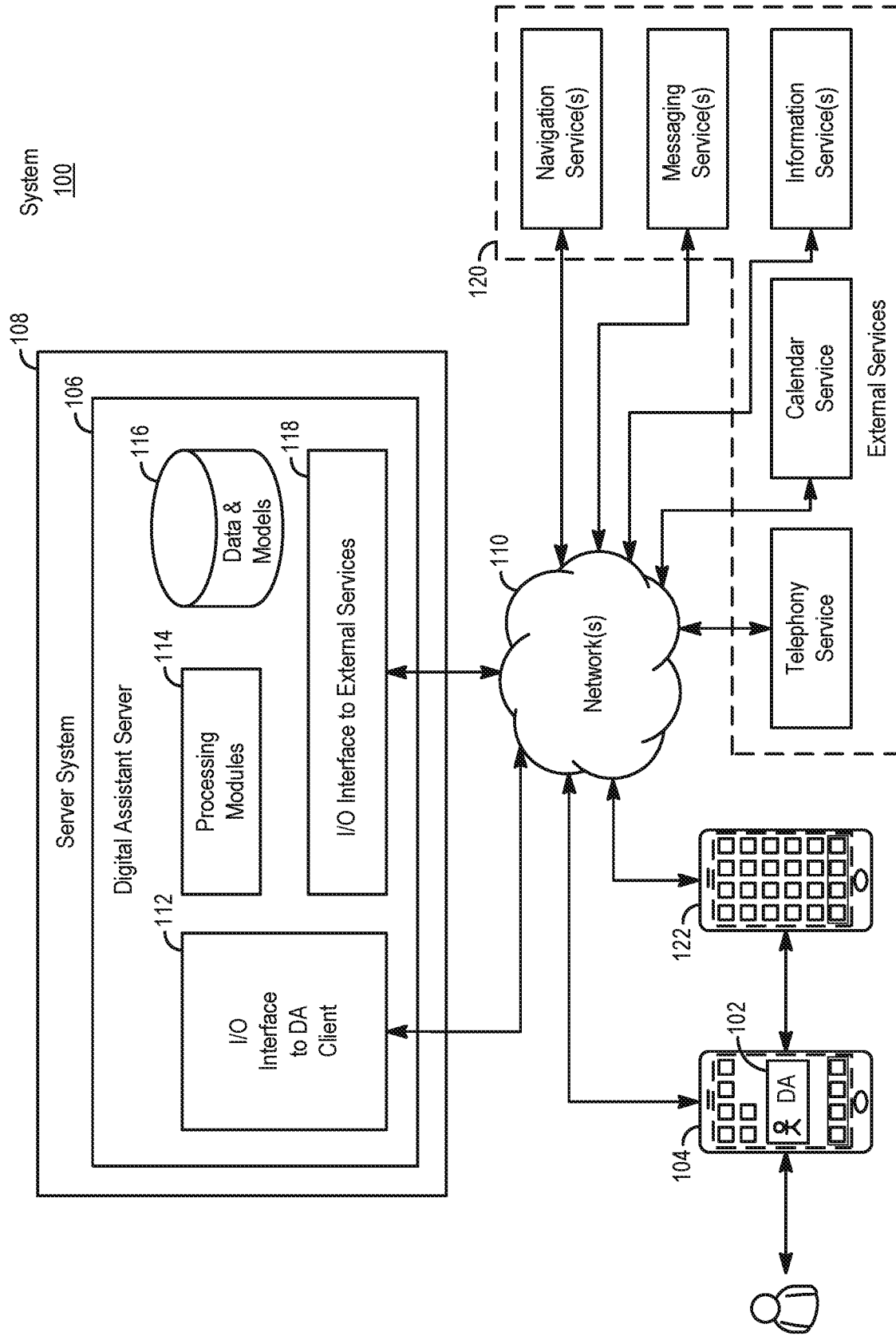


FIG. 1

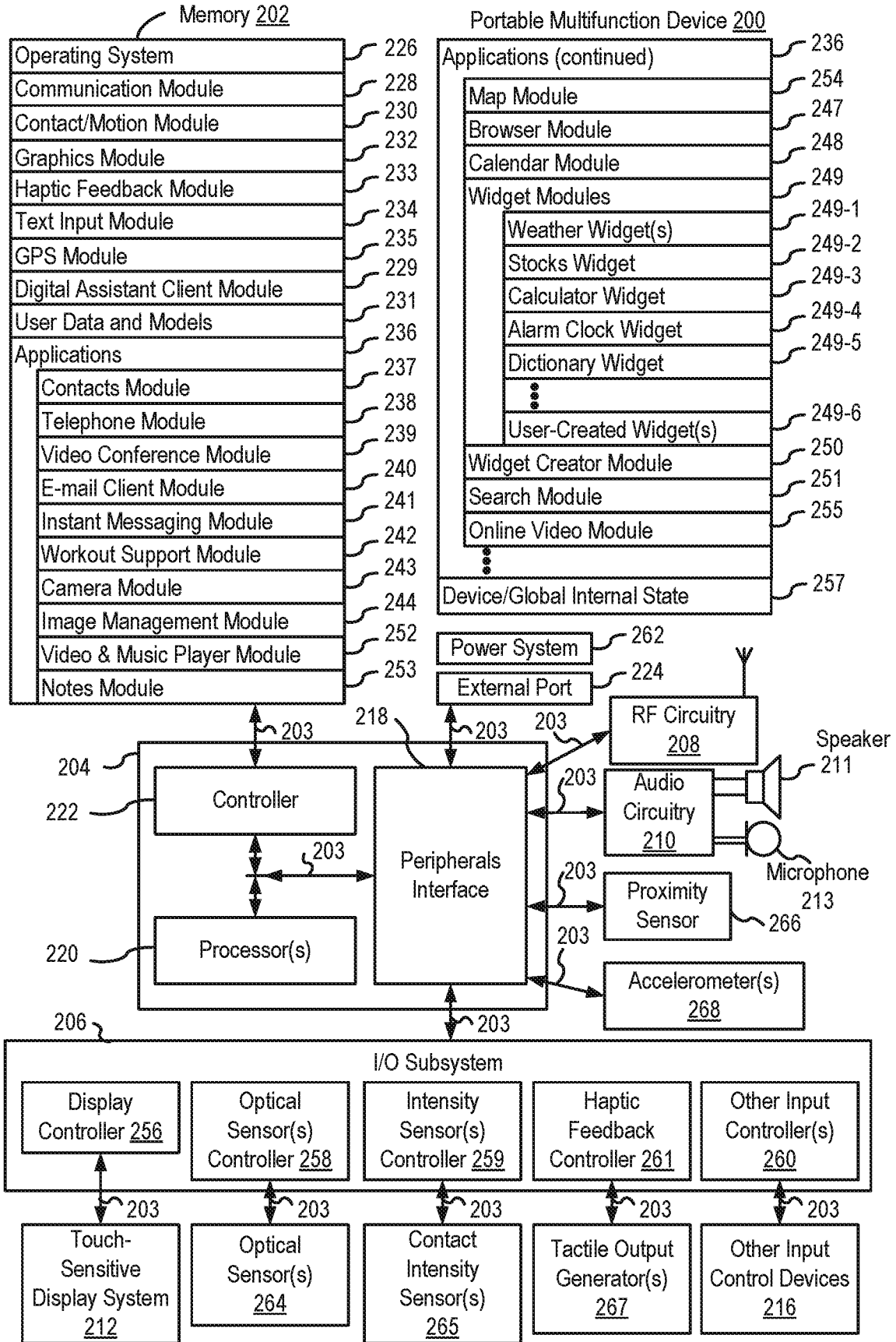


FIG. 2A

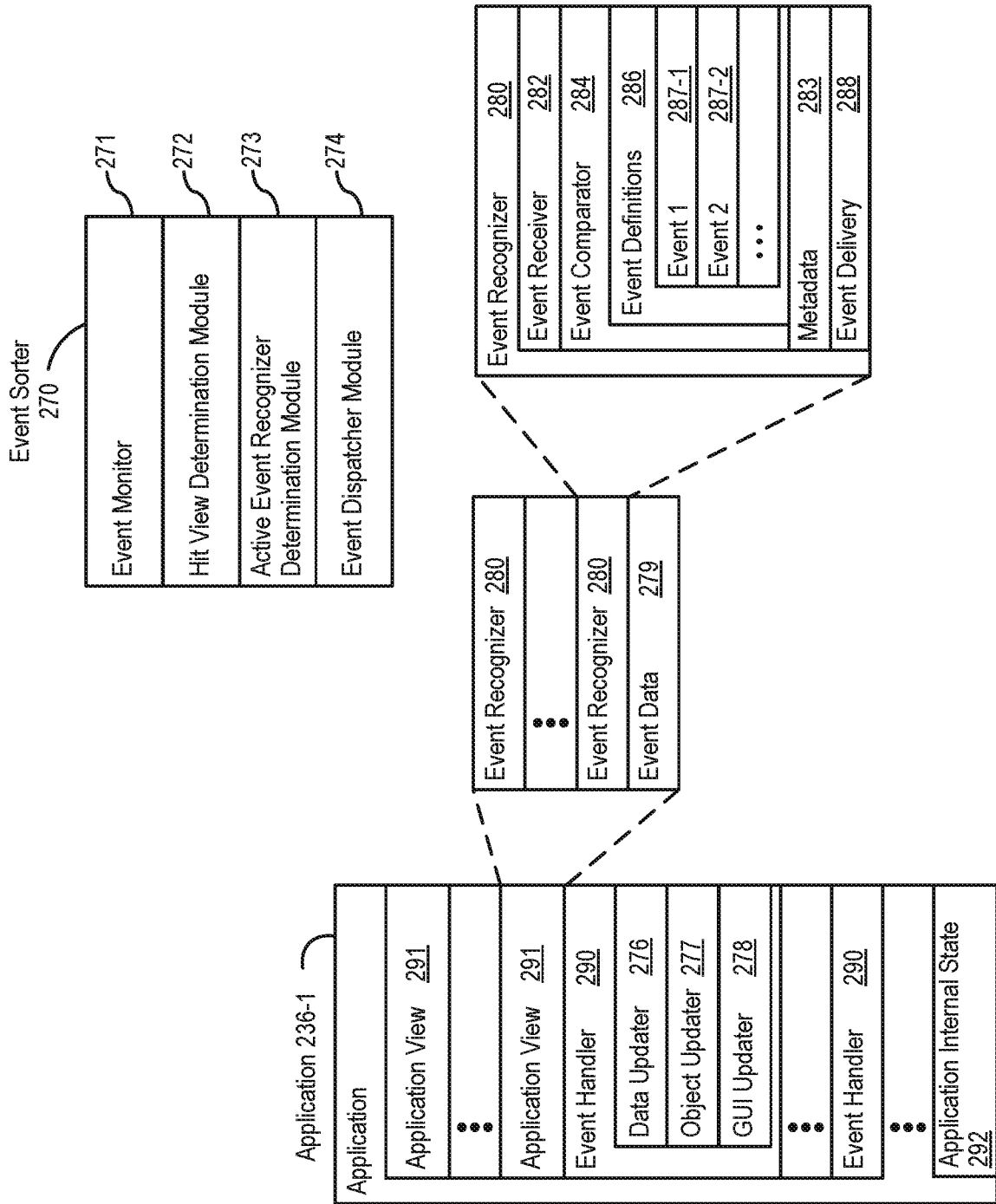


FIG. 2B

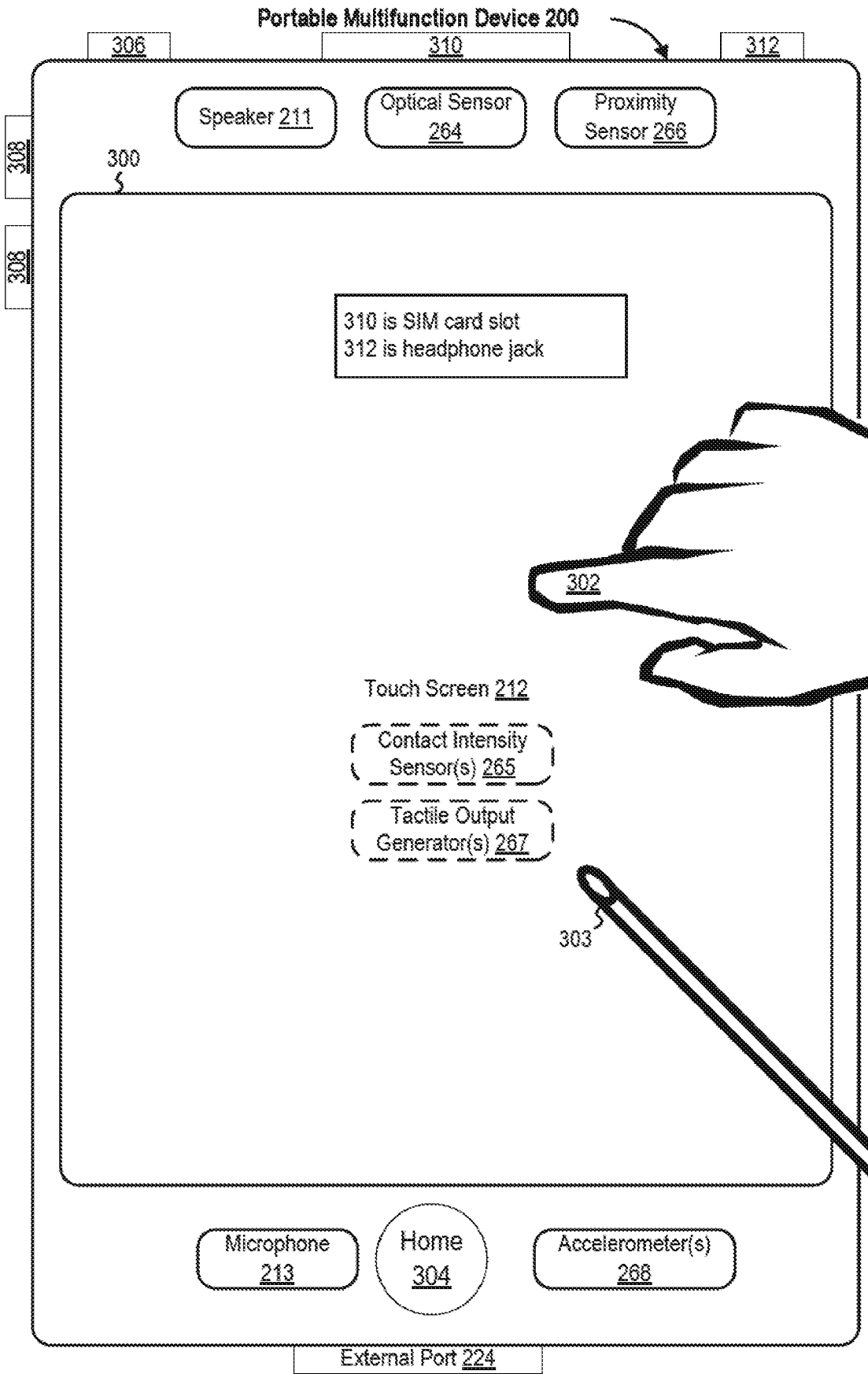


FIG. 3

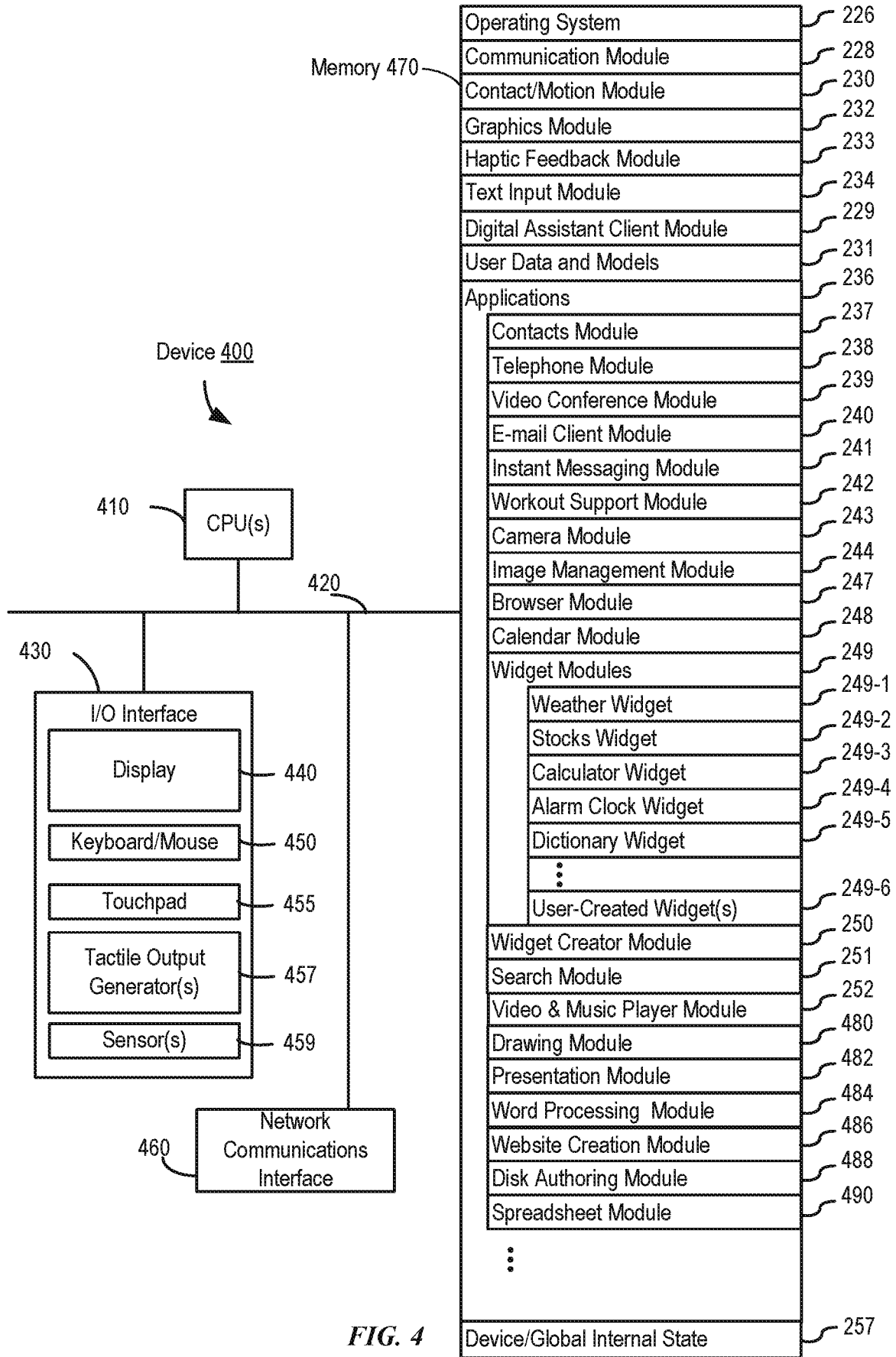


FIG. 4

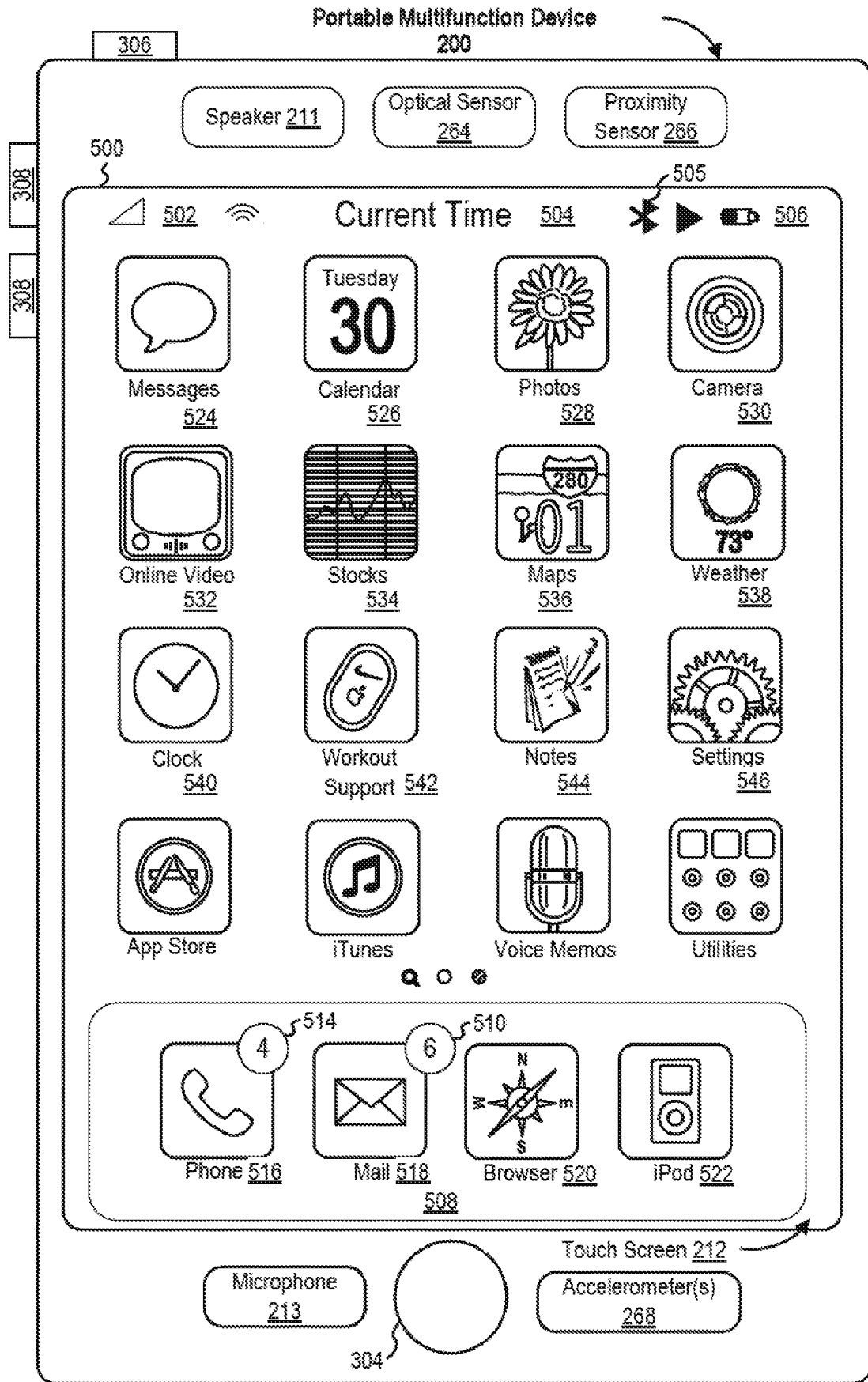


FIG. 5A

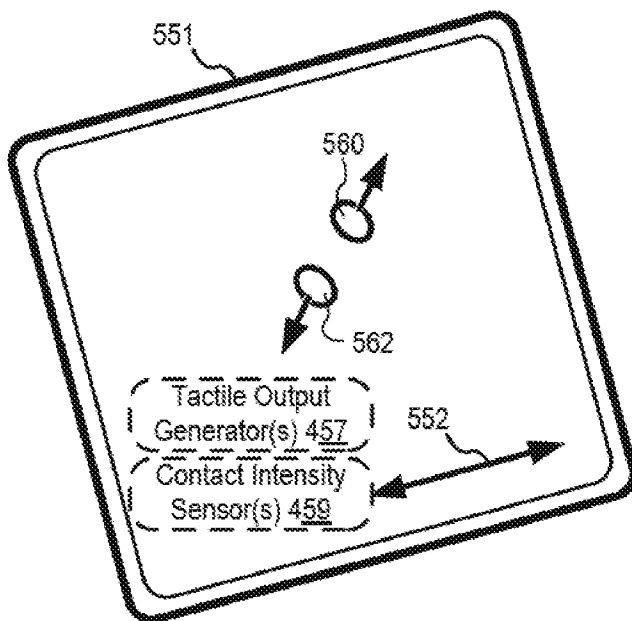
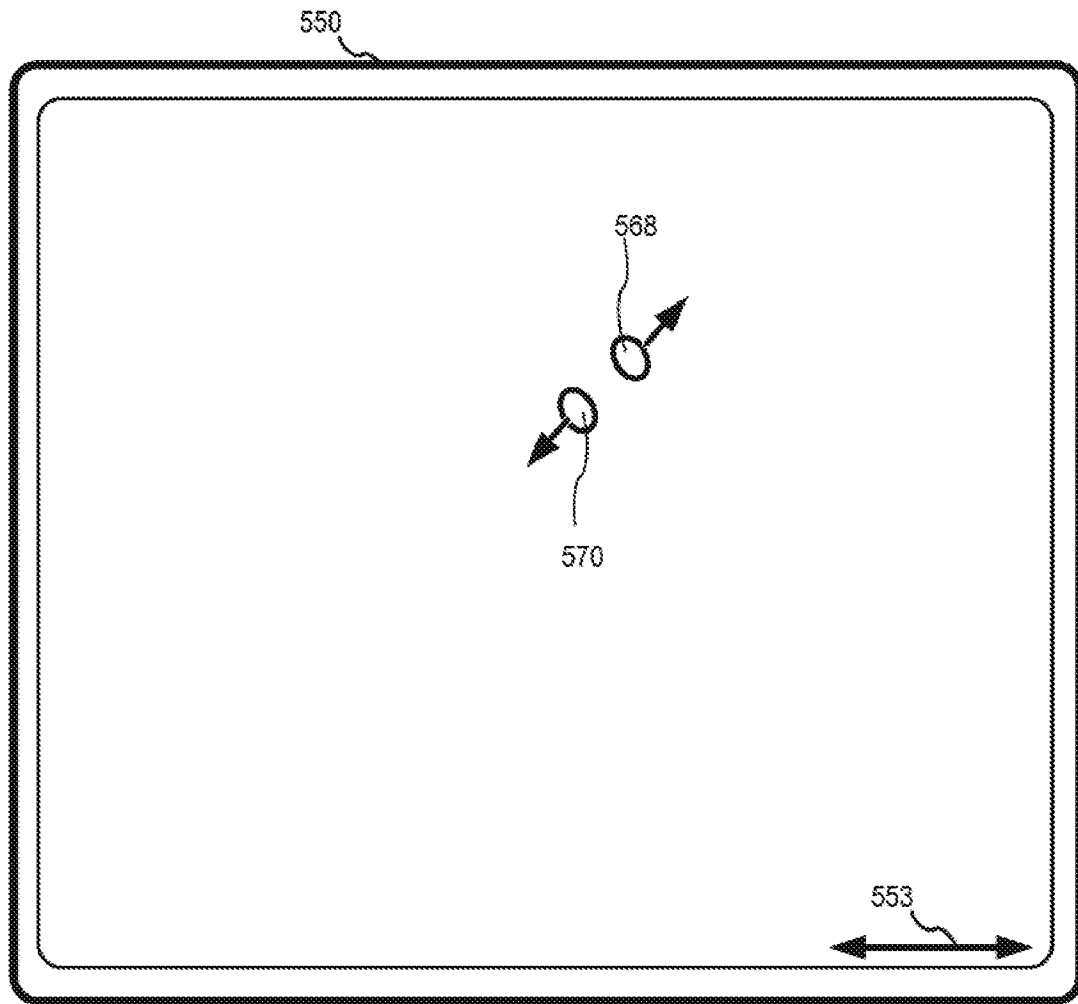
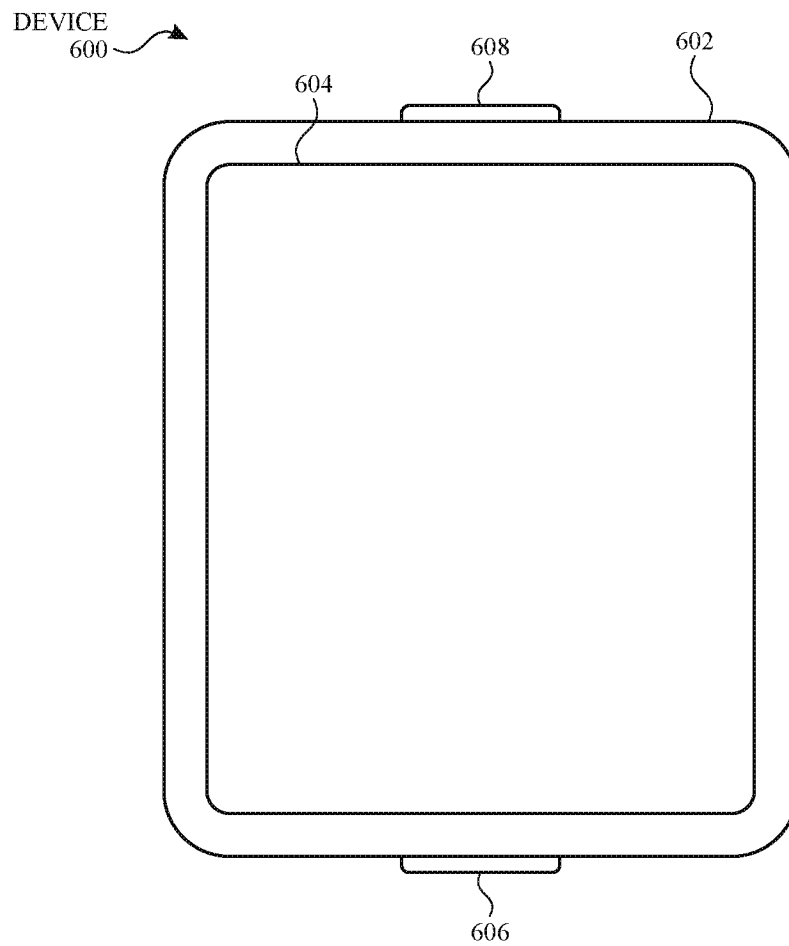


FIG. 5B

**FIG. 6A**

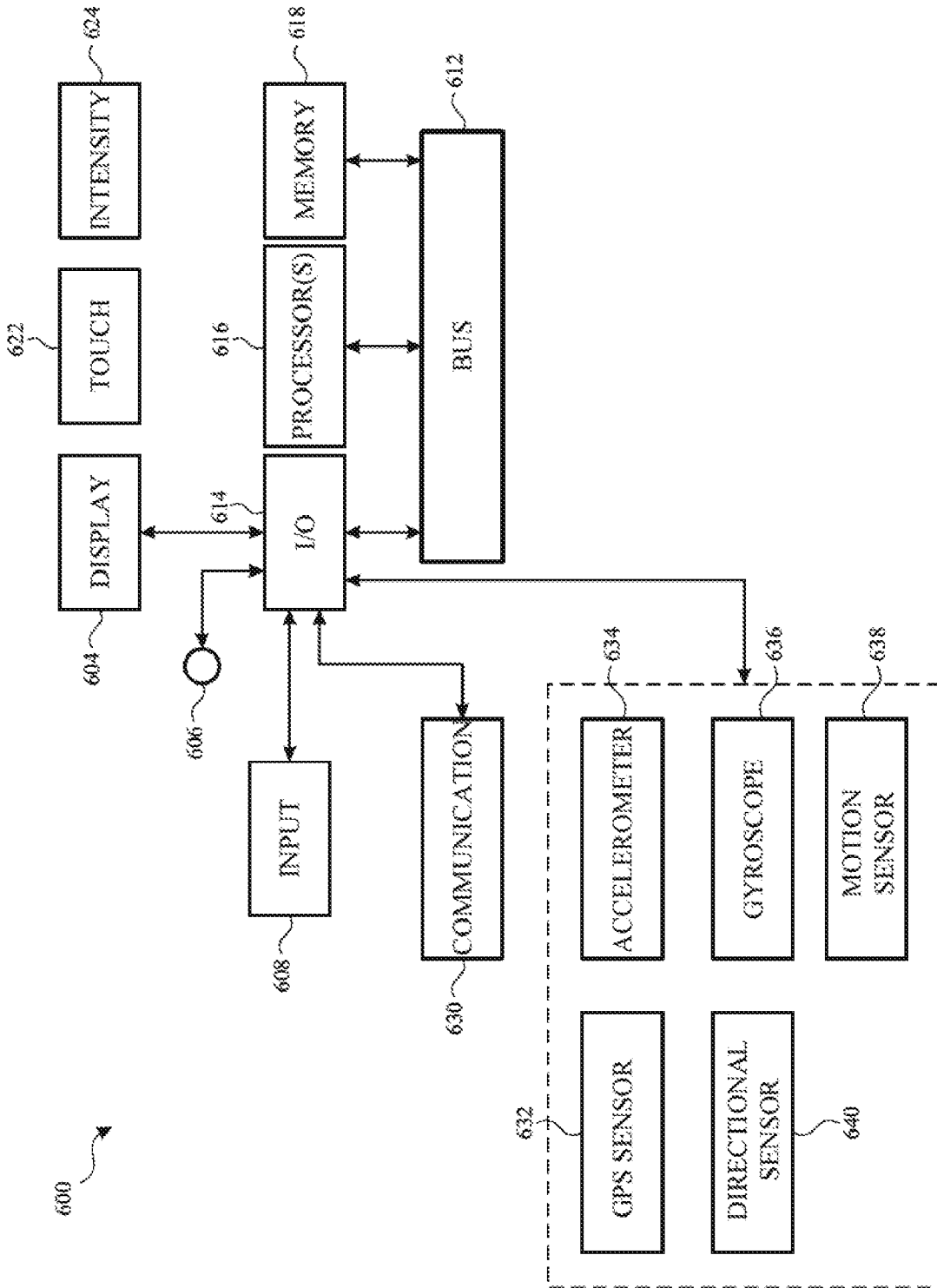


FIG. 6B

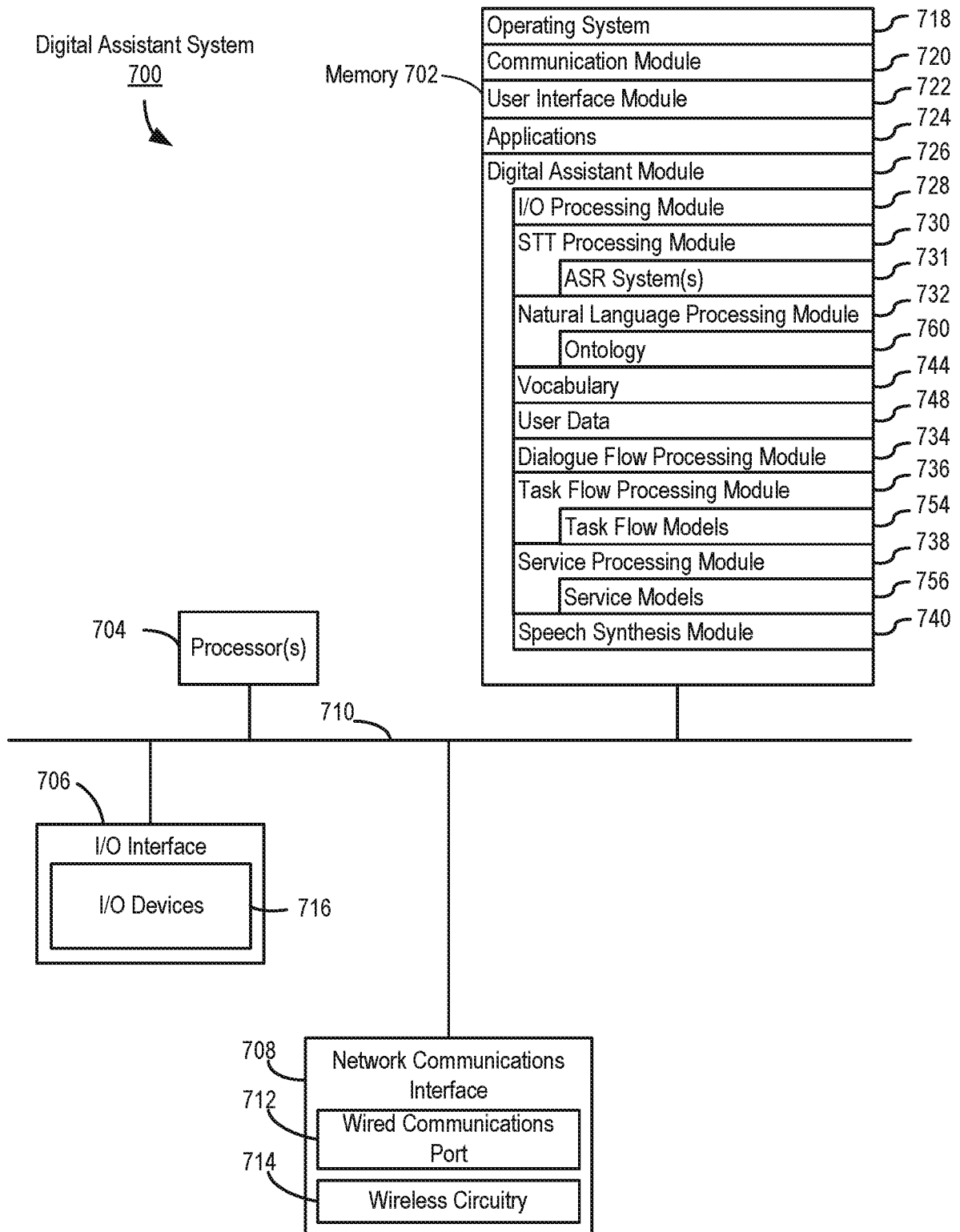


FIG. 7A

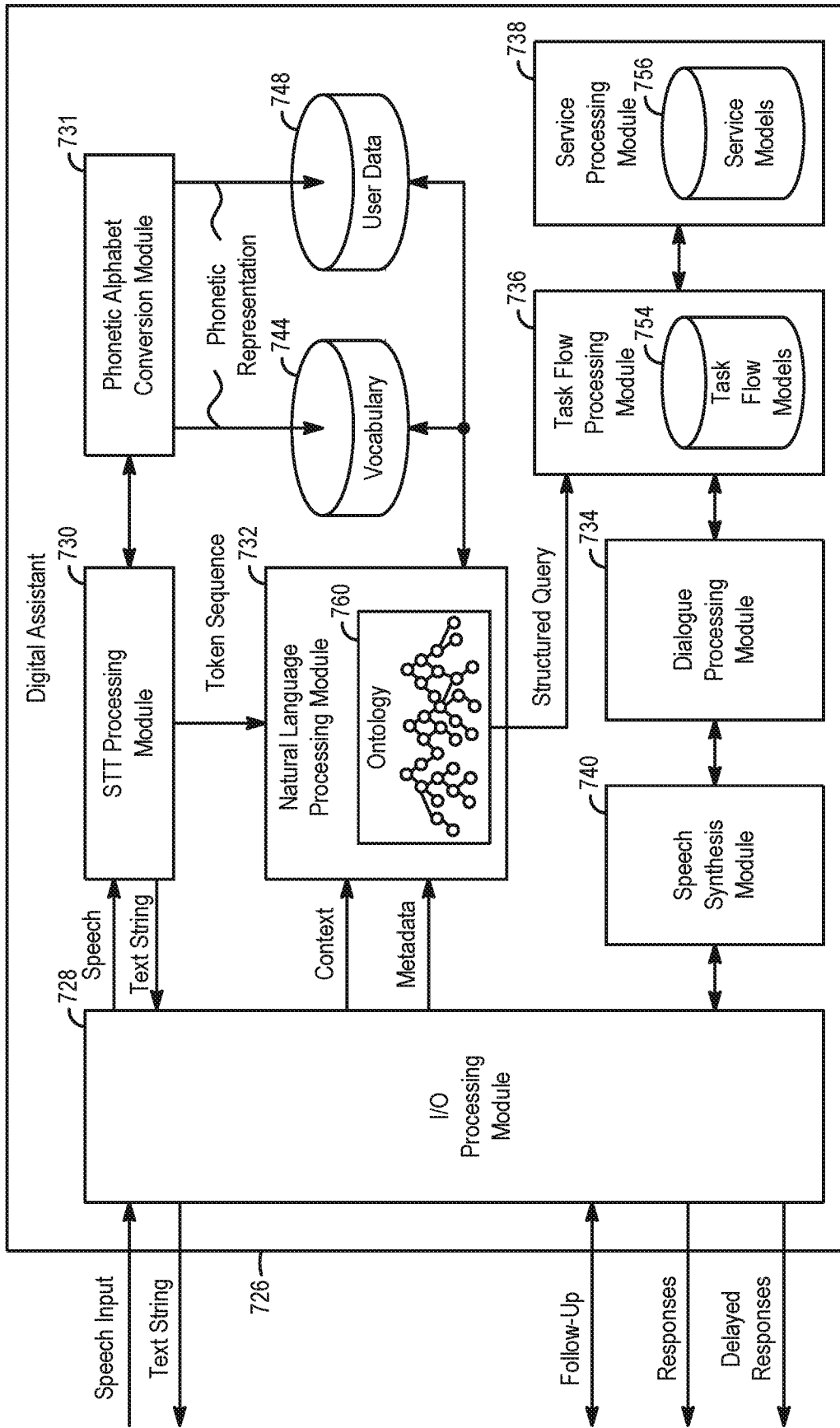


FIG. 7B

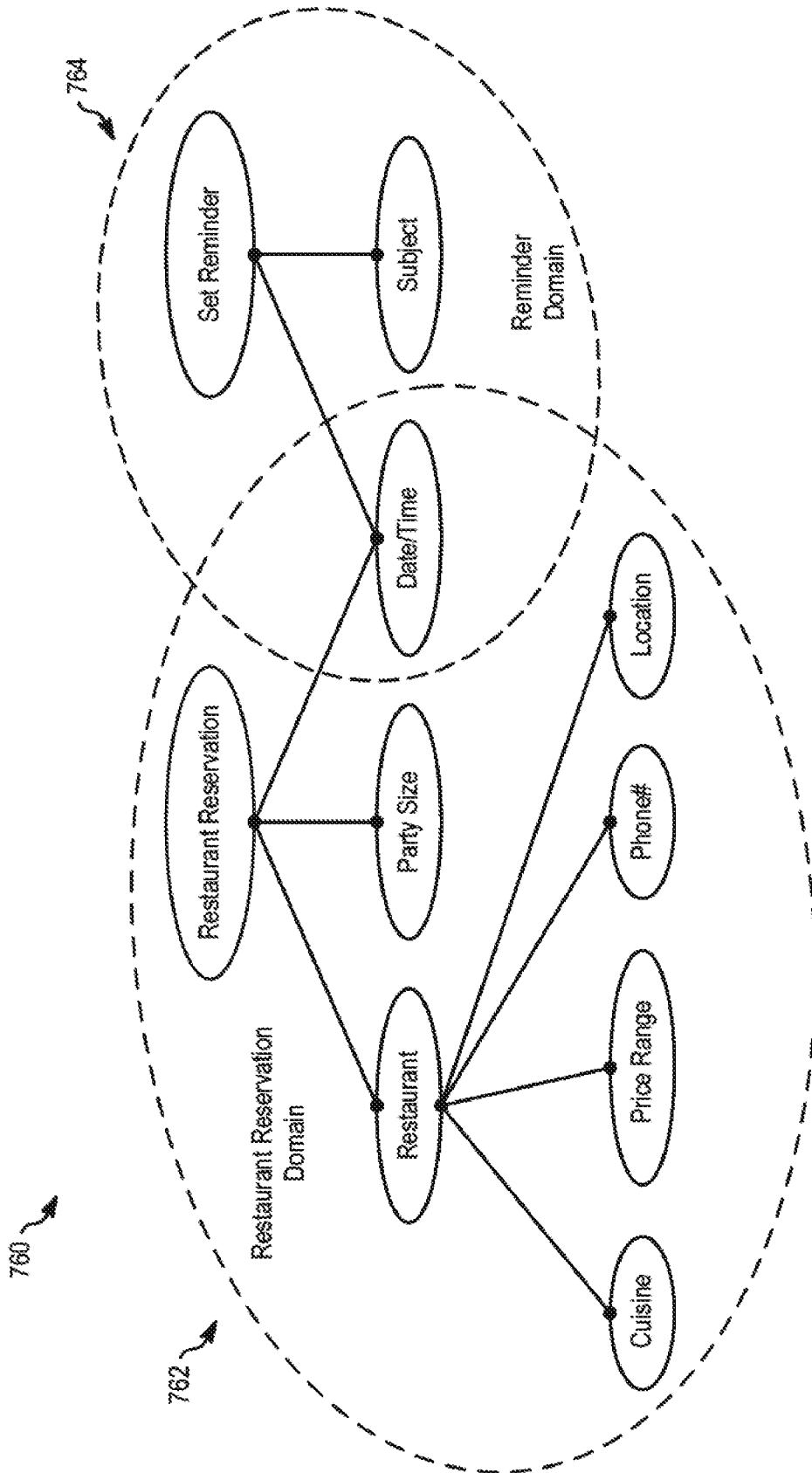


FIG. 7C

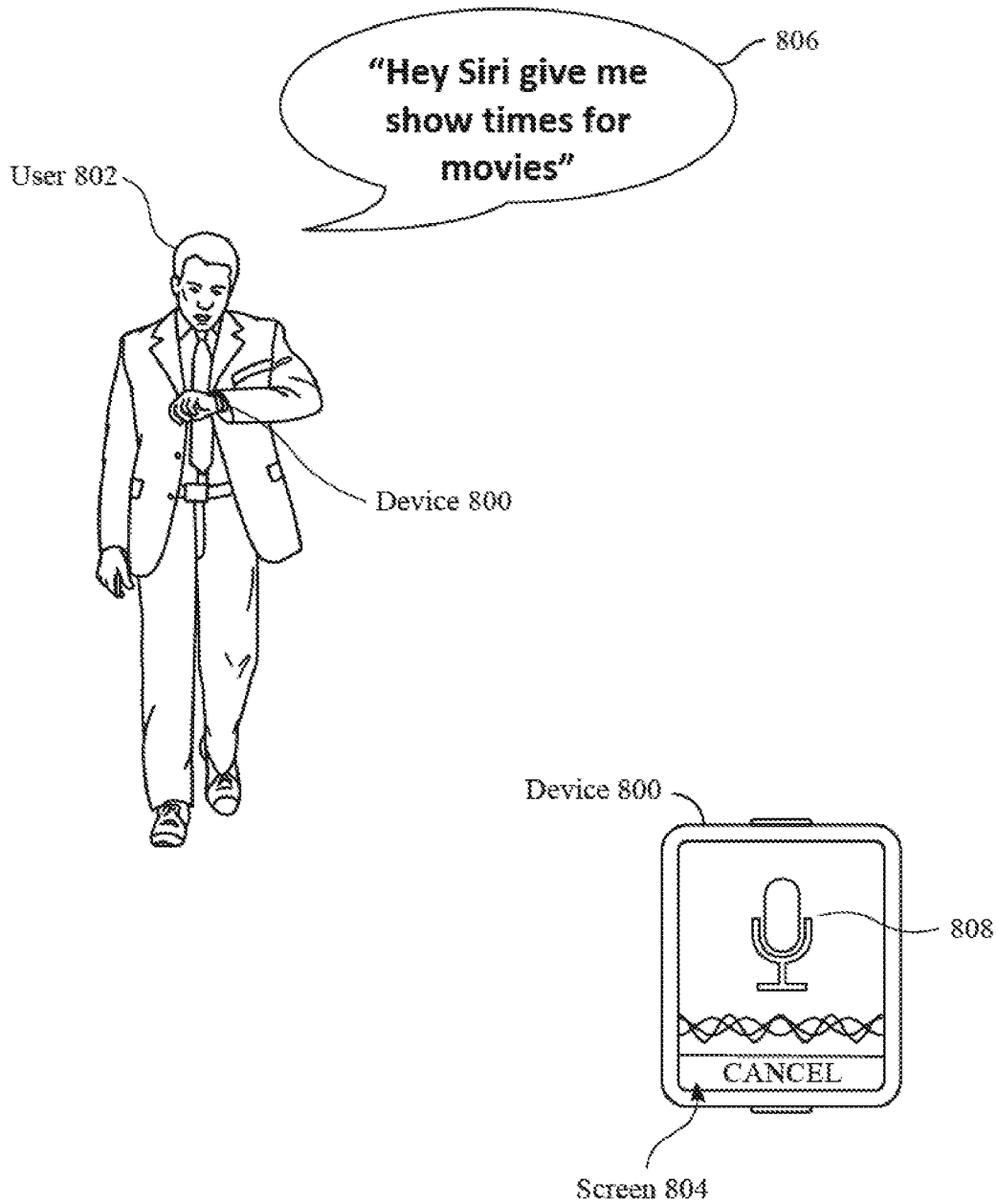


FIG. 8A

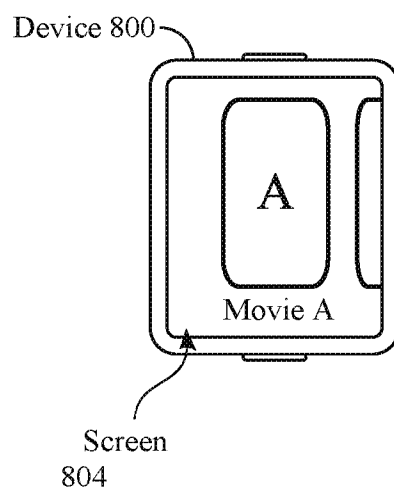


FIG. 8B

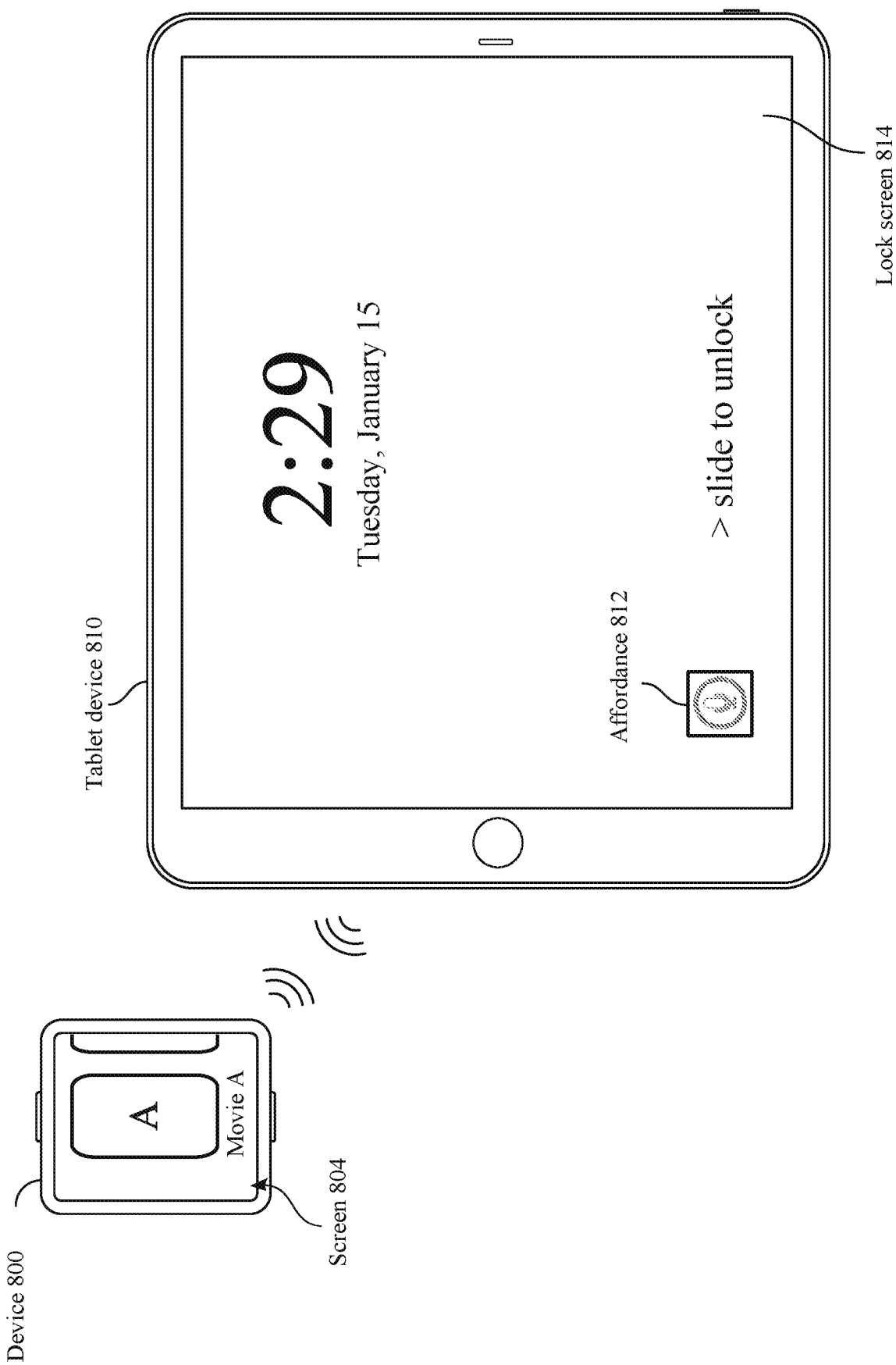


FIG. 8C

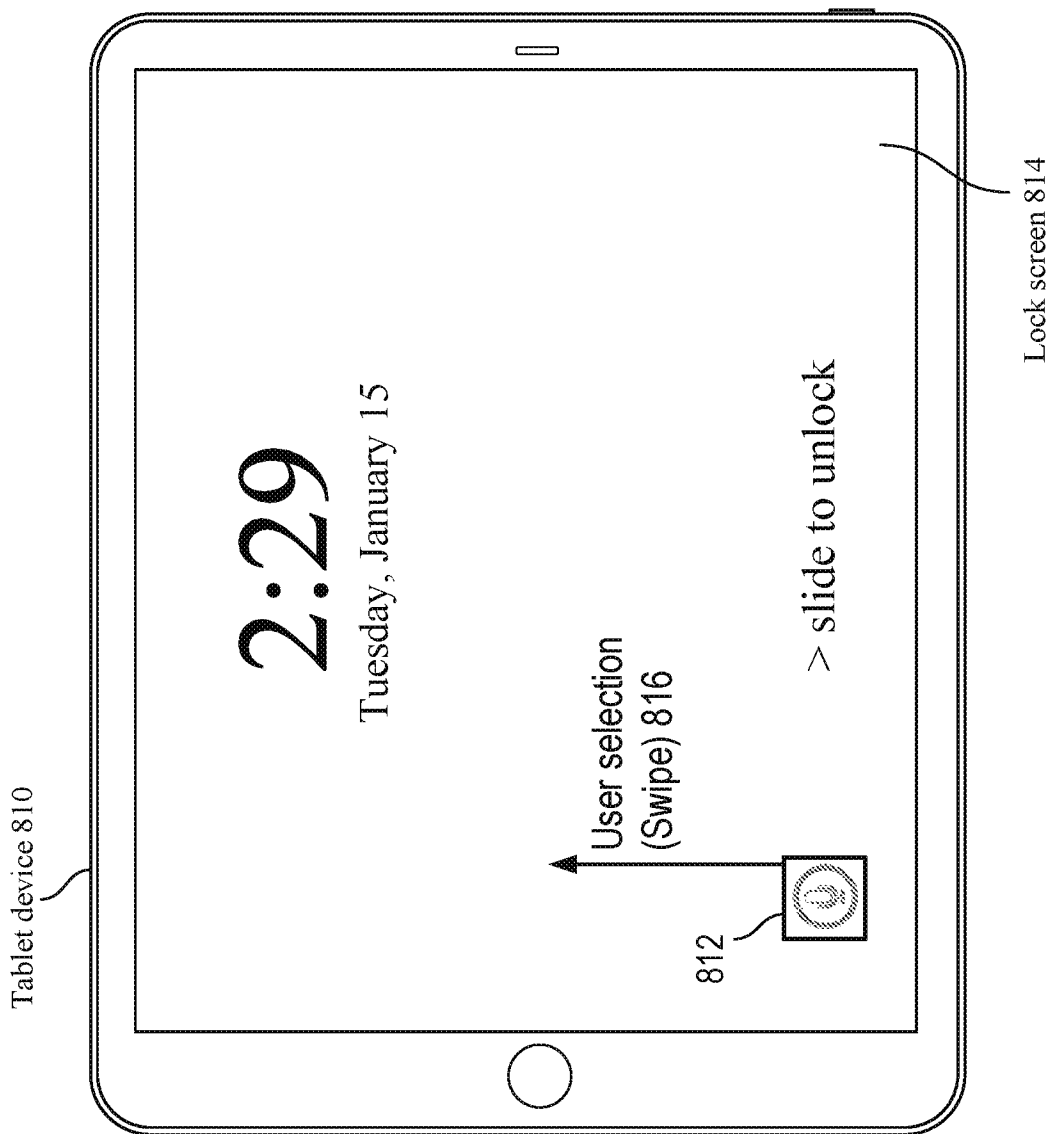


FIG. 8D

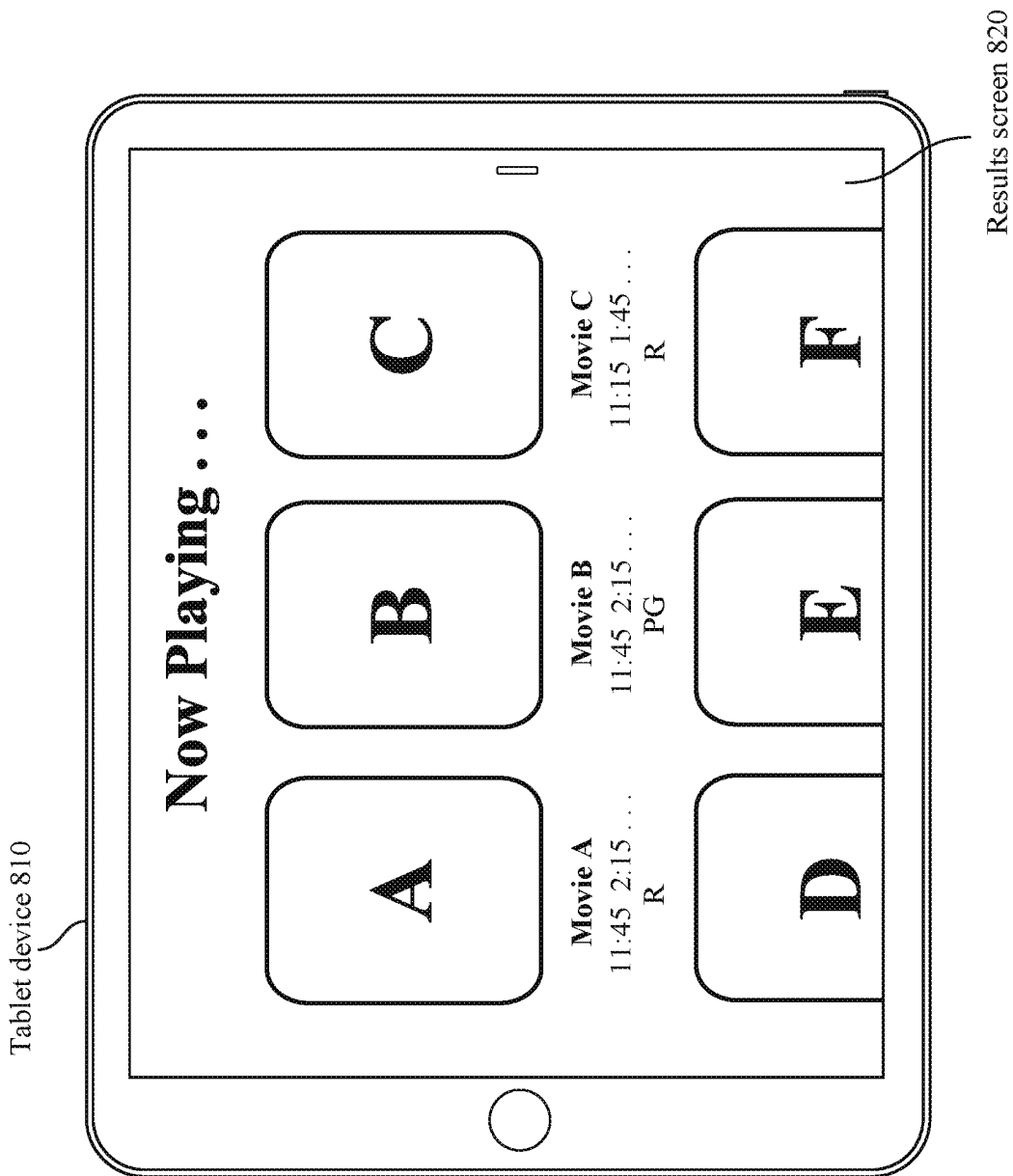


FIG. 8E

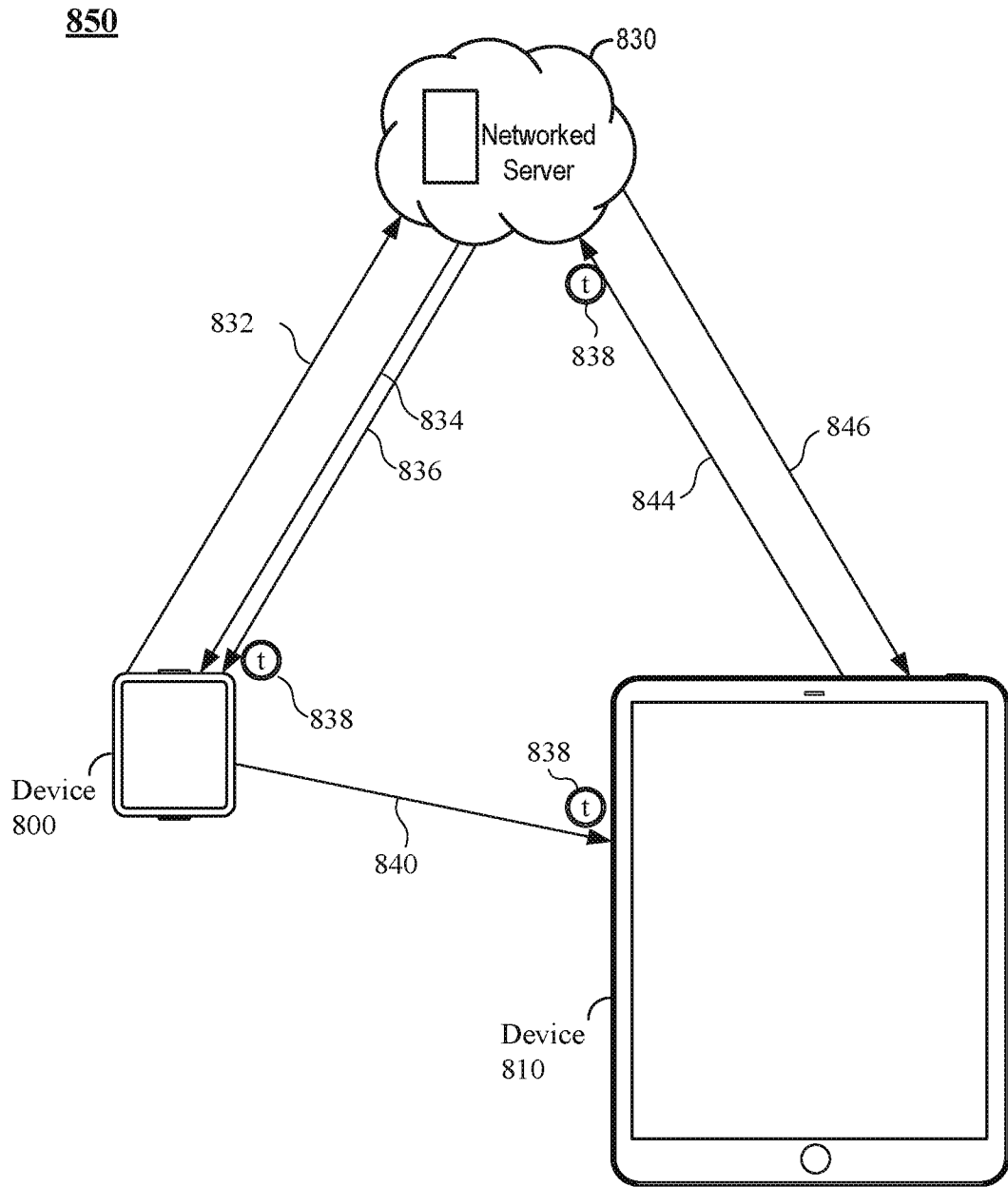


FIG. 8F

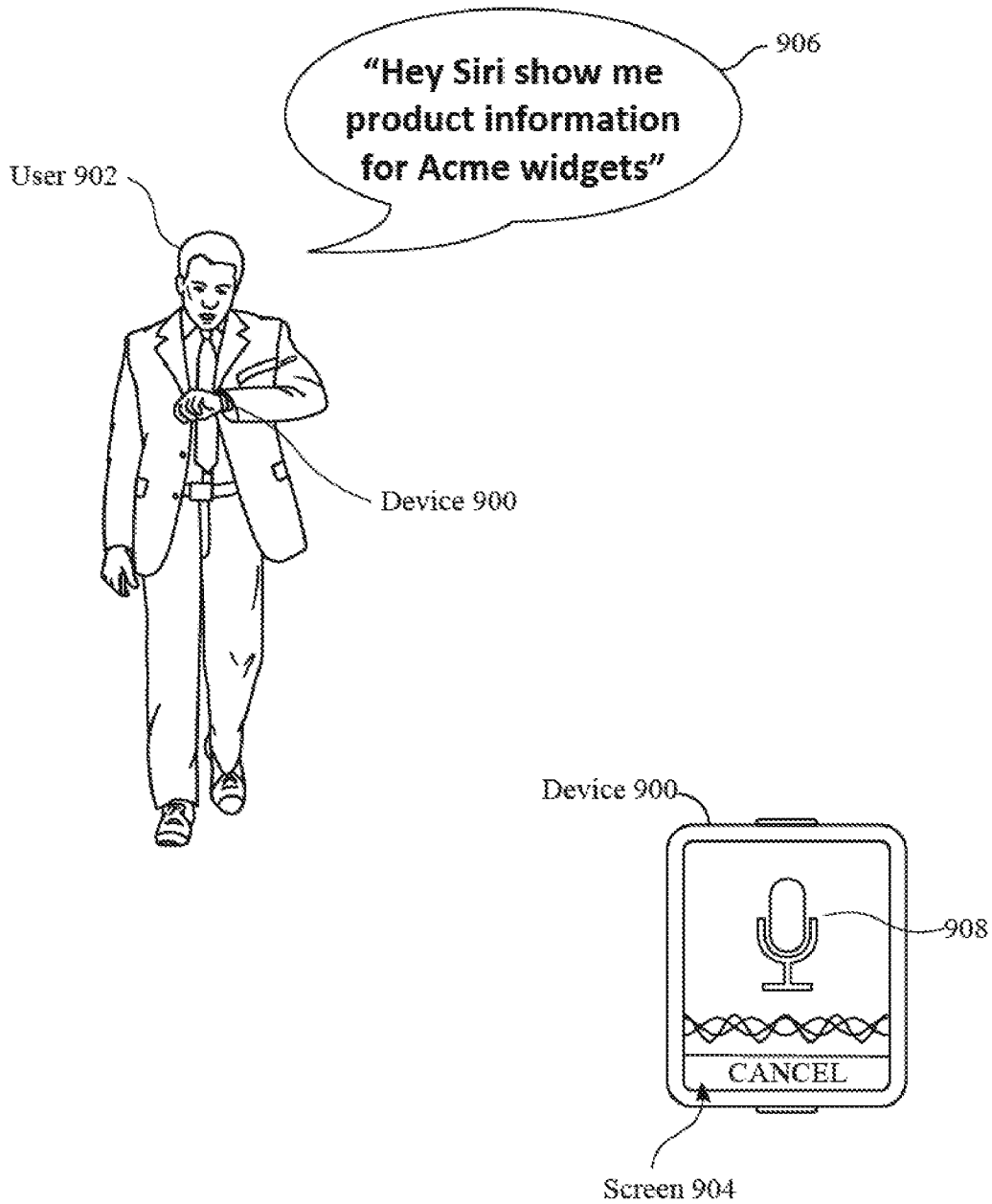


FIG. 9A

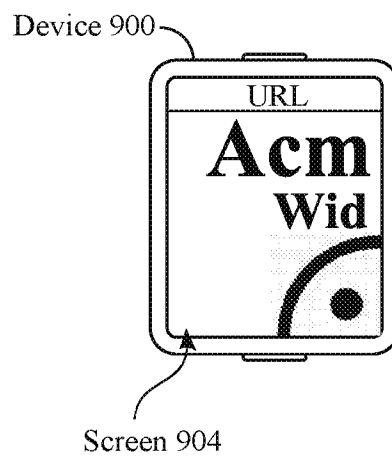


FIG. 9B

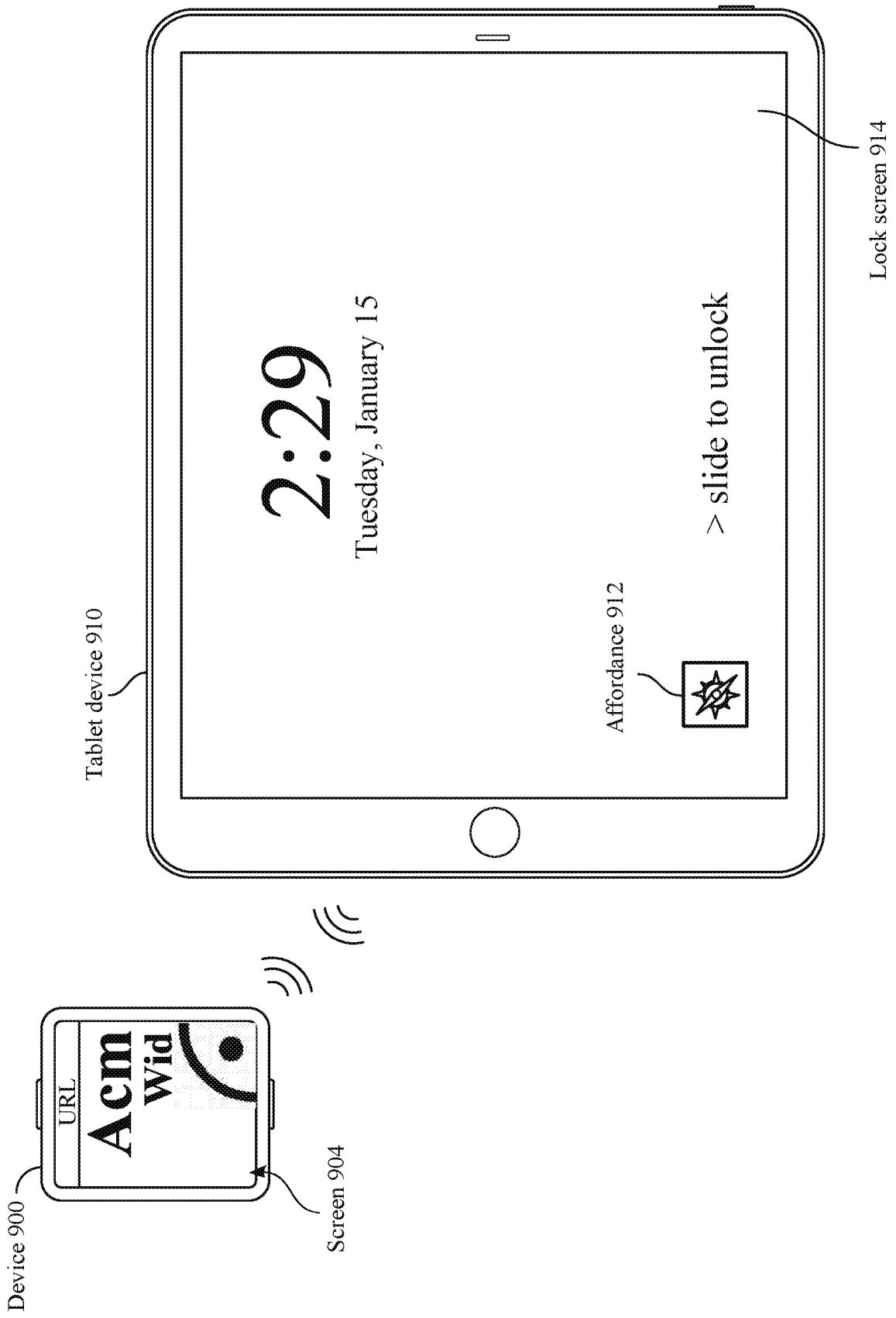


FIG. 9C

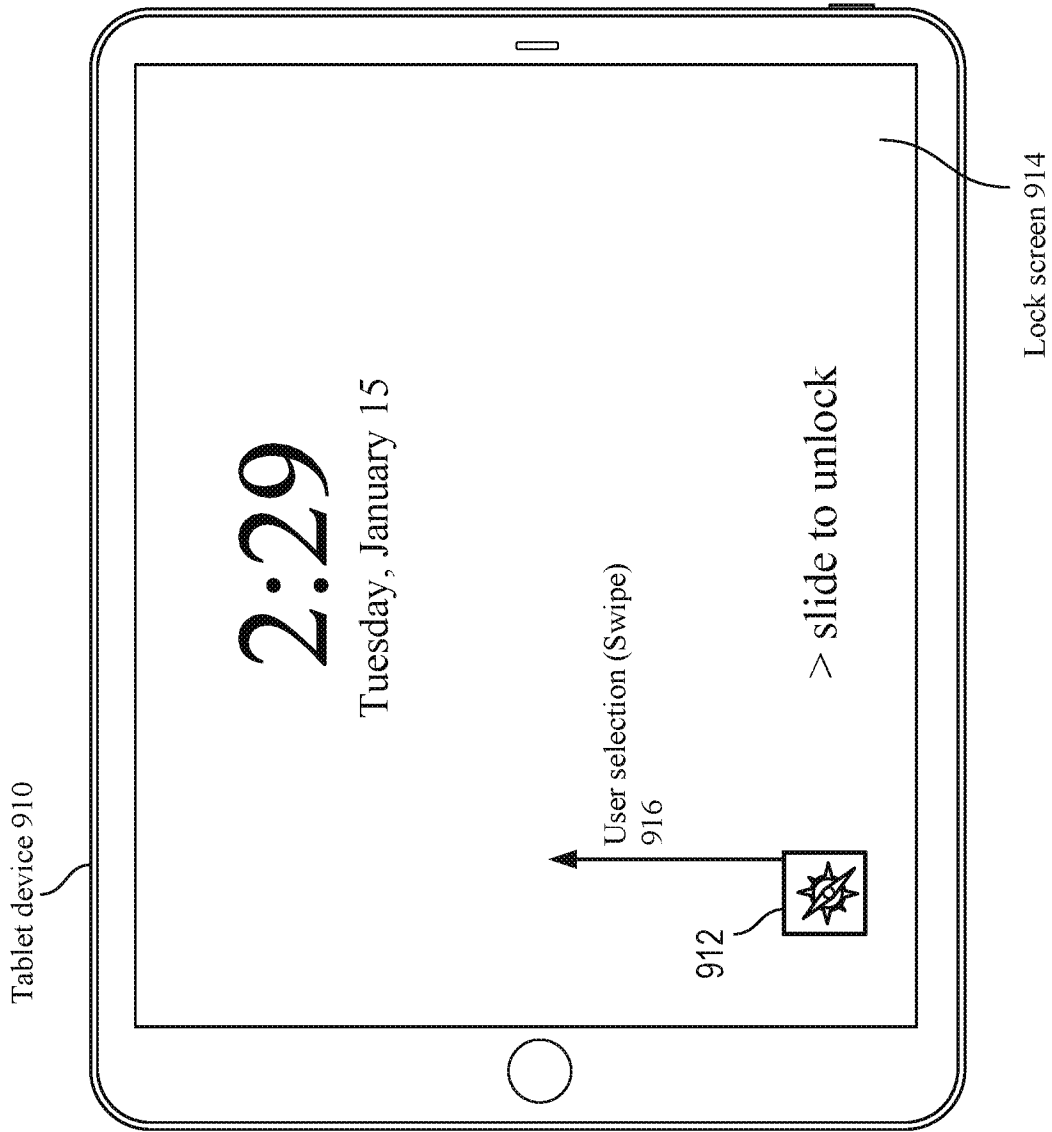


FIG. 9D

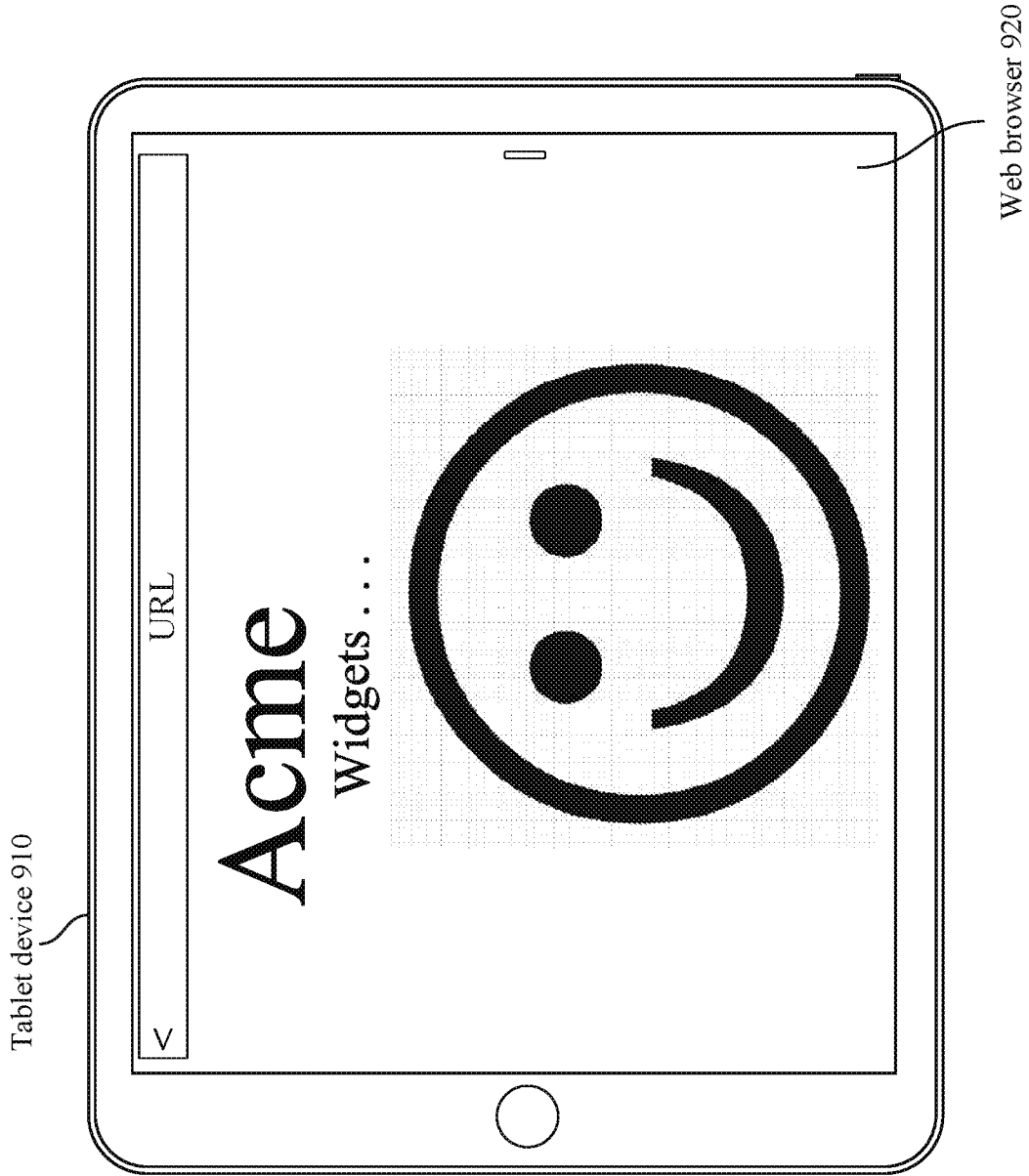


FIG. 9E

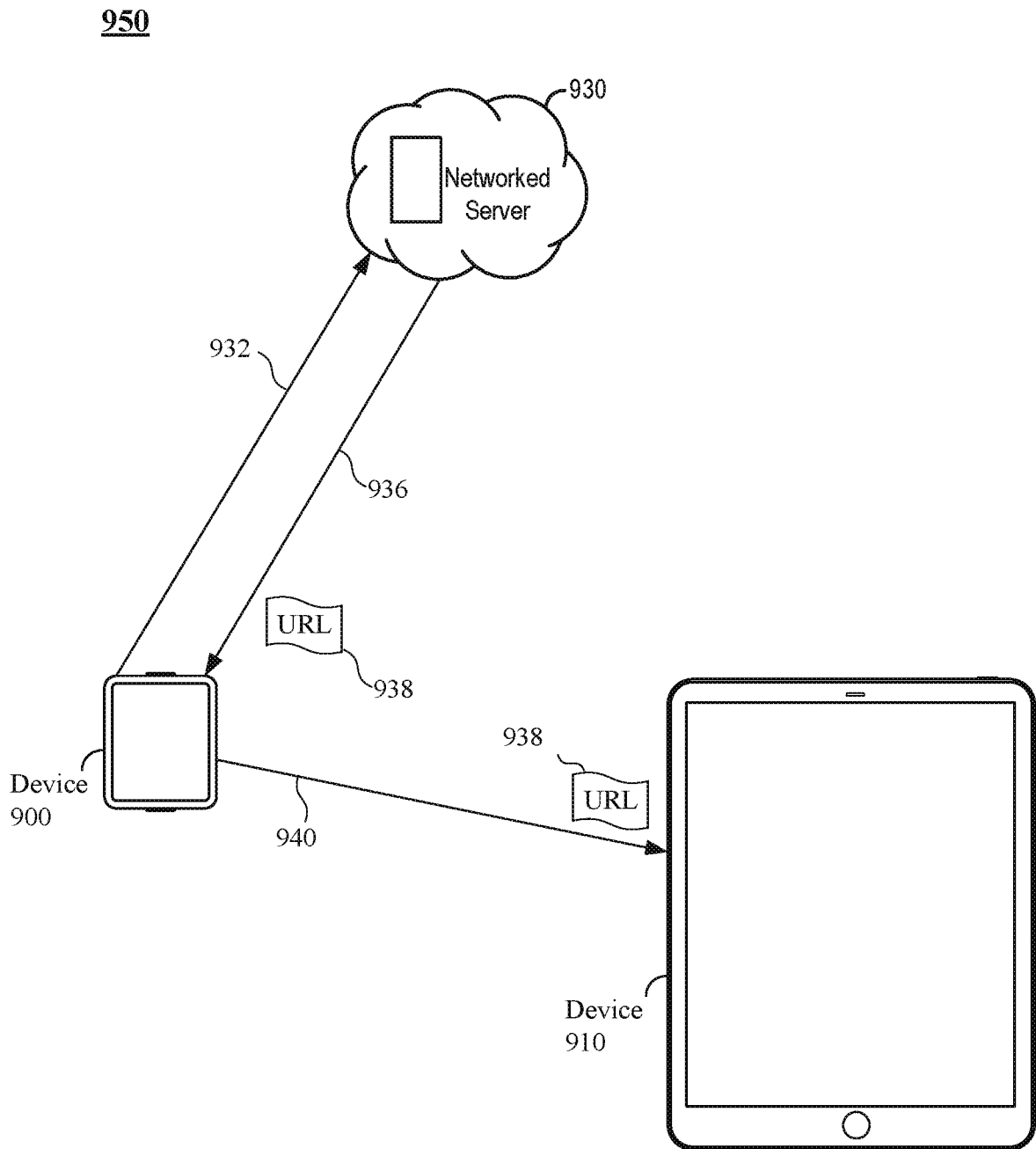


FIG. 9F

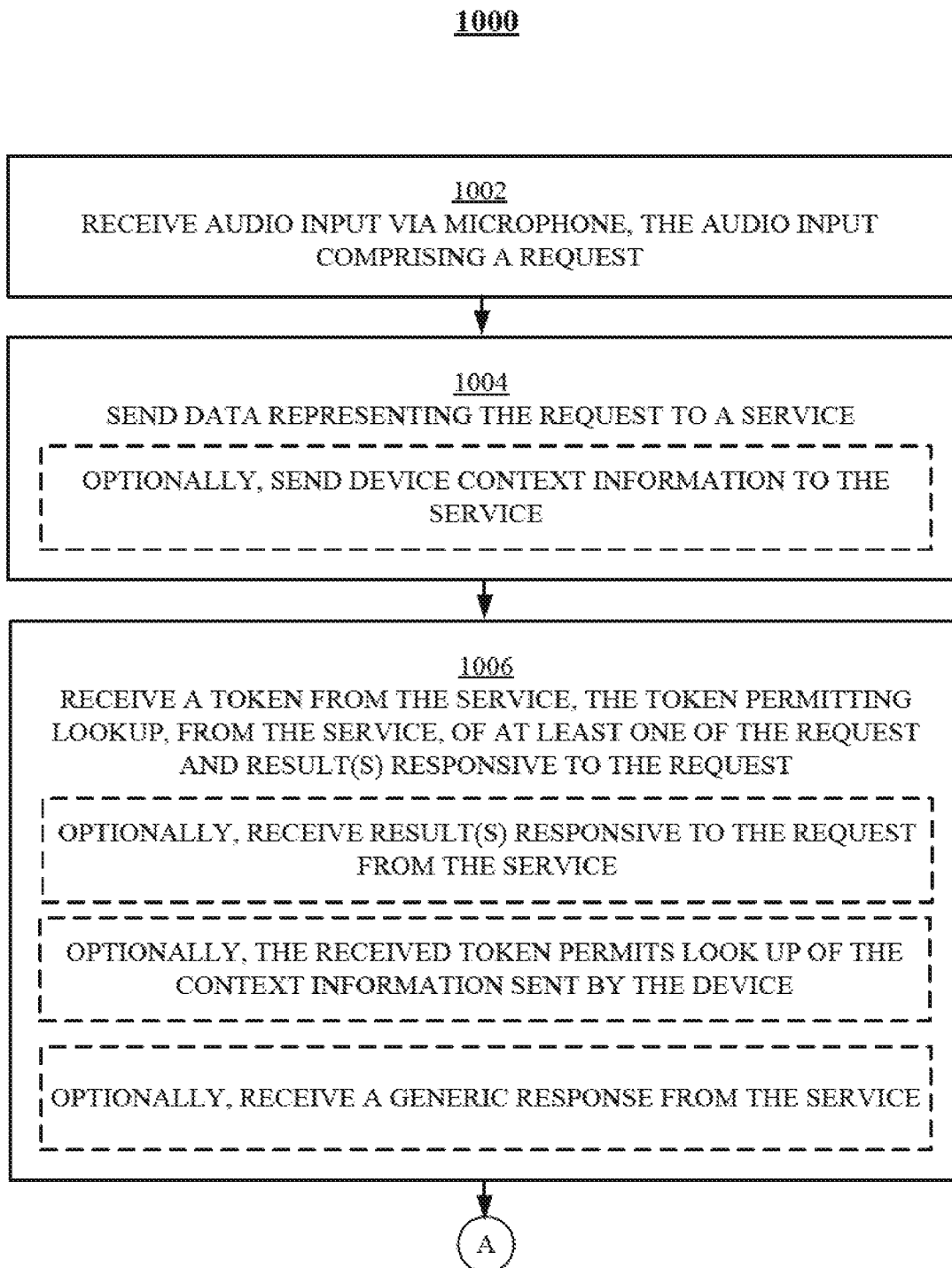


FIG. 10A

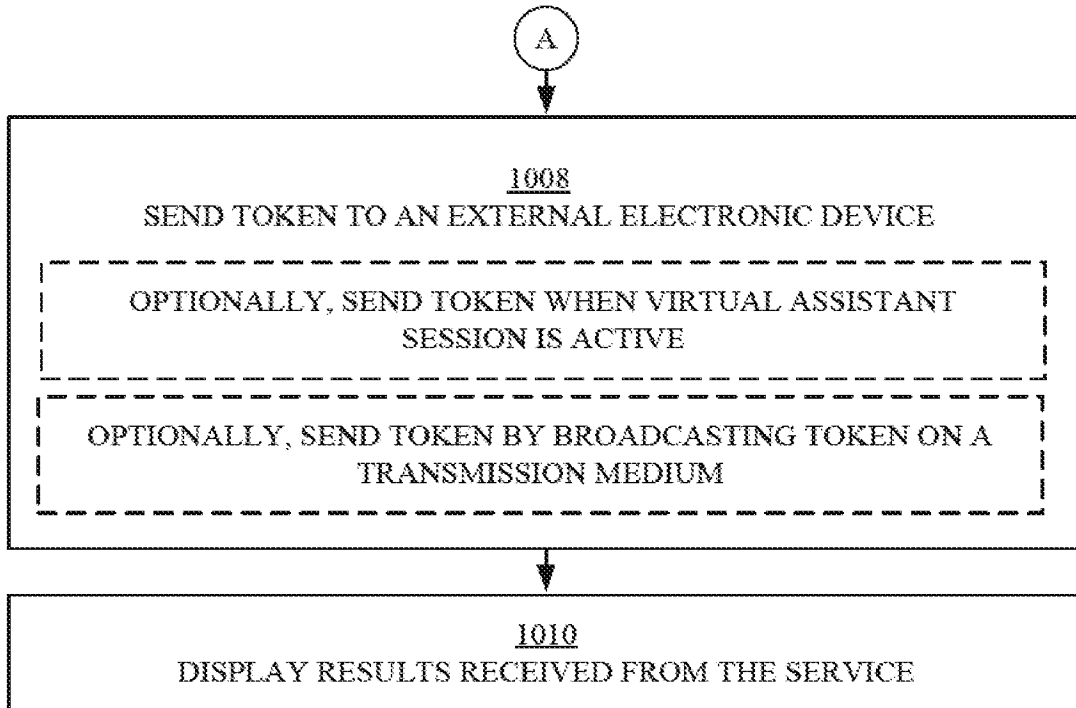


FIG. 10B

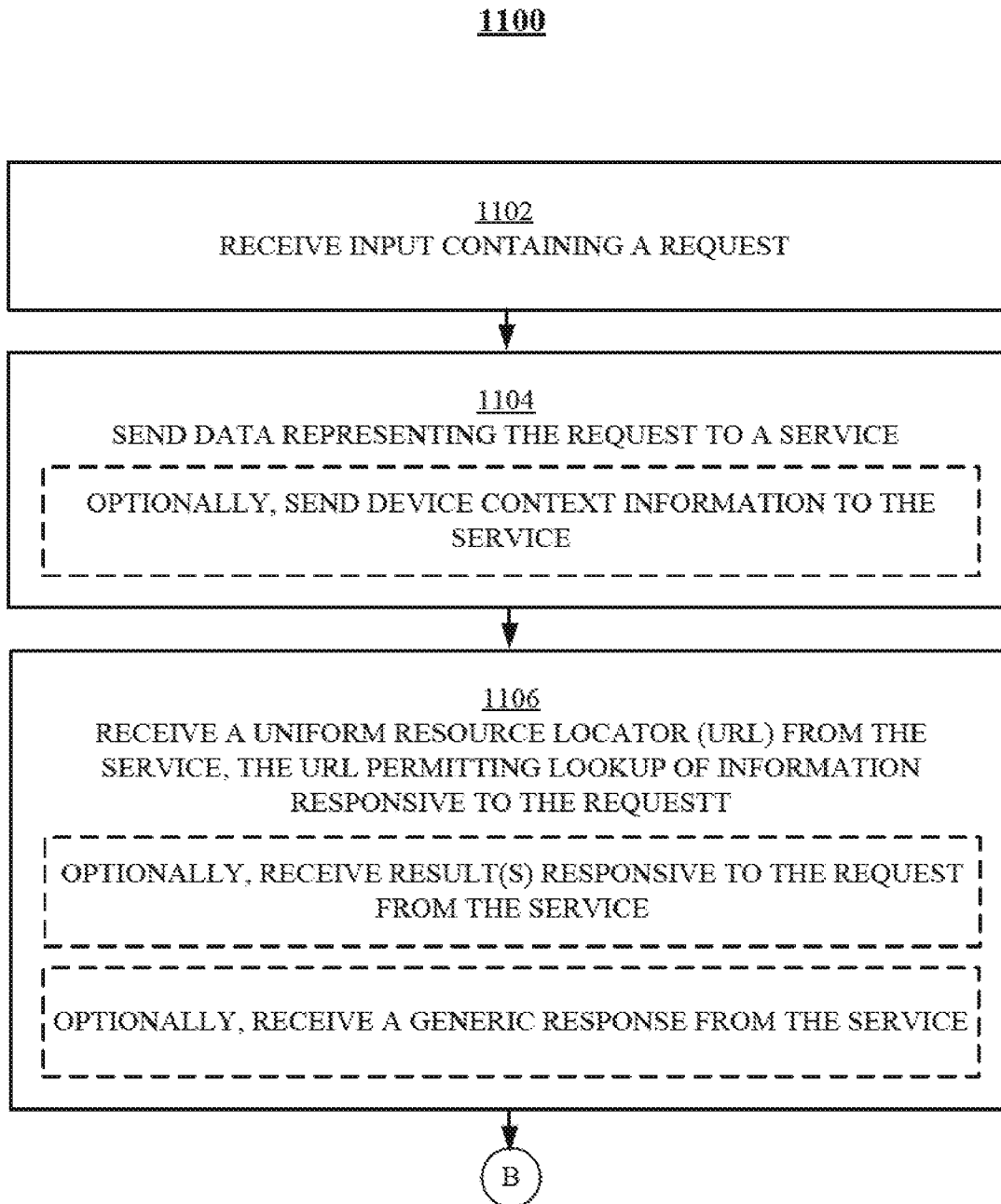


FIG. 11A

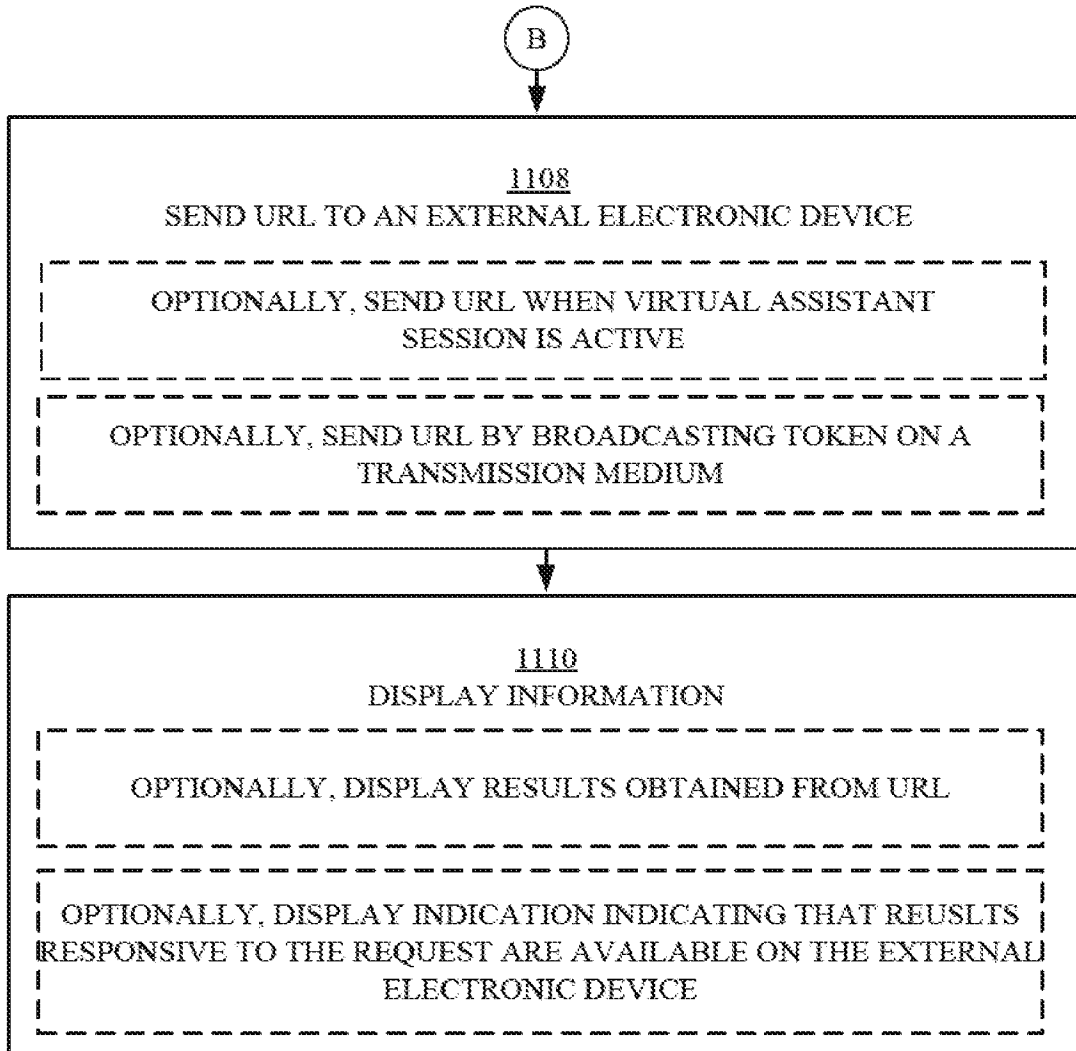


FIG. 11B

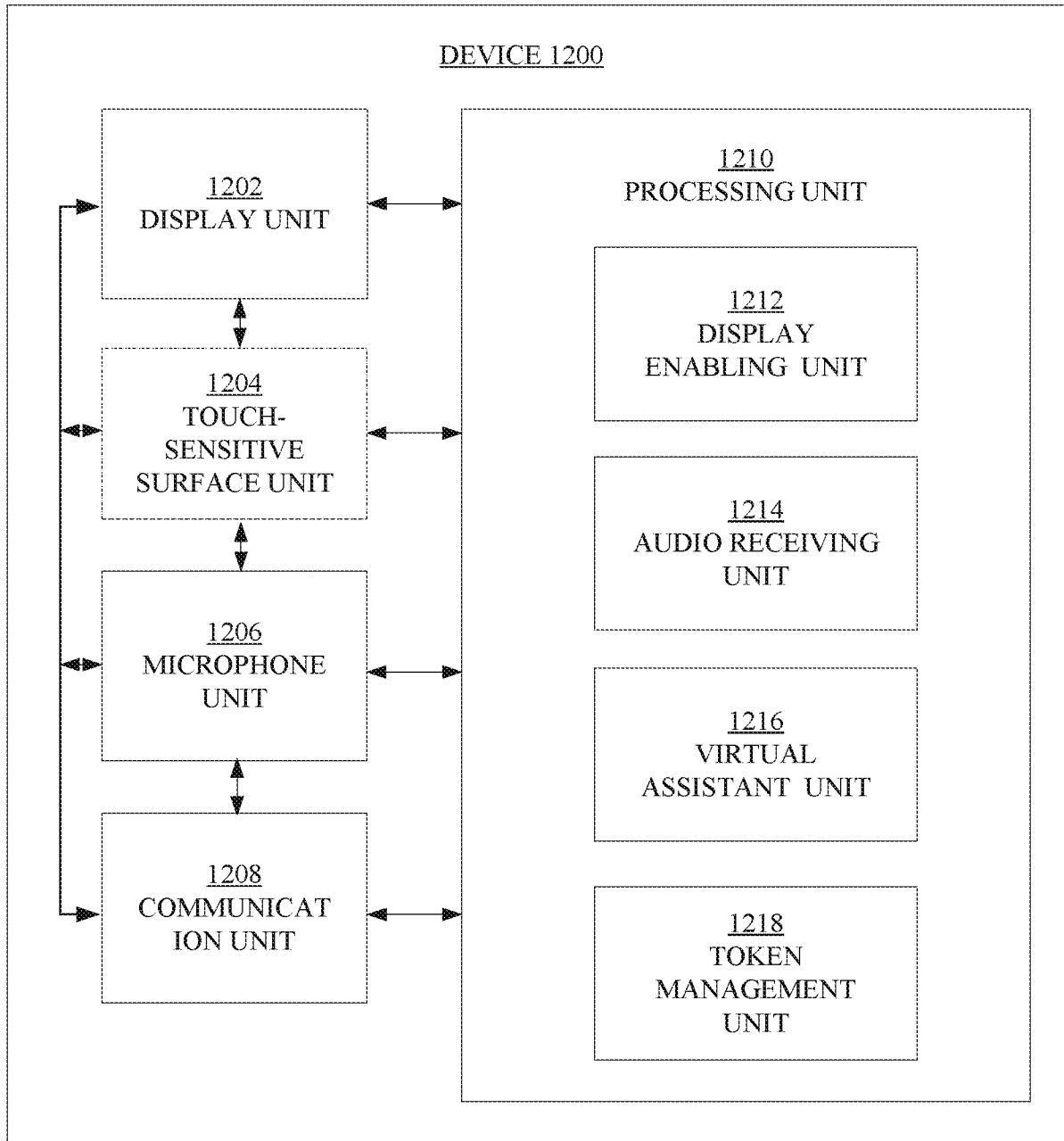


FIG. 12

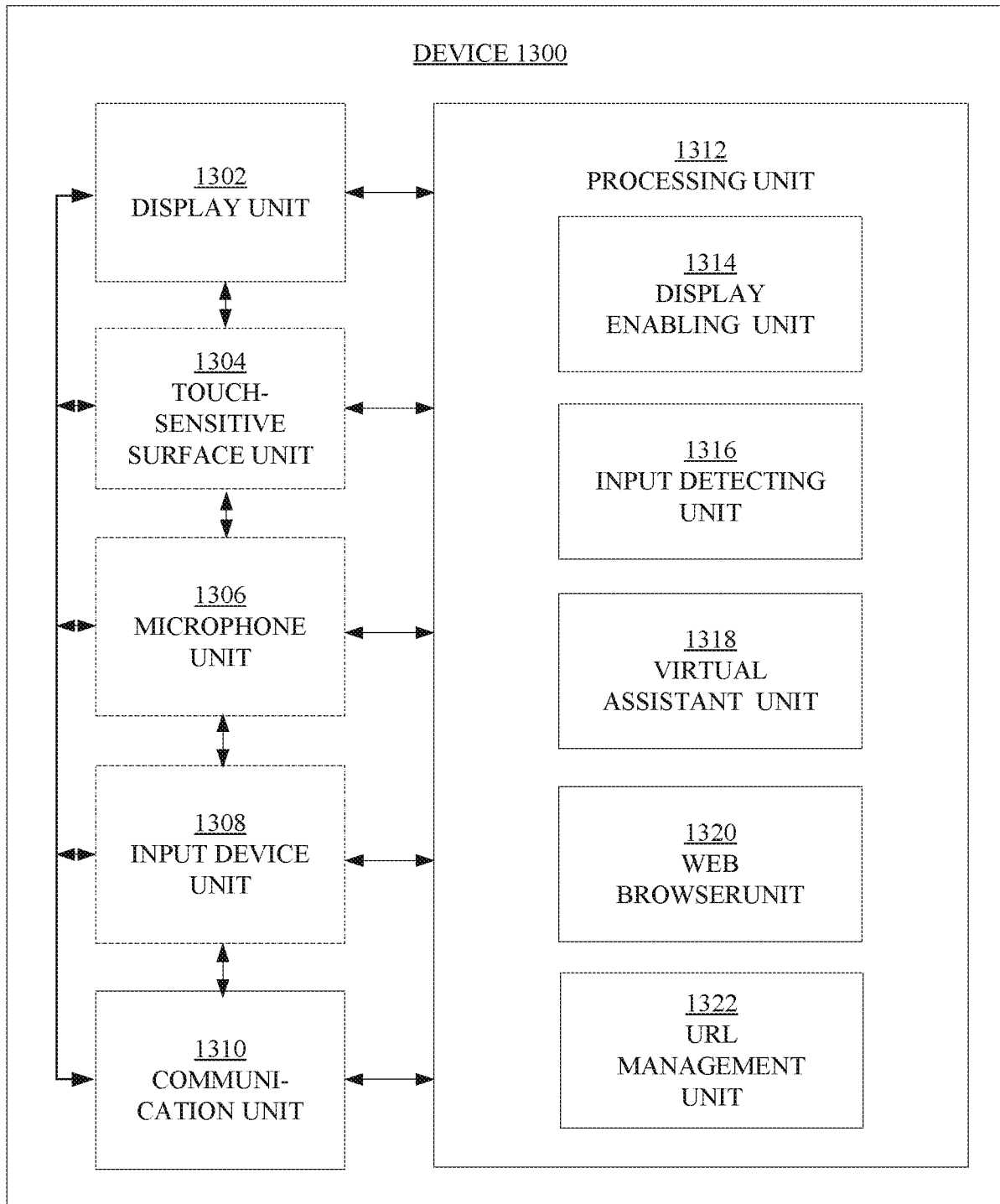


FIG. 13