EXHIBIT 134

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From: Constantin Koumouzelis </O=THEFACEBOOK/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=CONSTANTIN KOUMOUZELIS>
Sent: Friday, August 23, 2013 6:22 PM
To: Mike Vernal; George Lee; Ling Bao; Chris Daniels; Ilya Sukhar; Douglas Purdy; Kevin Lacker; Jason Clark; Vishu Gupta; Jeffrey Spehar; Harshdeep Singh; Eddie O'Neil; TR Vishwanath; Zhen Fang; Constantin Koumouzelis; Vladimir Fedorov
Subject: Message summary [id.1376659155890017]

Douglas Purdy:
>Constantin is going to own the spec for this (pushback if you disagree).

Douglas Purdy:
>To be clear, Constantin already agreed. :-)

Douglas Purdy:
>In terms of the SLAs, what is the Parse SLA today? Should we make these the same in some way?

Ilya Sukhar:
>Nah, our SLA doesn't really map well and varies depending on how much people pay us.

Douglas Purdy:
>Should it?

Ilya Sukhar:
>It's also entirely uptime focused. Is that the sort of SLA you're talking about? Or not-breaking-things?

Douglas Purdy:
>i think of those as the same thing.

Douglas Purdy:
>we release an api. your write code. the api is supported. it goes down. we have to fix it. the SLA is what we commit to.

Douglas Purdy:
>now there is a versioning window that may impact the SLA (old APIs get worst SLAs or no SLAs at all).

Ilya Sukhar:
>Yep, fair enough. I guess my point is that in reality the formal Parse SLA is quite lax for Enterprise customers and formally non-existent for self serve customers. We just over deliver. So, in that sense, yes -- they should be similar :)

Ilya Sukhar:
>I think we should definitely consider these new SLAs in light of a future where we might want to snap Parse to them. Totally up for considering that.

Douglas Purdy:
>i think you have something here however. we are still thinking of these these as bugs, they are not, they are service disruptions and an alarm should rung until we get it fixed. everything else is done is a version and that has long and graceful window. Beta can have bugs, Core has sevs.

Ilya Sukhar:
>(y)

Douglas Purdy:
>http://aws.amazon.com/ec2-sla/

Ilya Sukhar:
>Yeah, that's a good doc. We roughly copied them for our Enterprise customers.

Douglas Purdy:
>that is likely to extreme for us (as part of the social core) but i think that we should think about the SLAs in a way that we can 1) commit to broadly across all of Platform (including Parse) 2) be honest about what we can really do.

Vladimir Fedorov:
>I go to vacation and fun starts. I agree with dropping Platform 3.0, but I think we need better external name than Simplification:) internally sounds good. There is a ton of details here which are hard to follow on from vacation, so I reserve the right to randomize things a bit once I am back :) directionally agree

Harshdeep Singh:
>Marie prepared this list of Graph API endpoints, FQL tables and the permissions needed to access them. This would be useful for a detailed bucketizing of APIs into core, beta and partner/kill - https://www.facebook.com/download/500596376700650/API_FQL_Permissions_Dev_Segments_matrix.xlsx

Harshdeep Singh:
>This only covers the read APIs

Michael Vernal:
>Neither Platform 3.0 or Simplification are external names - to be clear.

Constantin Koumouzelis:
>Nor Stable Core correct?

Constantin Koumouzelis:
>I think all of these are good monikers for us to use internally but we should do a pass with content strategy / pmm before finalizing any names here

Michael Vernal:
>Correct.
>
>Also - yes, we should do an analysis of the write APIs as well. Generally, I think our principles should be:
>
>- We want people to be able to share into the graph (core value prop for us + them + people)
>
>- It's probably important that you can read-back what you wrote (e.g., Nike cheering stories, etc.). Maybe you can even read back _everything_ you wrote (e.g., maybe we should keep /me/feed but only scope it to your app_id)
>
>- There might be some minimal things you need to be able to query for sharing scenarios (people for tagging, places for tagging, etc.).

Douglas Purdy:
>Constantin: let's use this thread to review the Zuck deck.

Michael Vernal:
>Here's my sense of the conversation/narrative with Mark:
>
>1. The real goal of our Developer Platform work this half is to re-gain trust:
>
>- For people, that's fixing the login + sharing model
>
>- For Facebook, that means fixing the parts of platform that no longer make business sense
>
>- For developers, this means getting to a platform where they understand our incentives and business model, and we have confidence we can support it with high-quality.
>
>2. For users, we're changing login + sharing model. (Already discussed.)
>
>3. For Facebook, we're changing friend + feed model. (Already discussed.)
>
>4. For developers, the key problem is that we don't invest in APIs where there isn't a clear business model - we're just not good at it. Games + Neko are stable because they're clearly worth investing in. So is Login + Sharing (mostly). Everything else is half-broken, because it doesn't make sense to invest ~10 engineers in supporting Event APIs, or Group APIs, or Photos APIs, etc. This leads to broad developer mistrust, because they have no idea what APIs are safe to use, and which aren't.
>
>So we're proposing a pretty radical simplification of platform:
>
>- Mission: Help developers build, grow, and monetize mobile apps.
>
>- Core: Login, Sharing, Ads, Payments functionality. Will commit to 3-year breaking change policy.
>
>- Beta: Open Graph APIs, many Social Plugins, Insights, etc. Will commit to 3-month breaking change policy (should this be 6?).
>
>- Deprecated: Majority of the API surface.
>
>As we've already discussed, will also review all apps.
>
>5. How we roll this out?
>
>- Can hard deprecate stuff
>
>- Can declare a new version of the API, carry forward what we care about, commit to SLAs on new stuff and say old stuff will be unsupported (but work for 6-24 months).
>
>- Others?
>
>6. Timing
>
>Think this will take longer than we initially thought (probably into January). Can either:
>
>1. Launch in Jan, f8 in Mar.
>
>2. Launch at f8
>

>3. Slip f8
>
>Not 100% sure that's right (~4 minutes of writing), but I think that's the basic narrative/conversation we should have, Constantin.

Constantin Koumouzelis:
>Thanks Mike, this sounds right. Let me roll this into a deck and work out the broader narrative today. Will share once I have a first rev

Douglas Purdy:
>Have feedback on the details, but I think this is exactly how we should talk about it externally as well. This is actual net good for developers in the end (provided we honor our commitments, which we do). I strongly believe that all of our decisions need to be good for users, good for developers and good for us on balance, which is where we are finally landing with Platform 3.0. Feel really really good about this. I hope everyone thinks about this as the "reboot" we have all wanted.

Jason Clark:
>(y)

George Lee:
>(y)
>
>Small nit that we probably aren't going to want to deprecate any of the game APIs as we have plans to improve that piece at some point. It is really just scores and achievements.

Kevin Lacker:
>for SLAs, both the AWS and Parse SLAs are themed of, "here is how we define downtime and you get x money back if we have downtime." for a free api that doesn't really make sense because there's nothing we can really give to developers to make up for it when we miss the SLA.

Kevin Lacker:
>also, typically things called SLAs are goals like "99.95 % uptime" rather than "we fix all bugs within 2 weeks". the latter is great stuff to say externally, but there is probably something better to call it rather than an "SLA"

Douglas Purdy:
>I personally think that the core need to have an SLA. Any issue that causes an app to break, within the versioning window, is a service disruption.

Constantin Koumouzelis:
>Agree

Kevin Lacker:
>so for that you are talking about a goal in the dimension of "99.95% uptime"? or in the dimension of "we fix bugs within 2 weeks"

Constantin Koumouzelis:
>both

Douglas Purdy:
>I think an SLA is the former.

Kevin Lacker:

>in general, there are going to be bugs that will take long enough to fix that you won't spend 99.95% of the time bug-free

Constantin Koumouzelis:
>we've flighted an SLA that matches to the commit Mike mentioned earlier with partners

Constantin Koumouzelis:
>i.e Zynga

Constantin Koumouzelis:
>it took us a semester, but we eventually got to a 100% compliance rate

Constantin Koumouzelis:
>I think if we are aggressive with defining the scope of stable core, we can do this more broadly

Douglas Purdy:
>The Z example is a good proof point.

Douglas Purdy:
>That effort really paid off with them.

Constantin Koumouzelis:
>(y)

Douglas Purdy:
>That said, I still think that we have to think about core with a little more rigor. If we all agree regressions are service disruptions, I think this is ok.

Constantin Koumouzelis:
>if we look at the relationship we had with Zynga as a microcosm of the developer community at large - the SLA and bug commitment completely turned around our relationship

Kevin Lacker:
>i think we will end up in a state where some regressions count as service disruptions, but some count as bugs. a regression that is like "whenever you do a request on a user with this particular unicode character in their last name, there is now a 2% chance of an out-of-memory error.", you don't want to say "sorry we missed our SLA for March because of that." but a regression like "this api used to return the int 0 but now returns the string "0", that is a service disruption.

Kevin Lacker:
>part of what makes an sla valuable and what i hope we do is to make public how well we did against the sla. what was our uptime and how did we do on bugs

Douglas Purdy:
>This should absolutely be public, in my book.

Constantin Koumouzelis:
>I'm not sure I follow. If your API request is in the set of out of mem failures, that is a disruption in service and should affect our 99.9 SLA, no?

Tirunelveli Vishwanath:
>Yeah a lot of incoming bugs are non-mainstream cases/developer confusion/non repros

Tirunelveli Vishwanath:
>should #affected devs factor into what comes in SLA?

Constantin Koumouzelis:
>as a developer my app broke. I don't care if it's your infrastructure or you engineers fault

Douglas Purdy:
>Vish: I think we cut all non-mainstream scenarios from core

Constantin Koumouzelis:
>Pragmatically I think that makes sense re: affected devs but I think it's a slippery slope

Kevin Lacker:
>well, if you count every five minute period where some api request had an out-of-memory error as a period where we missed the sla, then we will have 0% uptime.

Constantin Koumouzelis:
>+1 to Doug

Constantin Koumouzelis:
>but isn't that exactly how a service oriented SLA should function?

Constantin Koumouzelis:
>if AWS is down for 5 min in a specific location, my app is still down

Kevin Lacker:
>no. if you look at the AWS SLA for example, they only consider it to be "downtime" when it's totally down

Tirunelveli Vishwanath:
>yeah I agree wrt slippery slope. I guess we need to completely specify behavior for core in our docs,

Constantin Koumouzelis:
>hmm, so cases when AWS crapped out in the entire north east, that did not count against their SLA?

Douglas Purdy:
>If the DynamoDB read API breaks, but write still works is that down?

Kevin Lacker:
>“Unavailable” and “Unavailability” mean:
>For Amazon EC2, when all of your running instances have no external connectivity.
>For Amazon EBS, when all of your attached volumes perform zero read write IO, with pending IO in the queue.

Kevin Lacker:
>that's the goal they hit 99.95% for

Kevin Lacker:
>basically, you can hit 99.95% for a goal like "It is not totally down"

Constantin Koumouzelis:
>so limited connectivity doesn't count then?

Douglas Purdy:
>I think dynamo or simple DB is a better comp

Kevin Lacker:
>for a goal like "it works perfectly", you can only asymptotically approach it

Douglas Purdy:
>That is why there are lots of 9s

Kevin Lacker:
>they don't have an SLA for dynamo

Constantin Koumouzelis:
>ie one virtual nic is connected, so it's ok if the others are not? I can understand that attitude, but I'm not sure it's the standard we should shoot for

Douglas Purdy:
>RDS?

Douglas Purdy:
>They have to have an SLA for their turn-key services?

Kevin Lacker:
>this is just how SLAs typically work

Kevin Lacker:
>SLAs are designed to cover total downtime

Kevin Lacker:
>they are not typically the mechanism used to describe quality of bug-fixing

Kevin Lacker:
>RDS is: "Unavailable" means that all connection requests to the running Multi-AZ instance fail during a 1 minute period.

Constantin Koumouzelis:
>Let's seperate out the bug commitment timing for a sec

Douglas Purdy:
>I assume that you have to be able to do something with the connection?

Douglas Purdy:
>Read or write?

Douglas Purdy:
>Or is a connection, just the open?

Kevin Lacker:
>in practice, the SLA is for your promise to keep the service running

Constantin Koumouzelis:
>the way I see this is: we maintain an SLA for uptime. Anything that causes a service disruption (bug or infra) affects our guaranteed uptime. If something breaks, we have the bug committ (24hr hi-pri, etc) to get it operational again.

Douglas Purdy:
>I like that frame

Constantin Koumouzelis:
>if we can provide a work around that does not require a code change (route to a different service, etc) that is sufficient to meet our commit

Constantin Koumouzelis:
>this is all completely orthogonal from our commitment to not change anything (intentially or unintentially).

Constantin Koumouzelis:
>effectively we are writing a contract for platform that contains all of this. The service SLA is one component

Kevin Lacker:
>the azure sla is another good example - http://www.windowsazure.com/en-us/support/legal/sla/

Kevin Lacker:
>to get a guarantee with a lot of 9's you have to be very minimal about what functionality you promise is working. in their case it's basically "you can connect to the database" rather than "every method works absolutely as documented with zero crashes". so, anyway i think we are on the same page, it's just the precise definition of "what counts as a service disruption"

Constantin Koumouzelis:
>(y)

Constantin Koumouzelis:
>Agree that there is a spectrum here between [the graph API responds to an HTTP request] and [method X returns the correct response to every request, every time]

Constantin Koumouzelis:
>I'm a little hesistant to snap to the graph API always returns an http status code, and 500 counts - this isn't really what we have a problem with today and I think developers can see through it

Constantin Koumouzelis:
>looking at the Azure SLA - the Storage section is closer to what I'm thinking we should offer

Constantin Koumouzelis:
>we guarantee crud operations

Constantin Koumouzelis:
>Just uploaded a first rev of the deck for Zuck review, outline based on this thread and our meeting yesterday. Feedback welcome

Constantin Koumouzelis:
>https://app.box.com/files/0/f/1101006748/1/f_9940797246

Constantin Koumouzelis:
>https://facebook.app.box.com/files/0/f/1101006748/Deck

Ilya Sukhar:
>When you say (limited) near Social Plugins, does that mean just a subset will make it to beta or something else?

Constantin Koumouzelis:
>a subset will be core - i need to make this clearer :)

Ilya Sukhar:
>Also is the gameplan still that we meet a bunch next week to nail this down?

Douglas Purdy:
>Yes.

Douglas Purdy:
>Sara and Rennie are scheduling.

Ilya Sukhar:
>(y)