

Technical Disclosure Commons

Defensive Publications Series

June 05, 2018

Triggering an Update of a Client Application by Selecting a Deeplink

Justin Lewis

Ruxandra Davies

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Lewis, Justin and Davies, Ruxandra, "Triggering an Update of a Client Application by Selecting a Deeplink", Technical Disclosure Commons, (June 05, 2018)

https://www.tdcommons.org/dpubs_series/1225



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Triggering an Update of a Client Application by Selecting a Deeplink

Abstract:

In some instances, a user of a mobile device may trigger an update of a client application by selecting, from an application other than the client application, a deeplink (e.g., a hyperlink or a uniform resource locator (URL)) that has context and that also links the mobile device to a host program. If the client application is out of date and not compatible with the host program or content provided by the host program, the mobile device triggers an update to the client application. During the updating of the client application, the mobile device maintains the context so that the user may complete initiating the client application in accordance with the context.

Keywords: Deeplink, application updating, mobile device

Background:

To maintain revisions of a client application on a mobile device, it is important for the client application to be compatible with other applications, such as a host program. If, for example, a video-playing application installed on a mobile device (e.g., the client application) is not current with a video-sharing application provided by a website (e.g., the host program), a user of the mobile device may not be able to view a video (offered by the video-sharing application) that he selects via the video-playing application installed on the mobile device.

In like fashion, the need for the video-playing application to be current extends to selecting the video via a deeplink. Assume, for example, that the user selects a URL, which

is intended to link the mobile device to the video-sharing application (e.g., the host program) with context specific to a three-dimensional video of an artist performing a song. If the video-playing application, (e.g., the client application) is current and capable of rendering the three-dimensional video, the video-playing application may immediately launch the three-dimensional video of the artist performing the song. If, however, the video-playing application is not current and not capable of rendering a three-dimensional video, the video-playing application is not able to launch the three-dimensional video and must be updated.

Currently, when a client application requires updating, it is common for the client application to query the user if he would like to perform an update. The client application often presents the query on a display of the mobile device via a selectable “Update Application?” icon, a graphical user interface window, or the like. If the user confirms his intent to update the client application, however, the client application will close, discard context, and update in a fashion that is agnostic to the context or the user’s request. In such an instance, the user may become disengaged with the client application and forget the context of his request. In the case of the current example of the video playing application, this would result in the user potentially forgetting his request and not viewing the video associated with the URL.

Description:

A user may trigger an update of a client application by selecting, from an application other than the client application, a deeplink that has a context and links the mobile device

to a host program. If the client application is out of date and not compatible with the host program, the mobile device triggers an updating of the client application. During the updating of the client application, the mobile device maintains the context so that the user may complete initiating the client application in accordance with the context. Furthermore, during updating of the client application, the mobile device may present status indicators to the user that remind him of the context and allow the user to access the client application upon completion of the updating.

Fig. 1, below, illustrates an example mobile device communicating with a host program.

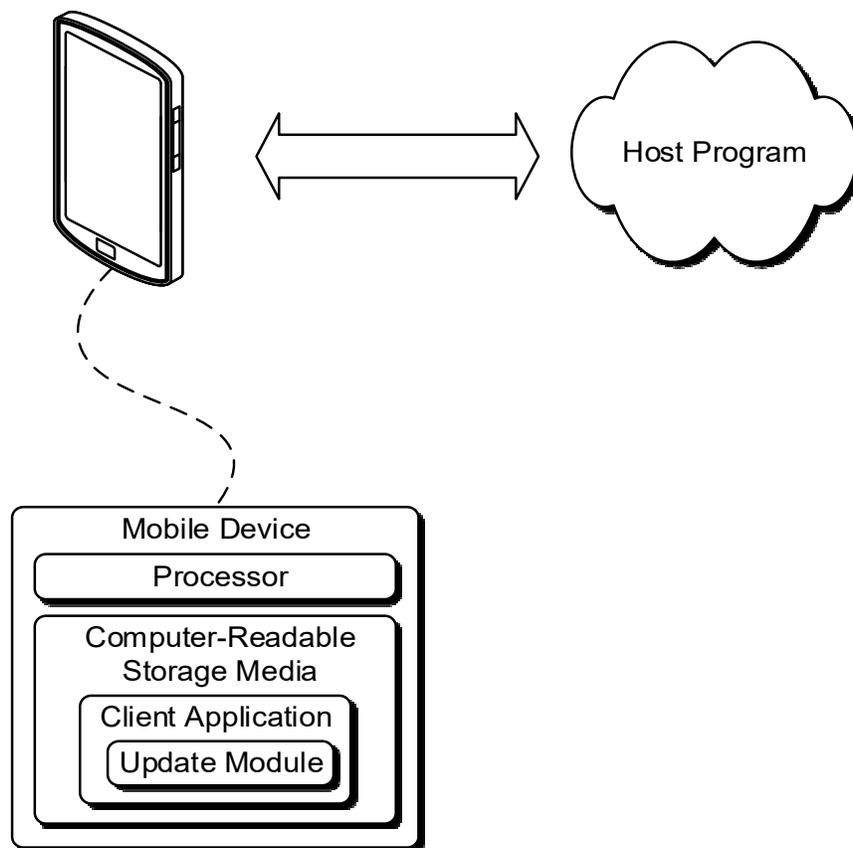


Fig. 1

As illustrated in Fig. 1, the mobile device includes a processor and a computer-readable storage media (CRM). The processor can execute modules of instruction or code stored by the CRM to implement various functionalities of the mobile device.

Stored in the CRM of the mobile device is a client application (e.g., an application that is a client of the host program with which the mobile device is communicating). Included in the client application is code in the form of an update module that, when executed by the processor of the mobile device, manages the update of the client application.

Fig. 2, below, illustrates an example method that the mobile device can perform as part of updating a client application.

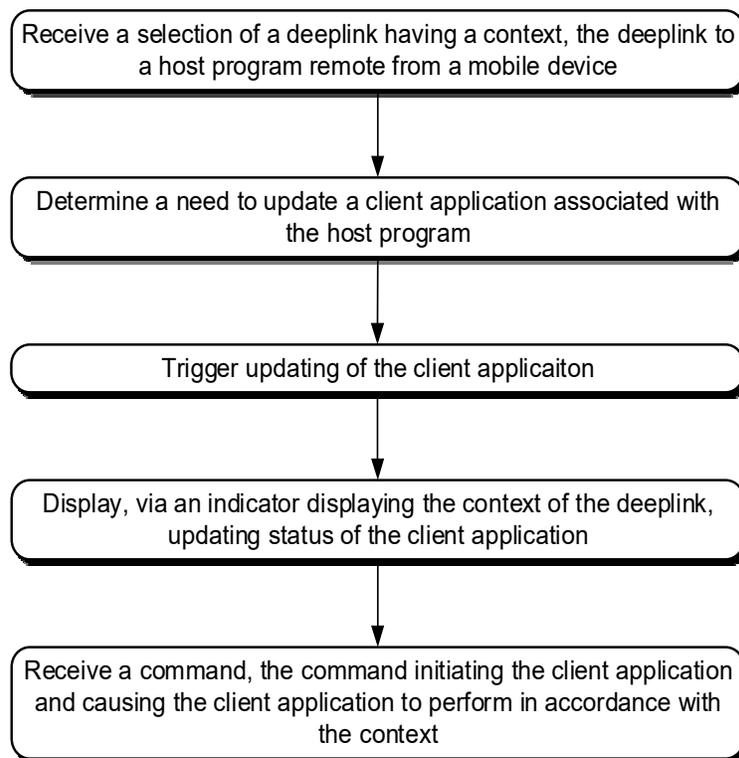


Fig. 2

As a first step in the method, the mobile device may receive a selection from a user of the mobile device who, for example, selects a deeplink embedded in an email, presented in a social media window, or the like. The deeplink may include context and link the mobile device to a host program that is remote from the mobile device.

Next, and based on data received from the host program, the mobile device may determine a need to update a client application that is stored in a CRM of the mobile device and is associated with the host program. As part of this determination, a processor may execute instructions of an update module included in the client application to analyze a header (e.g., data) that the host program communicates to the mobile device as part of establishing communications.

The mobile device may then trigger the update to the client application. After triggering, the mobile device may present an indicator on a display of the mobile device. The indicator may include not only the context associated with the deeplink, but also a status of the ongoing update (e.g., start at “active” and change to “complete”). Furthermore, the indicator may be in the form of a thumbnail, a slice, the entire display of the mobile device, or the like.

After the status of the ongoing update is “complete”, the mobile device may then receive a command from a user. The processor, in response to the mobile device receiving the command, initiates the client application, causing the client application to perform in accordance with the context. In one example instance, a selection of the indicator may be the command while in another example instance, an audible signal may be the command.

Figs. 3A and 3B illustrate an example mobile device and a display presenting an indicator. The indicator includes example context associated with a deeplink (Fred Gallium singing “Don’t Stop”) as well as an updating status.

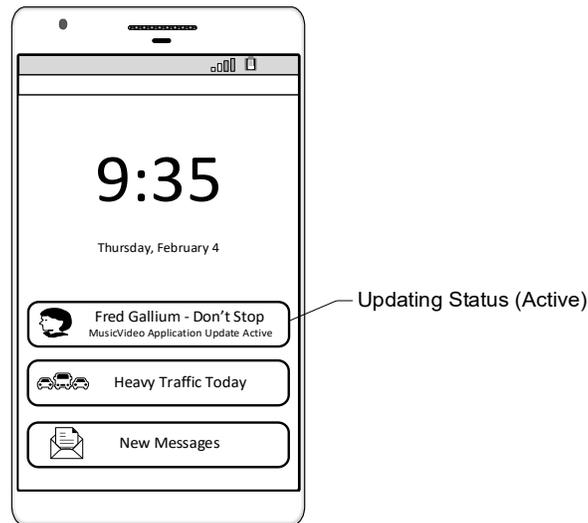


Fig. 3A

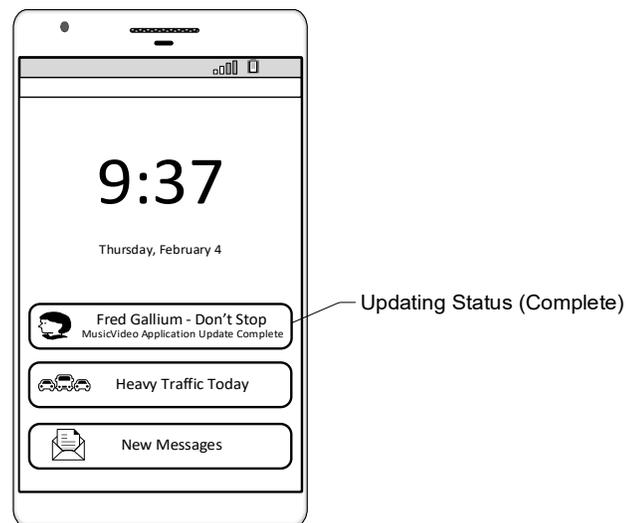


Fig. 3B

As illustrated in Fig. 3A, the display is presenting several indicators that may be associated with various applications. Such applications may be local to the device, remote from the device, or a combination thereof (e.g., a host program and a client application). In particular, the indicator associated with the client MusicVideo application is reminding a user of the mobile device that he has recently expressed interest in accessing, from the host MusicVideo application, the music video of Fred Gallium performing the song “Don’t Stop”. The recently expressed interest may have been the result of the user selecting a deeplink embedded in an email, present on a social media website, or the like. However, as the client MusicVideo application is outdated and not capable of rendering the music video available from the host MusicVideo application, the mobile device has triggered an updating of the client MusicVideo application and is actively performing an update.

As illustrated in Fig. 3B, the indicator associated with the client MusicVideo application continues to remind the user of his recent interest in accessing the Fred Gallium music video. However, now the update of the client MusicVideo application is complete and the client MusicVideo application is capable of rendering the music video. In this particular example, the indicator may now be selectable, and the user may simply select the indicator and directly access the music video of Fred Gallium singing “Don’t Stop”.

Additional variations of updating a client application, using similar techniques, are contemplated here. As an example, updating a client application may include triggering the updating locally. In this example, prompting information in a file that the mobile device receives or stores may act as a triggering mechanism. A user may receive a file (such as a file containing the music video of Fred Gallium) via an email or a purchase and

downloading of the file. However, continuing with this example, the file includes prompting information in the form of a file identification number, a compatibility identifier, a file version number, or the like. In contrast to analyzing a header that the mobile device receives from a host program, the processor may execute instructions contained in the update module to analyze the prompting information and determine that the client application needs updating.

Although the present disclosure teaches triggering updating of a client mobile application on a mobile device, the techniques are not restricted to mobile applications and devices. Such techniques are extendable to any application that is executed on any platform, including servers, workstations, personal computers, and the like.