

EXHIBIT 171

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From: Mike Vernal </O=THEFACEBOOK/OU=FIRST ADMINISTRATIVE GROUP/CN=RECIPIENTS/CN=MVERNAL>
Sent: Thursday, August 02, 2012 5:51 PM
To: Mike Vernal; Sam Lessin
Subject: Message summary [id.318712944877202]

Samuel Lessin:
>what was the GDP followup with mark? do we have a good plan there?

Michael Vernal:
>We're not going to talk about GDP - we're going to continue our conversation from yesterday. I had a long conversation with Mark last night, and it's you, me, and Zach at 10am.

Michael Vernal:
>I was planning on dialing you in.

Samuel Lessin:
>cool

Samuel Lessin:
>re: GDP tho, did you guys reach a conclusion / or are we just not discussing because we need more tools / answers before we proceed?

Michael Vernal:
>GDP is putting the cart before the horse. We have to agree on the product. GDP is how you turn it on/off.

Samuel Lessin:
>makes sense

Samuel Lessin:
>i am on the line

Michael Vernal:
>Will dial-in in 5-10. Assuming Mark is 10 late and have to go to candidate review for one candidate first.

Samuel Lessin:
>hah

Samuel Lessin:
>k

Samuel Lessin:
>some thoughts on OG business model... apparently we are going to talk
>with the BD about this... here is an outline / would love your reaction
>
>
>(1) Context - the platform business model
>

>- With Canvas

>the "trade" was clear -- we provided applications with tools for acquiring and re-engaging users in return for more inventory for ads and ego (growth and revenue). In many cases this was neutral or even negative for users given the free routes we opened for application distribution.

>

>- With Connect

>The trade was marginally less clear... We provided similar tools for acquisition and re-engagement, but did not get the benefit of more inventory, etc. We still considered this net positive because it made facebook more useful and more central for our users... But it also had negative user implications.

>

>- With Open Graph

>The trade of open graph is theoretically based on two sets of premises. The first part is that with structured data from developers... (1) we can create a more compelling/ positive user experience around self expression and discovery that will drive more usage (2) we will know more about our users and be able to better monetize them / target them with ads. The second part is that developers will give us that structured data because... (1) users will demand it from them / they will need to build good facebook integrations as a user benefit around self expression (2) we will be able to provide even better distribution for good apps with better structured data.

>

>

>(2) Fundamental problems

>

>- The Open Graph tradeoffs are not yet working / proving correct We

>have not yet built a user experience that users demand (actually quite

>the opposite - on one side we have to dramatically subsidize the

>stories in newsfeed for apps to get any distribution, on the other we

>have indication that passive sharing is pretty negative in terms of

>self expression), we have not been able to monetize any of the data

>meaningfully (there are some pretty severe issues here varying from a

>mismatch between developers publishing information and who would

>theoretically monetize it to the fact that good partners can't value

>the payout / payoff in giving us good data - beachmint, amazon,

>foursquare, etc)

>

>- we have gone through imprecise cycles of boosting and then pulling

>back distribution because we have no fundamental value model for information published by partners we are constantly trying to tune the ecosystem up and down in ways that make it hard for partners to rely on us / work with us.

>

>- We have a sometimes "tense" relationship with people we might view as

>competitive (or people who might rightly or wrongly view us as a competitor) Because there is no clear and positive value exchange with developers to evaluate situations against we get into tense situations where we feel like we are paying for / subsidizing the growth of competitors ...

>

>- *We have never been able to give out certain requested APIs For a

>long time we have been asked by developers to release APIs for a series of types of information and types of distribution, but we have been unwilling because we have viewed these access points as too competitively valuable to give away on our generic terms / the generic trade conditions.... Examples include information like coefficient between users, FOF paths, and validation of specific data fields. They also include distribution routes like messaging and notifications.

>

>(3) A proposed solution / our proposed next step... Economically

>rationalize platform for stability and growth

>

>- charge directly for information / APIs Charge developers per-call to
>our APIs, starting with just premium APIs for data where we are uncomfortable about the value exchange (open up coefficient, fof connections, eventually browse queries). Key factor here is that we will not be charging users for recalling their own data (that is theirs) we would be charging for our data about users / other people's data that you don't own...

>

>- charge directly for subsidization in feed / distribution across feed,
>notifs, messages we should always start by always generating for every user the "ideal" newsfeed with no subsidization, applications can then buy distribution in feed, where we charge differently based on engagement. The thinking here is that in a unified feed we are optimizing for purely engagement and revenue. High engagement stories are free, low engagement stories / messages can be shown, but they cost a lot for it to be worth our while to display them. We move messaging and notifications to the same model with different engagement vs revenue goals.

>

>- new platform terms and pricing
>in order to be able to do the above / feel like we are building a
>stable platform with a sustainable underlying logic we will need to
>introduce some new developer terms, and generally control the ecosystem
>much more tightly

>

>(1) we need a mutual information exchange policy, if you pull data from us, you need to push data to us. WIP concept here, will flesh out - and maybe discuss how we could retail for our partners data they write in??

>

>(2) no cache and no resale... If we allow you to get FOF paths, or
>coefficient, etc it is critical that apps agree not to cache the data,
>and to never release a competing set of APIs / resell our data

>

>(3) pricing will be different for different industries / partners. Likely the pricing on distribution can be off the rack, but our data is worth different amounts to different applications / developers. We need to therefore be prepared to negotiate pricing with different people differently.

>

>(4) market sizing
>we need a market sizing on all this, TBD

>

>(5) Questions / For Feedback
>There are a lot of questions here... Things that would be great to get feedback on...

>

>- How will apps / partners feel about this? What is good / what is bad / will they pay?

>

>- What APIs are most valuable that we are not offering - stack rank
>friend paths, coefficient, etc

>

>- How should we package the pricing for companies to be able to easily
>/ efficiently consume (per-call vs. per-acquisition, etc.)

Michael Vernal:

>1/ I would prefer we frame these as options vs. proposed solutions (you use stronger words than I would).

>

>2/ I really think we're wasting time talking about charging for APIs.

>I don't think it's a scalable business model and is, frankly, a waste
>of our time. That's not where the real money is, and it will have
>really weird downstream effects. We can include it, but if our
>business model is really selling data to developers, I think it's

>pretty uninteresting (and deeply problematic). (I'm fine doing the
>thought exercise, but becoming rapleaf is just going to be super lame
>and uninteresting.)
>
>3/ I am a fan of making feed more competitive and letting developers pay to increase distribution. I think a good principle here is that we should only be distributing content for free that actually makes Facebook better (e.g., Twitter content). If your content is neutral-to-negative (e.g., games), then charging to balance that seems reasonable.
>
>There are a few other things I think we should talk about.
>
>Concretely, we should understand what developers have money to spend. Anyone who sells something has money, and can pay us to make more money. Anyone whose business is advertising-based is basically competitive with us (we're competing for attention), and won't really spend money with us to acquire more users (advertising has network effects, so it'll never be cost-efficient for them to pay us to get more users).
>
>I think only two sets of developers have real money to spend with us today -- game developers + commerce companies.
>
>For game developers, we made a lot of money on canvas because we could tax their revenue (the most efficient model, in some senses). We lose that on mobile, and so I think we should talk about what we should specifically do about mobile games, since that's a huge chunk of potential 1-2 year revenue.
>
>I think there are a few models:
>
>1/ Publisher Model - we could say that we only distribute games published by us, and we take 10% (on top of OS's 30%).
>
>2/ Non-Publisher - we could not be publisher but still take 10%. Harder to police, but simpler in other ways.
>
>3/ Charge-per-install - we charge you for every users you acquire through us, even if it's organic.
>
>4/ ...
>
>For commerce companies, I think the real potential is in letting them advertise against other people's content (as we've discussed). I think we should talk about this, too.
>
>I guess, at a more meta level, it's easy to talk about the 5-10 year business model for Open Graph, but it's really hard to know whether that's right or not. The more fruitful conversation would be to talk about the 3-18 month business model (if any), as that will force us to be concrete.

Michael Vernal:

>Can you help shepherd gamer friends while I'm gone? I'm worried about it.

Samuel Lessin:

>Yes -- on it -- anything in particular you are worried about, or just product going off the rails?

Michael Vernal:

>Worried we'll do a one-off.

Samuel Lessin:

>Yah... You and I are at least always aligned in our concerns... ;). -- I will make sure that doesn't happen.

Michael Vernal:

>I think we can build this so it matches both use cases, but requires really broad perspective.

Samuel Lessin:

>Is Doug on board with the broader bit

Samuel Lessin:

>Are you just worried about implementation?

Michael Vernal:

>We're ending up with a compromised product. Don't think Doug is spending time on this.

Samuel Lessin:

>Ok --

Michael Vernal:

>Worried about product / abstraction mostly.

Michael Vernal:

>Eng too, but if abstraction is clear Eng is easier.

Samuel Lessin:

>Ok -- I will push that -- I care a ton about lists, so will stay on top
>of it