

EXHIBIT 43

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

Platform 3.0 Plan

The following outlines the changes that we are planning to announce with the roll out of Platform 3.0. We will announce all of these changes out at the same time in a single blog post that starts the clock ticking on a 90-day migration. This 90-day migration window is the standard that we adhere to in Platform and helps developers handle large changes in features (this is especially important for native mobile developers which have a much longer lead time). We will make all API/permissions changes available in the migration so developers can test against these changes. We will make the review tool and new APIs available before the migration takes effect. We will make all tooling/policy changes available on the day the migration is turned on by default.

Timing

We have the following rails that will inform the timing of our announcement and roll-out. Our next migration window opens on June 26. If we decided to snap to this date that would put these changes live on Oct. 2nd. In terms of implementation, looking at the schedule for app review and SWAG for the APIs, it appears that the needed functionality should come online in August. Based on the above, a reasonable schedule is the following:

1. June 26: Announce Platform 3.0 on the developer blog, making the API/permission changes available under a migration.
3. Aug ??: Bring the App Review tool and new APIs online start testing with a limited set of developers.
4. Oct. 02: Default the migration to on and review all new apps.
5. Oct. 14: Begin to review apps for policy violations.

[Note: Need to talk to Sean about the impact of these changes on games developers to see if this schedule works. We may need to accelerate the API development. Ideally we could be in position to have the new APIs in beta when we announce the migration.]

Changes to Permissions and APIs

Facebook Replacement APIs: We are removing the ability to request permission and read data from the user's stream, notifications, and inbox (message/thread). These features are used primarily for apps that attempt to recreate the Facebook experience on other devices. We will still make these APIs available to a limited number of select partners where it makes sense (phone OEMs, etc).

Friends Data: We will reduce the scope of the friends' data that a developer can request and access from users. Specifically, we will change the /friends Graph API connection to only return the user's friends that are already connected to the app. In addition, we will no longer support friends_* permissions or data access.

New APIs: In order to help developers address scenarios for which they utilize the above APIs, we are going to make two new APIs available: Suggested Friends and Recommendations. Suggested Friends will help developers pick the right app friends and non-app friends to invite to the app. Recommendations will help developers recommend various OG objects based on their friends likes and OG activity.

App Review & Reciprocity

Since the launch of Open Graph in Jan. 2012, we have moved toward an app review model where we review and approve an app's integration with Facebook social channels (News Feed, Timeline, etc.). We extended this model to the App Center. With this announcement taking the next step in this evolution. In 90 days, we will begin to review and approve all apps that integrate with Platform. This will ensure that we are maintaining a high-level of app quality and that our user and developer interests are aligned. Developers may continue to develop and test on Facebook Platform as they always have, but before they can take their app "live" to non-developers/testers, their app must be approved and reviewed by Facebook.

As part of this review process, we will examine the quality of the app, but also if the app is in compliance with our policies. In particular, we will determine if the app is following our reciprocity and duplicative functionality policy. All apps may use Platform for Login and Social Plugins, but if the app accesses extended user information such as the friend graph, photos, etc. the app must also make it possible for the user to bring their data from the app back to Facebook. In order to help developers with this requirement, we are releasing tools collectively known as Action Importers.

Further, for the small fraction of developers whose app may duplicate existing FB functionality, we can make this determination at review time, before the app launches, to ensure that they can work together to see if we can come to an equitable resolution.

Canvas Redirect Policy

We have had a long-standing policy prohibiting canvas apps from redirecting outside of FB. As the ecosystem has developed, we have seen more and more web sites create canvas app with the sole goal of gaining access to bookmarks and requests for their web apps outside of FB. In 90 days, we will begin to enforce this policy in earnest. Canvas apps that exist that chiefly exist to redirect to external web sites will be disabled. For developers that were relying on this mechanism to gain access to requests, we recommend that they utilize the Send dialog to implement their request/invitation functionality.

Page Apps (optional/for discussion with Mike/Dan)

There are a number of apps that are utilized to manage pages. Previously, an app could ask for the manage_page permission and access the page feed, post to the feed, etc. Moving forward, securing access to this permission and the Page API is limited to specific apps that offer compelling functionality above and beyond what is available on Facebook. For developers building page management apps, they can use this permission for developer/tester owned pages during development, but need to be explicitly approved via our App Review before they can ask any user for this permission or manage pages on their behalf.

Canvas as Game Platform (optional/for discussion with Mike/Dan)

As we have watched the development of the 3rd party ecosystem on Facebook.com, a clear pattern of usage has emerged. Canvas has become the default home for social games and Page tabs have become the default home for brand, contest, etc. apps. Moving forward, we are going to codify this within our product itself. Only new game apps will be able to leverage canvas (existing non-game apps are not effected). Non-games will continue to have access to page tabs.

Platform 3.0 Rules of the Road

As we work towards implementing the decisions that we made last year, which are now known as Platform 3.0, we need a common framework by which we can make decisions about what types of app to give access to Platform. This framework must address three key questions: what are the broad principles of our platform, how do these manifest in our products/policies and how do we communicate this to developers? This document answers these questions, constituting the Platform "rules of the road".

Principles

The fundamental principle that governs Platform usage is a simple concept: reciprocity. Reciprocity involves an equable value exchange between a 3rd party developer and Facebook. This value exchange involves one of the following from developers: high-quality experiences that FB users can use to tell great stories to their friends and family on FB and/or monetary value in the form of revenue sharing or direct payment. In return, Facebook offers a developers access to our Platform.

When considering the implications of reciprocity it is important to note that a second order principle quickly emerges: competitive access. There are a small number of developers whom no amount of sharing to FB or monetary value can justify giving them access to Platform. These developers do not want to participate in the ecosystem we have created, but rather build their own ecosystem at the expense of our users, other developers and, of course, us. That is something that we will not allow.

Platform Services

In order to outline how the above principles manifest in our products/policies, we need to identify the various parts of Platform. This is required because we have a disjoint set of product and policy constraints that apply to each of these different areas:

App services: these are paid generic services (storage, compute, etc.) that apps may use to build the core foundation of their app. At present, we do not have an offering in this space, but we are close to closing an acquisition that adds these services. As such, in order to be complete and future-proof we will outline the rules associated with these types of services.

Ads services: these are paid promotional services that enable developers to drive awareness and installations of their apps using News Feed and other paid channels. We have always had an advertising business the developers could leverage, but this is increasingly an area of focus for us with the transition to mobile.

Identity services: these are the traditional identity/social services that we have provided to developers since 2007. These services enable developers to build personalized app experiences for people and enable these people to share aspects of those experiences back to Facebook. [todo: payments is here]

Application

The following outlines the application of the above principles to the various kinds of platform services.

Strategic competitors: We maintain a small list of strategic competitors that Mark personally reviewed. Apps produced by the companies on this list are subject to a number of restrictions outlined below. Any usage beyond that specified is not permitted without Mark level sign-off.

Ad services: All developers, save strategic competitors (above), may use our ads services. The reciprocity for these services is clear: money in exchange for new or re-engaged users. In terms of oversight/policy enforcement, we follow the standard ads creative review process.

App services: All developers, save strategic competitors (above), may use our app services. The reciprocity for these services is clear: money in exchange for CPU, data storage and network bandwidth. In terms of oversight/policy enforcement, we will reactive handle any strategic competitors that we discover using these services.

Identity services: this set of services is the subject of much of the rules of the road. This is due to the fact that we have a variety of mechanisms for value exchange.

All developers, **including strategic competitors (above)**, may use the Login and Social Plugin features available within identity services. We permit this because we wish to see our core login service and basic sharing services used by users in any app, creating an equitable relationship with the all, including competitive, developers. To this end, we make these features available to developers with out app review.

The use of identity services, beyond Login and Social Plugins, is subject to app review. We review the apps usage of our APIs to determine if they are adhering to our principles. In particular, we look at the quality of the app's user experience and if there existing some equitable value exchange.

During app review, we determine the quality of the app by using the app and comparing the experience to our quality guidelines [link]. Apps that do not meet our quality bar are rejected.

During app review, we examine the APIs that the app uses in order to determine what the appropriate level of reciprocity. The guideline for this review is "take data, give data". The review tool is built to help with this assessment in that for every read API used by the app, we flag if the app has also implemented Action importers. Using this tool, as well as an examination of the user experience itself; we can determine if the app is reciprocating. If they are not, the app is a "data leach" and will be rejected.

Open Issues

There are a number of different fields like birthday, hometown, etc. that apps can request and there is no way for them to write back anything. We can do a couple of things: create an API to set this info (maybe not a bad idea for identity growth?), limit these data fields to just canvas apps (the value exchange is time on site and maybe payments), pull these fields, something else?

How do we think about the baseline level of value exchange of canvas apps due to time on site? Is that enough to forgo the "take data, give data" mandate for non-payment games/apps?

If you are offering real/world goods for sale on your web site or mobile app, in order to use identity services, you must use Facebook Wallet (Payments 3.0)?

Registration Plugin

Need to talk to the field about not selling FB platform as part of an

ads deal (this is where we are seeing non-games canvas use)

Group Management APIs? Event APIs?

Todo/Notes

Read

Login (uid, name, profile pic, small # of core fields) - anyone can get this. No a priori review.

User Data - requires user_* permissions. Ability to ask for those user_* permissions requires unified review.

Friend List - Requires unified review. If you access the friend list, you must conform with Social Reciprocity, as defined above.

Friend Data - we're removing this (removing friend_* permissions)

Core Facebook Features (News Feed API, Inbox API, Full Search API, etc.) - requires unified review. Generally only available with a business deal, generally limited to Facebook replacement apps.

Write

Share Dialog - anyone can use

Open Graph (defining actions, using the API) - requires unified review

Other Write/Management APIs (events, groups, etc.) - requires unified review. Generally requires a business deal, only available to Facebook replacement apps.

Distribution

Bookmarks - limited to canvas apps + mobile games. Requires unified review.

Requests - limited to canvas apps + mobile games. Requires unified review.

Notifications - limited to canvas apps + mobile games. Requires unified review.

App Center + Search - limited to apps that have gone through unified review.

Messaging (Invitations) - will require unified review. Available to anyone who abides by our rules (does not spam the channel).

Reciprocity language: "Facebook Platform provides an extensive set of APIs that allow users to bring their data with them to your application. If you use any APIs beyond the basic login APIs, then you must also allow users to bring their data from your app back to Facebook by implementing the Action Importer spec."