

EXHIBIT 114

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From: Ilya Sukhar </O=THEFACEBOOK/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=ILYAS8A7>
Sent: Sunday, August 25, 2013 4:54 PM
To: Bryan Klimt; Kevin Lacker; James Yu
Subject: Re: heads up about a rant

Apologies for the slow response. I was off the grid in Big Sur. My response would've largely been the same as Kevin's.

It does suck that Platform has been built in a way that led to this situation. I can assure you that your concerns are our concerns and they are not going unheard. I talk about this in every single meeting I have with Mike, Doug, Zuck.

Lobbying to reverse this particular decision is not the most productive thing we can do here. The underlying organizational issue would come up again and again in a bunch of different APIs in the future. There are beginnings of a plan to "reset" Platform to a state where the rest of the company is happy to provide a rock solid stability guarantee to developers.

So, I'm glad you're holding this rant because there's just a lot of things in flight day to day regarding this whole issue. We made a pretty good breakthrough in our lobbying late last week and should sync in person.

From: Bryan Klimt <bklimt@fb.com>
Reply-To: Bryan Klimt <bklimt@fb.com>
Date: Saturday, August 24, 2013 2:09 PM
To: Kevin Lacker <lacker@fb.com>, Ilya Sukhar <is@fb.com>, James Yu <jamesyu@fb.com>
Subject: Re: heads up about a rant

Thanks for the thorough response. I will delay posting this until after we've had a chance to talk about it. I think Tom and David would benefit from hearing your response as well, since they've had the same concerns as I have.

Sent via the Samsung Galaxy S™ III, an AT&T 4G LTE smartphone

----- Original message -----

From: Kevin Lacker <lacker@fb.com>
Date:
To: Bryan Klimt <bklimt@fb.com>, Ilya Sukhar <is@fb.com>, James Yu <jamesyu@fb.com>
Subject: Re: heads up about a rant

Hmm. Well I basically agree with 80% of this rant. But, my suggestion is to hold off on posting this rant, because I think I have the answers to a few questions that you are missing here and I think you will agree with my suggested strategy once I fill in more answers.

My initial reaction to the friends thing was also, hey this is retarded. But in the course of complaining I have gotten a lot more context.

"Stealing the graph" is precisely the reason. It is neither of the other two. The concern is specifically that big competitors like specifically Google and Wechat and the various Wechat-like competitors will manage to scrape the graph of friend connections.

"That sends a clear message to developers: Facebook Platform comes second to Facebook the Social Networking Product."

Well, it's awkward but that is just true. The internal deal that made it possible for platform to exist in the first place was that the social networking product could demand the revocation of any feature at any time, even if it sucks for developers. In return, platform could build whatever APIs they wanted. The deal is sort of an implicit thing, but it's how it works. The social networking product is now demanding the revocation of friends.get and also the ability to read one's feed. This will also break products like Flipboard that provide an alternative to reading Facebook content on Facebook.

"why should any of us work on a product that could be crippled at any time to benefit another team?"

Indeed, platform has been losing engineers for some time for this reason. This makes the remaining engineers unhappy, and one of the reasons I hesitate to say go ahead and post this is that I think this part of your rant feels like poking the remaining folks in the eye.

So, the things that suck, I agree that they suck. Where I disagree is the "ask". The rationale of "this makes Platform terrible" is not sufficient to make the social network side of the business take back their requests for this breakage. They would rather have the platform shut down entirely, and under the terms of the deal, they could have that if they wanted.

What I think we should do and what Ilya and I have been pitching and what I think is actually possible and going to happen is to revisit the terms of the deal between platform and the social network. We need to be able to offer features that do not break at any time, and we need the social network to accept that. We need features that, if they really are being revoked, at least we can keep them maintained for a period of years, and not break mobile developers' apps. In return, the other part of the deal will have to be stricter. Instead of "platform can build whatever APIs it wants", new features will have to be either pre-vetted by the social network side or there must be some fallback plan where they do not actually break if they are bad for the business.

How we get there from here is an interesting question, because the current surface area of platform is not something that platform can even maintain, let alone get the social network to commit to. But there are parts that the social network will commit to. So there are a whole lot of execution details here and perhaps at some point rather than counter-ranting it would be more effective for us to chat in person.

Tldr I hope this has been compelling enough to get you to delay posting this rant ;-). Let me discuss this with you in person first.

From: Bryan Klimt <bklimt@fb.com>

Date: Saturday, August 24, 2013 6:58 AM

To: Ilya Sukhar <is@fb.com>, Kevin Lacker <lacker@fb.com>, James Yu <jamesyu@fb.com>

Subject: heads up about a rant

Hi guys,

I just wanted to give you a heads up about a status I'm planning on posting later today unless I hear a really extremely compelling reason not to.

Also if you have any particular feedback about it, I'm glad to hear it.

Thanks,
-Bryan

#####

Hi folks,

I need your help. I'm trying to write a post about how bad an idea it would be to remove the api that lets you get a list of the user's friends from Facebook Platform. In order to illustrate my point, I'd like to satirically suggest removing some API that is so core to the developer experience that removing it would be ridiculous on its face.

For example, removing the Windows API method that lets you create a new window. Or removing the Twilio API method that lets you send a text message. Both suggestions are utterly insane. The problem is, for Facebook Platform, removing the method to let you get a list of friends literally is already that ridiculous. I can't think of an example more ridiculous to parody it with.

Before we discuss it in more detail, I'd like to clear up some misconceptions about the deprecation. I've heard some rumors floating around about why we are doing this. But many of them are clearly pabulum designed to make engineers think this decision has solid technical reasons. It does not.

1/ This API can be abused so we should remove it.

False. That is a non-sequitur. Lots of our APIs can be abused. Our whole product can be abused. That's why we have one of the best teams in the industry at detecting and stemming abuse. That team, plus Unified Review, is more than sufficient to deal with any theoretical abuse coming from this API. Even if this were true, who wants to be in that classroom where the whole class is punished for the transgressions of a few?

2/ It's okay to remove because we've provided alternatives for the common uses.

False. If you think that's true, then I don't think you realize why developer platforms exist. If we wanted to limit Facebook to the set of use cases we've already imagined, we could just do that ourselves, and not even have a Platform. The purpose of a Platform is to let people build new things on top of it. It's to enable the whole universe of ideas that anyone in the world could think of. Developers out there will have all sorts of crazy ideas. We want them to build those crazy ideas on top of Facebook.

Do you know why Facebook was originally built for the WWW instead of being part of CompuServe or AOL's proprietary networks? It's because the web is an open and extensible platform. It lets developers make their craziest dreams become reality. Tim Berners-Lee never imagined someone coming along and building what Facebook is today on top of it. If we make our Platform a rigid box and try to force developers into it, all we're going to do is encourage them to jump out of that box. And Facebook will be the next AOL.

So, if neither of those reasons explains why we are doing this, what's driving it? The only reason I've heard that makes sense it that we are worried about people "stealing the graph". We are doing this as a protectionist grab to make sure no one else can make a competing social network by bootstrapping with our social graph.

Okay, so let's assume for a minute that the social graph does belong to us, and not to our users. And let's even go so far as to assume that this is a real problem, although I'm not convinced it is. I mean, concerns that other companies will steal our friend graph may just be paranoia. But for the sake of argument, let's say it's not.

Then what? We're removing the core API in our developer platform. Out of concerns that someone will steal our social network product. That sends a clear message to developers: Facebook Platform comes second to Facebook the Social Networking Product. This has been a criticism all along with our Platform. When you go read the blog posts critical of our Platform, they all hit on this same point. When our APIs are subjugated to the whims of our other products, they can't be stable. And an unstable platform isn't really a platform at all.

So then you are left with 2 big problems.

1/ How do you convince external developers to build on a platform where the most basic core APIs may be removed at any time? I mean, the only big value we bring to the table right now is in distribution and discovery, and that's going to encourage developers to do only the most superficial integration with Facebook. Basically, they're going to do just enough to be able to use Neco ads.

2/ How do you convince internal developers to work on Platform knowing it's only ever going to play second fiddle to the rest of the company? I mean why should any of us work on a product that could be crippled at any time to benefit another team? If I worked on Platform, I would be seriously reconsidering my options if this API gets deprecated.

You might wonder why I care about this issue enough to write all this. After all, I work on Parse, and this doesn't directly affect Parse. It's simple. I care about developers. I care about Facebook Platform. And I care about Facebook. I want to be proud to tell people I work at Facebook and on the Facebook Platform. That means building something developers love. And I want Facebook to grow the way only a platform can. I want Facebook to be the next Windows, not the next CompuServe.

If you agree, feel free to reshare this post or a link to this post within Facebook.

Thanks,
-Bryan