Technical Disclosure Commons

Defensive Publications Series

August 2021

CLIENT APPLICATION COMMUNICATION WITH A HOST

Rafael Lima

CLIENT APPLICATION COMMUNICATION WITH A HOST

ABSTRACT

A projection application for projecting a graphical user interface (GUI) of a computing device (e.g., a smartphone, a tablet computer, smartglasses, a smartwatch, a portable gaming system, a laptop computer, etc.) to a head unit (e.g., an infotainment system) of a vehicle (e.g., automobile, motorcycle, a bus, a recreational vehicle (RV), a semi-trailer truck, a tractor or other type of farm equipment, a train, a plane, a helicopter, etc.) may process an input provided to a notification of an application installed at the computing device to launch the application. In accordance with the techniques described here, the projection application may attach a binder (e.g., an inter-process communication mechanism) to the notification with an active pending intent. The projection application may broadcast the notification with the binder to the application, which in turn checks the pending intent status of the notification. Responsive to determining that the pending intent is active, the application may leverage a library (e.g., a set of implementations of behavior that includes configuration data, pre-written code, subroutines, values, classes, etc.) to use the binder to issue a callback to the projection application that causes the projection application to launch the application. In this way, the projection application may communicate with the application despite the projection application being unable to access the contents of notifications for the application.

DESCRIPTION

FIG. 1 below is a conceptual diagram of a system 10 including a head unit 100 of a vehicle (e.g., an automobile, a motorcycle, a bus, a recreational vehicle (RV), a semi-trailer truck, a tractor or other type of farm equipment, a train, a plane, a helicopter, etc.) and a computing device 120 (e.g., a smartphone, a tablet computer, smartglasses, a smartwatch, a portable gaming system, a laptop computer, etc.). As shown in FIG. 1, head unit 100 includes one or more processors 102, a display 104, one or more communication components 106 ("COMM components 106"), and one or more storage devices 108. As further shown in FIG. 2, computing device 120 includes one or more processors 122, a display 124, one or more communication components 126 ("COMM components 126"), and one or more storage devices 128.

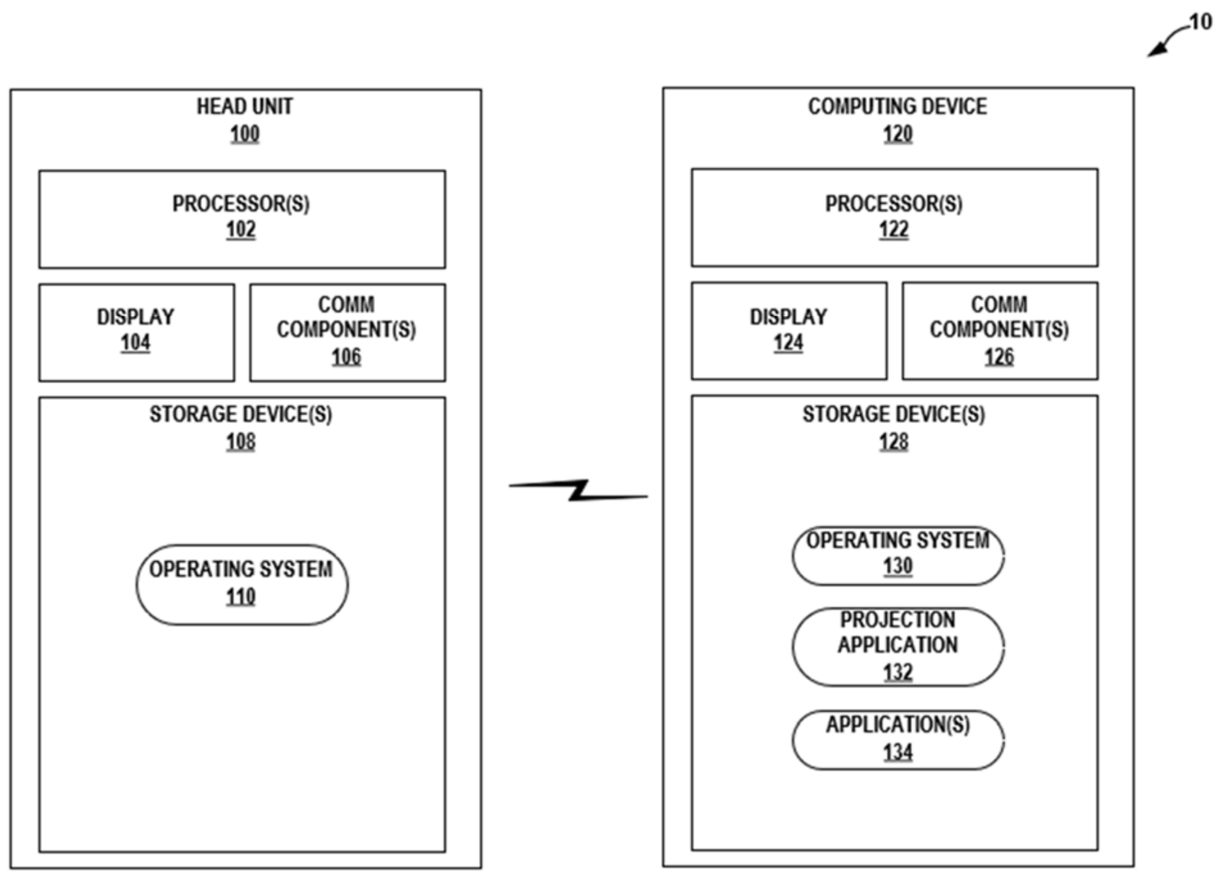

FIG. 1

One or more components (e.g., processors 102, display 104, COMM components 106, storage devices 108, etc.) of head unit 100 may be substantially similar to one or more components (e.g., processors 122, display 124, COMM components 126, storage devices 128, etc.) of computing device 120. As such, the description of one may apply equally to the other except for any differences described here.

Head unit 100 of system 10 may operate to assist, inform, entertain, or otherwise provide for interactions with one or more occupants of a vehicle. Head unit 100 may represent an integrated head unit that provides a user interface (UI), such as a voice user interface (VUI), a graphical user interface (GUI), etc. In general, head unit 100 may control one or more vehicle systems, such as a heating, ventilation, and air conditioning (HVAC) system, a lighting system (for controlling interior and/or exterior lights), an infotainment system, a seating system (for controlling a position of a driver and/or passenger seat), etc. Head unit 100 may be configured to establish a session with a computing device to permit data exchange. In some examples, an occupant of the vehicle in which vehicle head unit 100 is located may connect computing device 120 to head unit 100 to project (or otherwise, cast or stream) a GUI to head unit 100. For instance, a UI model in head unit 100 may be a thin client that supports projection of a GUI from computing device 120.

Processors 102 may implement functionality and/or execute instructions associated with head unit 100. Examples of processors 102 may include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device. Processors 102 may retrieve and execute instructions stored by storage devices 108 that cause processors 102 to perform the operations described in this disclosure.

Display 104 of head unit 100 may be a presence-sensitive display that functions as an input device and as an output device. For example, display 104 may function as an input device using a presence-sensitive input component, such as a resistive touchscreen, a surface acoustic wave touchscreen, a pressure-sensitive screen, an acoustic pulse recognition touchscreen, or another presence-sensitive display technology. Additionally, display 104 may function as an output (e.g., display) device using any of one or more display components, such as a liquid crystal display (LCD), dot matrix display, light emitting diode (LED) display, active-matrix organic light-emitting diode (AMOLED) display, etc.

COMM components 106 of head unit 100 may include wireless communication devices capable of transmitting and/or receiving communication signals, such as a cellular radio, a 3G radio, a 4G radio, a 5G radio, a Bluetooth® radio (or any other PAN radio), an NFC radio, or a Wi-Fi™ radio (or any other wireless local area network (WLAN) radio). COMM components 106 may be configured to send and receive information via a network (e.g., a local area network (LAN), wide area network (WAN), a global network, such as the Internet, etc.).

Storage devices 108 of head unit 100 may include one or more computer-readable storage media. For example, storage devices 108 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information used by head unit 100. In some examples, storage devices 108 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard discs, optical discs, solid state discs, etc. Examples of volatile memory devices include random-access memories (RAM), dynamic random-access memories (DRAM), static random-access memories (SRAM), etc.

As shown in FIG. 1, storage devices 108 may include an operating system 110 (“OS 110”) that provides an execution environment for one or more applications, such as projection application 132. OS 110 may represent a multi-threaded operating system or a single-threaded operating system with which projection application 132 may interface to access hardware of head unit 100. OS 110 may include a kernel that facilitates access to the underlying hardware of head unit 100, where kernel may present a number of different interfaces (e.g., application programmer interfaces – APIs) that projection application 132 may invoke to access the underlying hardware of head unit 100.

Projection application 132 may represent an application that provides a bridge between computing device 120 and head unit 100. Projection application 132 may facilitate projection of UIs, such as GUIs, VUIs, etc., for applications 134 to head unit 100. For instance, projection application 132 may provide via display 104 a GUI for applications 134 that includes information to the user in the form of text, images, etc. Projection application 132 may generate graphical elements to, for example, satisfy driver distraction standards and accommodate a variety of car screen factors and input modalities. Actuation of graphical elements may invoke corresponding functions of applications 134.

As noted above, projection application 132 projects a GUI for computing device 120 to head unit 100. When the user provides (e.g., via display 104) inputs to head unit 100, projection application 132 may receive the inputs. For example, if head unit 100 outputs via display 104 a notification for application 134 (which may be a third-party, vehicle-related application) and the user provides via display 104 a user input to the notification to launch application 134, projection application 132 may receive that user input and the corresponding notification. However, projection application 132 may be unable to process the user input because the notification may be opaque (e.g., not readable by) to projection application 132. Thus, in the above example, projection application 132 may be unable to launch application 134 (and thus be unable to project a GUI of application 134 at display 104) despite receiving the user input.

To address this issue, techniques described here may enable projection application 132 to communicate with application 134 by using a pending intent. For example, projection application 132 may attach a binder (e.g., an inter-process communication mechanism) to a notification with an active pending intent. Projection application 132 may send the notification with the binder to OS 130, which in turn may send the notification and the binder to application 134. Application 134 may check the pending intent status of the notification. Responsive to determining that the pending intent is active, application 134 may use the binder to issue a callback to projection application 132 that causes projection application 132 to launch application 134. In this way, projection application 132 may communicate with application 134 (e.g., to launch application 134) despite being unable to access the contents of notifications for application 134.

The pending intent for application 134 may represent a token that allows another application (e.g., projection application 132) to use application 134's permission to execute predefined code. In other words, application 134 may give a pending intent to projection application 132 to perform an operation specified by the pending intent, such as launching application 134. In some examples, the pending intent for application 134 may become active in response to user input. For instance, display 104 may display via projection application 132 a notification for application 134. The notification may include one or more graphical elements, such as buttons, icons, etc., that a user may actuate to invoke corresponding functionality. As an example, the user may tap a button of the notification to making a pending intent for launching application 134 active.

In some cases, only vehicle-related applications 134 may have a pending intent for launching applications 134. For example, a GUI for a notification for a social networking application may not include the graphical element for making the pending intent for launching the application active. At the same time, a GUI for a notification for a navigation application may include the graphical element for making the pending intent for launching the application active. In that example, a user may provide a user input, such as a tap to the graphical element, to make the pending intent active.

In any case, projection application 132 may receive the notification with the active pending intent. However, as noted above, projection application 132 may lack the permission to access the contents of the notification. As such, projection application 132 may not be able to process the pending intent to perform the operation specified by the pending intent. Hence, projection application 132 may attach a binder to the notification and broadcast via OS 130 the notification with the binder, which application 134 may receive via a broadcast receiver.

As the notification is for application 134, the notification may not be opaque to application 134. Accordingly, application 134 may access the contents of the notification to read the pending intent. Responsive to the pending intent being active, application 134 may leverage a library (e.g., a set of implementations of behavior that includes configuration data, pre-written code, subroutines, values, classes, etc.) to use the binder to send a callback to projection application 132. The callback 132 to projection application may cause projection application 132 to perform the operation specified by the pending intent. For example, if the operation specified by the pending intent is to launch application 134, the callback may cause projection application 132 to launch application 134.

One or more advantages of the techniques described in this disclosure include enabling a user to interact with a projection of a computing device being displayed at a head unit to perform operations with respect to other applications, such as launching those applications, installed at the computing device. In a vehicle setting, interacting with the head unit to launch the applications may be more convenient than interacting with the computing device. Furthermore, such improvements in the user experience may be particularly beneficial when the user is operating the vehicle by reducing distractions to the user, thereby potentially promoting safety.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2017/0031680A1. In another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2012/0081353A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in Google Android, “Build navigation, parking, and charging apps for Android Auto,” Using the Android for Cars App Library. In yet another example, the techniques of this disclosure may be combined with the techniques described in “How to use readPendingIntentOrNullFromParcelmethodinandroid.app.PendingIntent,” Tabnine.