August 29, 2019

Protecting User Privacy by Monitoring API Queries

Matthew Sharifi

Bernhard Seefeld

Protecting User Privacy by Monitoring API Queries

Abstract:

This publication describes a method for protecting user privacy from targeted application interface program (API) queries made to a prediction service on a computing device. More specifically, the method involves utilizing an API Query Manager to monitor API calls made by applications to the prediction service. If the API Query Manager determines a lack of positive feedback (*e.g.*, screen content presented on a display of the computing device does not match the personalized prediction returned from an API call), then the API Query Manager can throttle further API calls made by the application to protect user privacy.

Keywords:

Applications, application programming interface (API), computing device, user privacy, targeted queries, machine-learned features, query services, user profile, user information, user consent, API call, personalization, prediction

Background:

On-device, machine-learning powered features (ML features) are increasingly common inside mobile applications and operating systems of computing devices (*e.g.*, wireless communication devices, computers, tablets). Examples of ML features include smart reply, text selection, smart action suggestions, and the like. Such ML features utilize an on-device prediction service, enabling them to work privately (*i.e.*, without sending information to a server).

The on-device prediction service is accessible to applications through an application programming interface (API). Applications may access the API to obtain and/or process personalized predictions through the use of API calls. Figure 1 illustrates an API call from an application on the computing device.

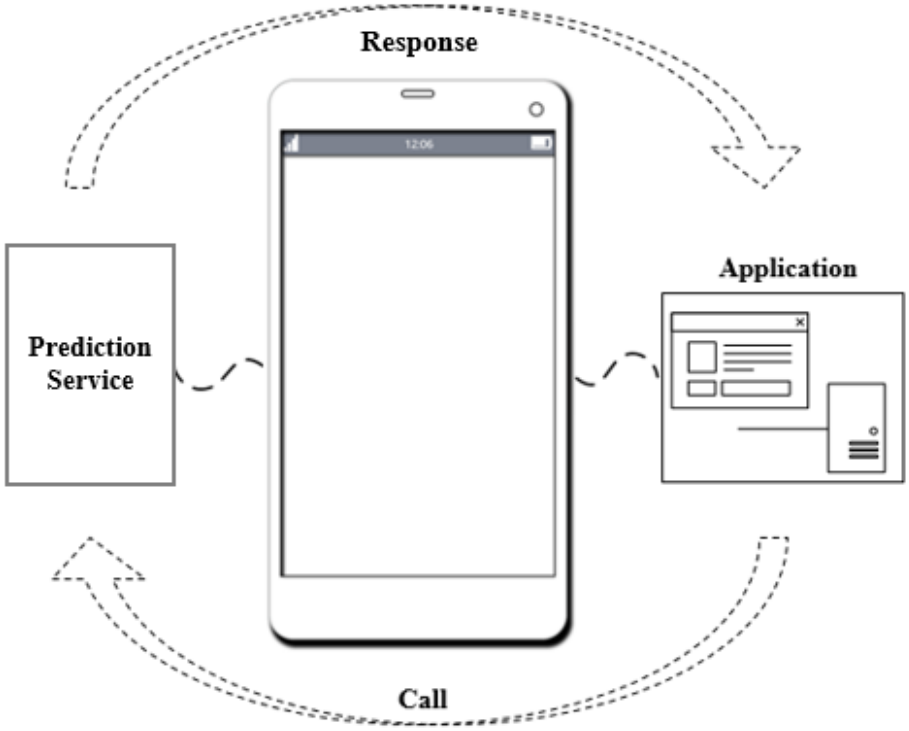

Figure 1

As illustrated in Figure 1, the computing device (*e.g.*, the computer-readable media of the computing device) includes a prediction service and an application. The application performs an API call to the prediction service, and the prediction service provides a prediction (*e.g.*, query results, an API response) to the application. The application may customize a service provided to the user utilizing the prediction. For example, a food service application may perform an API call to the prediction service for commonly searched entrees, to which the prediction service can

provide a prediction consisting of food query results. Thus, the application can suggest restaurants that the user may enjoy.

The risk associated with applications performing API calls and the prediction service providing predictions is an application may repeatedly query the prediction service in an attempt to extract user information (*e.g.*, a user profile). For example, the application may try to extract a user's preferred genre but may not need this information to customize services. It is desirable to protect user information by monitoring API queries to maintain user privacy.

Description:

This publication describes a method for protecting user privacy from targeted application interface program (API) queries made to a prediction service on a computing device. More specifically, the method involves utilizing an API Query Manager to monitor API calls made by applications to the prediction service, to the end that user privacy can be protected.

Figure 2 illustrates an example computing device and elements of the computing device that support the method of protection described in this publication.

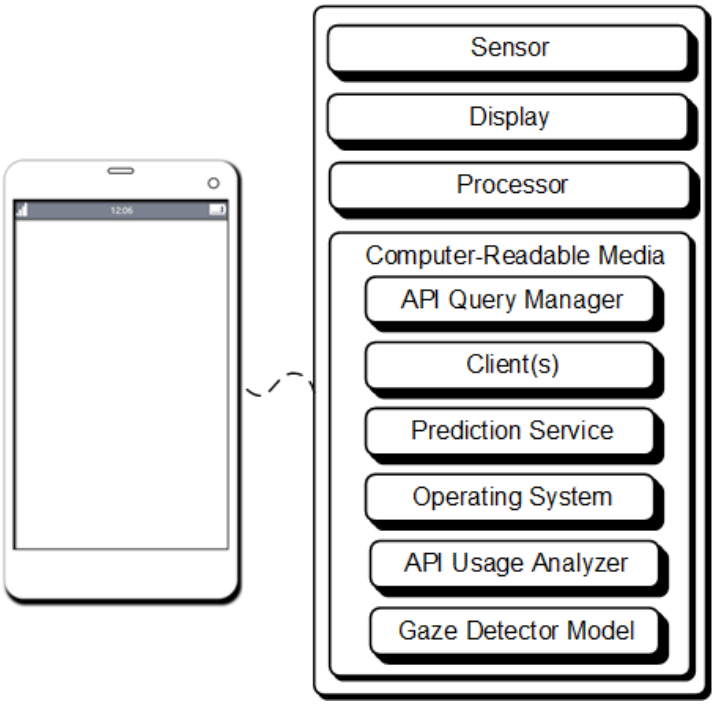

Figure 2

As illustrated, the computing device is a smartphone. However, other computing devices (*e.g.*, wireless communication devices, tablets, watches, laptops) can also support the methods described in this publication. The computing device includes a sensor (*e.g.*, a camera), a display, and at least one processor. The computing device also includes a computer-readable medium (CRM) which stores executable instructions. The CRM may include the operating system of the computing device, the API Query Manager, a gaze detector model (*e.g.*, an algorithm that detects the on-screen location of user gaze utilizing the sensor), and applications on the computing device.

The CRM further includes a prediction service. The prediction service hosts an API, enabling applications to access predictions generated by the prediction service. For example, the prediction service, based on inputs, may provide personalized predictions related to the user, such as commonly used words. When an application appropriately uses an API for personalization services, the results of the API call predictions typically result in changes to the information

displayed on the screen of the computing device. For example, the application, in response to the predictions, may present commonly used words as a smart reply suggestion.

The CRM also includes executable instructions of an API Usage Analyzer. The API Usage Analyzer continuously generates real-time information relating to application activity, information displayed on the display, and information relating to actions performed by applications. The analyzed information could include information within on-screen entities, application context information, and information displayed on the display of the computing device.

The API Usage Analyzer and API Query Manager work cooperatively to prevent applications from obtaining unnecessary information via targeted API calls to the prediction service. In more detail, between each API call made by an application, the information regarding the interaction of the application and the user (*e.g.*, screen contents, user input into the application, system-level events (*e.g.*, notifications)) is collected by the API Usage Analyzer and analyzed by the API Query Manager. Additionally, the API Usage Analyzer can log sensory information (*e.g.*, gaze detection) and the API Query Manager can analyze this sensory information (*e.g.*, the API Query Manager can use gaze detection to determine the presence or absence of user interaction with an application making the API call).

Utilizing this information, the API Query Manager determines if the application properly uses the prediction(s) provided in the API response. Proper usage includes the application utilizing the prediction(s) for customization (*e.g.*, to provide legitimate customized services or features to the user). Indications of customization include the display of some or all of the results from the API call on the screen of the computing device, indications that the user triggered the application to make the API call, and an external event/state change occurring (*e.g.*, an incoming notification). If the API Query Manager determines there is a lack of customization, then the ability of the

application to make API calls may be throttled, or the computing device may prompt the user for input (*e.g.*, confirmation that the user desires the sharing of the information with the application).

An illustration of the API Query Manager determining an application's lack of customization, despite the predictions provided by the prediction service, is illustrated in Figure 3 below.

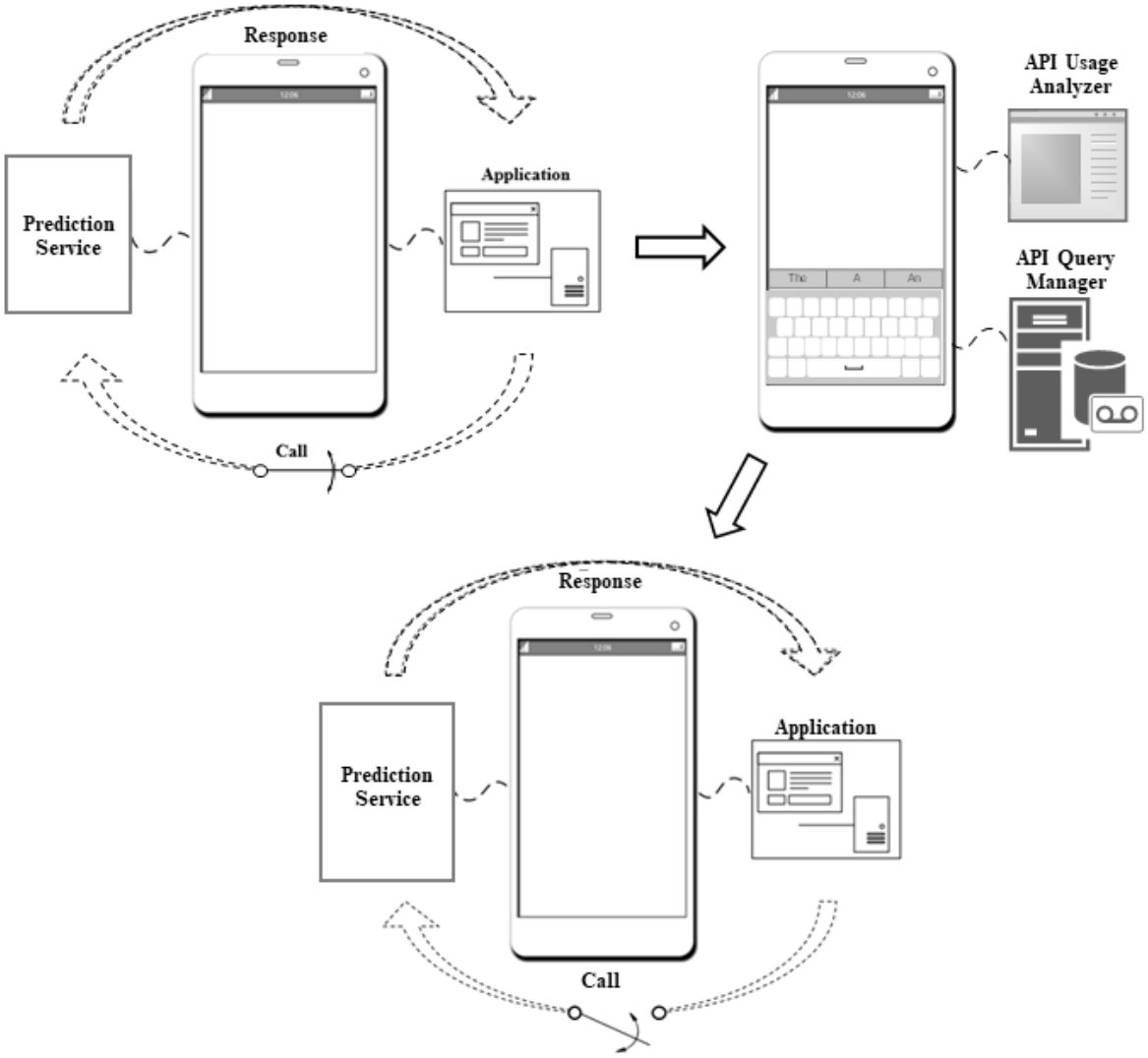

Figure 3

As illustrated in Figure 3, an application installed on the computing device from Figure 2 performs API calls and acquires predictions. The personalized predictions are not utilized to

customize the application, however. Lack of customization on the application's behalf, despite the prediction service providing personalized predictions, may manifest in many forms. For example, the application may perform an API call requesting commonly used words. In response, the prediction service can return a prediction consisting of commonly used words. The application may not display these words as suggestions; instead, the application may display generic words (*e.g.*, "The," "A," "An"). The API usage analyzer can log the on-screen information associated with application activity along with the returned prediction from the prediction service and provide all this information to the API Query Manager. The API Query Manager can identify an application's lack of customization despite the predictions returned. As illustrated, the API Query Manager can then throttle the number or frequency of API calls permitted by the application and, thus, protect the user's privacy.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, or features described herein may enable collection of user information (*e.g.*, information about a user's social network, social actions, social activities, profession, a user's preferences, or a user's current location), and if content or communications are sent from a server to the user. In addition, certain data may be treated in one or more ways before it is stored or used so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

In conclusion, the utilization of an API Query Manager working cooperatively with an API usage analyzer to monitor applications making API calls can protect user privacy.

References:

[1] Patent: US 10078803 B2. Screen-analysis based device security. Filing Date: June 15, 2015