October 2020

ADVERTISEMENT SOFTWARE DEVELOPMENT KIT UNBUNDLING

Giles Hogben

Chris Palmer

Nick Kralevich

Bram Bonné

Sudhi Herle

ADVERTISEMENT SOFTWARE DEVELOPMENT KIT UNBUNDLING

ABSTRACT

A system described herein enables advertisement software development kits (SDKs) to be unbundled from applications that use advertisement SDKs. The system provides SDK proxy libraries and advertisement services developed based on the advertisement SDKs. The SDK proxy libraries replace original calls to advertisement SDKs with inter-protocol communication (IPC) calls. The advertisements services receive the IPC calls and, in response, generate views that include advertisements. The advertisement services provide the views to the applications. Rather than the application determining the advertisement views and/or clicks, the system described herein enables the advertisement services to validate views and/or clicks of the advertisement separately from the application. In this way, the system described in this disclosure may prevent the applications from generating fake clicks or fake views while preventing the advertisement services from accessing sensitive data collected by the applications.

DESCRIPTION

Application developers may generate revenue from in-app advertising services, which require application developers to link third-party advertisement SDKs into their applications. Typically, when a developer incorporates advertisements into an application, the developer must embed a particular version of an advertisement SDK into the application at build time. By embedding the advertisement SDK into the build, it is more difficult for the advertisement SDK developer to update the SDK as it requires the application developer to update the application. Further, by embedding the advertisement SDK, the application developer can programmatically generate fake views (e.g., by hiding advertisements under other user interface elements),

generate fake clicks by calling a method indicating that the advertisement was clicked, or otherwise manipulate the advertisement SDK to generate more revenue for the application developer. Further, by embedding the advertisement SDK within the application, the advertisement SDK developer is granted the same permissions as the application, which may reduce the user's ability to limit what information is provided to advertisers. Rather than embedding the advertisement SDK within the compiled application, techniques described herein separate the advertisement SDK from the application by enabling the advertisement SDK to be separately installed from the application.

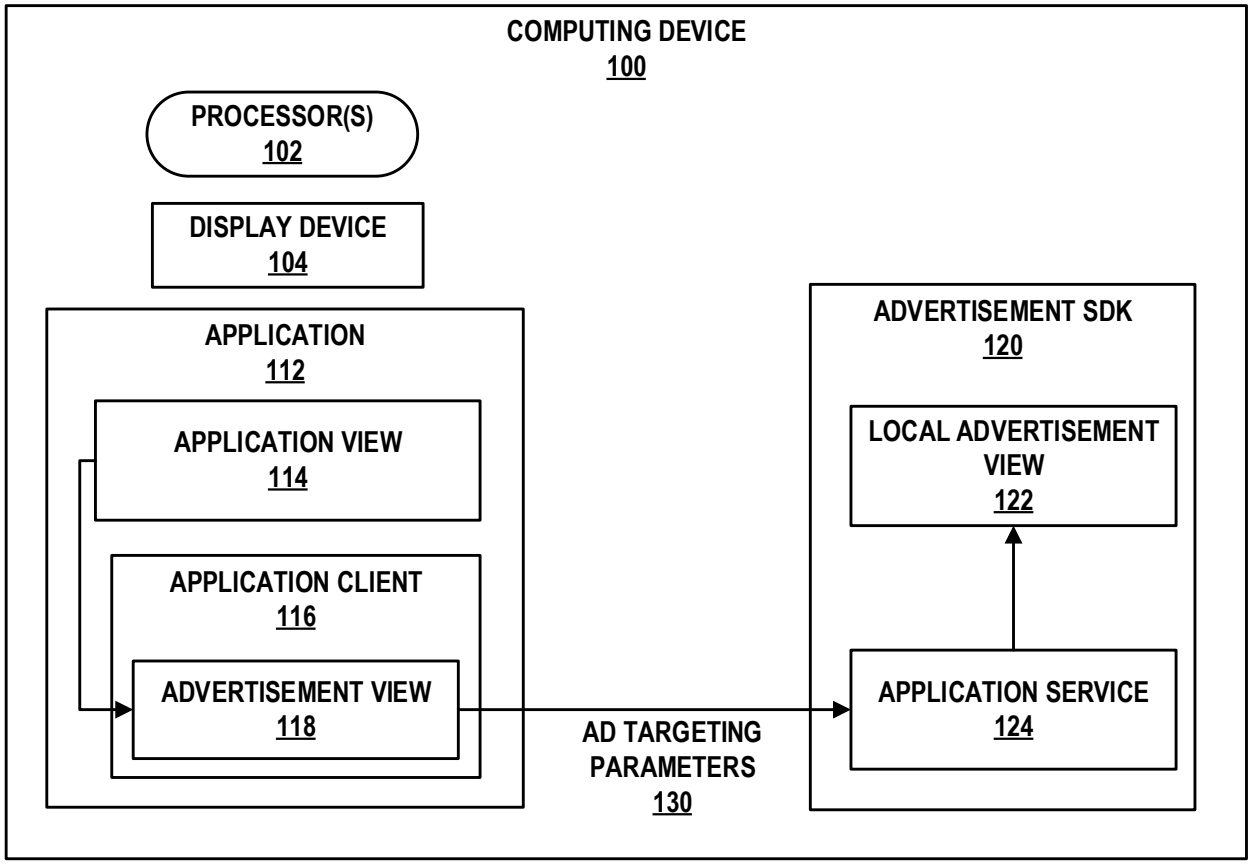

FIG. 1

Figure 1 illustrates a computing device 100 capable of running an advertisement software development kit (SDK) 120 and an application 112 that uses advertisement SDK 120 as separate

processes. In some examples, computing device 100 may represent a mobile device capable of downloading and installing one or more applications 112.

As shown in Figure 1, computing device 100 includes one or more processors 102, a display device 104 and one or more applications 112. Examples of processor 102 include, but are not limited to, digital signal processors (DSPs), general-purpose microprocessors, application-specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry.

Display device 104 of computing device 100 functions as an output device for displaying an application view 114 to a user. Examples of display include a liquid crystal display (LCD), a dot matrix display, a light emitting diode (LED) display, an organic light-emitting diode (OLED) display, e-ink, or similar monochrome or color display capable of outputting visible information to a user of computing device 100.

In some examples, application 112 represents a mobile application developed by an application developer. Application 112 may be configured to be downloaded and installed to a user device, such as computing device 100.

Computing device 100 may further include a communication unit capable of communicating with another computing device, such as a server device. Examples of communication unit include a cellular radio, a wireless network radio (e.g., WIFI™, BLUETOOTH ®, etc.), a network interface card (e.g. such as an Ethernet card), a cable modem, or any other type of device that can send and/or receive information. For example, computing device 100 may communicate with a server device via a network provider, such as a cellular provider, cable internet provider, or other network providers.

Advertisement SDK 120 represents an advertisement SDK developed an SDK developer, such as an advertiser. Advertisement SDK 120 is configured to be linked to one or more applications, such as application 112. Advertisement SDK 120 makes it easy for application 112 to request advertisements and display them in its application view 114.

As discussed above, it may be desirable to enable advertisement SDKs to be unbundled from applications that use the advertisement SDKs. At such, SDK proxy library 116 and application service 124 may be developed based on advertisement SDK 120 to enable application 112 and advertisement SDK 120 to run as separate processes. SDK proxy library 116 may be shipped together with application 112 to computing device 100 for installation. Application service 124 may be included in advertisement SDK 120. In some examples, a server device may receive advertisement SDK 120 and automatically generate SDK proxy library 116 and application service 124. In some examples, SDK proxy library 116 and application service 124 may be developed by an SDK developer.

SDK proxy library 116 is an embedded proxy SDK library configured to replace original calls to advertisement SDK 120 with inter-protocol communication (IPC) calls. In operation, when application 112 loads an advertisement, application 112 first generates a message requesting for an advertisement using the embedded proxy SDK library stored within application 112, such as SDK proxy library 116. Processor 102 of computing device 100 may then pass the generated message through an IPC call to advertisement SDK 120. This IPC call instructs application service 124 stored in advertisement SDK 120 to fetch the requested advertisement as well as contextual information about the advertisement. Application service 124 may create a local advertisement view 122 and store local advertisement view 122 in storage dedicated to advertisement SDK 120. For example, application service 124 may create a local

advertisement container, the control of which is delegated to advertisement SDK 120. Advertisement SDK 120 may create advertisement view 122 within its own memory space and context, which is displayed within the advertisement container. Application service 124 may then send data indicating local advertisement view 122 via an IPC call to the proxy SDK library. Processor 102 of computing device 100 may process the received data and instruct display device 104 to display an advertisement view 118 within a certain area of application view 114. Furthermore, advertisement SDK 120 can now verify that an advertisement is shown or clicked by talking to processor 102 of computing device 100 directly rather than relying on feedbacks received via application 112. In this way, application 112 and advertisement SDK 120 are configured to run as separate processes, as all of the privileged advertising-related operations would be performed by application service 124, which is stored on the advertisement SDK side.

In some examples, SDK proxy library 116 may be automatically generated by a server device. In this model, advertising SDK 120 may only access data with appropriate permissions. For example, SDK proxy library 116 may generate a message to be sent from application 112 to advertising SDK 120, where the message only includes the application ID of application 112 and unit ID representing the type of the advertisement to be loaded.

In other examples, SDK proxy library 116 may be personalized by an SDK developer. In this model, the SDK developer may program SDK proxy library 116 to specify data to be pulled from application 112. For example, SDK proxy library 116 may generate a message to be sent from application 112 to advertising SDK 120, where the message further includes additional parameters for targeting the application 112. For example, SDK proxy library 116 may generate ad targeting parameters 130 to be sent to advertisement SDK 120, where the ad

targeting parameters 130 includes location parameter indicating location of the user device and age parameter indicating age of the user for targeting application 112.

The described system addresses tampering threats such as click fraud. When applications and advertisement SDKs are bundled together, applications may manipulate advertisement views to hide advertisement displays in their entirety while simulating fake clicks that do not result from intentional user interaction. By using SDK proxy libraries and application services, advertisement views are generated and attested on the advertisement SDKs sides. Generating local advertisement view 122 on advertisement SDK 120 allows to prevent deceptive content from being shown over the top of local advertisement view 122, which prevents application 112 from manipulating with advertisement view 118. In addition, instead of tracking advertisement clicks through application 112, advertisement SDK 120 tracks advertisement clicks directly, which prevents application 112 from generating fake clicks.

The described system also provides users meaningful control over advertisement SDKs running on their devices. By using SDK proxy libraries and application services, the described system enables advertisement SDKs and a mobile application that use the advertisement SDKs to run as separate processes, which means a user may set different privileges for the advertisement SDKs and the application with which they are packaged.

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 by Paul Pearce, Adrienne Porter Felt, Gabriel Nunez, and David Wagner “AdDroid: Privilege separation for applications and advertisers in Android” available at https://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-59.pdf. As another example, the techniques of this disclosure may be combined with the techniques

described by Shashi Shekhar, Michael Dietz, and Dan S. Wallach "AdSplit: Separating smartphone advertising from applications" available at https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final101.pdf. Such a combination may be made for any suitable purpose, including, but not limited to, enabling advertisement SDKs to be unbundled from applications that use the advertisement SDKs.