

1 GIBSON, DUNN & CRUTCHER LLP  
 JOSHUA A. JESSEN, SBN 222831  
 2 JJessen@gibsondunn.com  
 JEANA BISNAR MAUTE, SBN 290573  
 3 JBisnarMaute@gibsondunn.com  
 PRIYANKA RAJAGOPALAN, SBN 278504  
 4 PRajagopalan@gibsondunn.com  
 ASHLEY M. ROGERS, SBN 286252  
 5 ARogers@gibsondunn.com  
 1881 Page Mill Road  
 6 Palo Alto, California 94304  
 Telephone: (650) 849-5300  
 7 Facsimile: (650) 849-5333

8 GIBSON, DUNN & CRUTCHER LLP  
 CHRISTOPHER CHORBA, SBN 216692  
 9 CChorba@gibsondunn.com  
 333 South Grand Avenue  
 10 Los Angeles, California 90071  
 Telephone: (213) 229-7000  
 11 Facsimile: (213) 229-7520

12 Attorneys for Defendant  
 FACEBOOK, INC.  
 13

14 UNITED STATES DISTRICT COURT  
 15 NORTHERN DISTRICT OF CALIFORNIA  
 16 OAKLAND DIVISION

17 MATTHEW CAMPBELL and MICHAEL  
 HURLEY,

18 Plaintiffs,

19 v.

20 FACEBOOK, INC.,

21 Defendant.  
 22

Case No. C 13-05996 PJH (MEJ)

**PUTATIVE CLASS ACTION**

**DECLARATION OF ALEX HIMEL IN  
 SUPPORT OF DEFENDANT FACEBOOK,  
 INC.’S OPPOSITION TO PLAINTIFFS’  
 MOTION FOR CLASS CERTIFICATION**

**Table of Contents**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

I. Identifying Proposed Class Members ..... 1

II. Share Objects ..... 3

    A. Overview ..... 3

    B. Share Objects and Global Share Objects and Messages ..... 4

    C. Variability in Connection with Share Objects ..... 6

III. Global Share Object Counters..... 8

    A. Overview ..... 8

    B. Variability in Connection with Global Share Object Counters..... 9

IV. Plugin Count ..... 10

    A. Overview ..... 10

    B. Plugin Count and Messages ..... 11

    C. Variability in Connection with Plugin Count ..... 12

V. ██████████ ..... 13

    A. Overview ..... 13

    B. ██████████ and Messages..... 14

    C. Variability in Connection with ██████████ ..... 14

VI. Nectar ..... 15

    A. Overview ..... 15

    B. Nectar and Share Scrapes for Messages ..... 16

    C. Variability in Connection with Nectar ..... 16

VII. Insights and Related APIs ..... 17

    A. Overview ..... 17

    B. Insights and Related APIs and Messages..... 18

    C. Variability in Connection with Insights and Related APIs ..... 18

VIII. “Link\_Stats” and Graph API..... 19

    A. Overview ..... 19

    B. “Link\_Stats” and Graph API and Messages ..... 20

    C. Variability in Connection with “Link\_Stats” & Graph API ..... 21

1 I, Alex Himel, declare as follows:

2 1. I have been employed as a software engineer at Facebook since April 2009, and my  
3 current title is Engineering Director. I am over the age of 18. From 2009-2014, I worked on  
4 Facebook's Developer Platform, and my work encompassed Facebook's Social Plugins and Insights  
5 features. I have personal knowledge of the matters stated herein and, if called as a witness, could and  
6 would testify competently thereto.

7 2. I provide this Declaration in support of Facebook's Opposition to Plaintiffs' Motion  
8 for Class Certification, and to explain certain facts regarding Facebook's software code as it relates to  
9 Facebook's share objects, Plugin Counters, [REDACTED] Hive table, "Nectar" logging of share scrape  
10 events, Insights (including the user interface ("UI"), application program interface ("API"), and  
11 dashboard, also referred to below as "Insights and Related APIs"), and other public APIs including  
12 "link\_stats" and Graph API. This Declaration also describes certain Facebook services related to  
13 these functions, particularly as they relate to uniform resource locators ("URLs") in messages sent  
14 and received through the Facebook platform.

15 3. I also understand that, on November 13, 2015, Plaintiffs filed a Motion seeking to  
16 certify the following proposed class:

17 All natural-person Facebook users located within the United States who have sent, or  
18 received from a Facebook user, private messages that included URLs in their content  
19 (and from which Facebook generated a URL attachment), from within two years before  
the filing of this action up through the date of the certification of the class.

20 I understand that Plaintiffs filed their action on December 30, 2013, and that therefore the relevant  
21 period for Plaintiffs' new purported class is December 30, 2011 to the present (the "Class Period").

22 **I. Identifying Proposed Class Members**

23 4. To my knowledge, neither Facebook nor any other entity possesses the data that would  
24 be required to identify all persons meeting Plaintiffs' class definition. Facebook does not possess  
25 records showing all users that sent or received a message that included a URL in its content and from  
26 which Facebook generated a URL attachment. For example, people who included a URL in their  
27 message, and for which a URL attachment was generated, but who deleted the attachment before

1 send, fall within Plaintiffs’ class as written. However, users’ messages on their face do not reflect  
2 that information, and therefore even a manual, message-by-message review could not identify these  
3 putative class members.

4 5. Additionally, as discussed below, determining whether any given person was  
5 subjected to the challenged practices would require a message-by-message inquiry. To my  
6 knowledge, neither Facebook nor any other entity possesses the data that would be required to  
7 determine whether any given person meeting this criteria was subjected to all the challenged  
8 practices.

9 6. In her report, Dr. Golbeck says that “to retrieve a list of class members, the Code  
10 process should be relatively straightforward,” and that “a database query could be used to select the  
11 Facebook user IDs of everyone whose actions had created an EntShare from a private message.”  
12 (Golbeck Report ¶ 103.) In the next two paragraphs of her report, she provides “sample” code that  
13 she contends would return a list of “Facebook user IDs of everyone whose actions had created an  
14 EntShare from a private message,” and, in her deposition, she said that such a list would identify the  
15 class members. (Golbeck Deposition Transcript at 331:2-8.)

16 7. That is incorrect. This query would return a list of users that is both under- and over-  
17 inclusive of the proposed class. For example, a share object does not include any information about  
18 the recipients of the message sharing that URL. Therefore a query for IDs in share objects will not  
19 include recipient class members. Also, Dr. Golbeck uses the “global” field in the share objects to  
20 eliminate any global objects from showing in the results of her query, but the “global” field did not  
21 exist in the share objects created early in the proposed class period, and would not have been  
22 correctly set for those older objects. Thus, this query might incorrectly return global share objects  
23 that were NOT generated by a URL shared in a message, or incorrectly exclude older share objects  
24 without correct information in the “global” field.

25 8. In addition, Facebook’s systems do not index the fields set out in Dr. Golbeck’s query;  
26 in other words, there is no existing table of the key fields she is searching for that can be searched  
27 quickly and efficiently. Instead, in order to accommodate her query, Facebook would have to load  
28

1 every share object in its databases into memory sequentially and examine each object's values to find  
2 those containing a particular individual's Facebook ID, in order to know whether that individual's  
3 message URL share had resulted in the creation of a share object. Facebook share objects comprise  
4 two trillion database rows, distributed across 50,000 database machines. Searching those objects  
5 would require writing queries to access every database, and to individually assess every share object  
6 (as described above) in an object-by-object analysis.

7 9. Further, the results of this query will include users who never typed a URL into a  
8 message, and instead merely chose to "Share" a URL through a "Share" button on a third-party  
9 website in a message. (Golbeck Deposition Transcript at 339:18-340:13.) The URL attachments to  
10 these messages were not generated from text typed into a message.

## 11 **II. Share Objects**

### 12 **A. Overview**

13 10. All information that users share through the Facebook platform, including messages  
14 and all other information, is received by Facebook and stored on Facebook servers. Facebook must  
15 receive and host all information shared on the site in order to provide its social-networking service.  
16 Facebook also anonymizes and aggregates certain data in order to help facilitate users' discovery of  
17 potentially relevant and interesting information on the web at large. For example, Facebook offers a  
18 "Like" button social plugin, which has been integrated into websites all over the world; if a user  
19 clicks on the "Like" button, Facebook displays a "story" of that action on the users' Timeline, and  
20 Facebook keeps a count of the number of times that webpage has been "Liked" and provides some of  
21 that data publicly in the aggregate.

22 11. Another way that users interact with webpages is by "sharing" the URL to that  
23 webpage, for example by copying and pasting the URL into a post or a message. Under certain  
24 circumstances "sharing" a URL in that way will involve some specific processing, which, if  
25 successful, may result in storing that event on a Facebook server as a "share." As explained in  
26 Facebook's Second Supplemental Responses and Objections to Plaintiffs' Narrowed Interrogatory  
27 No. 8, attached as Exhibit MM, Facebook stores "shares" (whether in a post or in a message) as  
28

1 “objects.” Generally, in computer science, “object” refers to data and software code grouped together  
2 to make the process of writing and running source code efficient and effective. The concept of an  
3 “object” is a basic element of what is widely referred to as “object-oriented code.” When certain  
4 types of data are configured into a limited number of classes in this way, the code that actually  
5 processes that data can be written more efficiently, which can improve speed and reduce errors.  
6 Facebook’s “objects” group together data in order to make the operation of its software more  
7 efficient.

8 12. Facebook’s source code creates, stores, and accesses “objects” for innumerable  
9 processes and purposes. For example, when a user copies and pastes a URL into a post, Facebook  
10 stores the information about that users’ action of sharing that URL in a post in a “share object.”  
11 There is nothing unusual or nefarious about the use of “objects”—which are merely a name for a  
12 certain way of storing data—in software programming.

13 **B. Share Objects and Global Share Objects and Messages**

14 13. As explained in Facebook’s Supplemental Responses and Objections to Plaintiffs’  
15 First Set of Interrogatories (attached as Exhibit NN), during the relevant period in this case, if a user  
16 typed a URL into the text field in the Facebook Messages product, and the user had JavaScript  
17 enabled in her browser, the JavaScript code running in the user’s browser may have detected the  
18 existence of a URL and requested information from a Facebook server in order to provide a preview  
19 of the typed URL (“URL preview”)—including a brief description of the URL and, if available, a  
20 relevant image from the website. At the time the request was sent to Facebook for a preview,  
21 Facebook assessed whether the URL was in its library of known malicious URLs. If it was,  
22 Facebook would not return a preview; if it was not, Facebook may have returned information already  
23 on a Facebook server to generate a preview.

24 14. Or, if information to generate the URL preview was not available already on a  
25 Facebook server, a Facebook server may have sent a request to the website, generated an image and  
26 description if available, and delivered those components to the user’s browser to generate a URL  
27 preview. Facebook sometimes refers to this function as its “share scraper.” When Facebook  
28

1 “scrapes” a third-party website for the first time, it creates a record for that URL in its system, called  
2 a “global share object.” The global share object stores the components necessary to generate URL  
3 previews across the Facebook website.<sup>1</sup> By storing these elements on a Facebook server, Facebook  
4 can avoid scraping the website every time it needs to generate a preview, thereby reducing traffic and  
5 allowing for faster and more efficient and effective preview generation. It also allows Facebook to  
6 generate previews even when third-party websites have errors or down time. The purpose of a global  
7 share object is not to track Facebook’s users’ interactions with URLs across the Web; rather, the  
8 purpose is to efficiently and effectively store data about the URL, to make it easier to write and run  
9 code that uses or presents information about that URL for myriad purposes ranging from site integrity  
10 and security processes, to URL preview generation, and for other features and processes that present  
11 information about that URL in compelling ways and at compelling times to the right users.

12 15. Where available, URL previews helped users verify the URL they were sharing before  
13 sending their messages. Whether or not a URL preview was generated depended on myriad factors,  
14 such as the configuration of the user’s browser, the type of URL entered, the number of URLs  
15 entered, and the speed of send, among other individualized factors. A URL preview is an attachment  
16 to the draft message. Once the URL attachment was created, changes to the characters in the text  
17 field of the draft message did not impact the URL attachment. Therefore, if someone using Facebook  
18 deleted the URL in the text field of a message after a URL preview had been generated, the URL  
19 preview would not have been deleted (unless the sender affirmatively deleted it).

20 16. If a URL attachment was successfully created (and not deleted by the user) before the  
21 message was sent, then, after the message was sent and the message and components were received  
22 and stored on a Facebook server, and if the message was not blocked in the course of abuse- and  
23 security-related processing, a “share object”<sup>2</sup> may have been created reflecting the fact that the

---

24  
25 <sup>1</sup> In Facebook’s storage system, “global share objects” are typically accessed in the form of  
26 “EntGlobalShares,” which represent the set of fields and data about that URL and about  
27 engagement with that URL.

28 <sup>2</sup> In Facebook’s storage system, “share objects” are typically accessed in the form of EntShares,  
which represent the set of fields and data about that share event.

1 message had a URL as an attachment. These URL share objects are stored in a system called TAO.  
2 If a URL preview was not created before the message was sent or was deleted by the user before  
3 sending, no share object was created. Similarly, if a malicious message or URL was successfully  
4 “blocked,” no share object was created. Share objects may also have been deleted if the user deleted  
5 his message or account, or if data was lost.

6 17. Another way to share a URL in a Facebook message was to click on the “Share”  
7 button on a third-party website, and choose (from the options presented to the user) to share the URL  
8 for that page in a Facebook message. In these circumstances, Facebook did not detect the URL from  
9 any text written by the user; instead a preview was automatically generated (where possible),  
10 generally with the URL for the page on which the “Share” button was displayed.

11 18. When a URL attachment is successfully sent with a message, the share object is the  
12 storage of the URL attachment to the message. When the sender or recipient of a message with a  
13 URL attachment chooses to view the message, Facebook displays the message and the attachment.  
14 The attachment is displayed from the information stored in the share object. If a share object is not  
15 created for a URL attachment, or the share object is deleted, no URL attachment will reach the inbox  
16 for the sender or the recipient. If a share object for a URL attachment is deleted, the URL attachment  
17 can no longer be displayed to the sender or recipient. In Facebook’s source code, the share object is  
18 necessary in order to deliver the URL attachment to the mailboxes for the sender and recipient, and it  
19 is necessary in order to display the URL attachment to the sender and recipient after it is sent.

20 19. The same processes for URL preview generation, and the same variability in those  
21 processes described below, are applicable to URLs shared in Facebook posts as well as Facebook  
22 messages.

### 23 **C. Variability in Connection with Share Objects**

24 20. A share object is not created every time a person includes a URL in a Facebook  
25 message. Rather, a share object will not be created, even if there is a URL in a message, unless there  
26 is a URL preview generated and successfully sent with the message. There are many circumstances  
27 in which no URL preview will be generated, including if the user does not have JavaScript enabled,  
28



1 there was more than one URL in the draft message (in which case a preview would only be generated  
2 for the first URL), the preview was blocked by Facebook’s anti-abuse or security processes, or the  
3 message was sent too quickly to generate a preview, among other reasons. There are also subtle  
4 differences in how the user enters the URL that may impact whether a URL preview is generated.  
5 For example, if a user copies and pastes a URL into a draft message, the process for generating a  
6 preview will begin, whereas if a user manually types a URL into a draft message, but does not add a  
7 space at the end of the URL before sending the message, no preview will be generated. Also,  
8 messages sent from Facebook’s stand-alone messenger mobile application do not generate URL  
9 attachments.

10 21. Additionally, even if a URL preview attachment was generated, a share object may not  
11 have been created. For example, if the user deleted the attachment before sending, no share object  
12 was created. Additionally, if the message or attachment was blocked in the course of abuse- and  
13 security-related processing, no share object was created. Share objects may also have been deleted if  
14 the user deleted his message or account or data was lost.

15 22. Taking all of this variation together, at a minimum, determining whether a user’s  
16 inclusion of a URL in a Facebook message resulted in creation of a share object would involve the  
17 following individualized inquiries for each message:

- 18 a. Was the user using a browser that is JavaScript capable?
- 19 b. Did the user have JavaScript enabled in her browser?
- 20 c. Did the user paste in a URL or type a message including a URL and press  
21 “space”?
- 22 d. Did the user send the message too quickly to generate a preview?
- 23 e. Did the user include more than one URL in the message?
- 24 f. Did the user delete the URL text in the body of the message before sending?
- 25 g. Was a URL preview successfully created?
- 26 h. Was a URL preview created, but deleted prior to the message being sent?
- 27 i. Was the URL or the message blocked by Facebook’s security-related or anti-  
28

1 abuse processing?

2 j. Was the message sent from the messenger mobile application?

3 **III. Global Share Object Counters**

4 **A. Overview**

5 23. Global share objects contain a field called “tracking information,” which contains five  
6 aggregate, anonymous counters named: share\_count, post\_count, like\_count, comment\_count, and  
7 click\_count. These internal counters are intended to aggregate the number of certain categories of  
8 actions taken relating to that URL object.

9 24. Share\_count is the internal counter that attempts to keep track of the total number of  
10 times a URL attachment was shared in a Facebook message or when a person chooses to send a URL  
11 in a message through the “Share” button dialog on a third-party website. Post\_count is the internal  
12 counter that attempts to keep track of the total number of times a URL is posted in a person’s feed.  
13 Like\_count is the internal counter that attempts to keep track of the total number of times people have  
14 clicked a “Like” button for that URL or “Liked” a story on a user’s Newsfeed with an attachment for  
15 that URL. Comment\_count is the internal counter that attempts to keep track of the total number of  
16 times people have commented on that URL. Click\_count is the internal counter that attempted to  
17 keep track of the total number of times a URL was clicked on after being shared.

18 25. The share\_count is an entirely anonymous and aggregate counter. It does not identify  
19 the people who shared the URL in a message, nor is there any association back to those people, the  
20 messages containing the URL attachment, or any information about those user-specific share events.  
21 A particular share object will have a field that includes the identifier for the relevant global share  
22 object, but the reason for this is to provide a way to obtain information about the URL. For example,  
23 to ensure that URL attachments in previously sent messages appear updated when a sender or  
24 recipient looks at them later, the field with the identifier to the global share object provides an easy  
25 way to get updated information about that URL (since the global share objects are updated regularly).  
26 The relationship is not reciprocal, however, and it is generally not possible to determine from  
27 share\_count which people shared URL attachments in Facebook messages. While the global share  
28

1 object has some fields for information about “associations” to specific persons who have engaged  
2 with the URL in a public way (for example in a public post), it does not include associations to  
3 people who only shared the URL in a message.

4 26. There is also no data from which to identify persons whose sharing of URLs in  
5 messages was included in the “share\_count” in the global share object.

6 **B. Variability in Connection with Global Share Object Counters**

7 27. For people using Facebook who sent a message with a URL attachment, only those  
8 people for whom the URL attachment actually resulted in the successful creation of a share object  
9 even potentially could have had their URL share included in the share\_count in the global share  
10 object. Specifically, the share\_count counter could have been incremented only based on the  
11 generation of a share object, which, in a variety of instances, might never have been generated. For  
12 instance, as discussed above, if the URL attachment were deleted by the user before sending the  
13 message, no share object would have been created, and the counter in the global share object would  
14 not have been incremented. Similarly, if the message or its attachments were blocked in the course of  
15 abuse- and security-related processing, no share object would have been created, and the share\_count  
16 would not have been incremented.

17 28. Additionally, even when a share object was created from a URL attachment to a  
18 Facebook message, that action did not always result in an increment in the share\_count in the global  
19 share object. For example, global share objects were not generated for Facebook webpages, and thus  
20 URL shares for Facebook websites did not increase the share\_count in any global share object. Also,  
21 race conditions, database failures, and database contention each may have resulted in failure to  
22 increment the counters in share\_count even where a share object was created.<sup>3</sup> “Race” conditions  
23 occur when multiple people share the same URL at the same time and Facebook only processes one  
24 increment to the count; this happens more frequently when many people try to share at once.

25 \_\_\_\_\_  
26 <sup>3</sup> Facebook’s system comprises tens of millions of lines of code, hundreds of thousands of servers  
27 supporting over 1.5 billion people, and handling over 1 trillion requests each day. As with any  
28 system of this size, it is expected that at least some machines will always be offline or not  
functioning properly resulting in some error.

1 Database failure or contention can occur for many reasons, including the interplay between different  
2 databases stored in different locations contributing to a single count.

3 29. Taking all of this variation together, at a minimum, determining whether a person’s  
4 inclusion of a URL in a message resulted in an increment of the anonymous, aggregate share\_count  
5 in the global share object for that URL would require the following individualized inquiries for each  
6 message:

- 7 a. When was the message sent?
- 8 b. Was a share object successfully created for a URL attachment in that message?  
9 (i.e., all the individualized inquiries in paragraph 22 above.)
- 10 c. Was the URL to a third-party webpage (as opposed to a Facebook webpage)?
- 11 d. Were race conditions present?
- 12 e. Was there database failure?
- 13 f. Was there database contention?

14 30. To my knowledge, neither Facebook nor any other entity possesses the data that would  
15 be required to ascertain the answers to the inquiries in paragraph 29, either on an individual or bulk  
16 basis, for putative class members.

#### 17 **IV. Plugin Count**

##### 18 **A. Overview**

19 31. During the proposed Class Period, Facebook offered websites “social plugins,” or  
20 units of embeddable code that allow people to share information using Facebook directly from third-  
21 party websites. For example, a third-party website may embed code for the Facebook “Like” button  
22 plugin on its website, enabling people using Facebook to directly “Like” the website and to share that  
23 action with their Facebook connections (without having to return to <https://www.facebook.com> or the  
24 Facebook mobile app to share the content).

25 32. The “Like” button plugin also may display an anonymous and aggregate count of all  
26 “Likes” for that particular website. At different times, this aggregate count next to the plugin  
27 (“Plugin Count”) may have included URLs (a) shared (in the NewsFeed), (b) commented on,

1 (c) liked, and (d) sent as an attachment to a message (and recorded as a share object). Or, depending  
2 on how it was configured by the site owner, it may have displayed the number of “fans” for that page.

3 33. Instructions for how to embed the “Like” button and Plugin Count into a website, and  
4 an explanation of the components of the Plugin Count, were disclosed publicly in Facebook’s  
5 developer guidance—one of the primary locations where Facebook explains the functionality of its  
6 service to the public. For a period beginning at least as early as March 7, 2011, the developer  
7 guidance included a section entitled “What makes up the number shown on my Like button?” and  
8 explains that the number is “the sum of:

- 9 • The number of likes of this URL
- 10 • The number of shares of this URL (this includes copy/pasting a link back to Facebook
- 11 • The number of likes and comments on stories on Facebook about this URL [and]
- 12 • The number of inbox messages containing this URL as an attachment.”

13 **B. Plugin Count and Messages**

14 34. From the beginning of the Class Period until December 19, 2012, if the Facebook  
15 Plugin Count next to a “Like” button for a third-party website was not configured to reflect its “fan  
16 count,” it displayed the combination of the following fields from the counters in the global share  
17 object record: share\_count, like\_count, comment\_count, post\_count.

18 35. During that time period, the share\_count reflected in the global share object record  
19 attempted to reflect all instances of sharing a URL using the external Facebook “Share” button dialog  
20 (and choosing to share in a Facebook message) and URL attachments for third-party webpages  
21 shared in messages sent from the Facebook website that resulted in the creation of a share object.  
22 Therefore, under certain circumstances, if a URL was shared in a message, a Plugin Count on a third-  
23 party website may have been incremented.

24 36. I understand that Plaintiffs’ expert, Dr. Jennifer Golbeck, has suggested that by  
25 removing URL shares in messages from the Plugin Count (after that practice was the subject of  
26 considerable national press attention) that Facebook simply “hid” that activity from the public.  
27 (Golbeck Report ¶ 85.) This is false. Facebook does detect (and has detected) URLs in draft

1 messages (as it does all over the Facebook website) for a number of purposes, including for the  
2 sharescape process (which facilitates Facebook’s security and anti-abuse related processes and then  
3 generates a URL preview for the sender) and in order to create a share object (which is necessary to  
4 deliver the URL attachment to the inboxes of the sender and recipient). There is nothing unusual or  
5 nefarious about Facebook taking note of its users’ experiences and preferences and the reaction of the  
6 press. Facebook is sensitive to users’ feedback and regularly incorporates that feedback into its  
7 design and engineering decisions.

8 **C. Variability in Connection with Plugin Count**

9 37. On December 19, 2012, I made a change to Facebook’s source code such that, if the  
10 Facebook Plugin Count for a third-party website was not configured to reflect its “fan count,” it  
11 displayed the combination of the following fields from the counters in the global share object record:  
12 like\_count, comment\_count, post\_count. As of the date of that change to the source code, Plugin  
13 Counts displayed on third-party websites did not include the share\_count from the global share object  
14 and therefore did not reflect URLs shared in messages.

15 38. As noted above, if a person using Facebook shared a URL through a message but no  
16 share object was created, the sharing of that URL did not increment the share\_count counter in the  
17 global share object, nor any Plugin Count on the destination website. Also, if the share\_count in the  
18 global share object did not increment, for one of a number of reasons discussed above, the sharing of  
19 that URL did not increment the Plugin Count on the website (even if a share object was created).

20 39. Similarly, if the destination website associated with the URL did not display a  
21 Facebook Plugin Count, the sharing of that URL did not increment the Plugin Count on the website  
22 (even if a share object was created). And, if the Plugin Count was instead configured to display that  
23 sites’ “fan” count, the sharing of that URL did not increment the Plugin Count on the website (even if  
24 a share object was created).

25 40. Additionally, in some cases, even if a share object was created and the destination  
26 website associated with the URL had a Facebook Plugin Count, and the Plugin Count did not point to  
27 its “fan” count, the Plugin Count on the destination website may not have been incremented (for  
28

1 example, if the URL a user included in a message was not exactly the same as the URL the developer  
2 passed to the plugin).

3 41. But even if sharing a URL in a message did result in an increase in a Plugin Count on  
4 a third-party website, there is no way to identify which user or what particular action resulted in that  
5 increment. The Plugin Count is aggregate and anonymous and cannot be traced to specific users or  
6 actions.

7 42. At a minimum, determining whether a putative class member's inclusion of a URL in  
8 a message ever actually incremented a Plugin Count displayed on a third-party website would require  
9 the following individualized inquiries for each message:

- 10 a. When was the message sent?  
11 b. Was a share object successfully created for a URL attachment in that message?  
12 (i.e., all the individualized inquiries in paragraph 22 above.)  
13 c. Was the share\_count incremented in the global share object? (i.e., all the  
14 individualized inquiries in paragraph 29 above.)  
15 d. Did the third-party website display a Plugin Count?  
16 e. Did the third-party website configure its Plugin Count to display its "fan" count?  
17 f. Was the URL included in the message exactly the same as the URL the  
18 developer passed to the plugin?

19 43. To my knowledge, neither Facebook nor any other entity possesses the data that would  
20 be required to ascertain the answers to the inquiries in paragraph 42, either on an individual or bulk  
21 basis, for putative class members.

22 **V.** [REDACTED]

23 **A. Overview**

24 44. For a period of time, prior to the Class Period, Facebook logged data regarding share  
25 objects in a table, [REDACTED]<sup>4</sup> in its Hive database, a data warehouse. The table was used for,  
26

27 <sup>4</sup> This is not the same as the [REDACTED] FQL table discussed in my June 1, 2015 declaration.  
28

1 among other things, Facebook's Recommendations social plugin, which I understand is addressed in  
2 the Declaration of Dan Fechete being submitted in support of Facebook's Opposition to Certification.  
3 Data logged in the "[REDACTED]" Hive table was deleted in 2011 prior to the Class Period. I  
4 understand that Plaintiffs have claimed that a particular printout of Facebook source code, a file  
5 called "[REDACTED]" (FB000014183), demonstrated that the fields in "[REDACTED]" contained a  
6 record of the user ID, the action the user took, the time of action, a pointer to data structures, and an  
7 indicator of whether it was publicly posted or not. However, as noted above, this table was deleted  
8 prior to the beginning of the Class Period. No URL shares by any purported class member were  
9 stored in this table during the Class Period.

10 **B. [REDACTED] and Messages**

11 45. If a person sent a Facebook message, and a URL attachment was successfully created  
12 (and not deleted by the user) prior to the message being sent, then, after the message was sent and the  
13 message and components were received and stored on a Facebook server, and if it was not blocked by  
14 Facebook's security and anti-abuse related processes, a share object may have been created reflecting  
15 the fact that the message had a URL as an attachment. If a share object was created from a URL  
16 attachment sent with a Facebook message, during certain periods prior to the Class Period, it may  
17 have been logged to the "[REDACTED]" Hive table. However, if no share object was created, the URL  
18 share from that message would not have been logged or stored in the "[REDACTED]" table in any way.

19 **C. Variability in Connection with [REDACTED]**

20 46. For people who sent a message with a URL attachment in 2011 prior to the Class  
21 Period, only those people for whom the URL attachment actually resulted in the successful creation  
22 of a share object even potentially could have had their URL share included in the "[REDACTED]" Hive  
23 table. For the same reasons discussed above in paragraph 27, there are a number of reasons why a  
24 share object may not have been created, and thus there would be no log to the "[REDACTED]" table.

25 47. Even when a share object was created from a URL attachment to a Facebook message  
26 during that time, that action did not always result in logging to the "[REDACTED]" Hive table, for the  
27 same reasons discussed above in paragraph 28 with respect to the global share tracking information.



1           48.     Additionally, if a message was sent with a URL attachment during the Class Period,  
2 which began after the [REDACTED] table was deleted, no shares were logged in [REDACTED]

3           49.     Taking all of this variation together, at a minimum, determining whether a Facebook  
4 user's inclusion of a URL in a message resulted in logging to the [REDACTED] Hive table would  
5 require the following individualized inquiries for each message:

- 6           a.     When was the message sent?  
7           b.     Was a share object successfully created for a URL attachment in that message?  
8                 (i.e., all the individualized inquiries in paragraph 22 above.)  
9           c.     Was the share successfully logged to the [REDACTED] Hive table? (i.e. the  
10                 individualized inquiries in paragraph 29 (d)-(f)).

11          50.     To my knowledge, neither Facebook nor any other entity possesses the data that would  
12 be required to ascertain the answers to the inquiries in paragraph 49, either on an individual or bulk  
13 basis, for putative class members.

14     **VI.   Nectar**

15           **A.    Overview**

16          51.     Nectar is a library, or a collection of scripts, that move data from one system within  
17 Facebook to another, including sending statistics to certain databases. It is one of a number of tools  
18 used to log certain activity or data generated in the operation of the Facebook website for purposes  
19 including to ensure site integrity and security, detect and prevent abuse, research and test new and old  
20 products, features, and functionality, detect and fix bugs, balance traffic and load processing, and  
21 learn about aggregate user preferences and patterns to inform site improvements and feature  
22 development. Logging can be highly specialized for its desired purpose; events logged, data fields  
23 logged for those events, rate of logging, duration of logging, and retention time can all be tailored for  
24 specific uses.

25          52.     When the unit of code that performs a share scrape (discussed above as part of the  
26 generation of a URL preview) is executed, that event may be logged using Nectar. Not all instances  
27 of execution of the share scraper code are logged to Nectar. Only a sampling of such instances is  
28

1 logged. From June 13, 2011 until June 20, 2012, only 1% of share scrapes were logged to Nectar.  
2 From June 20, 2012 to present, only 0.01% of share scrapes were logged to Nectar. However,  
3 sometimes the sampling rate depended upon the URL being scraped. That is, different URL domains  
4 had different sampling rates, such that share scrapes of different URLs had differing likelihoods of  
5 being logged in Nectar, assuming they were scraped at all (which may not have occurred, for  
6 example if the link was malicious or if the user pressed “Send” too quickly). During the Class  
7 Period, some domains had domain-specific sampling rates for Nectar logging of share scrapes for  
8 debugging purposes.

9 **B. Nectar and Share Scrapes for Messages**

10 53. Since the same share scraper used across the Facebook site is also used in Facebook’s  
11 messaging system, when a URL preview is generated for a Facebook message, prior to a user sending  
12 the message, if the components for the preview are not already on a Facebook server, and instead the  
13 server sends a request to scrape the third-party site, that event may have been logged using Nectar, if  
14 it was among the sampled actions.

15 54. Both the execution of the share scrape function and any logging of that share scrape to  
16 Nectar occur before a person sends the draft message.

17 **C. Variability in Connection with Nectar**

18 55. Because Nectar logging only occurred for a sample of share scrape events, and its  
19 sample rate varied both with time and with the URL in question, only a fraction of people may have  
20 been subjected to Nectar logging of share scrapes in connection with their messages that had a URL  
21 attachment.

22 56. At a minimum, determining whether a Facebook user’s inclusion of a URL in a  
23 message actually resulted in Nectar logging of a share scrape would require the following  
24 individualized inquiries for each message:

- 25 a. Did the user have JavaScript enabled in her browser?  
26 b. Did the user type a message including a URL?  
27 c. Did the user send the message too quickly to generate a preview? Did site  
28

1 integrity processes indicate that the URL should not be scraped because it  
2 matched a URL on the list of known malicious URLs?

- 3 d. Was the share scraper function called?
- 4 e. Was the information for a URL preview already stored on the Facebook site?
- 5 f. Did the Facebook server send a request to scrape the third-party site?
- 6 g. Did the Facebook server successfully scrape the third-party site?
- 7 h. Was a URL preview successfully created?
- 8 i. Was that incident of scraping among those sampled for logging to Nectar?

9 57. To my knowledge, neither Facebook nor any other entity possesses the data that would  
10 be required to ascertain the answers to the inquiries in paragraph 56, either on an individual or bulk  
11 basis, for putative class members.

## 12 **VII. Insights and Related APIs**

### 13 **A. Overview**

14 58. “Insights” is the name of a Facebook user interface (UI), accessible from a Facebook  
15 website, and a related Facebook application program interface (API). Facebook Insights and Related  
16 API provide the owners of particular websites (also known as URL “domain owners”) with data  
17 about interaction with and traffic to their websites. In order to access this information, a domain  
18 owner must provide authentication demonstrating that he or she does indeed own that particular  
19 website (URL domain) or webpage (URL). After authentication, the domain owner can use the  
20 Insights dashboard or APIs to obtain statistics and demographics about the domains/URLs they own.  
21 Specifically, Insights provides information about how effectively Facebook is generating traffic to  
22 their site and demographic information about the users who make up that traffic. It also included  
23 aggregate, anonymous statistics and aggregate, anonymous demographic information about the  
24 people who share links to that domain owners’ sites across the Facebook platform.

25 59. In 2011, Facebook created a new specialized Insights architecture designed to reflect  
26 data about activity as quickly as possible after that activity occurred (“Real Time Analytics”). The  
27 data store of activity to support the new Insights feature is completely separate from the other stores  
28

1 discussed above; the Insights system writes certain events to a separate Hbase data store with its own  
2 counters for different relevant actions.

3 **B. Insights and Related APIs and Messages**

4 60. When the Insights product was announced in April 2010, the aggregate, anonymous  
5 statistics and demographic information about the users who shared a particular URL in Insights  
6 included URLs shared in messages through the “Share” button plugin on third party websites (if a  
7 share object was successfully created and the Insights counters were successfully incremented).  
8 After the sharescraper was implemented in messages in August 2010, the aggregate, anonymous  
9 statistics also included URLs shared in messages through the Facebook website (again, if a share  
10 object was successfully created and the Insights counters were successfully incremented). When the  
11 new Insights system was introduced in 2011, URLs shared in messages were included in the new  
12 counters in the new Insights architecture. However, URL shares in messages were never listed  
13 separately, and were only included in a larger, aggregate count of shares across the site.

14 61. However, on October 11, 2012, I changed the code to no longer include URL shares in  
15 messages in the aggregated, anonymous counters visible to domain owners through Insights.

16 **C. Variability in Connection with Insights and Related APIs**

17 62. As stated above, URLs sent in messages have not been included within Insights and  
18 Related API since October 2012. Because Plaintiffs’ proposed class includes individuals who sent or  
19 received Facebook messages beginning in December 2011 to the present, only some proposed class  
20 members would have been exposed to the practice of including their URL shares in messages in data  
21 available to web domain owners. Moreover, whether URLs sent in messages were included in  
22 Insights and Related APIs during the class period still depended on the variability discussed above in  
23 generating a preview and share object (see paragraphs 20-21). Moreover, the counters stored in the  
24 special Insights architecture were subject to variability. If there was a server or database error, or the  
25 time to write to the Insights data store elapsed before finalizing, a share object may not be included in  
26 the Insights counter.

27 63. Further, it would be highly variable and impossible to know whether Insights or any  
28

1 Related APIs were ever actually accessed or viewed for any given website that was ever shared as a  
2 URL in a message. If the owner of a given domain never authenticated ownership nor accessed the  
3 Insights interface (or called the related API), the data may never have been used for that purpose or  
4 displayed to anyone, and Facebook has no records from which it can ascertain that information for  
5 the class. Thus, even when URL shares in messages were reflected in aggregate demographic data in  
6 Insights during the relevant period, it would be impossible to determine whether that data was ever  
7 actually accessed or displayed for each share or whether any person ever actually used the API to  
8 view the data about the share counts for that URL that included (among other things) URL shares in  
9 messages.

10 64. At a minimum, determining whether a person’s inclusion of a URL in a message was  
11 ever actually included or viewed in anonymous, aggregated Insights data or any related interface or  
12 API would require the following individualized inquiries for each message:

- 13 a. When was the message sent?
- 14 b. Was a share object successful created for a URL attachment in that message?  
15 (i.e., all the individualized inquiries in paragraph 22 above.)
- 16 c. Was the Insights counter successfully incremented?
- 17 d. Did the domain owner for that URL ever provide authentication?
- 18 e. Did the domain owner ever access Insights or Related APIs?

19 65. To my knowledge, neither Facebook nor any other entity possesses the data that would  
20 be required to ascertain the answers to the inquiries in paragraph 64, either on an individual or bulk  
21 basis, for putative class members.

## 22 **VIII. “Link Stats” and Graph API**

### 23 **A. Overview**

24 66. The original [REDACTED] FQL table I introduced in my June 1 Declaration (attached  
25 hereto as Exhibit OO) was renamed to be called the “link\_stats” table. Facebook exposed certain  
26 data in the table such that individuals outside Facebook could write queries or use APIs to obtain  
27 certain aggregate, anonymous statistical information contained in the internal “link\_stats” table.

1 Among the available information were the aggregate, anonymous counters reflecting shares of URLs  
2 through the Facebook platform stored in the “tracking info” in the global share objects. Herein, I  
3 refer to these publicly available queries or APIs as “link\_stats” queries or APIs. Facebook provided  
4 these APIs publicly in order to allow for the development of products and features that incorporate  
5 engagement statistics—products that take into account what people are interacting with now.

6 67. Graph API is an API that allows third-party apps to read and write to Facebook’s  
7 “social graph”—a general name for a store of data about users and their activity that Facebook has  
8 made available to developers in certain ways to facilitate the creation of products and features that  
9 interact with the Facebook platform in both directions. Developers and their users can learn about  
10 other users’ engagement with different information and contribute their own data to that effort, and  
11 build products that incorporate that information in useful ways. Developers can use the Graph API  
12 to, for instance, query data, post stories, upload photos, and perform other similar activities.

13 68. During the proposed Class Period, the “link\_stats” API and Graph APIs showed the  
14 same aggregate counts in Facebook’s global share objects, but were named differently and more  
15 aggregated when exposed externally. “Link\_stats” and Graph API only exposed externally a “share  
16 count” that included URL shares in both posts and messages. They did not expose any count of only  
17 URLs shares in messages.

18 **B. “Link Stats” and Graph API and Messages**

19 69. For a limited period of time between August 2010 and October 2012, if a share object  
20 was created from the URL share attachment sent with a message, the internal “link\_stats” table may  
21 have recorded that fact in an anonymous aggregate count, and thus the externally-facing “link\_stats”  
22 APIs would have likewise shown an increment in the anonymous count for URL shares (which  
23 included both shares in messages and in posts). However, if no share object was created (which  
24 could have occurred for a variety of reasons, described above, including certain race conditions, site  
25 integrity functionality blocking the message or the URL attachment, and so forth), the URL share  
26 shared in that message would not have been included in counters that could be queried through  
27 “link\_stats” or the Graph API.

1           70. Further, to the extent that URL shares in messages were included in the anonymous  
2 count made available to developers through the “link\_stats” or Graph API, the number of URL shares  
3 sent through messages (as opposed to generated through profile posts, comments, “Like” clicks, etc.)  
4 was never publicly available; it was only ever included as part of larger, aggregate counts, such as the  
5 “share\_count,”<sup>5</sup> which also included posts, during certain periods of time.

6           71. Facebook stopped including URL shares in messages in any public counters in the  
7 “link\_stats” and Graph API after October 16, 2012.

8           72. I also understand that Plaintiffs claim that objects created from URLs in messages  
9 “populate Facebook’s Social Graph,” thereby “increasing the quality of its ability to provide  
10 predictive value, and, consequently, increasing Facebook’s advertising revenue and value.” (Mot. at  
11 22.) This is incorrect. To my knowledge, none of the tools that Facebook provides externally call  
12 objects created from URLs in messages, nor any associations or nodes from those objects. The only  
13 relationship of URLs in messages to Facebook’s externally exposed “Social Graph” was, for a short  
14 period of time until October 2012, their inclusion (under certain circumstances) in some aggregate,  
15 anonymous counts of user engagement with certain URLs. This additional information about  
16 engagement is not valuable. Users share information publicly on Facebook, and Facebook can use  
17 that information to understand what users are interested in and how to most effectively tailor the  
18 Facebook service, including its targeted advertising. To my knowledge, information about URLs  
19 shared in messages was never used in that way, and did not “increase[] the quality of its ability to  
20 provide predictive value, and, consequently, increase[] Facebook’s advertising revenue and value,” as  
21 Plaintiffs allege without apparent support.

22           **C. Variability in Connection with “Link\_Stats” & Graph API**

23           73. “Link\_stats” queries and Graph API queries would have reflected the same  
24 anonymous counters stored in the internal Facebook global share objects in a more aggregated form.

---

25           <sup>5</sup> This externally named “share\_count” in the publicly available “link\_stats” table (which  
26 attempted, for a time, to include shares in both messages and posts) is not the same as the  
27 internally named “share\_count” in the global share object discussed above (which did not include  
28 posts).

1 Accordingly, as stated above and in my Declaration dated June 1, 2015, until August 2010, URLs  
2 sent in messages from the Facebook website were not included in these counters and thus would have  
3 not contributed to any counts returned by a “link\_stats” or Graph API query.

4 74. Similarly, after October 16, 2012, URL shares from messages would no longer have  
5 been included in the anonymous aggregate counts returned by a “link\_stats” or Graph API query.

6 75. Accordingly, anyone who only sent messages with URL attachments between October  
7 17, 2012 and the present would not have had their URL share from a message included in this  
8 aggregate, anonymous count and would not have had their URL share from a message included in the  
9 results of a “link\_stats” or Graph API query. In other words, only people who sent a message with a  
10 URL attachment between December 2011 and October 16, 2012 could have been subject to these  
11 practices during the Class Period.

12 76. Further, even for people using Facebook who *did* send a message with a URL  
13 attachment between December 30, 2011 and October 16, 2012, only those for whom the URL  
14 attachment actually resulted in the successful creation of a share object could have had their URL  
15 share included in the results of a “link\_stats” or Graph API query. The aggregate counters in these  
16 results could have been incremented only based on the generation of a share object, which, in a  
17 variety of instances (as detailed above in paragraphs 21 and 22), might never have been generated,  
18 and, for a host of other reasons, may not have been counted in any event (as detailed above in  
19 paragraph 28).

20 77. Moreover, it is highly variable whether anyone ever actually accessed or called the  
21 “link\_stats” or Graph API for any given URL that a class member shared in a message during the  
22 Class Period.

23 78. At a minimum, determining whether a Facebook’s user’s inclusion of a URL in a  
24 message was ever actually included or viewed in “link\_stats” or Graph API queries would require the  
25 following individualized inquiries for each message:

- 26 a. When was the message sent?
- 27 b. Was a share object successfully created for a URL attachment in that message?



(i.e., all the individualized inquiries in paragraph 22 above.)

c. Was the counter in the global share object incremented?

d. Did anyone ever query or call the API for that URL? When?

79. To my knowledge, neither Facebook nor any other entity possesses the data that would be required to ascertain the answers to the inquiries in paragraph 78, either on an individual or bulk basis, for putative class members.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct and that this declaration was executed on January 14, 2016, in Menlo Park, California.

\_\_\_\_\_  
/s/ Alex Himel

Alex Himel

**ATTORNEY ATTESTATION**

I, Christopher Chorba, attest that concurrence in the filing of this Declaration of Alex Himel has been obtained from the signatory. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct. Executed this 15th day of January, 2016, in Los Angeles, California.

Dated: January 15, 2016

\_\_\_\_\_  
*/s/ Christopher Chorba*  
Christopher Chorba

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28