Technical Disclosure Commons

Defensive Publications Series

February 14, 2017

Browser Based Multi-Account Notifications

Justin Lewis

Joseph Cohen

BROWSER BASED MULTI-ACCOUNT NOTIFICATIONS

ABSTRACT

A system and method are disclosed for accurately delivering notifications to devices with multiple users. The system uses a notification server and operates using cookies on the user device. When a user enables notifications, the device is registered with the server backend. Such info is then copied to the client device and stored in a cookie. At the time of notifying a particular user, the notification service will retrieve a list of devices registered to that user id. It then sends a notification fingerprint indicating the notification payload and intended account to the device. When the client device receives the push notification, it pings the server requesting the payload contents, the ping including data from cookies describing the multiple accounts that are logged in, and their active status. The server then sends the notification to the device which then distributes the notification to the specific users active on the device.

BACKGROUND

Multiuser sign-ins are becoming increasingly common on devices. Many operating systems and web properties allow users to remain signed in with multiple accounts at the same time. When multiple users are signed in, there is a significant risk they'll be delivered redundant notifications. For example, with multiple signed-in users, if a video upload alert is sent to one user, it will be sent to the other signed-in users if they both subscribed to the same channel. This not only wastes bandwidth, but also frustrates the user with multiple notifications for the same event. This user frustration will lead to notification blocking, which will reduce user engagement. Currently some websites handle this issue in web by only delivering notifications to the primary user. While this eliminates redundancies, it reduces the opportunities to engage with the person using the device. As multiuser sign-ins become more prevalent, the issue of redundant notifications will grow.

Unlike mobile OS notifications, browser notifications do not have account level ACLs. This is for two reasons: first, most web applications do not support multi-sign in features, and second, web applications use a diverse range of authentication techniques (most of which the browser is not aware of). Browser notifications in older browsers contain a payload. Instead, the notification pings the client device which must then do a round trip to fetch the payload contents. This is significant because there is no way to determine on the client side who the notification was intended for, making it difficult to deliver notifications to multiple users on the same device. In newer browsers where payloads are supported, the issue of similar notifications being received for different users on the same device remains.

DESCRIPTION

A system and method are disclosed for accurately delivering notifications to devices with multiple users. The system uses the notification server and operates using cookies on the user device. The server-side operations for creating a custom notification are illustrated in FIG. 1. First, when a user enables notifications, the device is registered with the server backend. The presence of this registration can then be copied to the client device and stored in a cookie for faster access. When a new user logs in on the device or when the device is first registered, a mapping from the user identifier to the device identifier is then registered on the backend. The registration is written to the user cookies for convenience (to easily determine if the account and device are already registered).

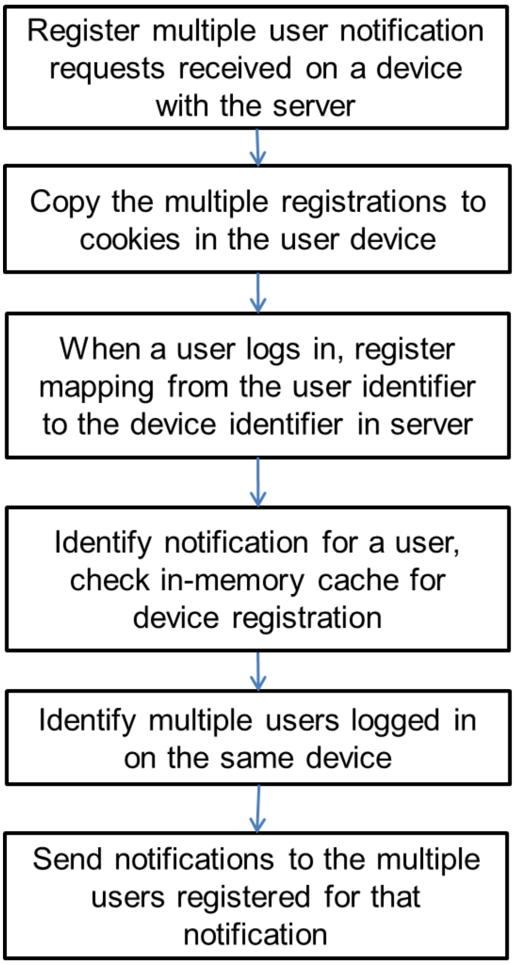
Register multiple user notification requests received on a device with the server
Copy the multiple registrations to cookies in the user device
When a user logs in, register mapping from the user identifier to the device identifier in server
Identify notification for a user, check in-memory cache for device registration
Identify multiple users logged in on the same device
Send notifications to the multiple users registered for that notification

FIG. 1: Server-side registration for sending custom notifications to multiple users on a device

At the time of sending a notification for a particular user the notification service will retrieve a list of devices registered to that user id. The server and device side processes for sending a notification are further illustrated in FIG. 2. The system will first check an in-memory cache table entry for registration of the client device. If the device is registered as a key in the cache table, then it will check each of the notification fingerprints attributed to the device. If the notification fingerprint is already present, then it does not need to send a notification to the client device so it will only append the user identifier to the notification data identified by the notification fingerprint. If the device is not in the cache table or the notification fingerprint is not attributed to the device, then the notification service will send a notification (without a payload) to the client device and record the notification fingerprint under the device. This recording also includes the notification payload and intended account.

When the client device receives the push notification, it pings the server requesting the contents of the payload. This ping includes data retrieved from cookies on the device describing which accounts are logged in and which account is active. When the server receives the payload request, it determines which notification payload to send, from the multiple notifications stored on the server. To do this it looks up the device id in the server side cache and retrieves the list of notifications registered.

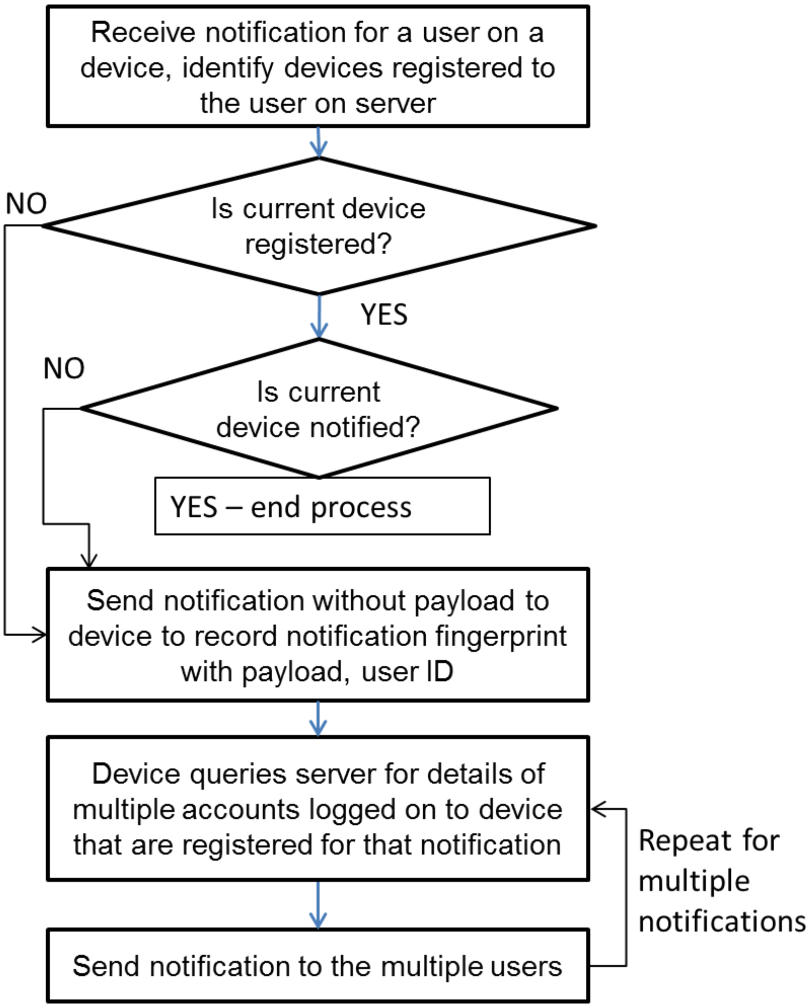
Receive notification for a user on a device, identify devices registered to the user on server
Is current device registered?
NO
YES
Is current device notified?
NO
YES – end process
Send notification without payload to device to record notification fingerprint with payload, user ID
Device queries server for details of multiple accounts logged on to device that are registered for that notification
Repeat for multiple notifications
Send notification to the multiple users

FIG. 2: Multiple user notification method

For each account starting with the currently active account it then loops through each of the notifications listed and checks if the account is listed as a potential recipient. If there is match between the signed-in account and an intended account listed in the notification list, then the notification payload is sent down. At the time of sending down the notification payload, the notification details will be cleared from the local cache.

In some instances, an identical notification may need to be sent to multiple accounts. For example in a group chat two users might be recipients of a notification of a chat from a third user. In these cases a user identifier is included in the notification fingerprint so that the two notifications do not share notification details and de-duplicate each other. One advantage of the disclosed method is that multiple identical notifications are only sent to one user account. For example if 2 user accounts are subscribed to the same channel, and that channel uploads a new video then only one of the two user accounts needs to be notified for the notification to reach both the intended recipients. With this method only the user account that has been active most recently will be notified, thus reducing redundant traffic. In some implementations, all relevant notification payloads are sent down with each payload request. Relevant payloads still have to be de-duplicated and a signed-in user still has to be authenticated before the payloads can be sent down. In these implementations, the client can receive multiple notification payloads and simply render them as independent notifications. The client or server might group the notification payloads based on the intended recipient account.

The system and method are a powerful way to extend user engagement with relevant content on devices where multiple users are logged in. The system also reduces multiple notifications to the same user and improving communication efficiency, both by combining notifications for either a single user or multiple users.