

EXHIBIT 78

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

Login v4 (+ Platform changes)

Update: 1/27/2014

Speaker Notes for Slide 1

[theme] User Trust
1/ Login v4 [permission x-out]
2/ Login review

[theme] Developers
1/ Versioned APIs
2/ Commit to 2yr stability for core Login, Sharing, Payments, Ads APIs and SDKs [iOS, Android, JS, PHP]
3/ Bug fix SLA [UBN, hi-pri]
4/ Sticks
= remove friend data APIs
= remove APIs for News Feed, Timeline, Notifications, Inbox, managing Friend Lists
= opaque user IDs for all apps
= non-app friends only available by approval
5/ Carrots
= Social Context API [e.g. facepile - answers questions like "which friends are connected to Batman via which edge types?"]
6/ API deprecations [mostly dead products]
= Checkins
= Locations
= Pokes
= Subscribers / Subscribedto
= Ouestionations
= Instant Personalization
= Start Now
7/ API calls w/o access tokens [going to try this, might not be possible]

8/ for games on canvas that use credits
8.1/ Invites API [for custom MFS]
8.2/ Access to all friends for cross app promotion

9/ Other
9.1/ Pivot the developer docs around Facebook Sharing as a product [sounds like this launched Tuesday]

Goals

1. Increase User Trust

1. Give people control of info they share [permission x-out]
2. Increase quality of FB integrations in apps [Login Review]
3. Minimize surprise [policy + friends model changes]

2. Increase Developer Trust

1. Versioned APIs
2. Stabilize core APIs [2 year stability per Platform version + bug fix SLA]

3. Protect the Graph

1. Make it difficult to connect graphs between apps [opaque IDs]
2. Limit data available to apps [remove friend APIs, privatize high-value APIs]

API Privatizations

- Available via whitelist / contract
 - News Feed
 - Timeline
 - Inbox / messaging
 - Notifications
 - Requests
 - Friend List management

API Deprecations

- Access to friend data [likes, photos, checkins, etc]
- Questions, Subscriptions, Checkins, Pokes
- Apps reading public posts for TOS'ed users

Affected Apps

	Total	> 1,000 MAU	> 10,000 MAU
API callers [last 30d]	1.4M	17K	3,206
Affected apps	27,019	7,744	2,532
Affected games	3,111	1,475	521
Affected Credits apps	338	315	253
Affected Ad Spenders	89	82	60
Affected Salesforce apps	1,639	1,262	812
Affected Salesforce games	458	375	253
Affected Salesforce non-games	1,181	887	559

Speaker Notes for Slide 5

<https://our.intern.facebook.com/intern/argus/view/193070>

Key apps

	# of apps	% requesting read_stream
Mark's friends	31	76%
Sheryl's friends	66	62%
Generating TPV	332	51%
Neko spenders	831	59%
Noisy	23	82%
T0 / T1 partners	160	77%

All on a list for pre-launch outreach

Model changes: highlights

1. App scoped user IDs

- Each partner has a unique ID space for users
- Makes on-trivial to connect graphs across apps
- Makes it possible to audit data leaks

2. By default, apps can only read app friends

- Common case for most apps
- More difficult to grow Lulu, Circle, Klout, BranchOut, etc.

Speaker Notes for Slide 7

Example: A TOS's w/ all friends, B TOS's w/o friends. if app doesn't know NAF, /A/friends didn't already include B

Affected Apps

1. Difficult / impossible to build [without contract]
 - Alternate FB clients [Flipboard]
2. Hard to grow
 - Messaging apps, contact sync apps, horoscope apps, birthday notifiers, gifting apps [ex: Wrapp]
 - Lulu, Klout, BranchOut
3. Good apps
 - Venmo

Speaker Notes for Slide 8

Wrapp: <http://crunchbase.com/company/wrapp>

Impact

1. Canvas Games that use Credits:
 1. Adopt app scoped IDs
 2. Adopt new model for cross-app promotion
 3. Adopt new model for custom MFS
 4. Loss of ranking signals for friends

Impact

2. Non-game apps / non-Canvas games

1. Lose access to non-app friends
2. Adopt app scoped IDs
3. [harder to build] FB replacement clients [e.g. Flipboard]
4. [harder to grow] Lulu, BranchOut, Klout, messaging / contact sync / gifting / horoscope / birthday notification apps
5. Casualties: Venmo, etc.

Speaker Notes for Slide 10

Wrapp: <http://crunchbase.com/company/wrapps>

Login v4



- First screen
- “App can’t post” text
- Second screen
- People have line item veto on permissions [except public profile]

Launch Timing

- ~March 12: launch Login v4
- April 30: developer event [f8?]
- Want 6-8 week separation between events

Questions

1. Acceptable to deprecate Feed given broad impact?
2. How quickly should apps lose access to Feed?
3. Acceptable to make medium-term trade of Trust for Games?
4. Commit to never removing basic Login from an app?

Appendix

Canvas Games using Credits

1. Goal: minimize impact to Canvas games
2. App-scoped IDs for new users
3. Access to all friends for custom MFS [invites]
4. Data
 1. ~900 financial entities [80% associated w/ multiple apps]
 2. ~2,300 active payment enabled apps [last 7d]

Speaker Notes for Slide 15

Changes from Vernal:

1/ use a Social Context API for cross app promotion

2/

Non-game apps / non-Canvas games

1. App scoped IDs for new users
2. Apps limited to reading app friends
3. Review
 1. [low bar] Access to API for tagging people
 2. [very high bar] Access to non-app friends
 1. Venmo: yes
 2. Lulu, Circle, BranchOut: no

Speaker Notes for Slide 16

Investigate a functional Invites product

Developer Trust

1. API versions [GET /v2.0/me/likes]

2. Bug fix SLA

UBN	24h
Hi-pri	10d
Mid-pri	30d

2. Won't publish TATs

3. Near term, likely won't meet mid-pri SLAs

Bug backlogs

Team	10/20/2013	1/5/2014	Growth	% in SLA
Ads	80	89	+11%	?
API	170	218	+28%	?
Devsite Content	33	51	+55%	?
DevX	38	56	+47%	?
Mobile Platform	50	49	-2%	?
Platform UI	106	152	+43%	?

Speaker Notes for Slide 18

<https://tableau.thefacebook.com/views/EngSLA/SLABacklog>

<https://tableau.thefacebook.com/views/EngSLA/Dash>

Core APIs

User Fields	APIs	Other
id	GET /permissions	iOS / Android / JS SDKs
name	POST /feed	PHP SDK
first_name	POST /photos	Login
last_name	POST /videos	Payments
picture		Like Button
gender		Ads
locale		
age_range		
link		
timezone		
currency		
birthday		
email		
location (current city)		

Login Review

1. Heavyweight review for high-value data [ex: photos]
2. Lightweight review for fields on user object [ex: current city]

3. Volume

1. 110,000 apps need to go through Login review w/in 6mo
2. 1,800 new apps / day need review
3. Currently review ~200 apps / day for OG / AC

4. TAT

Managed Partners	< 24h
Good apps	< 48h
Others	3-5 days

5. Tiering by classifier key to handling volume

App usage of friends

App	Use Case	Issue w/ friends usage
TripAdvisor	Show friend or fof reviews / star ratings	Uses FOF
Nike+	Compete with friends / friends leaderboard	
Spotify / Rdio / Stitcher	See friend listening activity	
Sosh	See friend places / activities / bookmarks	Friend permissions
Foursquare / Songkick	Tag friends when publishing an OG story	
Foursquare / Shazam / Deezer / Spotify / Mixcloud / EyeEm	Find friends and subscribe to their activity feed	
Wrapp	Buy a friend a gift on their birthday	Uses non-app friends
Tinder / HotOrNot	Show mutual friends given a profile	
Vamos	Event suggestions based on events popular among your friends	Friend permissions
Kickstarter	Find + fund projects that friends are funding	
Bandisintown	Concert recommendations based on friends. Find friends you can go to shows with.	
Strava	Compete with friends. Show support for friends' workouts.	
Waze	See friends driving. Compete against friends for status.	
Instagram	Find + follow friends	
Delectable	Tagging friends	
Venmo	Send / receive money with friends	Uses non-app friends
In general	Tagging non-app friends	

User of high-value perms

Permissions to privatize [won't be publicly available]	Total # of apps requesting these permissions / day [%games, %partners, %PMDs]	# of apps requesting perm from > 1,000 users / day [%games, %partners, %PMDs]	Unique user+app pair perm impressions / day
	41,191 (54%, 1%, 0.4%)	384 (33%, 15%, 4%)	4,974,471
friends_*	13,350 (14%, 3%, 0.9%)	482 (29%, 20%, 4%)	7,013,87
read_mailbox	1987 (25%, 3%, 0.6%)	246 (52%, 7%, 4%)	1,495,538
read_requests	951 (11%, 5%, 1.2%)	87 (72%, 12%, 2.3%)	497407
read_friendlists	6304 (12%, 2%, 0.5%)	179 (50%, 11%, 1%)	1,344,731
manage_notificatio ns	983 (16%, 6%, 1%)	93 (73%, 8.6%, 1%)	442,259
manage_friendlists	661 (15.5%, 4.5%, 0.6%)	66 (90%, 4.5%, 0%)	249,425
create_event	1,411 (80%, 3.40%, 1.40%)	70 (860%, 4.20%, 0%)	262,122

Data returned for NAF

Redacted – Source Code

APIs returning friends

	Use Case	ID Space	API	Can be cached	# of apps	Notes
1	Login [no friends]	A	N/A	Y	10 ⁶	In the limit, we want all apps using Login
2	Login w/ friends [app friends + NAF]	A	/me/friends	Y	App Friends: 10 ⁵ NAF: 10 ²	Currently used for cross app promotion
3	Tagging	B	/me/taggable_friends	N	10 ⁴	Requires approval
4	Social Context	B	/me/social_connections	N	10 ⁴	
5	Invites	A	/me/inviteable_friends	Y	10 ³	Response sorted by likelihood of conversion Available only to Canvas games that use Credits

Social Context API

- GET `/ {id} /social_connections`
- Callable if TOS'ed user grants `user_friends`
- Returns detailed data for AF. Summary data for NAF.
- May end up with legal / policy requirement to disclose access to this data :\

Example: Social Context API

```
GET https://graph.facebook.com/ironman/social_connections?  
fields=name,action_type,friends.summary(1)&  
access_token={user-token}
```

Redacted – Source Code

`/ {user-id}` supports
mutual friends today

Issues

1. Complexity of migrating to opaque IDs
2. Perception of omitting NAF from /me/friends
3. Game adoption of Login v4
4. Semantics of x'ing out user_friends
5. Does not address Page API data leaks [major issue]
6. Login v4 support for web / Canvas
7. Documentation

New Constraints

1. [policies] Login
 1. No infinite loops
 2. Additional password only if required, must be explained
 3. Cannot re-prompt for data available through FB using empty form
2. [policies] user_friends
 1. If apps can access NAF: for users that x-out friend permissions, apps cannot use data from FB to bootstrap social connections between users
3. [pTOS] Once apps can read X% of the graph, they must talk to us. Akin to the MAU cap.
4. Contract required to read NAF

Login v4

1. Model for x'ing out user_friends

1. Two choices

[assume A TOS's w/ friends. B TOS's w/o friends.]

1. If apps get app friends: B is never in A's friend list
2. If apps get all friends: B is in A's friend list. Policy prevents apps from auto-wiring $A \Leftrightarrow B$ [ex: auto-follow, TOS notifis]

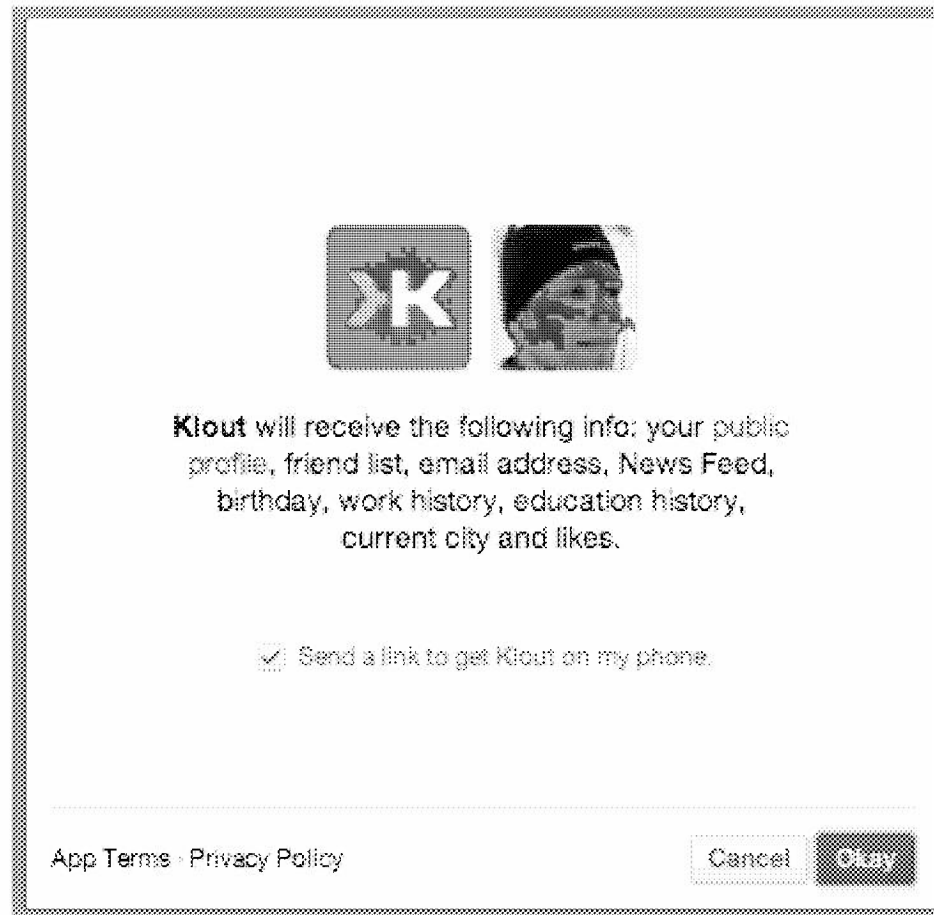
2. Users who revoke can be found in app's search + followed

3. Users who revoke cannot send Requests via Custom MFS

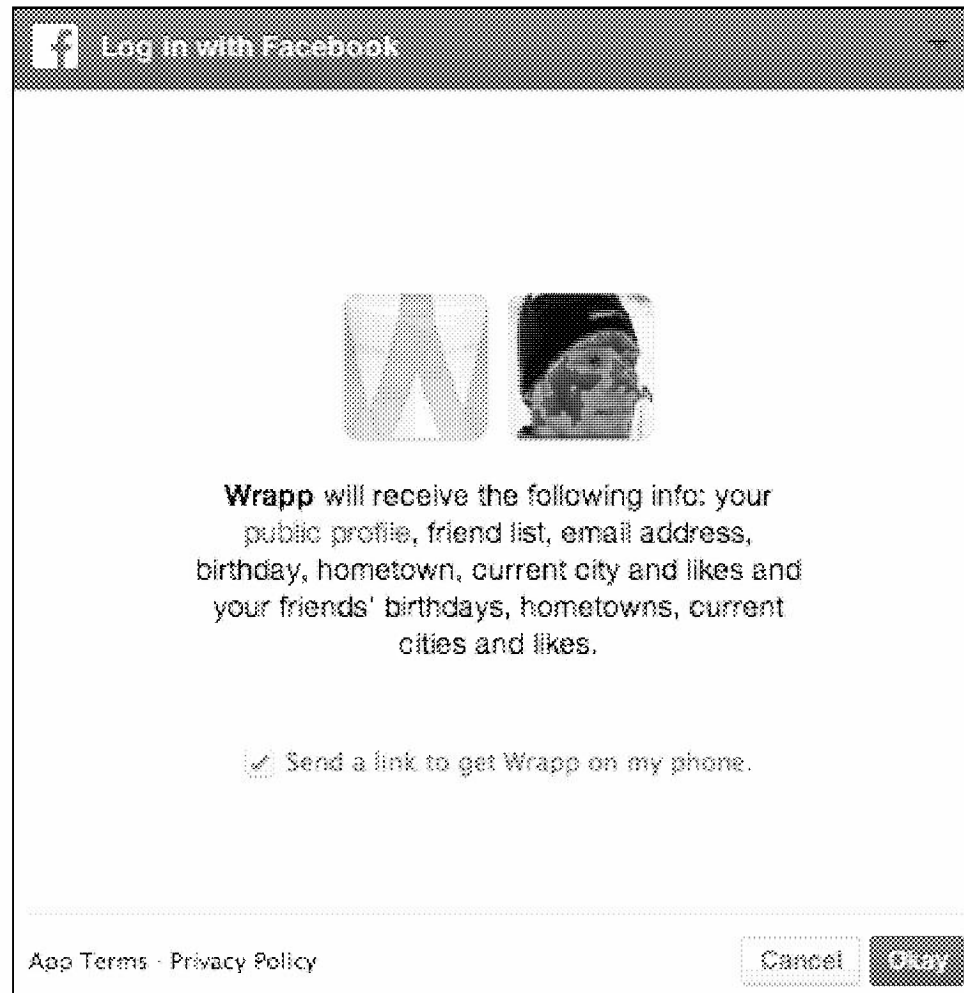
4. For A: B.installed = true

5. iOS Login Dialog disabled for v4+

Klout Login Dialog



Wrapp Login Dialog



Launch Timing

