EXHIBIT 29

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From:	Sam Lessin </O=THEFACEBOOK/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=SAM LESSIN>
Sent:	Wednesday, July 25, 2012 9:05 AM
To:	Evernote Publish
Subject:	FW: some thoughtsŠ

From: Sam Lessin <sl@fb.com>
Date: Tuesday, February 21, 2012 11:36 AM
To: Mark Zuckerberg <zuck@fb.com>
Subject: Re: some thoughtsŠ

I think that the business models / concepts are related, but are not the same when stretched to their logical extremes.

In an information business:

- We could thrive even if not a single person ever directly visited our site / newsfeed / timeline, but instead all apps / interfaces wrote to /read from us to optimize the experiences they provide to their users. So we make money / have a healthy platform because there is return on scale in information & we know better than anyone else what story (sponsored or otherwise) to deliver to whom at any given moment by knowing everything about that person.
- E.G. — if we are an information platform a future business like 'pinterest'
  - Input side: would have a front end user experience for collecting data, write all their data to our graph (potentially as their only data store at some point in the distant future)
  - View side: would have a custom interface for viewing their content, but would actually query us for the best information to show their users (which we can do better than they can by mixing their specific inputs & data with everything else known about the viewer, poster, etc. in the world)
- In this model of the world, over time Facebook eventually becomes purely a central information exchange, and someday might not even have a consumer brand.... Apps / partners have the attention, we have the information that powers them.
- Our value as a company isn't the ability to 'reach' a customer (where there will be more and more competition / a flatter and flatter marketplace with more and more 'scale' players), it is to know a customer/user (where there is real return on scale).

If we are a distribution / attention business

- We need people to actually pay attention to our platform for us to succeed / read NF through our interfaces, etc.
- E.G — if we are an attention business a future business like 'pinterest'
  - Input side: would have a font end user experience for collecting data, and perhaps copy their data to the graph
  - View side: we would always be competing with pinterest for the attention of their users...
- I have heard the argument made that the value of FB is that we can aggregate a very large audience (just like the premium that exists for TV) -- that just seems like a pretty poor and long-term indefensible advantage.

My belief is that short term there isn't a huge difference, we can pursue a distribution strategy and an information strategy at the same time (they are short term symbiotic) -- I do think that long term the information business is more scaleable and more defensible vs. an attention business with a finite number of hours in the day.

That said, if you wanted to draw an immediate distinction, I would say that an information business focus would say that we should be focusing on knowing more about everyone than anyone else (bits of information per-person, etc), and then creating ways for other businesses to leverage our knowledge without appropriating our core value / dataset… if we are attention / distribution focused then we should be focused on getting the maximum number of DAUs and care less about knowing who they are.

From: Mark Zuckerberg <zuck@fb.com>
Date: Tue, 21 Feb 2012 09:24:32 -0800
To: Sam Lessin <sl@fb.com>
Subject: Re: some thoughtsŠ

I really agree with your point about being able to articulate our business model around data. However, one thing I still don't really get is the difference between being a distribution platform around people's attention vs being an information platform. Those just seem like two ways of describing the same thing to me since in both views we're helping people get content into the system and then creating utility and revenue by showing people the best content/information.

If you think these approaches are actually different, can you list off some concrete differences in what our approaches would be if we adopted one model vs the other?

On Feb 20, 2012, at 10:57 PM, "Sam Lessin" <sl@fb.com> wrote:

> Zuck,
>
> I have been working on a set of ideas / trying to form an opinion re: the top level strategy course we are currently on… simultaneously, I really took your note last week to heart and wanted to respond to it a bit more specifically than I did before. I wanted to write you something very polished, but after working on thinking it through for two flights and a few other odd hours last week, I realize that polish will take far too long for the value — so instead here is a somewhat edited / but not fully polished view of the world as I am currently thinking about it.
>
> I am going to do this in two parts — the first is about strategy / where we are and the second is about me & the identity team. I am going to be really direct / tell you all the things I am thinking in the back of my head — and I do worry in writing this out rather than speaking it that the tone will not come through — but the flip side is that in person I worry that tone but not content might come through -- so please as you are reading it keep in the back of your mind that I am really excited about what we are doing / our mission, the opportunity to help lead, and I really do respect and trust you as a leader enormously. I just really want to win / and see Facebook become a full expression of its promise… and I think that the best thing I can offer is directness. So, with that scary windup done…
>
> **Strategy:**

Let me start with a few key things I believe / a lot of which is informed to a greater or lesser extent by listening to many of the recent conversations of which I have been a part. I know that some of these stances are controversial / are not consistent with the current wider message to the company, but I will state them pretty bluntly nonetheless:

- (1) We are primarily a web company & the 'desktop' web is clearly not the future - this needs no explanation. I think we all know this is true.

- (2) Native apps will dominate over mobile-web for a long time (maybe forever) and we cannot prop up HTML-5 / are not strong enough to lead a shift - The mobile OS makers have a strong incentive in native apps performing better / working better than the web... so theory / what is possible aside, native apps will work better & be better experiences than the mobile web. I might go so far as to say that the free and open web was an anomaly caused by immature desktop OS platforms who failed to develop the distribution and payments capabilities needed to support good business models for developers and therefor lost the advantage to the web briefly... iOS, and to a lesser extent android, has reversed this — the tower of babel is falling / the unified language will go away for a while and the value (or a relative if not absolute basis) will end up tucked away on proprietary 'platforms'.

- (3) I do not think we will be able to build a successful 'phone' anytime soon – I don't have all the details on firefly, and It is possible, but to me it feels like a hail-mary to be trying. It isn't in our DNA to build a phone / manage carrier deadlines, battery life, etc. and as a result I am skeptical that we can do it well enough to get any meaningful part of the market from our competitors... Our processes and culture is just not set up to win at this, and for most of the traditional 'phone' functions 'social' is not enough of an advantage to overcome all the other cards stacked against us. Further, I don't think we are setup to maintain & improve a phone platform even if we built one.

> *Two caveats here* -- (1) if we had a business model innovation which completely changed the game people might use our phone even if it wasn't as good as the others (e.g. we were talking about advertising supported free phones about a year ago) (2) if we didn't build a phone, but instead built a simple social 'device' (a newsfeed machine if you will... a generation one blackberry device that just did NF and was really cheap / a second/third device) -- I could see that working to some degree. I really just don't believe we can go toe to toe with apple or google and win.

- (4) The 'Open Graph' is fundamental but a long way from being set up for success — I am ridiculously excited about the OG long term — I think it is the holy grail, and I really look forward to a world in the future where the OG is the primary data-store for many applications... but I think it will be years before it is really good, not months. The immaturity of the OG might be fine if there was a gold-rush mentality about the platform it as there was originally, but I really think that the developer world has grown up a lot since the original platform — and with options like iOS out there, I think we need to be in a far more polished state to get people to use us in a serious way;

- **As a 'language'** -- it lacks all sorts of key pieces we are aware of, and many which we probably aren't— e.g. allowing use of Facebook objects (profiles, pages, etc), per-edge privacy, 'private' objects (or, more specifically, maybe public objects with private attributes), internationalization systems (which is a huge but relatively un-discussed issue), intransitive verbs (which turn out to matter a lot for self expression), etc. More abstractly, we talk about wanting convergence of language / different apps making claims / associations between a common vocabulary of verbs and objects; however, the current structure of the OG with 'custom domains', etc. encourages a diaspora and lowers the value of having the OG.
  - This might not seem like a huge deal given that the OG does support many use-cases even now, but because the language is relatively immature & incomplete, developers who are tying to be at all innovative are running into issues & getting frustrated in their attempts to buy into the promise we sell.
  - Further, I would argue that in many ways we did enough of the OG to support the types of games, etc. that we expressly are less happy with, and did custom mapping for certain

    partners, but we have not done enough to support most of the good use cases we would want (e.g. pInterest can publish back the 'follow' relationships on their service, but we have no understanding directly or indirectly of how those map to our actual profiles / people from which they are drawing)

- **As a value proposition for developers** -- when there are other options out there we need to provide clearly demonstrable value for developers in a way that they can understand is symbiotic so they don't question our motives. Right now, we provide no clear monetization, and because developers can't tell themselves a clear story about what OG is doing for us, we can establish little trust in the distribution patterns we do provide. Right now those that use the OG are looking at it basically just for distribution / as a way to drive people back to their own apps... This mentality encourages people to look for ways to get the most distribution in return for giving us the least data / least well structured data. It means no one is betting the farm on it, and app developers will tend to act parasitic instead of symbiotic.
    - This is probably the biggest deal in my mind. The world is breaking down in my mind into two classes of application developers. One class are small teams / mobile developers, who find that iOS allows them to be much closer to a revenue stream then we do (even if that revenue stream is not ultimately all that great / the app store model limits upside). The second class are larger teams / VC backed startups (path, evernote, etc) that are game for a longer term & higher return play, but because they are longer term focused will not deeply integrate deeply us both for trust reasons on the value / volume of distribution we send AND worry deeply about the value of their data / are telling themselves a story about how valuable the data they are building up really is.
- **As a developer interface** — documentation, submitting / approving apps, etc. I don't have much first hand experience here, but from poking around a bit myself and talking to people at Nike, watching Waze try to do it, etc. it sounds like there is a lot of growing up to do.
    - This probably isn't deeply fundamental, but I do think that we probably loose people and good will here on the margin constantly / and that many people end up going for less ambitious / interesting integrations than they might otherwise want because they are just confused or don't know how to best leverage us.
- **As a user value proposition (publishing side)** – self expression via data aggregations and reports might be the right direction / I am pretty excited about it, but we aren't even close yet to users really demanding applications publish to their timelines. As you have pointed out many times, we might never be able to win at this vs. very good / vertically focused apps, but we need to at a minimum offer a certain level of quality.
    - I also think that we are ahead of the market on people wanting to use data / information to tell stories. Timeline I think has started to do a decent job abstractly getting people focused on Facebook as a store of value you build up over time rather than a stream that passes in the night... but I actually don't think that even if we had great visualizations most of the world is ready to be really excited about Feltron reports (even if they will in the future / it is the right fundamental underpinning to be going after)
- **As a user value proposition (consumption side)** — many people seem to be pretty ambivalent about what they are seeing / feel like a lot of the stories are noise / clutter in newsfeed.... And in many ways I think this is fair until things like browse exist / there are more interesting pivots to do on the data
    - I again feel we have the right fundamental story here around discovery, but clearly we aren't even close on things like Newsfeed and Ticker.. And I don't think there are quick fixes, I think we need a more powerful / deep approach.
- **As a permissions model / GDP** — we have a lot to work on here. I think we need to dramatically lower the barrier for applications being able to publish in a safe way to users' timelines (aka, we don't technically need GDP to publish / publish only me). On the model for apps requesting 'read' permission, without more nuance (e.g. You can have my 'friends' data, but not my 'only me' data) I fear our model will not scale to support communities / more private sharing (just think what would happen / how scared we would be as FB employees if we had not added the 'hack' so that FB-only & messages between FB employees don't go out when we auth an app)

- *UPSHOT — I don't think the above is 'patchable' -- I don't think we can fix all the issues quickly or on the fly as we also push the OG out. I feel like it is going to take some time to be in a good place…*

- (5) We generally to date have had a monolithic approach to our user interfaces (one website, one iPhone app, etc.), but (A) bite-size experiences will win on mobile platforms (B) having a monolithic approach to interfaces means you end up building ever more complex interfaces that are harder to make 'good' & fast - - On mobile people want simple experiences that they can consume very rapidly / in intermediate time rather than constantly in the background… our mentality & app are not tuned for this. The fact that people love instagram speaks to this in my mind. Further, from an interface perspective, the larger an interface the more corners you hit / the harder it is to achieve 'quality' and optimize things like speed. In some ways, even setting aside consumer expectations / etc. I fear our monolithic approach is making it too hard for us to be 'good' even on the web (and I think you are seeing evidence of this in complex interfaces like 'timeline' on the web)

- (6) We have a ton of ideas on what to improve in our core model / business, but not the resources to do it — I feel like every week we are calling out really important and leveraged product improvements we could make (like your email re: path this last weekend) -- but because we are so heavily invested in a few very large products right now, we just aren't even close to being able to actually execute on these social changes which are core to our business.

So… the question becomes — where do we go from here / what strategy should we be running as a company? Here are a few bullets I would call out for discussion:

- (1) Invest significantly more in the core, and become a 'API first' company we talk about being a mobile first company, but I actually don't think that goes far enough… I think we need to be an 'API first' company and focus heavily on our core abstractions / enabling ourselves and others to build all the experiences we want on top simply and efficiently… perhaps even to the degree that we release new APIs for things even before we release our own interfaces to make them useful for end consumers. The reason I think this is important is that I think that for us to really win, we need to be thought of by developers first and foremost as effectively a service layer (AWS) for structured information… meaning, they put data in because the social functions and structured stores for their data that we provide are more efficient, and add social context and interfaces we add on top makes their data (& them) more valuable. I think this specifically speaks to a few projects that exist, but we need to balance resources towards / invest much more heavily in & move faster on:

- (1) BROWSE:
    - To me browse is the single most important abstraction we lack. Direct queries like 'music my friends like', 'friends who like running', 'engineers who work at google' is the cornerstone to making our app a real utility (and making Open Graph useful)… When you allow users to turn these queries into feeds & notifications, etc. -- and then allow app developers to layer on top of them, I think we will unleash an unbelievable amount of value. I would be betting the farm on this right now as a fundamentally missing part of how you look at the graph & how users and app developers should be thinking about and using Facebook.
- (2) TAGGING & GRAPH CLEANUP
    - Browse only works well if tagging works well / efficiently for end users and apps. There is a lot to fix here, we are in a pretty poor state in terms of what is possible & the current setup of OG with custom domains and us needing to do custom mapping between things. Further, we don't really have good generic tools for crowd-sourcing mapping / our current machine based de-dupe approach seems insufficient. We also really need to introduce the equivalent of 'hash-tagging' / true arbitrary structure into FB, and I think building editor plugins (so you can markup websites with FB tags, etc. would be huge coupled with browse)… I think we mostly know what to do here, but again I think we are majorly under-resourced.
- (3) LISTS / COLLECTIONS

    - For privacy / publication control, and to enable collections — you can think of this as tagging or lists, but we are missing pieces here as well that aren't as fundamental as #1 and #2, but are close. I don't think we are crazy off here, but it seems on the back burner and like we are doing it to produce a few specific user-use-cases, not a generic concept that apps can use differently.
- (4) NOTIFICATIONS
    - I think notifications is a critical abstraction that we currently have basically no people working on... you can make arguments that we will lose this because of mobile platforms, but I think we need it as a core abstraction (and am happy to talk through why in more detail)
- (5) AD NETWORK & PAYING DEVELOPERS
    - We need to figure out a way to pay developers. We have the pump running on an advertising market, our data-centeric model / targeting means that we SHOULD be able to do this naturally better than anyone and provide application developers with a better business model than anyone else. We need this to be true to win -- (and, on the flip side, as we discussed, paying developers for information fits on the other side of this transaction curve)
- (6) 'ROLODEX'
    - I put this towards the end of the list, partially because I am know I am biased / heavily personally invested in this.... But I think that this is critical. Our current model where nodes can only describe themselves limits our platform, and I believe is causing all sorts of pain that we don't even fully feel (including things like lacking good ways to do graph cleanup).... We bend over backwards in weird ways, etc. to try to fit revers claims into a self expression model, but we need to call a spade a spade and build a useful low-level model for claims about others
- (7) PAGES
    - To me another real hole in our core model is that pages cannot act the same way as profiles, especially WRT OG (but also tagging, etc).
- (8) SPEED & RELIABILITY
    - I don't have a lot of first hand experience here... but we should be investing heavily in speed & reliability on our APIs. My sense is we are getting better but have a long long way to go here.

(2) Build & expose a clear model for how we make money given information for ourselves and for our partners – For people to build on us we need to be predictable & trusted. The most important way to get there is to make our business model really clear to developers in the ecosystem. Right now, I believe that it is just too opaque... which means they don't trust us not to change the deal in the future (APIs , Access, etc.), and will not do serious business with us.

- The way people will trust us is if they fundamentally understand how we make money / the trade involved in using our platform. The only way we will have a stable platform is if the core economics work and scale well. I do not believe that we can win long-term at an attention based economic model... I think we can only win long-term at a data-centric model where our targeting surpasses everyone both in terms of delivering useful information when given a query, and modifying that delivery with payment / advertising buys as a core input.

(3) Become a mobile 'second' company (after APIs) and build <u>tons</u> of single serving apps, but stop trying to do 'takeovers'

- As I stated above, I think that our 'takeover' strategies like Firefly and min-Firefly will pretty likely not work. I would love to be proven wrong... But I think doing a good job with Firefly is currently beyond our reach, and I am skeptical that a highly incentivized and very well funded competitor will not figure out how to shut down / cripple mini-firefly before we really get it off the ground… so, in my mind a 'Facebook phone' is something that we should not be focused on.

- The next question is monolithic applications. I just don't believe in them because I think that apps are too hard to build / make fast / maintain / and use when they are really big… we end up going too slowly and worrying too much / in too many compromised states because we are making tradeoffs between functions which end up with us being great at nothing. One way this manifests itself is this zero sum game we seem to play / be talking about with bookmarks… which is just un-winnable in my mind / I think of as an interface hack that will never fully scale out.
- This leaves a model of federated inter-connected small apps built on our common backbone… which I really like as an approach. When I say this, I actually mean this to the extreme, where we build tons of interfaces which we treat almost as disposable… Let each team build things independently like a browse app, a newsfeed only app, a notifications app, messenger, a timeline publisher app, a groups app (or one app per group), a profile app (or one app per person), a 'likes an interests' app, a local app, apps which are just specific browse queries, etc. Don't even require any parity between android, iPhone, etc. except in really basic cases where you need 100% coverage for the network effect to work properly (most notably messenger and the most basic profile, NF functions)
  - Doing this well would force us to emulate developers using our apps / work almost on a level playing field with them (or at least deeply understand their pain)… which I actually think would be really helpful (especially if we could get our business model to the point that we truly want our partners to be wildly successful — which because of our current business model I am at least currently pretty ambivalent about)
  - To be able to do this we need to prove to ourselves that we have a huge ability / beyond anyone else to distribute new applications and monetize across all the apps… a tall order, but something I think we must be investing in.
  - To be able to have this scale we need to all be working against the same exact abstractions and APIs / we can be hacking too low into the fundamental model of Facebook.
  - The risk here is that we get too attached to any one app / are not willing to abandon certain ones / not upgrade them at all after a certain point over time. I think if we were good we would be willing to make this commitment & be strategic about a portfolio of apps.

- (4) Be OK with releasing different apps / interfaces with pretty significantly different feature sets based on the platform & use the web as our more 'complicated' interface of legal record also mentioned above, but I want to again call this out as a separate top level point. In many ways, especially as the desktop web dies, I think it is OK for it to be close to complete against our APIs and as a result complicated, and likely as a result a bit slow and a bit buggier than mobile (meaning, you can do ANYTHING on the command line that is allowed by our fundamental model, you can do 99% on the web, you might only be able to do 20% or 10% in any given app)… This requires a mental shift for us where we really are building single serving apps / are not duplicating all functionality in all places… and letting the web be a bigger monolith which is a fall back management interface and interface of record rather than something we expect to be pristine in the future.

- (5) Be simpler in our core model / do not create corner cases even when it is convenient at the core (but of course let them exist / leverage them on the surface) I think that as a company we systematically and massively under-estimate the cost of having multiple different sets of rules and abstractions at a low level / the cost of avoiding policy pain. This goes for decisions we make around things like project kid, social ads labeling, how we rolled out subscribe in an opt-in way, etc. I think that for developers to understand us, and for our engineers and product people to understand us themselves (and build good things), we need to have a really really simple basic model which we do not break / create corner cases or exceptions around just because it seems convenient in the sort term. I think we are improving here… but we need to continue to work really hard at simplifying our core model and then get really really good at executing against it (and put real headcount against smoothing out a lot of the more crazy corner cases we continuously find in legacy interactions)… If / as we find products we want to build that break the core model, then we need to holistically re-evaluate our core instead of hacking things together or making a new exception.

- (6) Focus on key social product improvements we know provide value our mantle is people interacting with each other. We should be very focused on wins which are meaningful directly to social, like autocheckin, notifications when a photo is uploaded by someone and looks like you, 'views' as social feedback, etc. and not lose an advantage on what we should be best at to any app.

- (7) Rapidly slow our investment of time and energy in large / established revenue sources that do not have a future (even if it is painful) and immediately and heavily hit the accelerator on unproven things in which we believe This seems like a ridiculously hard situation to manage. Perhaps your strategy here is to wait for sponsored stories / good business models to show enough growth that we can signal externally a simultaneous story of declining legacy businesses coupled with growing / better businesses, but it seems like this is going to be really hard whenever we have to deal with it, and I worry that in the meantime it is a straight (and actually pretty expensive) tax on our development.

- (8) Slow our roll on the OG in its current form Given all the above, I just think we are premature on really pushing the OG aggressively — we could spend an infinite amount of time optimizing the last mile of OG right now and trying to light up partners, but I think we have more fundamental work to do before we can do this successfully… and I think doing it now comes at the expense of our resources and the quality of the experience for users -- which sucks, but is the spade I just want to call here… I don't think we can build the platform that we want & get application partners to grow & be good right now.

- (9) Slow our roll on firefly Implied by the above. I don't know the exact numbers, but I do know that a pretty large number of our most talented people are focused on firefly right now… and while I am really excited about some of the designed interfaces I have seen, I just don't see how we are going to win here. I know that stepping away now would be an enormous organizational challenge, but I don't think we can afford the resources now or on an ongoing basis yet as a company. We can do new audacious products from places of weakness or strength, to me this feels like a project born of weakness…. And I would rather double/triple down on the information business that is our core / what we have a natural advantage in.

- (9) Buy some social tangents we admire that should be on OG this is going to be my craziest suggestion / point… but I want to put it on the table as a thought experiment. What if we took a relatively sizable percent of our value and made some major acquisitions. Twitter, Tumblr, Foursquare, Pinterest, Path, etc…. Consider this a placeholder for more discussion in person… but the quick version is that integrating them onto OG would be an interesting and valuable exercise & in my mind worth a lot.

Upshot — I think we are currently in an 'uncanny valley' of product… and we should instead go for a high-road / low-road approach where we build bullet and fully articulated low level abstractions up through partner-application facing APIs, and then move towards more vertical / more bite-sized experiences for consumers on the edge…

I am so excited about how close we are / what we can do by building a great information platform. We should be the fundamental layer of the world's information because we provide the best store, the best business model, and the best consumer experience / demand in the world. I worry that we are missing the ball on this not because we don't have a clear path to do it, but because we are fracturing our resources at a critical time. I think we are reacting to a big shift by trying to emulate the people we see as competitors with low leverage rather than double down on our core strengths where we can get huge return on being who we are / the network we have built. This leads me to think we need to invest in our core (and factor our system for the next generation of social) and then build effectively flexible / small / semi-disposable interfaces on top of that core engine.

-- I am really tempted to try to articulate the full model here as I see it, but I will refrain… instead I will just say that I really don't think we are a 'technical platform' (which I really don't understand why we would want to be or how we would be), nor do I think we are an attention/distribution business at our core… I think we are an information medium / or 'platform' if you will — and I think we should embrace that as our core strength and point of leverage and get very aggressive about being that to our core.

Team:

here is the other half of my brain in a snapshot — which is on my team & how I am learning to interact with the m-team / in the m-team conversations. I will try to walk through this with your note to me in mind / as a reference point for discussion.

(1) Design / frustration with our interaction design process I feel like there are some tactical and some broader things to discuss here and that this is a pretty high order bit…

- (A) The tactical bit is that this is all about people…. Felton is good, but as you note not an interaction designer either in terms of skill or interest. Ryan is a good helper / can do basic things, but he isn't a great designer, nor is Jez (our very young designer from a graphic design background, and Tom *can* be good — but is both getting married and having a child and isn't working that much / as much as he should. We need a design manager to control quality and mentor / build up the team, and it looks like we might get one soon (offer out and high probability that she does identity if she joins), -- but if you look at the non-manager design hiring priorities we aren't even close to the top — which means in my mind we will not get more real execution horsepower for months.
- (B) A broader point is that we really just need to decide how much autonomy / design oversight we allow felton / the designers to take. The more we give them clear requirements and then let them work out the decisions on their own the faster we can move… the more that they feel like you want to approve each change / you signal that you want to approve each change to them, the more they will wait for your feedback / our weekly discussions to do a full 'turn' on designs & the slower we are going to be able to progress —
  - By way of example on this — I feel like our discussion on 'only me' has been really frustrating (and I suspect you feel the same). It has gone way way too slowly, and sucked hours and hours away from a much higher order bit on the OG work. I fully understand if you want to approve each change & really own the design process here (and we can try to factor more efficiently around that type of work/feedback cycle), but I think we will be sacrificing a lot of speed / nimbleness — and I am not convinced that in general it is the right tradeoff (though happy if that is your explicit call)
- (C) A final borader point on design is that there is this tradeoff between building design 'systems' which are flexible and scaleable vs. perfecting a given interaction. I feel like as a company we should be in the systems business, and only apply tweaks in the last mile where absolutely critical…. I know Felton feels this way… but I don't think we always make the right tradeoffs here
  - Again, by way of example — from our 'only me' discussion… I would argue that we need a 'system' for only me that is going to work in a variety of context, and that having one will make us move faster, lower the number of bugs, and generally be a helpful abstraction — and then for a specific case like 'shoebox' if we want to layer something on top of that it is obviously fine as a one off, but we should treat it as a one off that we apply in the last minute… not something we should be building up from as a first principle.

(2) Timeline Quality & Perf: As I frequently express, I feel like I don't have a lot of credibility with you on this, partially because I am not an engineer by background, and because the products I have worked on here and elsewhere have not ever been really polished…. That said, here are things I think about:

- (A) The user-interface complexity of the timeline product is probably bigger than anything else Facebook has done. The fact that so many different product pieces exist / interact in combinatorial ways both leads directly to 'bugs' and makes it extremely hard for the team to know when we will be 'done' (in terms of things feeling fast / good). Take by way of small example what we have done with ads — probably the correct call to optimize for revenue, but there are about 20 different ad configurations to manage on timeline, which is pretty insane / will lead to bugs on the margin and always hurts perf on the margin. The team is working hard and we are doing better, but we certainly lag the rest of Facebook as a result.
- (B) The team really is focused on this stuff / cares a lot about the product, but I think that a lot of times people like Paul / Steve, etc. hear us asking for perf improvements, but then also asking for

product interactions / changes / complexity that trade off against perf focus (at least in the short term) and feel like their job is to do the best we can / get us to parity with the rest of FB... but there is only so much they can do...

  - As one example, take ranking of months vs. moving to 'show all' as an example here. We got scrolling perf to a pretty good place before launch in the normal use pattern / better than it is now, and then made the 11th hour call to stop ranking the months section... which ended up erasing a lot of scrolling perf gains because it ends up loading way too much content into the DOM. With ads, transparencies, moving reports to the right, etc. perf and product do tradeoff against each other.. and necessarily slow down the process of building/rolling things out and making them faster. Again, I don't mean to be making up excuses on any of this at all, I am just trying to explain where we are with some specific examples.

- (C) re: timeline, I think there are a few specific things we should do here to help get to a better place.
  - -- (1) we should obviously keep people like steve totally dedicated to perf, and if we are really focused on it dip back into sitespeed team / though I am not sure that is the right company optimization
  - -- (2) Wherever possible on the margin we should trade away from product and towards perf... that means that even if the cost is minimal, we should not do things like transparencies on OG stories that run directly contrary to our perf goals for moderate/small product wins. I am not sure there is much more to do here.
  - -- (3) If perf and being bug free is the prime directive, I really wonder if we can't remove some of the product complexity we have built up and trade back towards perf & quality. The biggest change we could make here that would instantly return big wins would be to step away from a two column layout (or, less drastically, do more of the height calculations server side and tolerate occasional mis-ballancng of the two columns)
- (D) 'Quality' overall / more broadly — The thing I do worry about here is what we mean by 'quality' on our current existing product vs. building systems and making product decisions which set us up for having high quality in the long term. I actually think that the reason we have 'quality' issues in our product today across the boards is because we have over-optimized specific interactions / product functions at the expense of an overall maintainable and nimble system in many places. This is part of the longer strategy conversation above... but suffice it to say that the fewer abstractions that exist and pixel optimizations we do / the more we have a unified core / the faster we can move. This goes for everything from design all the way through to the settings we use and our policies.

(3) Timeline Rollout Speed -- I am personally pretty frustrated that we are only as of last week turning on viral channels / are still weeks away from a full rollout. At the beginning of Jan / the last time we really talked about timing / we said we would be ready to force in by the end of FEB / at a quality level to do that — I don't think we are massively off on that, but I do think we are off by a few weeks. I know getting timeline out 'well' is really important to the company (and my credibility) — I was a bit blindsided by revenue as a core factor in this as of early Jan, and I was also blindsided by early data-quality and then later-braking test bias issues... but I do think we are still doing pretty well here in terms of product reception / even if we are not moving as fast as we would like.

(4) Timeline Team Leadership Overall — I am pretty concerned about where we are with this... let me roll through each discipline:

- **Product Management:** Justin and Mike are good / I think are really meaningful assets, but they are not right for the timeline. I think they would both personally under-perform but for different reasons --- Mike is too new and Justin isn't detail focused enough. I have tried to put them each in contexts where they will be engaged and will succeed — and my belief is that having them on timeline will accomplish neither. I don't think either of them will end up bing a product head for the group that we want (though I could be proven wrong) -- I think they are both top notch / the best ICs that can do great things if given the right framework to succeed.

    - I will also note that I am really having a hard time getting a good PM resource into timeline from within Facebook. I thought Josh Williams (who is very quality and design oriented) would have been great, and in one quick conversation with him he seemed theoretically really excited about the role, but Schrep (and Peter) really wanted him on location stuff, and so that is where he went… though I am not sure that was the right call for him or the company
    - I am hoping to get one of our more senior / experienced new-hires, but because timeline is seen as last year's product and Mobile / OG is where all the sex is, it seems that it will be where good people will gravitate.
    - I know the primary mandate / agree with the idea that we should be hiring in great people rather than resource shifting, but especially for PM around identity I feel like we face an issue which is that we think people are good if their projects are good /going well… so the people we want we have a hard time getting, and we don't want the others.
- **Eng:** Boz is more engaged, but I don't think he is ever going to go deep with the identity team from a technical perspective / be deeply invested beyond as a cultural leader. While there are still rough patches, our relationship is getting better / there actually is some real mutual respect building, but given his view of himself and his role at Facebook I just don't see him ever leading on the ground / working really hard or getting into the weeds of problems (timeline or otherwise). I don't think this is me / or the org change — by my understanding it is consistent with how he was functioning reporting to Badros as well. I think he is the right partner for now, but I think it is highly likely that he will either leave or be effectively a lame duck after a point. I already feel like he isn't working enough hours / hard enough.
    - In terms of building a bench around him / the next step, I do think that some people like Paul will in time be really good, but as you have noted they probably are not there yet
    - Hiring outreach for more senior Eng leadership that Boz has been mostly doing hasn't been super hot so far… Again, to state a truism, the people that are really good are the people that are hardest to get.
- **Design:** Already discussed / hopefully we get this candidate in, and then she proves to be really good and will help scale out a team.

(5) My interactions with m-team / strategy vs. execution - I think there are two intersecting problems here… especially with my contributions to specifically OG conversations… one is the 'topic' / focus area, and the other is personalities. I will work on both, but let me just express where I think I am first.

- **On personalities —** especially within the 'small group'
    - My honest assessment is that Cox and I have a strong relationship / see eye to eye and I feel like I can have an open and clear discussion with him. I don't think there is an issue where I can't have a voice with him and I think he respects my opinion — He might not always agree, but I don't think there is a communications issue. I also think that Badros and I have worked together enough that there isn't a communications issue really.
    - I do not feel the same way about my relationship with Brett (or to a lesser extent Schrep)… I think there are a few things at play here but (1) I just haven't spent as much time with them, which I need to try to fix / perhaps set up one on ones with them for a while to try to build more of a relationship (2) I think that cox and I tend to think much more similarly as a default, which makes bridging the gap with Brett / Schrep harder.
    - I agree with your assessment that the only thing I can do here to build trust and respect is launch timeline and do other tactically good things here. Not much more to say on this.

- **Content / holding back too much on mobile & platform discussions**
    - The first reason I have been a little restrained on these topics is I don't really have a sense of how many degrees of freedom we are talking about with regards to mobile or platform in our current strategy. I don't know if any of the above points are even worth bringing up, or if we are in more of an execution phase right now. I am always trying to feel out when I am 'derailing' conversations vs. when I am actually helping them along, and clearly have more to figure out here.

    - The second reason I feel like I have been a bit hesitant in these discussions is that I really have felt to date like our decisions on how OG integrates with timeline (and the changes we are going to make to boost up OG) have been relatively straight forward / we have already made the calls on them for the immediate term… and that I don't know all the nuance when talking from other perspectives tactically (NF/OG core/etc) There is no question that we have things from a timeline perspective to contribute…. But I also do really feel like newsfeed & ticker are the main distribution points for OG that are more in question (and more short term) -- and so to me it makes more sense that a lot of our discussion has not been very 'identity' team focused. Again, correct me if you think I am off base here.

- **Separately, on our hiring conversations of the last two weeks…**
    - On the senior level I have a bit of a disadvantage here because I don't know a ton of senior people at google / in the valley that are easy to draw on. I do actually know a ton of startups / people from the wider ecosystem and have been trying to press on the areas I know and from a broader perspective I think have been funneling good prospective acquisitions into the pipe in several places at a minimum (people like chartbeat, waze, etc.)
    - I also feel a bit at a disadvantage because even more than distribution, and certainly more that mobile — it is a pretty rare breed that is going to be ideal to slot directly onto the identity team. We don't have a clear external corollary in my mind for what we do / want to do. It clearly isn't microsoft passport folk. This makes it hard to target people in particular — though to date people are really interested in what we do / excited about it — which, of course, is useful.
    - That said — again, I am working on this / want to be as leveraged as possible and will continue to do what I can.

Perhaps in overall closing, I know that this isn't a polished assessment in many places… but I really wanted to share a lot of my thinking with you. I am really concerned about the company overall being over-extended and not focused on the areas where we have the greatest natural advantage. On the team side, I am working hard on building a culture around the identity team, and I think it is going pretty well with the core members -- but I think we have a long way to go and it will take us a while to be at full steam….

Obviously happy to discuss everything / anything with you on this at any point — and, despite some of the doubts I express to you — I really would follow you on any strategy you outline / I hope you know I am fully all in & committed.

Sam