EXHIBIT 183

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From: Peter Martinazzi </O=THEFACEBOOK/OU=EXCHANGE ADMINISTRATIVE GROUP (FYDIBOHF23SPDLT)/CN=RECIPIENTS/CN=PMARTINAZZI>
Sent: Tuesday, January 07, 2014 8:08 PM
To: Douglas Purdy
Cc: Mike Vernal; Vladimir Fedorov; Ilya Sukhar; Eugene Zarakhovsky; Javier Olivan; Naomi Gleit; Luke Woods; Raymond Endres; Alex Schultz; Kevin Lacker; Eddie O'Neil; Vishu Gupta; George Lee
Subject: Re: Thoughts on a Messenger Platform

- Mark
+ Vishu, George

Based on our conversation earlier today, I wanted to summarize a few of my thoughts.

I think **the overarching goal here should be to change the trajectory for Messenger**. If Messenger doesn't become the defacto communication solution for 1:1 and small groups, then it really doesn't matter what platform is built on top of it. So the goal of whatever platform things we do over the next year should be to change the trajectory for Messenger. (If something increases sends by 2% that doesn't matter, if we think it will doubles sends, it's worth working on)

Within that, the next question is, how can a platform of some sort help out there. The thing that comes to mind for me is **it can scale things in a way a team our size cant do internally**. I'm not sure yet whether that is valuable or not, but it's something that I think is worth thinking about applying to everything in Messenger and seeing whether that could be high impact. If we scale that thing in some way, does that make it a differentiator or something that can materially change our trajectory. (I'm interested in hearing other high level things it could do to beyond scaling stuff, because I haven't thought much about platform).

--- Brief aside about the direction of messaging ---

I think it's important we scale the product Messenger will become, not what Messenger is right now. 6 Months ago, FB Messaging was "1:1 text chat while browsing FB". With 3.0 our focus was to change this to "1:1 text chat *anytime, mobile-to-mobile*". Now our focus is to transition to "*(1:1 and small group) communication (with different content types: text and photo and voice and stickers ...)* anytime mobile-to-mobile". We want to become **the defacto every day communication platform**, something that makes SMS look like smoke signals.

In Messenger there are **audiences**. Right now the main audiences that users interact with is a 1:1 convo with their friend. We plan on building out a spaces (groups) product where groups are an as important audience as people, and in the future an audience will be either a person or a small group.

Right now, you choose an audience and then you communicate with them, this is **audience-first sharing**. You do this by sharing **content** that can be different **content types.** Right now, text is the main content type we focus on, but we support others (photos, stickers, voice clips). In the future we plan on making it even easier to share these other types of content so that often times it's less friction to send a photo of you with your friend than it is to type 'I'm with raymond'. For some content types there are different pieces of content you can share within them. For example, the stickers content type has lots of different pieces of content inside of it. This would apply to other things like memes and links that aren't really created when you share them but are more picked when you share them.

In addition to the current model of audience-first sharing we want to enable **content-first sharing** directly in the app. This is when you have a piece of content and then blast it to audiences. For example, I could send a funny link to Mike, Luke, and my group that is my brothers. This would generate 3 copies of that content and in turn three conversations around that content. This would also be something I want do if I want to reshare a link that was sent to me.

To add visuals to these concepts:

- Here is a potential new version of the composer after you've selected an audience that makes it easy to send different content types: https://facebook.box.com/s/uva38ikio2aaaekz3iu3
- Here is a potential version of content-first sharing from within the app: https://facebook.box.com/s/m92bzimvzka86fxgy8d2

--- Now Back to Scaling Things ---

So given all of that, what kinds of things are their to scale?

1. Entry points for content first communication
2. Entry points for audience first communication
3. Content within a given content type
4. Content Types
5. Other audiences aside from friends, contacts, and groups of them

1) Entry points for content first communication

This is like the Android intent or the Send Plugin currently being worked on. These are things that are attached to a given piece of content on your phone and make it easier to share that to multiple audiences.

I'm not sure of the impact this will have (given we already have an Android intent) but seems relatively easy and we are already working on it and has the potential to get more content into the system. (Some data we should look at here is see how many photos are shared via the Android Intent, and how many messages in general are shared through it).

I'm not sure of other platform-based ways to scale this out.

2) Entry points for audience first communication

This would be something like, if I'm playing words with friends with Doug, there can be a "Chat with Doug" button, that when I click, pops open a chat head with Doug, so I can easily talk to him while playing the game. On iOS it would fast app switch to Messenger and then we could have a back button that brings you back to the app afterward.

I'm not entirely sure if other apps would want to implement this, but it generally seems positive as well.

3) Additional Content within a given content type

Some content types have a bunch of pieces of content inside of them that you pick out and share to express yourself. Stickers is the one example we have right now, but this would also apply to things like Memes, Links, Animated Gifs, etc… And even to more unique content types like games if we had them.

As more content types come to exist it's really important to nail content discovery and the picking out experience. For example, it needs to be easy to find the right sticker pack you want to download, and it needs to be easy to choose the perfect sticker to express yourself. This is an important thing to solve.

In addition to that, we could scale this out in a few ways. One way is with a content-creator approach, where it's really easy for official content creates to create or curate content. For example, professional illustrators can make stickers and a curator of some sort could pick out the 20 'angry face' animated gifs. Another way to do this is to take it further and crowd source it to all users. This would allow for tons of content of the types we've chosen to flow through the system and be easily choosable from as a way to send.

It would be helpful to understand here how much value each additional sticker pack provides and if we think scaling it out could provide significantly more (my gut is it could, but it would be nice to have more data here).

We can do this for Stickers as well as future content types that this makes sense for.

4) New Content Types

It's important that we have a nearly frictionless experience for the head types of content. These are things like text, photos (including selfies), video clips, audio clips, stickers, and location. I actually think there are a few more we would do ourselves as well if we were making a platform.

But there are way more potential types of things you could share beyond that, where the friction to create it is less of a problem. For example, a marauders map type of content that constantly shares everyone in the threads location once they opt in for x amount of time, or if I want to send a poll where everyone in a group can vote, or if I want to share a todo list with the group. These are things that aren't as common so its ok if they aren't easily accessible and low friction, but do provide enough value where it's still probably worth a few extra clicks to do them.

Enabling ways for more content types is something that could be good as well but I imagine its a lot more important to nail the head first.

5) Other audiences aside from friends, contacts, and groups of them

This would be enabling interaction with pages or apps. This is the kind of stuff where you can talk to your electronics, place an order at a restaurant, or .... I haven't thought through this as much (I want to think through, what would it be? would my mom do it? Would my 13 year old cousin do it? Do I think it will change our trajectory or be a real differentiator?)

Any other ideas of types of things to scale from within the base product? Or of ways to scale things?

I think a lot of these could make sense eventually, but I'm not convinced right now is the right timing to build/release them. I'd personally choose to invest heavily eng-wise in getting out the future base product out there (spaces, content first, the head content types, etc...) since every month we fall further and further behind versus building it with platform stuff in parallel. But regardless, we should think through them.

Beyond that, what other things could a platform type mentality unlock. Something that doesn't really fit into scaling is building a separate chat plugin (that doesn't use fb messages at all) that developers can use to easily embed chat in there apps. Any other ideas?

From: Douglas Purdy <dmp@fb.com>
Date: Tuesday, December 17, 2013 9:55 PM
To: Mark Zuckerberg <zuck@fb.com>
Cc: Mike Vernal <vernal@fb.com>, Vladimir Fedorov <vladf@fb.com>, Ilya Sukhar <is@fb.com>, Eugene Zarakhovsky <eugenez@fb.com>, Javier Olivan <jolivan@fb.com>, Naomi Gleit <naomi@fb.com>, Luke Woods <lukew@fb.com>, Raymond Endres <raymonde@fb.com>, Alex Schultz <aschultz@fb.com>, Peter Martinazzi <pmartinazzi@fb.com>, Kevin Lacker <lacker@fb.com>, Eddie O'Neil <ekoneil@fb.com>
Subject: Re: Thoughts on a Messenger Platform

Mark:

A few of us got together today to discuss Messenger Platform and your note. We have some time reserved to discuss this topic with you tomorrow, but we don’t think we made enough progress for that discussion quite yet. Our plan is to continue meeting as a group and then meet with you early in the New Year. That said, happy to keep the time if there is additional context you want to share or if you want discuss our progress (highlights below).

#1: Person to Person: Send in Messenger

Platform has been slowly working on this feature and we should be done with a cut of the product in January to review with you + Mike. Not likely to immediately help with reach (# of people using Messenger) or richness (most requested content types), but it should help with branding/awareness (similar to the Like Button) and have modest uplift in sends. Latest mocks: https://facebook.box.com/messenger-platform.

#2: Messenger Content Platform

This divides up into two areas: Sticker Platform + Games Platform. The Messenger team has done some work on Sticker Platform, but found that they could get all the 3rd party stickers they wanted without it, now focusing on the top 1st party content types (video, selfies, etc.). On the Games front, we are still thinking through what it would be mean to play turn-based games in Messenger. Doing some technical and design work to see what is possible in terms of content types and game play.

#3: App to Person: Messages + Agents

We talked about a couple of different scenarios here including Uber (car is here), sales (10% off), etc. The one that seemed the most compelling was local (maybe we even have a "Places" tab in Messenger). We need to spend a little more time discussing this idea as a group before we have definitive next steps.

#4: Messenger Agents

It seems like we could build #3 and then layer in #4. We think the natural language element of this would be the most difficult, so you could imagine only supporting a domain specific language. We are going to see if we can make progress on #3 and then revisit.
On Dec 9, 2013, at 12:45 PM, Mark Zuckerberg <zuck@fb.com> wrote:

I've been thinking a lot about what a Messenger Platform might look like and wrote up all of my thoughts on this. I wanted to be thorough, so this is very long. Hang in there.

This work falls between two teams -- Messenger and Platform -- and as such I'm worried that we're not thinking about it enough or prioritizing it appropriately.

Given how important messaging has become relative to News Feed where our platform primarily focuses, it seems like there is a large opportunity here, both to extend Messenger in a unique and differentiated way compared to other messaging products and to extend our development platform to another extremely important surface.

For Messenger, I think differentiation is extremely important and something we haven't focused on yet. We've spent the past 6-12 months catching up to WhatsApp and competitors on table stakes like performance, reliability, pushability, etc. This work isn't done and we will continue to do it, including catching up in areas like groups.

But to get people to ditch WhatsApp and switch to Messenger, it will never be sufficient to be 10% better than them or add fun gimmicks on any existing attribute or feature. We will have to offer some new fundamental use case that becomes important to people's daily lives.

The reason a platform is attractive to me is that it is completely new and brings different kinds of utility and content into the Messenger experience. Just like News Feed started out as friends content only but eventually expanded to include more content that is now critical to everyday engagement, I think there's an opportunity to do this with Messenger as well. We will face some of the same criticisms like "this experience should just be about my friends", but with the right defaults, controls and separation between the different types of content, I think we can make this a massively net positive addition without meaningfully reducing signal to noise or making the app worse for messaging friends. I will get into details below on the options for doing this.

As a side note, just like with News Feed, I think expanding beyond just friends content will eventually be the single key to turning Messenger into a business. We're obviously not focused on this yet, but it's worth considering because if this assumption is true, then it suggests others will eventually evolve in this direction too. We have meaningful advantages in infrastructure, SDK deployment and experience in executing a platform direction now. But if we wait too long, those advantages will go away.

For Platform, the key reason to focus on this is opening up another valuable surface beyond News Feed. This is important not only to expand, but because News Feed and therefore our current platform may actually be getting *less* valuable over time as more behavior moves to more private mediums like messaging.

Our current platform plan feels like we're going to invest a lot of energy to improve the Platform / News Feed experience for little marginal gain. The surface feels mature to me already. We're focused on quality because we think the issue is that the cost of using our platform is too high, when the bigger issue is that the value is too low compared to emerging alternatives. We do have real quality work to do, but we must also acknowledge that no amount of quality fixes will increase the potential value of using our platform.

Focusing now on opening up a new surface seems like a much better strategy. Perhaps the biggest reason we're not considering it as much as we should be is that organizationally the Platform and Messenger teams are very disconnected, so it is not within the Platform team's scope to unilaterally decide to do this. This is an unfortunate reason not to explore this.

The other major reason we haven't considered it sufficiently is because fear of spam. This is a completely reasonable fear given the experience with our existing platform when we rolled it out. That said, we have since gotten platform to a place where it is not spammy and it is a small part of our overall signal to noise complaints, while still building it into a $1 billion business.

This proves to me that it is possible to build something extremely valuable that is not spammy. In fact, if we had gotten the quality balance right from the beginning and not repeatedly thrashed our ecosystem, our platform business and engagement from it would almost certainly be much larger than even the $1 billion it is today.

So not only do I think it's possible to learn from our mistakes and do this well, but I also think this is probably a much bigger opportunity than we intuitively estimate if we can pull this off.

Below, I'm going to outline four basic Messenger Platform ideas. They go from smallest and simplest to biggest and most complex. They are not mutually exclusive, so we can and probably should do multiple of them.

1. Send a message API / button

Many apps have Like and Share buttons, encourage you to write on friends' timelines and use other broad social distribution tools. However, sharing is becoming increasingly private and many more people would use these to if you could just send a message to the right people privately with the content instead. If something like this was widely deployed, people would probably share a lot more content from apps. Surprisingly, there is no easy way to do this.

This is the simplest and easiest thing we can do. It would probably be meaningfully positive for developers, somewhat positive for differentiating the Messenger brand as more of a app-wide utility, positive as a feature for people when they need it, and it would not meaningfully change message volume. Still, this is low risk and positive for all constituents, so it seems like we should push further on this.

Historically, we might have had to worry about apps incentivizing people to send spammy messages to their friends, but now we effectively enforce our policies and can limit any behavior we don't want. Because of this, I don't think spam is a meaningful risk.

One nice feature of this is it's quite defensible and uses our installed developer base.

The closest thing that some apps do is import contacts and encourage you to SMS, but that's very heavyweight if the app has no other reason to import your contacts. By contrast, most apps have the Facebook SDK and if the person signs in then the app can get this capability for free.

SMS will not get meaningfully simpler over time because it is decentralized and nobody has an incentive to improve it. In some countries it also costs money, which is a fundamental disadvantage.

WhatsApp has no developer platform. I'd be surprised if they didn't develop one over time, but our deployed base is a meaningful advantage here.

2. Content Platform in Messenger

Stickers feel like a toy implementation of expression. They're valuable and get at a fundamental need for lighter weight emotional expression and engagement, but having single companies curate sets of stickers seems very suboptimal.

Over the long term, it seems like independent artists must be able to submit and sell their work, and there must be some kind of store that more effectively surfaces new content such that the best content can win. This will increase the overall quality and breadth of the content and make it a completely universal feature that everyone uses everyday.

It also seems thinking about this as just stickers is very limiting. This should include other forms of expression and entertainment, including memes, videos, games and other kinds of content that can be consumed within a message thread. The long term state of these will be to have an open market for them as well, rather than a list of content curated by individual companies.

It will be pretty easy to do this for stickers, memes and videos. For stickers, this could be a real business for artists and many will also want to distribute their work for free. For memes and videos, I assume they will be free and people will distribute them for free for fun. This will likely increase message sends and engagement on Messenger, perhaps meaningfully, but I doubt it's fundamental enough to get people to switch apps.

Where this really gets interesting is with in-message games and apps. It feels a unreasonable today to expect people would play many games in messages instead of switching to the game itself, but as messaging becomes and increasingly central part of the mobile experience, this inversion seems quite possible to me.

I actually think it's a great aspirational goal to make messaging central a lot of how people interact with content on their phones. The more ambitious ideas below will focus on extending this theme, but for now we'll focus on games.

Intuitively, it's easy to imagine simple games like the rock-paper-scissors prototype built in a hackathon to be played in messages. Sending stickers back and forth often feels like a game, and this feels like extendable behavior.

We could probably build a simple game platform based on the idea of people being able to send stickers (as "moves"). Whenever a sticker is played, the app the sticker is associated with gets its server pinged so it can update the game state, and then the app can send back an image or video of the game state that gets embedded in the thread.

This would be simple enough to make many fun games like rock-paper-scissors, hangman / Words with Friends, all of the popular quiz games and essentially any game with an interface that only requires tapping on an item to play it.

This would likely be pretty easy to build and would be very fun. Given how popular these games are, I think there's a meaningful chance this turns into a big growth driver for us.

Keep in mind one key advantage this games platform has: you don't need to install a game to play it. If someone sends you a sticker, you just get the stickers you need to play immediately. Games will spread extremely virally. Given how big these quiz apps and things like Words with Friends have gotten, it's not out of the question that this could drive 100 million more people to use Messenger.

That said, there are a few big limitations that will make it harder to expand beyond this in games:

- Due to policy, we can't build a full executable environment within our apps. This prevents apps with richer UIs from working in this way unless we develop a workaround.

- Lack of business model will make it less likely that the best developers invest in this. This isn't hopeless and there are possible ideas to explore like paid stickers with special powers similar to what you'd pay for in a normal game. Even without this, indie developers will still be able to make a lot of good simple games anyway.

- Even if we solve these issues and enable more complex games, at some point it will always just be better to play a more complex game in its own app. So it's probably worth investing to move the line of what people play in messages somewhat, but not a huge amount.

Overall though, building the simple games platform described here should be very easy and high leverage, so I think we should consider doing this soon. I also think moving any content we offer -- including stickers, memes and games -- into a better designed and more open market that anyone can contribute to.

3. App-to-person messages and agents

Those are the basic ideas. Here's where I think it actually starts to get really powerful.

Beyond enabling person-to-person messages and interaction, another obvious mode to consider is app-to-person or public figure-to-person messages and interaction.

Making this leap is somewhat analogous to News Feed transitioning from being just about friends to then being about both friends and public entities. This will fundamentally change the definition of what Messenger is, and that will come with some tradeoffs, but ultimately I think it will be overwhelmingly positive if we pull it off.

I also think it is one of the only truly game changing ideas I've heard that is a fundamental enough differentiator to make every single person need to use Messenger in addition to SMS or WhatsApp if we can make this work.

Before getting into how this would work, I want to address the obvious issues of spam and signal to noise.

A key difference from our initial Platform launch is that now we have the right tools and we've learned the right lessons to make this a much better experience than our News Feed / notifications platform ever was right out of the gate.

To prevent spam, the biggest thing we can do is make it easy to turn off (after you turn it on). We never got this quite right on desktop and even iOS and Android struggle with this. But it should be easy in Messenger.

The first time you get a message from a public entity, we should give you a prominent button to turn off receiving any more messages from that entity. After that, we should just make it so that if you ever delete the thread, we either kill the entity's ability to message you again or at the very minimum ask you if you want to do that. That interaction is much easier on Messenger than it ever was on desktop, so I expect people will use it.

Those controls -- plus only allowing apps and entities you've connected with to send you a first message to begin with -- should be sufficient to make sure people don't get messages they don't want.

It's also worth noting again that we are much better at manually enforcing policies than we used to be, so we can now set policies like "no mass messages to all your users" if we want. I'm not sure we'd want that rule specifically, but we now have the capacity to enforce this effectively if we did.

After spam, the next question is signal to noise. There are several variables we can play with here including having a different push sound for entity messages, not badging for entity messages, no push sound at all for entity messages or theoretically even not pushing entity messages at all and only having them be visible when you open the app. That last option is not recommended long term if we want this to be viable, but it's an example of the full spectrum of options.

So that's a short version of why this doesn't have to be a bad experience. Before getting into why I think this could be awesome long term, I want to call out two competitive near term issues we face.

The first is WhatsApp adding a feature like this for public figures. Line and Kakao already have something like this for public figures, so this is completely reasonable to expect and we've even heard they're working on it. If the space is going to move in this direction, being the leader and establishing the brand and network effects matters a lot. This alone should encourage us to consider this soon.

Another anecdote is from Twitter. One day I was in a restaurant and heard one guy tell his friend that he just got a text from Jay Z. He hadn't. He got a tweet and was subscribed such that he got notified about it. The reason this is interesting to me is that we have this religion around thinking about these channels as completely separate things and most people do not.

There is no rule that public figures go in feed and private messages go in Messenger. Over the long term, I think public figures and apps will go in Messenger just like friend messages, and private photos from friends will go in News Feed as well.

When the world shifts like this, being first is how you build a brand and network effect. We have an opportunity to do this at scale, but that opportunity won't last forever. I doubt we even have a year before WhatsApp starts moving in this direction.

Now, I'd like to finally get to what I think this could be.

In its most basic form, this is just another notification channel or where for apps and public entities to reach people. This is not why this is exciting to me, but it's worth calling this out because it is extremely powerful. This alone, even with many limits and controls, would add a lot of meaningful content to Messenger and would make every single app and many public figures want to integrate with Facebook much more.

Like I said above, I think we have the controls and experience to make this a good experience. An app or page would only get the ability to send you a first message when you connect to it, so there will be no viral spam. And it will be easy to turn off inline, so there shouldn't be direct spam either. This is simple but powerful and we shouldn't overlook this.

What's more exciting is not just using this as a one way push channel, but making it so you can interact with apps through Messenger.

Going back to the games case from above, it is conceivable that messaging will become the backbone of all of your activity on your phone and that apps will start to become a feature of messaging rather than messages being a feature of apps. This has happened with many important services that were once believed to be small features. People thought search was a small feature of websites but now most things are features of search. People thought social networks and communities were features of sites, but now most types of content are features of Facebook. It's conceivable that as messaging grows in importance, many apps will become features of messaging.

The simplest possible UI for this is that you send a message to an app and it does something and sends a message back. This could work for pages and places too.

For example, I could send a restaurant a message asking for a reservation and it could just reply with whether I got the reservation and when. From a user experience perspective, this is a great UI. I don't want to have to call a restaurant and the apps to do this are terrible. Also, the natural language processing to do this works with modern techniques so this isn't a huge technology challenge either.

I could see this working for ordering food, getting movie tickets and so on. Nobody wants to talk to people for these tasks, but the most efficient UI by far is language.

I could also see this working for having limited interaction with public figures. There are probably certain basic things that people want to ask public figures, and those could easily be programmed in to facilitate a more personal two way interaction.

With a mature version of this system, there would probably be lots of new use cases that don't exist yet.

That said, one significant limiting factor of this system is that you'd have to know about an entity before you could message it. So if you wanted to get a cab, you'd need to know to message Uber. And by that point, for apps with good UIs like Uber, you might as well just use their app and not learn a new messaging behavior. I will address this problem in the final and most ambitious idea below.

Overall though, there are several powerful ideas in app-to-person messaging that are simple and valuable. Basic push messaging with good controls is an important feature and may soon become a point of competition. An API for apps and pages to reply to messages would also be simple to build and could unlock some interesting use cases.

4. Messenger Agent Platform

The ideal agent would not require you to message a different thread for each type of task you want to complete. Instead, the ideal agent would be able to take any natural language input string and do what you intended.

One reason I believe in this so much is because I find I'd much rather communicate with Andrea via Messenger than use different apps to complete tasks myself. Natural language is a great common UI for any task you can imagine, and having an agent that can handle new kinds of tasks -- including ones you've never dealt with before and don't have an app for -- is extremely valuable. If Messenger came with an Andrea for everyone, that would clearly be amazing for the world.

This is more of a Platform / Utility challenge than a messaging one, but if we could enable this it would definitely drive adoption and engagement with Messenger.

This agent problem is similar to building Siri or Google Now, except I think it's an ideal problem to use a platform approach to solve, which is very different from what Google and Apple are doing.

Google approaches the problem from the perspective of search. Their default is to show you links where you might find your answer. Sometimes they give you an answer to a question directly, but they don't carry out commands for you. They're building conversational search to fit more into this agent-like use pattern, but they view that as a layer on top of search rather than the core product.

Apple approaches the problem as a closed system, which is typical for them. I actually think they are doing some things very well, especially in designing Siri to execute commands in addition to returning information. However, they're very limited by a couple of things, including not supporting text (only voice) and only supporting services they integrate themselves (closed approach).

I think there's an opportunity to use a platform approach to build a superior execution-oriented agent.

The way it would work is that developers could register their own agents with us that could handle certain types of requests. When we get a request, we'd figure out which developers' agents might be able to handle that request and we'd send it to them. They'd each return whether they needed more information or were in fact able to handle the request, their suggested response and maybe a confidence score for how likely they think they are right. We'd then pick the one we think is best (calibrated by people's feedback) and then we'd ask the person to confirm. After confirmation, we'd execute the plan using that developer's agent.

For example, you might say "get me two tickets to see Hunger Games at 9pm". We'd parse the input enough to understand it's a movie ticket request and then pass it off to Fandango and others. Fandango might require a theater to be specified but another agent might just use your location and has a good history of positive results, so we'd go with that one and reply with a message like "I can get you two tickets at Shoreline at 9:10pm. Is this right?" After that, you just hit the Like button and we'd pass your credit card number off to the service to complete the transaction and maybe send you another message with a link to the record of the transaction.

This idea is clearly the most ambitious and riskiest of the ones I've mentioned, but I think this approach could work well and be quite leveraged. With a relatively small team at Facebook, we can build a system that leverages a lot of work outside. We can build a developer community that builds agents for virtually any task you can imagine, ranging from running errands to looking up shuttle times or other information.

To recap, the four ideas discussed above are:

1. Send a message API / button
    - Enabling devs to promote person-to-person messaging in their apps

2. Content Platform in Messenger
    - Turning stickers into an open market
    - Adding memes and videos to the market
    - Building a simple in-message games platform

3. App-to-person messages and agents
    - Enabling apps and pages to send messages, with strict controls to ensure quality
    - Enabling people to send messages to apps and pages and get replies

4. Messenger Agent Platform
    - Building a meta-agent and a platform to let agents register to take on tasks

If we want to develop some of these ideas, we would need to determine which teams will be building which pieces. I imagine that the Platform team would do most of the work for (1) and (4), but the Messenger team would perhaps do most of the work for (2) and (3). All of these will likely require some work and full support from both teams though, so a good next step is to figure out which of these we are excited about building now.

I know this was very long, so I appreciate that you've read all the way down to here and are taking the time to consider this carefully. If it makes you feel better about having to read all of this, just think about how long it took me to write on my phone! Please take the time to think through this, because I wouldn't have spent hours writing it up if I didn't seriously believe there was something important here to consider now.