June 21, 2019

Handling a User's Preferences on Permissions Required by an Application Software

Victor Carbune

Sandro Feuz

Handling a User’s Preferences on Permissions Required by an Application Software

Abstract:

This publication describes an operating system (OS) that uses a privacy layer for handling a user’s preferences on permissions required by an application software (application). The privacy layer injects mock data into application programming interface (API) calls when the application requests a permission-protected resource. As described herein, a permission-protected resource may include user data (*e.g.*, calendar, contacts data, notes, user-biometric data) and access to user equipment (UE) hardware (*e.g.*, gyroscopes, magnetometers, barometers, global navigation satellite system (GNSS) technology, proximity sensors, touchscreen sensors, biometric sensors, heart-rate sensors, thermometers, humidity sensors, radar technology, cameras, microphones) that the user does not want to grant access to the application. The OS gives the user the choice to enable the privacy layer for each application or enable the privacy layer across the OS. This privacy layer protects the user’s privacy and allows the user to utilize the application without sharing information that the user wants to keep private.

Keywords:

Application software, application, application programming interface, API, user preference, permission layer, privacy layer, user permission, application permission, noise injection, simulated data, fake data, modulated data, privacy, user data, user equipment, UE.

Background:

Operating system (OS) developers, application software (application) developers, application software markets (application markets), and user equipment (UE) manufacturers increasingly offer users more product features. A widely-used UE, such as a smartphone, enables the user to call, participate in a video-conferencing session, text, email, bank, invest, shop, search for information, consume several types of media, participate in social networking, play games, navigate to a location, and use a plethora of other applications. In addition, UE manufacturers often integrate accelerometers, gyroscopes, magnetometers, barometers, global navigation satellite system (GNSS) technology (*e.g.*, global positioning satellite (GPS)), proximity sensors, touchscreen sensors, biometric sensors, heart-rate sensors, thermometers, humidity sensors, radar technology, cameras, microphones, and various other sensors in or on the UE, which enhance the user experience and may play a role on the functionality of various applications. Furthermore, the OS, the application market, the UE, and applications help manage user data, such as contacts data, short message service (SMS) data, notes, calendar data, user biometric data (*e.g.*, fingerprint data, voice recognition data), credit card numbers, and other user specific data, which may also play a role on the functionality of various applications.

When the user installs an application, the application may request permissions to access user data (*e.g.*, calendar) and certain UE hardware (*e.g.*, microphones, cameras, GNSS, accelerometers). Figure 1 illustrates how the OS may prompt the user to accept or deny permission requests by an application when the user installs an application downloaded from the application market.

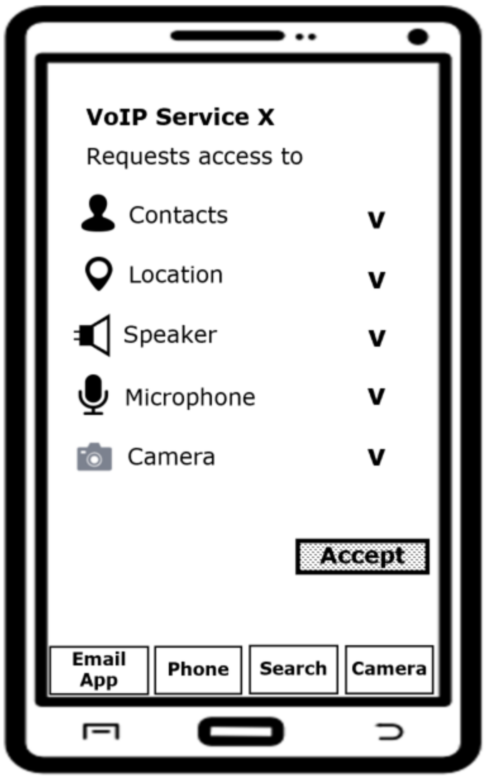


Figure 1

Assume Jane downloads a voice over internet protocol (VoIP) application software called *VoIP Service X*. Jane's friends are increasingly using *VoIP Service X*, and Jane wants to start using it to stay in touch with her friends. Also, assume Jane is aware that *VoIP Service X* allows users to make calls, send messages, track each-other's location, and communicate using video-conferencing sessions. As Jane downloads *VoIP Service X*, the OS and the application market let her know that this application requests permission to access her contacts data, her location, the UE's speaker, the UE's microphone, and the UE's camera, as illustrated in Figure 1. Jane understands that *VoIP Service X* requests permissions to access her contacts data, the speaker, the microphone, and the camera, and her location, because *VoIP Service X* supports features that require access to the requested permissions. Jane may be thinking, "I wish I can use *VoIP Service X* without granting access to my exact GNSS-tracked location because all I want to do is call, send messages, and participate in video-conferencing sessions."

The example *VoIP Service X* application, however, requires Jane to grant access to the user data and the UE features illustrated in Figure 1 before Jane can start using the application because part of the appeal of *VoIP Service X*, is for friends to keep track of each other when they are attending a busy event, such as downtown on a Friday night, in a concert, *Oktoberfest*, and so forth. Jane, like many other users, decides to avoid using *VoIP Service X* due to the required permissions upon installation or usage. Jane, however, can see the value that *VoIP Service X* can provide to her social life. Therefore, it is desirable to have a technological solution that enables users to install and use applications while protecting their privacy.

Description:

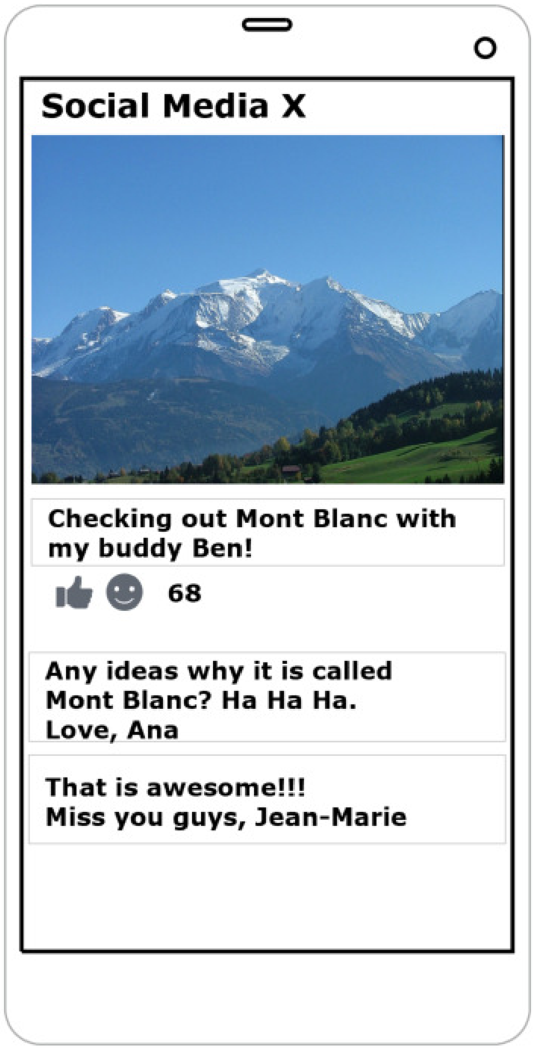
This publication describes an operating system (OS) that uses a privacy layer for handling a user's preferences on permissions required by an application software (application). The privacy layer injects mock data (*e.g.*, simulated, modified, truncated, pre-recorded, fake, noise) into the system application programming interface (API) calls when the application requests a permission-protected resource. As described herein, a permission-protected resource may be user data (*e.g.*, calendar, contacts data, notes, user-biometric data) and user equipment (UE) hardware (*e.g.*, gyroscopes, magnetometers, barometers, global-positioning system (GPS) technology, proximity sensors, touchscreen sensors, biometric sensors, heart-rate sensors, thermometers, humidity sensors, radar technology, cameras, microphones) that the user does not want to grant access to the application.

The OS gives the user the choice to enable the privacy layer for each application under an easily-accessible user setting. The user controls which permission-protected resource they want the privacy layer to protect. The OS may inform the application when the user chooses to protect access to certain user data and hardware. At this stage, the user, the OS, and the application are aware that certain data may be mock data in order for the user to utilize application features and protect the user's privacy. If, however, an application developer refuses to allow the user to use the application when the privacy layer is enabled for reasons that are not clear to the OS or the user, the OS may make the use of the privacy layer opaque (instead of transparent) to the application.

The OS supplies mock data on a case-by-case basis. For example, a navigation application requires exact GNSS-tracked location in order for the user to safely use the application. As another example, a racing game application requires real accelerometer-produced and gyroscope-produced data for the user to play the game. In such cases, the OS does not give the user the choice to enable the privacy layer on access to user data and hardware that are required to use the application as intended. The OS provides true data to the application when the real data is essential to use the application or the user has decided to grant the requested permissions.

Recalling Jane's example in Figure 1, the example *VoIP Service X* application requires Jane to grant permissions to access her contacts data, her location, the UE's speaker, the UE's microphone, and the UE's camera. The OS prompts Jane to select which permission-protected resource she wants the privacy layer to protect. If Jane wants to protect her location, the privacy layer provides real data to the *VoIP Service X* application from the UE's speaker, the UE's microphone, and UE's camera and provides mock data, such as Jane's city or a fixed location in a larger vicinity (instead of her GNSS-tracked location). If Jane wants to use *VoIP Service X* to only make and receive calls, the OS provides real data to only her contacts data, the UE's speaker, and the UE's microphone and provides mock data for her camera and her location.

In addition to the user's choice to enable the privacy layer for each application, the user may choose to enable the privacy layer across the OS. In that case, the OS uses the privacy layer depending on the features of each application. To show how the privacy layer may work when enabled across the OS, consider the example illustrated in Figure 2.



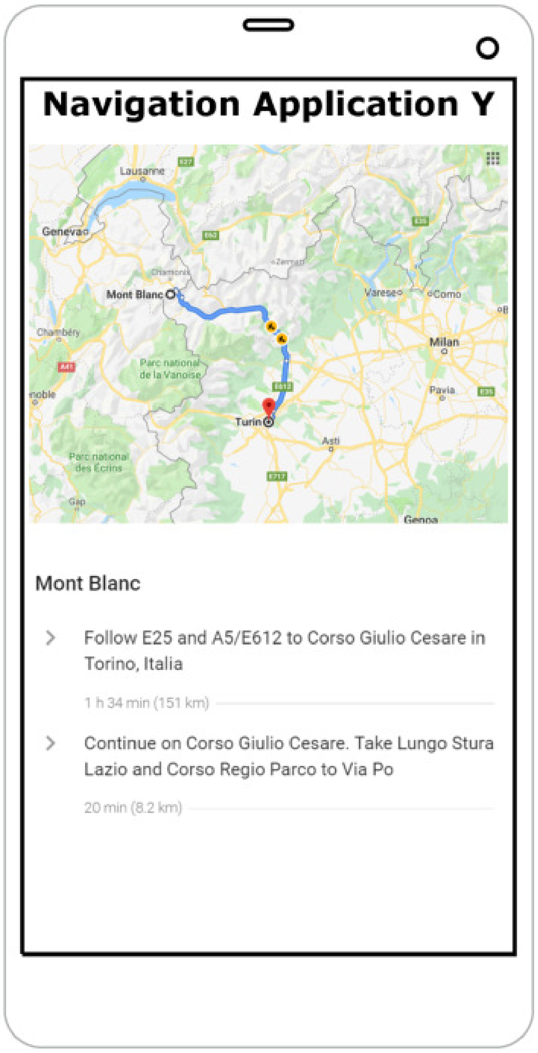
Social Media X

Checking out Mont Blanc with my buddy Ben!

68

Any ideas why it is called Mont Blanc? Ha Ha Ha. Love, Ana

That is awesome!!! Miss you guys, Jean-Marie



Navigation Application Y

Mont Blanc

Follow E25 and A5/E612 to Corso Giulio Cesare in Torino, Italia

1 h 34 min (151 km)

Continue on Corso Giulio Cesare. Take Lungo Stura Lazio and Corso Regio Parco to Via Po

20 min (8.2 km)

2A) Social Media X **2B) Navigation Application Y**

Figure 2

Assume Tom and Ben are on a road trip, and Tom is using his smartphone during the road trip. Tom has enabled the privacy layer across the OS of the smartphone. Tom and Ben have decided to leave their hometown, Geneva, Switzerland, to spend one night in Turin, Italy and a few days in Milan, Italy. As Tom and Ben get close to the border of France and Italy, they stop their car when they see *Mont Blanc* — the highest and the most famous mountain of the Alps. Tom decides to take a photo and posts it in the *Social Media X* application, as illustrated in Figure 2A. The example *Social Media X* application requires users to grant permission to access their location before they can use the application. The privacy layer, however, supplies a fake location in accordance with the user's wishes because location is not an essential part of the *Social Media X* application's functionality. Differently said, the *Social Media X* application does not know Tom's and Ben's location unless Tom explicitly decides to share his GNSS-tracked location.

As Tom and Ben drive from Geneva, through France, to Turin, they use a navigation application called *Navigation Application Y*, as illustrated in Figure 2B. Given that the GNSS-tracked location is essential to the *Navigation Application Y* functionality, the OS supplies real GNSS-tracked location to the *Navigation Application Y* application. Therefore, the *Navigation Application Y* application knows Tom's location, but the *Social Media X* application does not know Tom's location even though both applications require Tom to grant access to his location.

In conclusion, the privacy layer for handling a user's preferences on permissions to access user data and hardware required by an application protects the user's privacy and allows the user to utilize the application without sharing information that the user wants to keep private.

References:

[1] Chitkara, Saksham, Nishad Gothoskar, Suhas Harish, Jason I. Hong, and Yuvraj Agarwal. "Does This App Really Need My Location?" Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Vol. 1, No. 3, Article 42 (September 2017): doi:10.1145/3132029.