EXHIBIT 28

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

LEADER TECHNOLOGIES, INC.,	)	Trial Volume 1
Plaintiff,	)	C.A. No. 08-862-JJF-LPS
v.	)	
FACEBOOK, INC., a Delaware corporation,	)	
Defendant.	)	

July 19, 2010
9:00 a.m.

BEFORE: THE HONORABLE LEONARD P. STARK
United States District Court Magistrate

APPEARANCES:

POTTER, ANDERSON & CORROON, LLP
BY: PHILIP A. ROVNER, ESQ.

-and-

KING & SPALDING, LLP
BY: PAUL ANDRE, ESQ.
BY: LISA KOBIALKA, ESQ.
BY: JAMES HANNAH, ESQ.

Counsel for Plaintiff

both a fair shake.

And I know you can do it. I have every confidence in you.

I represent Facebook. I'm assuming that you know what Facebook is.

Facebook is a social network. And the evidence is going to show you that Leader-to-Leader is a business and enterprise software company. Social networking, business and enterprise software.

During the entirety of the trial, it's going to be one version and another version. I'll give you two simple ideas. You can adopt them if you want, if you don't come up with your own.

Generalities, specifics. Confusion, clarity. You choose.

As you listen to the arguments of the lawyers and this overwhelming evidence, think for yourself, listen to what I say. Please take it into consideration, but you think for yourself.

You make up your own minds about what that patent covers, what Facebook does and

whether that patent is valid. Because we believe and we will attempt to show you that Facebook does not infringe the patent.

Remember the video they showed you where the man expressed the idea of a patent being like a deed. We sometimes use a surveying term, the metes and the bounds of the patent.

Well, if you have a corner lot and kids run over the corner of your lot all the time, you have a right, as the property owner, to control that. But the deed confines your property rights to the deed. That's what a patent is like.

And everything Mr. Andre said could be true about the inventive process on their side of the house. But it could be just as true that we do not trespass on that property.

So the question you're going to have to be grappling with is, not listening to a bunch of snippets of things thrown at you: What does the patent cover? What does Facebook do?

And are you satisfied at the end of the day that they have carried their burden

to show that Facebook practices the invention? The power of the patent holder is to exclude anyone from using it. That's a weighty decision. That's a weighty decision.

If Facebook infringes the patent, it cannot use that invention. Please give us your full, and fair and undivided attention.

I think it's going to be an interesting week for you. This is the patent. I know you've seen it, but I want to actually take some time to go through it.

That's a big number. You know what that number means? Seven million other patents out there.

Remember the video where you saw the guy run it through the mail room of the patent office, stuff was everywhere. It's a busy office. There's a lot of things that have been invented.

This is the title of the patent. And as I was listening to the tape, I wrote this down. The man on the tape said a title that describes the invention.

Look at the title. Nothing about

networking. It refers to the Dynamic Association of Electronically Stored Information with Iterative Workflow Changes.

That describes what this invention is about. It says nothing about social networking.

There are the two inventors, Mr. McKibben and Mr. Lamb. There are these figures in the patent, and they are intended to show you the logic, if you will, of the invention or the way you could put together a system or a method.

And as you listen to the evidence come in, think about when you go deliberate looking back at this evidence and looking at these figures for yourself and looking for things.

As the Court has instructed you, the next section of the patent is called a specification. Now, this is where they describe what was leading to the invention, what the background is, how they got there, what they were trying to accomplish, another good place to look.

And then as you heard, there are

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

LEADER TECHNOLOGIES, INC.,	)	Trial Volume 2
Plaintiff,	)	C.A. No. 08-862-JJF-LPS
v.	)	
FACEBOOK, INC., a Delaware corporation,	)	
Defendant.	)	

Tuesday, July 20, 2010
9:00 a.m.

BEFORE: THE HONORABLE LEONARD P. STARK
United States District Court Magistrate

APPEARANCES:

POTTER, ANDERSON & CORROON, LLP
BY: PHILIP A. ROVNER, ESQ.

-and-

KING & SPALDING
BY: PAUL ANDRE, ESQ.
BY: LISA KOBIALKA, ESQ.
BY: JAMES HANNAH, ESQ.

Counsel for Plaintiff

And that changed over time, too, as we came up with other applications that we built into that, we added that to the mental pictures of what Leader2Leader was in the product.

Q. Sometimes when you talked about Leader2Leader during your time at Leader, did that include things like LeaderPhone?

A. Yeah, so LeaderPhone was one of the products I developed, helped develop, led the team in developing at Leader Technologies.

Q. Is there any other names that come to mind that would have --

MS. KEEFE: Objection. Beyond the scope.

THE COURT: Overruled.

THE WITNESS: Smart Camera was another application that stood out as something that we didn't conceive of when we originally started, but then later on, hey, this would be a cool addition to throw that in.

Q. Turning to the technology that you developed that you understand is the invention of the '761 patent, when you implemented it, did

it have the ability to share photos?

A. Yes.

Q. Did it have the ability to share videos?

A. Yes.

Q. And how do you know this?

A. It had the ability to share any file by design. It was intentionally trying to not just solve the problems that we knew of at that point, but every, you know, every data problem ever, so if someone came up with a new video format or a new image format or a new 3D we didn't have to know, we built it to handle any of that stuff so all of the stuff that existed at the time, images, photos, video would all have been supported by the file application.

Q. Now, just turning to your deposition, you mentioned that you have made some clarifications?

A. Yes.

Q. What made you think you could make clarifications or corrections to your deposition?

A. At the deposition, the Facebook

lawyer that took my deposition instructed me that I could review it and make those corrections. So when I read it, saw it in print, felt that that clarification was more accurate, I felt that made sense to do that.

Q. So did you just want to make sure your system testimony was really accurate and precise?

A. Yes.

MS. KEEFE: Objection.

THE COURT: I'll strike that answer. The objection is sustained.

MS. KOBIALKA: I have no further questions.

THE COURT: Okay. Thank you. You may step down.

THE WITNESS: Thank you.

MR. ANDRE: Your Honor, at this time, we were going to be playing some videotape deposition. I don't know if you want to start this up before lunch or take the lunch break early.

THE COURT: We'll keep going until 12:30, so you can go ahead and play what you

need to play.

MR. ANDRE: At this time, ladies and gentlemen, we're going to be showing you a videotape deposition of one of Facebook's senior engineers by the name of Josh Wiseman.

He's going to talk about the Facebook technology for the first time.

(Beginning of videotape excerpt of Joshua Wiseman:)

Q. Would the court reporter please swear in the witness?

Q. Good morning.

A. Good morning.

Q. Can you please state your full name and address for the record?

A. Yes. It's Joshua Wiseman. And my address is 1523B Church Street, San Francisco, California 94131.

Q. Are you currently employed, Mr. Wiseman?

A. Yes.

Q. Where are you employed?

A. At Facebook.

Q. And how long have you been working

at Facebook?
A. I've been there for a little over three years.
Q. And what is your current title at Facebook?
A. My current title is engineering manager.
Q. Do you understand today that in addition to your personal testimony, you'll be testifying on behalf of Facebook for certain technical topics?
A. Yes.
Q. Facebook maintains a website; is that right?
A. Correct.
Q. And that website is found at www.facebook.com; is this correct?
A. Yes, that's the URL.
Q. If I was on the internet, correct on a computer and I went www.facebook.com, what would happen?
A. So if you enter www.facebook.com into a web browser, you would arrive at the Facebook website.

By Mr. Hannah:
Q. And what would be displayed?
A. So if you are arriving for the first time, you would see what we call our log-in screen, which presents you with a prompt to enter your email and password. On subsequent visits you might be -- you might be within a logged in experience, which would be accustomed to whatever user you are.
Q. And if a user enters a user name and password, then what happens?
A. Then they are taken to what we call the Facebook home page.
Q. So you said after a user logs in, then they're taken to the Facebook home page; is that right?
A. Yes.
Q. And what does a user see on the Facebook home page.
The Witness: A user on the -- when they're looking at the Facebook home page would see various pieces of content that are relevant to that user within the Facebook experience.
Q. Like what pieces of content?

The Witness: So they would see -- on the top, they would see the Facebook logo. They would see a number of navigational links.
Similarly on the left, they would see navigational links to reach various portions of the site. In the mainframe of the page, they would see what we call our News Feed, and on the right side of the page they would see various advertising units and other features of the site.
By Mr. Hannah:
Q. And what is the -- what is the code that is used to generate the Facebook home page?
The Witness: So the code that is used to generate the home page is, again, what we call our web code. So it's mostly a PHP code that's processing the request and spitting out the home page.
By Mr. Hannah:
Q. Can you recall any specific files that are used to generate the Facebook home page?
A. Yes.

Q. And what are those?
A. So one that I can recall is HTML/Home.PHP.
Q. Any others?
A. That's the most clear one that I can remember off the top of my head.
Q. All right. From the user's home page, you mentioned that there was several different areas that a user can go to; is that right?
A. Correct.
The Witness: Yes.
By Mr. Hannah:
Q. And is one of those profile?
A. Yes.
Q. What happens if a user clicks on the profile tab?
The Witness: There is no profile tab on -- on the home page. There's a profile link or button?
By Mr. Hannah:
Q. Okay. So what happens when you click on the link or button?
A. You are taken to what we call the

profile page, which is another rendering of your information on the site.

Q. And what is displayed to a user on the profile page?

The Witness: So like the home page, there is a top navigational element you can use to get to other parts of the site. You will also see what we call a profile picture, a photo of the person who's profile you're looking at.

There is a tabbed interface for browsing to various subparts of the profile. And depending on whose profile you're looking at, there will be different information displayed by defaults on that page.

By Mr. Hannah:

Q. What code is used to generate the profile page?

The Witness: So, again, like the home page, our web code, our PHP code is used to generate the profile.

By Mr. Hannah:

Q. Do you recall any specific PHP files?

A. Yes.

Q. What are those?

A. HTML/Profile.PHP is the main file used for rendering the profile.

Q. You mentioned there was a photo on your profile page that's displayed; is that right?

A. Correct.

Q. How does a user upload a profile photo?

The Witness: There are several ways a user can upload a profile photo. They're all accessed by -- in a current interface by hovering over the profile photo with your mouse. There's a drop-down menu where you can choose from a few different options.

By Mr. Hannah:

Q. So what are those options?

A. Let's see. So you can -- one option is called something like upload a photo, which will allow you to upload a single photo from your hard drive. There is an option to take a photo, which is referring to like a webcam functionality.

There is an option to choose an existing photo from one of your photo albums -- actually, I believe it's only from your profile photo album, which is a special album of existing profile photos.

Q. So if a user clicks on the upload a profile picture link, what happens?

The Witness: If -- okay. So if a user clicks on the upload link in that -- in that -- in that hover menu, they'll be presented with a browser file chooser, which will allow them to choose a file on their -- on their file system.

By Mr. Hannah:

Q. And, presumably, that file would be a picture; right?

The Witness: So the file chooser that is displayed shows you potentially every file in your file system, but it's restricted to only allow you to select one of several common image formats.

By Mr. Hannah:

Q. If a user selects an image format and uploads it, then what happens?

The Witness: So they are not selecting an image format, but --

Q. Oh, sorry. If they select a file which is in an image format, what happens?

A. So if they select a file and using whatever file system chooser is presented, they click that.

Okay. That file would then be -- would go through a multistep process to upload it to our servers and then return the result to the browser.

Q. And what is that multistep process.

The Witness: So the file is part of an HTML post request to our server. So the bytes of the file itself are piggybacked on an HTML request, which arrives at our web code that processes photo uploads.

From there, the photo is stored in our -- in our -- one of our filer systems that -- that stores the actual bytes persistently.

The data about the photo is stored in one of our databases. And then the URL to the image on -- which will be served from one of

our content distribution networks, like basically one of our web caching partners, will be returned to the user, such that the profile picture on their profile can be replaced with a new one.

Q. You mentioned the photo is stored. Where is the photo stored at?

The Witness: So the photo itself is stored in essentially a big file system on -- in our -- one of our data centers.

By Mr. Hannah:

Q. Is it fair to call that the photo database?

The Witness: That's not what we would call it. We would call it a filer.

By Mr. Hannah:

Q. The photo filer?

A. Correct.

Q. You also mentioned that data about the photo was stored; is that right?

The Witness: Correct.

By Mr. Hannah:

Q. And where is that stored?

A. So the data about the photo is

stored in one of our user databases.

Q. What data about the photo is stored in the user database?

The Witness: So in the user database, we will store the -- we will store a handle to - that will allow us to reference the actual file in the filer. We will store an ID that represents that photo, a unique ID.

We will store the width and height of the photo. And we will store a -- essentially, a number that tells us through which technology the photo was uploaded. So be it a file chooser or our Java uploader, which we use for other parts of the site, in this case for a profile photo, it would just be the file chooser as the main option.

Q. How is the photo associated with the user that uploaded the photo?

The Witness: So we also store the -- what we call the owner ID of the photo as well on the database. That was a field I forgot to mention.

By Mr. Hannah:

Q. Are there any other fields?

A. Yes.

Q. What are those?

A. There are fields that can be stored for other types of photos. So not -- they wouldn't be used for profile photos, but they might be used for some other type of photo.

Q. The handle, the photo ID, the width and height, the technology through which it's uploaded, the owner ID, are those the only fields that are stored with the profile photos?

The Witness: No. It is also possible to store a caption with a photo, but the -- through the interface I explained, there's no way to enter that caption in the UI. There is a way to enter that on other parts of the site.

By Mr. Hannah:

Q. Any other fields that relate to the profile photos?

The Witness: So there is -- and there is an album ID field in this table I'm describing in the database. I'm not sure whether or not it's used for profile photos.

By Mr. Hannah:

Q. Any other fields you can think of?

A. Yes. There is a time field as well which we used to store the time at which the photo was uploaded.

Q. Any others?

A. I think that's it.

Q. You mentioned earlier that there are two different types of photos. There is a profile photo and there's other types of photos; is that right?

The Witness: So profile photos are a type of photo and there are other types of photos.

By Mr. Hannah:

Q. So what are the other types of photos?

A. There are photos which are placed in other contexts around the site. So an album would be one example.

Q. How would a user upload a photo to a regular album using the create album photo path?

The Witness: So you can create an album, again, from -- from several locations I

Page 486

mentioned. So the first one I mentioned was the -- from the photos application, there is a create -- I can't remember exactly what the text is, but something along the lines of create album or upload photos.

Once you enter that flow, you enter the album details, such as the album name and then after filling out those details, you have a choice to use either our robust uploader or our simple file chooser uploader.

Q. Sure. We can just do the simple uploader.

The user clicks on a file and chooses to upload an image file off their hard drive, then what happens?

The Witness: So upon submitting that file in the UI, the file will -- the bytes of the file will be sent along with a -- like HTML post request to our servers. One of our photo-related PHP end points will get that request.

It will be able to parse the file as well as the information on the request, which would be the user ID of the person uploading.

Page 487

And -- and depending on what context you're using might have some other information.

It will take the bytes of the file and store them in our photo filer. And then it will create a database entry that represents that photo.

By Mr. Hannah:

Q. You said depending on the context, different things might happen. What do you mean by that?

A. So in the example we were talking about, you're uploading to a new photo album. So the ID of that album would be on that request as well.

Q. Any other information?

A. Can you clarify?

Q. Sure. You said that the user ID, the album ID would be -- would be stored. Is there any other information regarding the photo that would be stored?

The Witness: Yes, that we would store the width of the photo, the height of the photo, whether the user entered a caption on the photo.

Page 488

By Mr. Hannah:

Q. And the technology associated with it?

A. Correct.

Q. How does the flow change if you upload a photo using a group, for instance?

The Witness: So are you referring to when you upload a photo directly to a group?

By Mr. Hannah:

Q. Right. Does it change? Does the flow change?

The Witness: The photo upload code will be very similar. The context that is passed in the request will contain the group ID rather than, say, the album ID.

I'm not actually familiar with the exact details of the database entry that's created, but I know that there will be a database entry for the photo that is similar to the other entries that we talked about.

By Mr. Hannah:

Q. And that would be stored in the user database as well?

A. Correct.

Page 489

Q. Can a user import a photo from the regular photo album into a group?

A. Yes.

Q. And how does that happen?

The Witness: Are you referring -- are you referring to the UI or in the code?

By Mr. Hannah:

Q. Well, let's start with the UI.

A. Okay. So when you're looking at a group profile, there is a photo section and you can click a link to add photos to that group. Upon clicking that link, you have a choice between uploading photos in -- from your file system, from your hard drive or choosing an existing photo.

If you choose an existing photo, you're presented with an interface that allows you to browse all of the photos in your albums, and then you can select the one or more photos you want to add to that group.

Q. And then if I select a photo and add it to the group, what happens on the user interface?

The Witness: On the user

interface, after that request is issued to our servers and returned, you will see that photo shows up in the photo section of the group.
By Mr. Hannah:
Q. Now, what happens in the code?
A. So in the code, there is a PHP end point which receives those types of requests. The requests will contain, again, the user ID, the group ID. In this case, it would also contain the photo ID since it's an existing photo.
The code would -- the database entry of the photo wouldn't be touched at all. The code would store in a separate table. It would store the user ID or the photo ID and the group ID, such that when you're viewing the group, it will look in that table to find out which photos have been attached to the group.
Q. And would that be stored in the user database?
A. Yes.
Q. Mr. Wiseman, if a user imports a photo from a regular photo album into a group, does that generate a story?

The Witness: Do you mean a News Feed story?
By Mr. Hannah:
Q. Yes.
A. Yes, it does.
Q. And how does it do that?
A. In the PHP -- in the PHP request that processes that action, similar to the way all other News Feed stories work, we create a set of data which contains all the relevant IDs for that action. So the user ID, the group ID, and in this case the photo ID, as well as the time and the type of action that it was, in this case adding a photo from an existing album into a group.
We package that data and send it in a single request to the News Feed service, which can then serve as a story later.
Q. Does that action also generate a story for Mini Feed?
A. Yes.
Q. And how does it do that?
The Witness: It uses that same set of data with the user ID, the photo ID, the

group ID, and it inserts it as an entry into the Mini Feed table?
By Mr. Hannah:
Q. Do you know where each of these databases is located?
The Witness: You mean where they're physically located?
By Mr. Hannah:
Q. Physically located, yes.
A. Yes. I know where the data centers that house all of our -- our -- our databases are.
Q. Where is that at -- or where are they?
A. We have several data centers on the West Coast of the U.S., several on the East Coast. And, yes, all our databases are spread between those -- those two areas.
Q. They are all in the United States?
The Witness: All of our databases are in the United States, yes.
By Mr. Hannah:
Q. How about your filers, where are those located?

The Witness: They are similarly spread between the East Coast and the West Coast of the United States.
By Mr. Hannah:
Q. They're all in the United States, though?
A. As far as I know, yes.
(Conclusion of videotape excerpt of Mr. Wiseman.)
MR. ANDRE: Your Honor, that concludes the videotape deposition of Josh Wiseman.
THE COURT: I take it the next videotape would last longer than two minutes.
MR. ANDRE: I believe so, yeah.
THE COURT: Then we'll let the jury go for lunch now and I'll remind the jurors not to, during the break, discuss the case and return in time to be back in your seats at 1:30.
THE CLERK: All rise.
(Jury leaving the courtroom at 12:27 p.m.)
THE COURT: And we will stand in recess until 1:30.

excerpts. You're seeing only portions of the deposition. And, therefore, these materials have been edited so that you can see the portions that I am permitting you to see. And that explains the I suppose jerky nature of what you're seeing and you can expect to see more of that as we go forward.

Mr. Andre.

MR. ANDRE: Thank you, Your Honor. We're going to be playing another video excerpt of a Facebook engineer by the name of James Wang.

(Videotape deposition:)

Q. Good morning, Mr. Wang.

A. Morning.

Q. Can you please state your full name and address for the record?

A. James Howard Wang. I live at 24 Walter Street, San Francisco, California, 94114.

Q. Are you currently employed, Mr. Wang?

A. Yes.

Q. Where are you employed?

A. I'm employed at Facebook.

Q. How long have you been employed at Facebook?

A. A little over four years.

Q. And what is your current job title at Facebook?

A. My current title is engineering manager.

Q. Do you understand that you are testifying today as a fact witness on your own behalf?

A. A fat witness?

Q. Fact witness.

A. Fact witness. Yes.

Q. And you also understand that you've been designated for certain technical topics on behalf of Facebook?

A. Yes.

Q. When did you start working at Facebook?

A. I believe my start date was in February of 2006.

Q. Have you ever wrote on someone's wall?

A. Yes.

Q. How many times have you done that?

A. Hundreds, maybe thousands. Probably hundreds.

Q. When's the last time you wrote on somebody's wall?

A. Probably a couple days ago.

Q. What happens from a user perspective when you write on someone's wall?

The Witness: From a user perspective, I then see my wall post published on that person's wall, and then I -- I may or may not see a story on my own wall that just shows recent activity that I wrote on this other person's wall.

By Mr. Hannah:

Q. Well, let's start with a user at the home page. How would a user go and write on somebody else's wall?

The Witness: If I knew the person that I wanted -- whose wall I wanted to write on -- first of all, they need to be my friend. But so then I would go and I would go to my search box and start typing their name, and because we have a search type ahead that looks first at

your friends, they probably show up in one of those boxes and then I would click that box.

It would navigate me to their profile. I would go to their wall table, and at the top, I would type my message into the composer interface and then I would hit post.

By Mr. Hannah:

Q. Do you know what happens from a technical perspective?

The Witness: Are you speaking about the -- at the PHP script level or from the database level?

By Mr. Hannah:

Q. Well, start with the PHP script level.

A. Okay.

The Witness: So this is -- I don't own this code. I didn't write this code, but I'm pretty sure that what goes on is you -- you make a form submission, so you make a -- you make a post request.

And actually no, I -- that's one implementation. I think now it uses asynchronous requests. So the page doesn't

refresh, but under the hood we're still sending a web request to a Facebook end point, which contains all the relevant data in the post query parameters that includes the content of the data that's being submitted.

Also, because I'm a logged in user, it's submitting all my user cookies that tells Facebook who I am, so then when Facebook receives this incoming asynchronous request, it's got the user that's logged in, it's got the variables for who -- you know, for the message itself. It's got the target user ID whose wall I'm writing on.

It's also got some additional security checks to make sure that this request isn't spoofed, and then -- and then, basically the script processes it, it calls a function that -- that is in charge of logging and storing the wall post, and then, you know, the appropriate entries into the database are written and then the function returns.

By Mr. Hannah:

Q. Do you know what's written into the database?

The Witness: So I believe that -- and this is like -- and I defer to the code, but I believe what's written on the target user's database is in their wall table we have the "from" the ID of the user that wrote the wall post, we have the time stamp of the wall post and we have the content of the wall post, and we probably also have a unique identifier for that wall post, a global unique identifier.

And then on the user who wrote the wall posts, in their own database, as mentioned earlier, we'll probably have an entry in their -- in their Mini Feed table of the action they performed. And we went over the metadata that was stored there.

By Mr. Hannah:

Q. Just so the record is clear, what metadata would be stored there?

The Witness: So the data that I believe is stored on the Mini Feed data -- the Mini Feed table of the home user's database would be their own user ID, a unique identifier for that Mini Feed story, the story ID type, which you know, identifies this as a wall post

story, the time stamp of the action, and -- and then, I think, in the -- there's like a free-form section that has additional data, would probably include the target -- the recipient of the wall post ID. I think that may be it.

Again, I would defer to the code.

(Conclusion of the videotape deposition excerpt of Mr. Wang.)

MR. ANDRE: That concludes Mr. Wang's videotape deposition. We're going to show you a couple more small clips.

The next clip is of a Facebook vice president and it will be very short, about a two-minute clip.

(Beginning of videotape deposition excerpt of Mr. Dan Rose.)

The Videographer: Would the court reporter please swear in the witness?

Q. Could you please state your full name and address for the record?

A. Dan Rose, 448 Addison Avenue, Palo Alto, California 94301.

Q. When did you start working at

Facebook?

A. 2006.

Q. What was your title?

A. What is my title?

Q. What was your title in 2006?

A. I joined the company as director of business development.

Q. Did that title -- has that title changed since to the present?

A. Yes.

Q. What is it changed to?

A. Vice president of the business development and monetization.

Q. When did that title change?

A. It changed to vice president of business development in 2006, and it changed to the current title in 2008.

Q. Does Facebook collect any kind of information regarding its users and what they are doing on the website in terms of clicks that they might have, making connections, becoming a fan?

The Witness: When a user becomes a fan of a page, we -- that information gets added

to their profile. So, by definition, we're tracking once people become fans of pages.

By Ms. Kobialka:

Q. Are there any other examples that you track what a user is doing on the Facebook website that you can think of?

The Witness: We -- when something shows up on the website, by definition we're tracking it. We have to track it in order for it to show up on the website.

I don't know what you're referring to by tracking, but logging an action in a database somewhere so that we can present that action on the website is something that -- that's what we do. That's what the site does.

(Conclusion of videotape deposition excerpt of Mr. Dan Rose.)

MR. ANDRE: That concludes the videotape deposition of Dan Rose. And our final videotape deposition clip is of another Facebook engineer by the name of Andrew Bosworth.

(Beginning of videotape deposition excerpt of Andrew Bosworth:)

The Videographer: Would the court

reporter please swear in the witness?

By Mr. Hannah:

Q. Can you please state your full name and address for the record?

A. Andrew Garrod Bosworth, 120 Kingsley Avenue, Palo Alto, California 49301.

Q. Are you currently employed, Mr. Bosworth?

A. Yes.

Q. Where are you employed?

A. Facebook.

Q. How long have you been working at Facebook?

A. To what degree of specificity? Four years, two months, a day.

Q. And what is your current title at Facebook?

A. Manager of engineering.

Q. Do you understand that in addition to your personal testimony today, you have been designated for certain technical topics for -- on behalf of Facebook?

A. I do understand that.

Q. Do you understand that one of

those technical topics is the News Feed system?

A. Yes.

Q. You stated News Feed launched in the fall of 2006; is that right?

The Witness: Yes. Does fall begin October, September? Let's say in the last -- last third of the year of 2006.

By Mr. Hannah:

Q. So at the time that Multi Feed was launched in June 2008, can you describe to me how News Feed as a service operated?

A. Sure. So, again, with the ongoing caveat that to be 100 percent certain of this, I would have to look at code.

To the best of my knowledge, the way it would have operated was that actions that -- that actions users took, that generated Falcon log events, would be logged via Falcon and stored in the Falcon logs. Aggregated -- not aggregated, but collected in the Falcon logs.

Multi Feed would then be tailing those logs and loading the logs into memory, essentially creating a memory log as opposed to

a non-disk log. And at that point, when some user -- you land on a page that prompted generation of stories, such as their home page, they would go -- the News Feed code in the front end would query to Multi Feed via RCP, receive data in return, apply privacy to that data to make sure that all of the things that can -- it could possibly show would be shown.

So, you know, already ranked in relevance, now it's going to apply privacy. Having gotten to this core set of stories that it's going to display for each story, given the core data, would go and fetch the additional data required to display that story to the user.

And then go about the business of rendering, as we've discussed kind of a couple of times.

Q. And when you say rendering, that's displaying it to the user?

A. Specifically, rendering is generating the HTML and text and CSS such that when the browser receives it, it's able to parse it and format that data to the user and proper setting and also fetch any images that are

specified by that HTML.
Q. And what is your understanding of how the News Feed service operates today?
A. So, pretty much identical. The only exception being that rather than exclusively tailing the Falcon logs, logs can also be loaded directly -- you can also load logs directly into memory from user actions basically, if that makes any sense.
Q. And then -- and then what happens?
A. So from there, the same process that I have just described applies again. Do you want me to go over it again?
Q. Sure.
A. Fair enough. Some user lands on a page that prompts the generation of News Feed. It's their home page.
That -- the News Feed code that exists in their kind of PHP layer will then fetch the RPC for Multi Feed data that's relevant and ranked and ready to go.
Apply privacy to that data to ensure that the user does not see anything that they're not supposed to see. At which point, it

will then endeavor to fetch additional data required to render and display the story in its full kind of rich glory.
I'm getting better at answering this question the fourth or fifth time, start using flourishes. And then, having -- that data will go about the business of rendering HTML, CSS in such a way and text, in such a way that when the browser receives that, it will be able to display in rich fashion the story to the user.
Q. Are you familiar with Mini Feed.
The Witness: Yes.
By Mr. Hannah:
Q. When was Mini Feed first available to users?
The Witness: It was first available at the same time as News Feed.
By Mr. Hannah:
Q. That was the fall of 2006 or October 2006?
The Witness: I believe the answer is October 2006, but, yes. The last half of 2006. I was there. I just -- it was a long

time ago.
By Mr. Hannah:
Q. How did Mini Feed operate in October 2006?
The Witness: Again, what level -- what are we dealing with here, what level of answer would you like? From a product perspective?
By Mr. Hannah:
Q. Yeah, technical perspective.
A. Technical perspective. So not a product perspective.
Q. Let's start with a product perspective then --
The Witness: From a product perspective when any user would view a profile that they had the permission to view, according to privacy, they would see a series of recent actions taken by that user, provided, of course, that each individual action was also a publicly visible kind of creation of some new content.
By Mr. Hannah:
Q. And how did it work from a technical perspective?

The Witness: When a user took an action is on the site that was something we felt could be displayed -- well, had, you know, kind of a public aspect to it that could be discovered, we would log to that user's database, that action, with data similar, though not necessarily identical to what was discussed earlier for logging to Falcon.
So that would be at that point in that table. At some point later, some other user, really any other user that had access to that profile, could come to that profile and upon arriving would trigger the generation now of Mini Feed stories.
And, again, the story describes what is displayed to the user. The data behind it is the kind of data I mentioned in the Mini Feed table of that user's database. So when some user, any other user arrives at that profile, the Mini Feed code would fetch from that user's Mini Feed table, the data, which is just kind of a little data, as little as necessary, using that -- or apply privacy, make sure it's allowed to be seen.

And then go and fetch whatever additional information is needed in order to display that to the user.

And again, the display is rich, ntedia rich, so the rendering of it is HTML, CSS, links to HTML, links -- links to images, which when the browser receives them, it will go and fetch that image and render it onto the screen for the user.

So that's -- that's how it works.

Q. Had there been significant changes to the Minifeed since its launch?

A. Not that I know of.

(End of videotape.)

MR. ANDRE: Your Honor, that concludes the videotape deposition of Mr. Bosworth.

THE COURT: You may call your next witness.

MR. ANDRE: Your Honor, we'll be calling Dr. Giovanni Vigna. He's out of the courtroom. Can we pass out jury binders?

MS. KEEFE: Go ahead. I was going to say subject to our previous objection, it's

fine.

THE COURT: You have may do so. Thank you.

THE WITNESS: My name is Giovanni Vigna. G-I-O-V-A-N-N-I, V-I-G-N-A.

GIOVANNI VIGNA, Ph.D.,
the deponent herein, having first been duly sworn on oath, was examined and testified as follows:

THE COURT: Good afternoon.

THE WITNESS: Good afternoon

DIRECT EXAMINATION

BY MR. ANDRE:

Q. Good afternoon, Dr. Vigna. Would you please give us your educational background.

A. I received a masters in electronic engineering and a Ph.D. in electronic engineering in 1994 and 1998 respectively from the Politecnico Milano in Italy.

Q. What has been your employment history since that time period?

A. So I was post doc for two years at UCSB and then in 2000 I became an assistant

professor at the University of California in Santa Barbara. In 2004, I became an associate professor at the same university. And in 2009, I became a full professor at that university.

Q. Do you have any awards or honors for your research?

A. I have several that are based partly on my research and partly on my teaching. I have best paper awards or awards due to how my work was influential in certain areas. And also I got teaching awards, for example, from the Academic Senate, University of California Santa Barbara.

Q. Have you received any funding for your research?

A. That's part of my job is to get funding so I can pay students who do research for me, I should say with me. And I received around $10 million in funding so far.

Q. And do you have any publications relevant to the computer science industry?

A. Yes. I have a number of publications. I have journal publication, I have conference publication, around sixty, that

are the main way in which we publish our research and our results. I have book chapters and workshop publications.

Q. And have you given any tutorials on the computer science industry outside of your classroom work?

A. Yes, several times I have been asked to give tutorials on different topics within computer science.

Q. And are you a member of any professional organizations related to computer science?

A. I am a member of the IEEE which is the Institute of Electronic and Electrical Engineers. I am a member of the ACM, which is the Association For Computing Machinery. I am member of USENIX and a member of The Computer Society.

Q. Do you have any editorships?

A. I was on the editorial board on a number of journals in my field. And I also edited proceedings of conferences and books.

Q. Which technical program committees have you been involved with?

A. Well, I have been involved with very many as a committee member, as a chair of a committee, I have been involved with three, namely RAID and NDSS and IEEE Security and Privacy which is the main conference in my field.

Q. Have you been on any committees at the University of California?

A. Yeah, I have been on several committees, both at the departmental level and at the college level.

Q. What type of classes have you taught at the University of California?

A. I have taught sort of a range of classes in computer science such as natural computing, operating systems, computer security analysis, things like that.

Q. How many graduate students do you supervise?

A. Currently I directly supervise six Ph.D. students. I also have two post docs, so they are students that after the Ph.D. continue their research work with me. And sometimes I work with undergrad students.

MR. ANDRE: Your Honor, at this point we would like to tender Dr. Vigna as an expert in computer science.

MS. KEEFE: No objection.

THE COURT: He is so recognized.

MR. ANDRE: Thank you, Your Honor.

BY MR. ANDRE:

Q. Dr. Vigna, you have been retained by Leader Technologies to provide an opinion in this case; is that correct?

A. That is correct.

Q. And what were you retained to do by Leader?

A. So I was asked as an expert to determine whether or not Facebook was infringing the '761 patent.

Q. And before you were asked to be an expert in this case, did you have any opinions on this topic?

A. No, absolutely not.

Q. And what did you rely upon in forming your opinion?

A. Well, I relied on a number of things. Of course I examined directly the

Facebook website. Then I was given access to the Facebook source code. I had access to documentation, public documents that are available to everybody, help files, for example, description of the API with which people can interact with the website. And the deposition of key people within Facebook. And I would say that's pretty much it. I might have left something out.

Q. Did you read the construction order?

A. Absolutely I did.

Q. When you're talking about the different types of documents that you are talking about, were you able to look at some developer documents as well?

A. Correct. So I had access both at publicly available documents and at documents that were used internally to describe particular aspects of the Facebook websites to the developers at Facebook.

Q. And based on all that you reviewed, what did you come to the conclusion of?

A. Well, my conclusion was that indeed Facebook infringes on the '761 patent.

Q. Before we get started, could you explain to the Court the setup you have in here in front of you. A lot of computers going on. Can you explain what's happening?

A. Yeah, especially the cables, there is way too many cables. This is a setup that I will use during my testimony to sort of explain why Facebook infringes in my opinion. There are two computers. There is a computer down here whose display will appear here and of course in front of you as well.

And this computer contains Facebook's source code, so the source code that actually makes the website work and do whatever it does. This is my personal computer, and I have some material on it that I will use to demonstrate several aspects of this issue.

And since we don't have internet, I will actually show you some movies that I recorded when I did have internet, and I used in this particular case a couple of tools to make apparent what goes on under the hood in a way.

When users interact with a website like Facebook, they usually have a browser. And the browser presents some kind of pretty picture of what the website wants the user to see. Of course, there is a lot that goes on in the background in terms of the information that is captured and exchanged with the website.

And thank God we don't see that, because it's not pretty. And in general we should not be concerned with that part.

In fact, in this particular technical matter, we want to understand exactly what happens between the browser and the server. And, therefore, I create a setup where there is an interceptor, it's like a component that sits between the browser, that is the tool used by the user, and the Facebook side. So that every time an operation happens, this interceptor can stop the request going to the Facebook website, allow us to see what is going on really under the hood, and then let it go and perform the actual portion.

So it's sort of like taking a snapshot of something that would otherwise move

very fast away and would not allow us to understand what is going on.

In addition, I will use something that allows us to inspect the exact code in terms of HTML code that is sent to the user. And I will, of course, explain to the best of my capability everything when we get there.

THE COURT: Hold on. Ms. Keefe is on her feet.

MS. KEEFE: Your Honor, at this time I would like to reflect for the record this objection that we made earlier this morning and just maintain that.

THE COURT: That objection is on the record, but my ruling stands. Go ahead.

MS. KEEFE: Thank you.

BY MR. ANDRE:

Q. Just before we get started, could you just give an understanding, a description of source code that we can probably understand, what is source code?

A. Yes. Well, source code is a series of instructions in a specific language that tell a computer what to do fundamentally.

And in particular, we will see a lot of PHP code. It's one type of code. There are many languages that are used in computer science. In this particular case the source code that we will review is PHP. So this is the code that actually implements the behavior of the Facebook application.

So at the very end is the data that represent the executable behavior of the application.

Q. And we've heard some terminology here. I would like you to explain it as well. What is a UI?

A. A UI is a user interface. So as I said before, there is a lot that goes in the background, we don't want to know, we would never like hey, you want to float a picture or two on a web application, look at this code. Nobody wants to do that. I don't care about that. So the UI is the user interface that is the only thing that the user uses in order to interact with the application.

And usually the prettier the UI, the easier to use, the better. And whatever

happens in the background is for the user irrelevant.

Q. And what is an API?

A. The API is more complex to describe. It's an application programing interface. So the idea is that there are applications that provide some kind of functionality. And at a certain point they will use this functionality internally, but they might want other people to use it. So how should other people use this functionality. Well, in a stylish practice in computer science is to create one of these API and say okay, these are all the way in which you can use my application.

To make an example with the Facebook website, suppose that I want somebody to be able to come to my personal account and find out about all my friends. Well, Facebook provides a function so that you can find all the friends of a specific user given the particular ID of the user. So it's like a way to -- a well-defined way to interact with an application, for example, Facebook to get

information out of it.

Q. Now, I would like to draw your attention to PTX 1, the '761 patent. And can you generally just tell me what the '761 patent discloses?

A. So the '761 patent describes sort of a collaboration tool in which users can share data, sort of they have a share of the world, and when users do things in this world, they are tracked. And there is information that is maintained about these users so that they can interact with each other and share data effectively.

Q. And what is an online collaboration tool generally speaking?

A. Well, generally speaking it is anything that allows multiple people to do things they want to do together. They want to share pictures, they want to send messages to each other, that's a collaboration tool.

Q. And generally, what is the Facebook website?

A. Well, the Facebook website is a collaboration tool, and it performs the

functions described in the '761 patent.

Q. You prepared some demonstratives. And I would like for you just to walk through with the jury and explain how the system captures the context and tracking information, how Facebook system captures context and tracking information?

A. Okay. So we will revisit this concepts a number of times, so if something is not clear, I'm sure that we will repeat a number of times.

But the main idea is that there is the concept of an account and a particular context or environment in which a user operates. And, for example, in this particular case we're in the home page of John Vineyard, which happens to be the English translation of my Italian name.

And go to the next one. In this case, for example, I want to upload some user defined data. For example, there is a picture of me that I want to upload to the website.

And you can see that there is additional information in addition to the

picture that is captured. This is context information that, for example, is the album where this picture should appear, the name of the year, the ID of the creator and so forth. And this information is captured and is stored in metadata as context information.

Go to the next.

And as a side effect, for example, my picture is now uploaded to the site and appears there and can be accessed whenever I access my profile.

Then I can move from one environment to another. For example, in this case, I -- if you go to the next, you will see that I moved to the page of a friend, in this case Mary Smith, and we're friends of each other.

Actually it's difficult to see in this particular case, but there are little pictures down there of each other. I have a laser pointer. Let's see if I can do this without hurting anybody. This is the picture of me, and this is the picture of her. And we're friends. So I can move and go to a different

context, a different environment which is the profile of this other user.

And, for example, I can perform some actions in that second environment, for example, I can post on the wall that person a message that says, "How are you?"

When I do that, what happens is for content, it was created in the first environment is used in the second environment. You can see here, for example, that in this second environment, the content that I just introduced in this first environment is used together with the data that I just introduced, the comment how are you.

And this is also generating tracking information that says that something happened. And this is stored in the metadata down here. The tracking information contains several kind of data, like the user ID, the type of event or type of story that is being created and so forth.

So the aspect of the patent is that there is this concept of capturing context information that is stored in the metadata,

tracking a user as it moves around, and then generating tracking information as whenever certain actions happen, like writing on the wall, joining a group, uploading a photo to an album and so forth.

Next. And you can see that this tracking information is also reflected had back in the original account because there is a news feed and a Minifeed which is two ways which this information is presented to this user. In this particular case in my original file a note appears that John wrote on Mary Smith's wall. I think we're done.

Q. Now, let me ask you some even more fundamental questions. How do you get to the Facebook website?

A. How do you get there?

Q. Yes.

A. You open a browser and you type a URL in the browser, and you actually are sent to the website.

Q. Do you know the website address?

A. It's www.Facebook.com.

Q. And how do you get to an account?

A. Well, usually when you start interacting with the Facebook website, if you don't have an account, if you have never been on it, or maybe you have an account but you're not logged in, and therefore you will get a page that invites you to either join Facebook and create a new account or to log in with the account that you already created.

MR. ANDRE: Your Honor, at this time I'd like to go set up a white board next to the witness. May I approach?

THE COURT: Yes, you may approach.

MR. ANDRE: Thank you. Is that okay?

THE COURT: Yeah, as long as the jury can see it. And Ms. Keefe, if you need to move so you can get a better view, that's fine.

MS. KEEFE: I'll have to move. Too many things in the way, Your Honor. Sorry.

BY MR. ANDRE:

Q. All right. Dr. Vigna, let's look at the claims of the '761 patent that's been asserted against Facebook.

A. Yes. Okay.

Q. First of all, is your understanding that in order for a product to infringe, it must meet all the elements of the claim?

A. Yes.

Q. Is also your understanding that you only look to the claims to determine infringement?

A. Yes.

Q. If you look at the Claim 1, element one, the context component; do you see that?

A. Yes.

Q. Can you put the screen up?

Sorry, I realized that she was standing there. I thought she was going to sit back down.

MS. KEEFE: I thought you were going to do something with it.

THE COURT: I think if she -- if you believe she's going to need to stand, if you're going to direct us to the board, feel free to bring a chair over so that you can -- you don't have to stand for the whole time.

MS. KEEFE: Thank you, Your Honor

BY MR. ANDRE:

Q. Dr. Vigna, would you please briefly describe the elements of Claim 1?

A. So the first element says that there is a computer-implemented context component of the network-based system for capturing context information associated with user-defined data created by user interaction of a user in a first context of the network-based system, the context component dynamically storing the context information in metadata associated with the user-defined data, the user-defined data and metadata stored on a storage component of the network-based system.

Q. Could you give us a -- your understanding of what that claim element is referring to?

A. So this claim element describes in very technical terms basic concept that there is a context component. Whenever a user wants to provide some data, it will capture that data, plus other data, some context information.

Take both these things and store

them in a storage using a storage component into metadata which is additional data about a certain data. Okay.

So it is rather abstract. So it describes a generic component like that can be implemented in many different ways, but the gist of it is that there is some data of a user, for example, a personal picture and there is something else that is captured of that particular environment, which that data is entered and this information is stored as metadata on a storage component.

Q. Now, I'd like to show you the court order for the claim interpretation in this case. I want to direct your attention to the term component.

Do you see that?

A. Yes.

Q. Do you recognize this as the order from the Court interpreting the claims?

A. Yes.

Q. And could you read what the term component means?

A. So in this document, it say the

term component means a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution.

Q. Now, what does that mean to computer scientists?

A. Well, in this particular case, I would say --

THE COURT: Hold on. There's an objection.

MS. KEEFE: Objection. Your Honor, that's the definition, not what it means to him. It's what it means to the Court and the Court's construed it that way.

MR. ANDRE: I'll rephrase it that way, Your Honor.

THE COURT: Sustained. Sustain the question.

MR. ANDRE: I will.

BY MR. ANDRE:

Q. When you're talking about hardware, what's that referring to?

A. Well, it's referring to any kind of equipment, group of equipment, it could be one CPU. It could be a CPU on a disk.

It could be an array of disks. It could be a network system like a distributed system. It could be even spread across the nation.

That would be hardware. It's -- it's a composition of hardware elements.

Q. And when you see one skilled in the art when they see that the word in combination of hardware and software, what would that mean to you?

MS. KEEFE: Same objection, Your Honor. I mean --

THE COURT: We will see counsel at side-bar.

MS. KEEFE: Your Honor, it's the Court's claim construction. The Court's claim construction is what it is.

And it seems like we're trying to reargue claim construction by redefining what the construction is.

THE COURT: Mr. Andre?

MR. ANDRE: Your Honor, the claim construction is determined based on one skilled in the art. Words in construction have special

meaning to those skilled in the art. I'm just asking what those words are and what they mean.

THE COURT: I think in this case, the jury needs some translation into English essentially to understand the concepts. And that's my understanding of what these questions are seeking to elicit, not reconstruing claims. But just trying to help the jury understand what it is that the Court's construction says.

MS. KEEFE: I think he's going a little bit far, Your Honor. We are talking about words that are supposed to have plain meaning. This is the definition they propose. It comes from the patent.

THE COURT: I'm overruling the objection.

(Conclusion of conference held at side-bar.)

BY MR. ANDRE:

Q. Dr. Vigna, go back to my previous question. What does it mean when there's a combination of hardware and software?

A. Well, usually a combination of hardware and software is a system that is

Page 546

composed of hardware and software that usually determines how the hardware is going to behave. And this can be any kind of composition.

It could be in an embedded system like cell phone is a combination of hardware and software. And it could be a large-scale system composed of thousands of computers.

There are compositions of hardware an software that makes them work together.

Q. I think we all got a pretty good feel of what software is, but what is software in execution?

A. Well, software in execution is when the code that determines what the software is. So it receives the instruction, it's telling hardware what to do. They are effectively executed and the hardware does what the instruction tells the hardware to do.

Q. And would source code be just software?

A. Yeah, software usually refers to source code -- not source code. Software usually imports instructions for the hardware. So it could be source code. It could be

Page 547

interpreted source code like in the case of PHP.

It could be compiled source code in the case of libraries. And in this case, the source code becomes executable code in manner of form.

Q. Going back to Claim 1, there's a claim element called the storage component at the end of Claim 1. Do you see that?

A. Yes.

Q. Does the storage component of the '761 patent have to be a single server?

A. Well, I think that --

MS. KEEFE: Objection. Sorry, Your Honor. Objection.

Again, the Court's claim construction says what it says and we have a definition for component.

THE COURT: Mr. Andre?

MR. ANDRE: Your Honor, I'm asking: The storage component in this claim has been inferred in counsel's opening to be a single server. And I don't believe that's the case.

Dr. Vigna can support his opinion

Page 548

based on the specification of the patent.

THE COURT: I'm going to sustain the objection to the question as asked. You can ask the expert what his opinion is as long as he's applying the Court's construction.

BY MR. ANDRE:

Q. Okay. Dr. Vigna, do you have an opinion as to whether or not a storage component has to be a single server?

A. I do. In -- the storage component is an architectural concept in the design of an application. It is not bound to one particular implementation of one server or multiple servers. And actually I think if you pull up the patent itself, and in the specification of the patent, I think it's in Column 5 towards the bottom.

So it's going to be around -- yeah. In this case, there are -- go -- they say one or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

Page 549

And I think that what it's trying to say here is that this component is, like it is often done in computer science. It is a way to express a functionality.

The way in which you perform the functionality is largely irrelevant, meaning if it's a storage component, it's something that, you know, allows you to store something and later when you want it, it's still there. You can get it out.

THE COURT: All right. Doctor, forgive me for interrupting you.

Is it the same objection?

MS. KEEFE: I very much apologize. It's a slight tweak on the objection.

The problem here is that we have a claim construction and the law on claim construction says that you do not need limitations from the specification into the claims.

THE COURT: I'm overruling the objection. The record will note your objection to this line of questioning.

THE COURT: Go ahead.

MR. ANDRE: Thank you, Your Honor.
BY MR. ANDRE:
Q. Did you finish your answer?
A. Well, I was almost close to finished.
Q. Okay.
A. Let me go. So the storage component typically can use different types of storage, can use a memory cash, can use a disk, can use an array of disks organized in a certain way, can use a database, can use a set of federated database, a database that talks to each other so that they can hold even more information.
The basic concept is that a storage component is something that stores data so that you can retrieve it afterwards.
Q. I'm going to show you what was marked as a demonstrative in this case earlier. I know you haven't seen this document, though Dr. Vigna, but when you see server one, server two and server three -- or strike that question. Let me try another one.
What would be the storage

component in this figure?
A. Well, in this here as I interpret it, I would say that the three servers together perform the function of a storage component. Because I'm here seeing that there are database tables, activity logs, tracking information and there is raw data.
So these are all things that you want to store so that later you can retrieve them. So altogether, regardless of the type of information, would present a storage component.
Q. Going back to the first element of Claim 1, is it your opinion -- do you have an opinion as to whether or not Facebook's website contains a context component?
A. Yes, I do. And I think that it does.
Q. Now, you said earlier that you had a program on your computer in which you could show the Facebook website in action under the hood you said; correct?
A. Yeah. That is correct.
Q. So just making sure I understand, when you send a request to Facebook from your

computer, your computer stops it from going and shows you what you're sending; is that correct?
A. Correct.
Q. And then after it goes to Facebook's servers in California or on the East Coast wherever they are, and it comes back, before it shows it on your screen, your computer stops it again?
A. Okay. Yes. That's correct.
Q. I just wanted to make sure I understand.
A. Yeah.
Q. Okay. And you have a representation of that on your computer?
A. That is correct.
Q. Okay. Could you please demonstrate to the jury how that actually happens?
A. Okay.
Q. Before you begin that, let me just ask you one more question.
A. Yeah.
Q. Why do you think Facebook has a context component?

A. Well, it has a context component because it wants to capture user data, plus additional context information so that it can use it to share it with other people.
Q. Okay. Okay. Could you demonstrate that on your program here?
A. Yes. It's here.
So here what you see is, you know, of course, not Facebook, another well-known website. And we see the user go into Facebook. You can see it on the top bar.
And this particular case, the user goes to the log-in page and puts in the email address of the user and the password to be able to log in.
And once the user logs in, it's sent to the home page that is in this case empty. Then the user goes to the profile.
And here you can see that the moment -- let me stop here really fast. Okay. What you see here is that I switched from -- you can see here the moment here at this point I clicked on profile up there to go to the profile page of John Vineyard account.

And then I moved to this interceptor, which is the tool that I was telling you about that is able to intercept the communication between the user and the Facebook website. In this particular case, the name burp.

It's not the most inspiring name, but it's a tool that's very useful that I use routinely in my work. So what you can see here is that this is the raw data that is sent to Facebook whenever the user clicks on that button.

So the experience of the user is I click on -- I want to go to profile. What happens under the hood is that an PHP request, and it's just a way to say communication is performed and the content of the communication is displayed right here.

Now, it's like a bird left from the browser and is going to Facebook with a message. And I just grab that bird and I'm opening up and say, Okay. What are you sending to Facebook? Let me look at the message.

And if it's okay once I'm done

looking at it, I can let it continue towards Facebook. Okay.

So in this particular case, for example, I see that by pressing that button, there is a request for the module Profile.PHP. And it is -- I highlight it up there.

You can see this is getprofile.PHP passing a number of parameters. And also there is a cookie that as we will see it's all that data that I highlighted that is important. And we will see later when we talk about tracking how that is important.

So now, I will just slide okay forward, let the bird go and go to Facebook. I get the results back and here I am in my profile page. Okay. So this simple interaction shows you that you, that UI, which would be the one with the button to say profile interact with the user, actually the inter -- users interact with a UI.

There was a click. There was some information that was sent to Facebook. And this page is sent back and is displayed to the user.

Okay. So that's pretty much how

you interact with the Facebook website. And so if now I go on, you can see that there is my picture here. I say, I don't like it very much and I want to upload a new profile picture.

So I press another button, so another action is generated. And here I switched again. I grabbed that communication before it reached Facebook, and I'm going to examine it.

For example, in this particular -- in this particular place, we can see that the ajax profile picture upload.PHP file is requested passing some parameters.

And as a result of this, what happens is that what is sent back to me is this pop-up dialogue that asks me for a new picture. And when it selects one, another request is performed. And that is the request that is performed.

And you can see up there that there is -- that there is a bunch of data, in particular, at the bottom -- sorry to scroll it. You see all this gibberish at the button is -- let me just go back for a second just to -- so

all this information that you see here right here, this is the raw information about the picture that I'm up loading.

And this information is sent together with other information, and I'm going to click. I want to explain to you, I click on params. You can see that there are those four headers here.

There is one that say raw, one say param. So when it show you the raw version, I show you exactly the communication that happened. When I clicked on params, what the tool did is actually parsing that data and is presenting you that same information in a much -- in a much clearer fashion where it can show you exactly the different information that is sent.

So the two information that I show is the content is exactly the same. It's just shown to you in two different ways.

And, for example, here you can see that there is the ID of the user, the type, the profile, picture. And so this information is sent to Facebook. And as a result, now we get

back information from Facebook and my picture is actually updated.

And this shows how by interacting with the Facebook website, we're able to capture not only the raw data associated with the picture itself, but now they're -- but other context information like the type of information, the idea before the user that this picture refers to and so forth.

MR. ANDRE: And, Your Honor, we're going to try executing the source code at this point, so I think we're a few minutes early for the break, but maybe we could do that and close the courtroom.

THE COURT: Right. It will be a good time for our break. And when we come back we'll close the courtroom. Let the jury step out first.

MR. ANDRE: I'm sorry.

THE COURT: We'll be back in 15 minutes.

(A brief recess was taken.)

THE COURT: We're ready to have Dr. Vigna take the stand again. We'll bring the

jury in.

For the record, has the courtroom been closed? Is there anybody that --

MS. KEEFE: There are people that I don't recognize that aren't even in the law firm on the other side.

THE COURT: Counsel confer with one another after the jury gets settled.

(Jury entering the courtroom at 3:20 p.m.

THE CLERK: Please be seated.

THE COURT: Welcome back everyone for the last session of the afternoon.

Is there any issue?

MS. KEEFE: No.

THE COURT: You may continue.

MR. ANDRE: Thank you, Your Honor.

BY MR. ANDRE:

Q. Dr. Vigna, when we took a break you had just shown an example of a profile photo using your interceptor program.

A. Yes.

Q. Could you show us in the Facebook source code where that's taking place as well?

A. Yes. I will follow my expert report just because I don't know by heart all the things that I have to show you.

So what you see here as I was telling you at the beginning, this is the source code computer, and let's see here. So this is pretty much the -- one of the snapshots of the code, and it's the one that I mainly reviewed. So now I'm going to open an editor.

Q. What is an editor?

A. Sorry. An editor is just -- this editor is called Emacs and it's used to review any type of text, including source code that in this case is text. So it's just, if you're familiar with the Windows operating system, I could also use Notepad. It would be exactly the same thing.

So, if you remember when I show you the uploading of the picture, let me just go back for one second here. You see up here there is this pic upload.PHP. So this is actually testimony source code component that receive information of the picture, profile picture in addition to more information.

And now I will show you what pic upload actually is. As you will see it is one of the most unglamorous things that you can imagine.

Q. Is pic upload, is that the link or folder name? What would that be?

A. Sorry. Say again.

Q. Would the pic upload, would that be the name of the file?

A. Correct. It's the name of the file. I mean, what I show you, so in this is technically pic upload is technically the resource that is requested by the browser, but of course that is mapped to a source code file on the Facebook server.

So what I'm going to show you now is that Facebook. Sorry, this is not the best environment to do this. But I'll do my best to not be completely -- so that request I show you will actually invoke this source code. Okay? And, of course, this source code to I think most of the people in this room doesn't say really anything, but what this source code represents is a series of instructions that represents

software that when executed will perform a particular task.

So I always explain software sort of like a recipe, when you read a recipe, you say okay, take out three eggs and scramble the eggs and put this butter on the pan, so it's a series of tasks that have to be executed in a certain order in order to achieve a certain goal.

Software is exactly the same. You can see every line here has a particular meaning and altogether one instruction after another will end up executing some kind of task to achieve a certain goal.

In this particular case, the source code in the pic upload.PHP file will receive that image and in addition will capture context information and store this in the metadata.

So this is a rather complex thing, but what you can see is, for example, this file gets executed, and this file calls other files, actually it invokes a number of pictures, you can see that there are a number of instructions,

and this, for example, call this pic upload function that you can see right here. And this function calls -- so what's a function? Sorry. A function is like a group of instruction that does a specific task.

And, of course, we could have one very large file with everything that has to be done, but human beings are not really good at handling very complex, very large scale set of instruction. We like to, you know, put little snippets together to achieve a specific subgoal and so that we can understand it.

So, for example, there is -- if I say, for example, prepare some scrambled -- no, prepare some scrambled eggs is an overall goal, but if I take the eggs and heat them, there are a number of subtasks that are involved and I just put them together and say prepare the egg.

So a function, it's sort of taking a number of instruction and package them so they can be reused multiple times.

So this is, for example, a function to upload a picture. And this function will call other function that will call other

functions in different files until there is a function. And I will spare you having to track all these functions across because it's excruciatingly painful, but I'm going to get to one important point is this important point is a function called add photo which is in a different file called list photos PHP, so execution of this code eventually will get to this file which is another not very interesting source code file.

But if we go to -- there is a lot of code here, and this code all specifies some kind of behavior. But at a certain point when we get somewhere, a function called add photos. This is the function add photo. And you can see that at a certain point, we'll get there, there is this construction that you see right here. You see this. This is something that say insert into photo ID, album_FBID, user, creator ID, link, order, visible, et cetera, et cetera.

What this is is another piece of code in a different language called sequel.

Now, this is just to make things a little more complicated. There is this source

code that is preparing another invocation of a different type of code called sequel code actually store in the storage component some data.

Q. When we talk about the sequel and this code, this is all Facebook?

A. This is all Facebook code.

Q. Okay.

A. And so in this particular case, the net effect of this instruction is to store in the database context information as metadata.

So, for example, who created a certain picture. In what album we want to create a picture. All this information is context information that has been captured by Facebook in addition to the data. So the data as I show you -- if we go to the data, the raw data that we see is this gibberish right here, all this data represents actually the image that you want to upload.

But we go back to the source code, in addition to that data, there is all this additional data that is captured by the context component and stored as metadata in the storage

component along with the actual data of the user. And this is where in the code the context component is.

Q. Now, going back to your next program, could you give us another example of uploading a photo perhaps into an album or something?

THE COURT: Let me stop you there. I apologize. I want to see counsel at side-bar real quick.

(Side-bar discussion:)

THE COURT: I'm sure I'm being unnecessarily fearful, but there is a number of people that are milling about outside the courtroom trying too peer in through the windows. I can't image they can see and copy it down off the board, but I don't know if you have any concern about that. If you want me to ask court security to do anything.

MS. KEEFE: I do, because I don't want to see the code showing up in the Wall Street Journal or the New York Times.

THE COURT: I have seen a number of --

MS. KEEFE: That wouldn't be good for anybody. Could we put a piece of paper over the window.

THE COURT: I think so. It is going to be a little annoying at the moment, but probably I'll have the court security officer to ask him to step back to the door for now and if he has the resources to cover the windows now, fine, if not, are you comfortable with going forward for the hour or do you want to take a break to let him do it?

MS. KEEFE: I have got paper and tape.

THE COURT: Let's talk to them. Stay here.

I'm just concerned because I have seen a number of people trying to peer in through the windows back there, and this is very confidential information on the screen, so if you could --

THE MARSHAL: Do you have any letterhead paper I could tape up?

THE COURT: I think counsel does. Could you also go outside and just tell them

that they should not be peering through those windows.

THE MARSHAL: Okay.

THE COURT: Thank you. Just to let the jury know, as you may be seeing, we're going to be covering up the windows that are in the back of the courtroom just to be extra sure that the confidential information that you all are hearing is not going to be disclosed to the public. So that we'll just take a brief moment.

MS. KEEFE: Thank you, again, Your Honor.

MR. ANDRE: And Your Honor, we're about ready to get out of the source code and go into something that's not as confidential, but we're going to come back to the source code in a few minutes. But we can do this.

THE COURT: I see it's done. Go ahead.

BY MR. ANDRE:

Q. Okay. Dr. Vigna, would you walk us through another example using interceptor program?

A. Yeah. For -- in this case, what

I'm going to show is the uploading of a picture to an album instead of using -- before we saw how one would change his own profile picture. In this case, we're going to see how somebody can upload the picture to an album.

So here we are again. This is John Vineyard profile.

He decides to go to an album that contains pictures of his recipes. And here clicks on add more photos and decides to try the simple uploader. Browse and decide to click on this granita picture.

And click on upload photo. So whenever it clicks on this pic button, again a request is made to Facebook. And I want to clarify that there are other requests that are made under the hood that I'm not showing here because it would be excruciatingly boring to see.

Everything I'm showing you, only the ones that are directly related to uploading a picture and that makes sense in this particular context.

But websites are very complicated.

And for each action, multiple requests are performed.

So just to be absolutely complete, this is the important one that I'm showing you. So in this particular case, photos upload.

PHP is invoked again. And as you can see below here, when I show the parameters, a number of things are captured. For example, the raw information, the ID of the user, the album ID in which the picture is loaded, and there is also a cookie of the user that plays an important role as will be described later.

So as a result, the picture of this delicious granita is now part here -- is part of the album and has been uploaded. And that's pretty much it.

Q. And could you go back to the source code and show us where that happens in the album when you upload a photo?

A. Yeah. This is at the -- slightly different, but very similar sort of like flow. As you can see, again, we saw this pic upload execution, and this function eventually calls again the photos.php. And in particular, the

add photo function.

And this is the code that inserts the photo and then specifies exactly the album ID as captured by the context component. And can take -- no, we're covered, so there is no problem.

Q. Okay. Is that it for the source code?

A. Yes, thanks.

Q. Okay. Now, I wanted to start directing your attention to some of the documents that were produced in this case by Facebook that reflect what you're talking about as well.

A. Yeah.

Q. Could I get you to go to what's been marked as PTX-628? And these will be in the jury binders as well.

A. Yes.

Q. Dr. Vigna, are you familiar with what's marked PTX-628?

A. Yes.

Q. And what is this document?

A. This describes their terms of use

and for the Facebook website, at least it's a version around September 23rd, 2008. It says many things, and one interesting thing is somewhat later in this where it says that -- I think it's -- maybe I can't.

Q. Page 2?

A. It says that Facebook can make copies of -- I think maybe that. Let me check in my copy.

Yeah, exactly. The second paragraph, it says when you post user content to the site, you authorize and direct us to make such copies thereof as we deem necessary in order to facilitate the posting and storage of the user content on the site.

So this show that actually the data is captured and stored on the website as part of their terms of use.

MR. ANDRE: Your Honor, I'd like to move exhibit PTX-628 into evidence.

MS. KEEFE: No objection.

THE COURT: That's admitted.

BY MR. ANDRE:

Q. Okay. Also, could you look at

PTX-629?

A. Yeah, this is the terms of use. I think you want to --

Q. I apologize. I believe that may have been inadvertently omitted from the binders.

Do you have a -- could you go to Page 3 with that?

A. Again, the second paragraph shows that Facebook, you know, make copies, capture the user information, make copies and stores those copies so they can be used by the site.

MR. ANDRE: Your Honor, I'd like to move PTX-629 into evidence.

MS. KEEFE: No objection. Just note for the record that these are old, 2008.

MR. ANDRE: Yeah.

THE COURT: Yeah. They are admitted or it is admitted, I should say.

BY MR. ANDRE:

Q. Dr. Vigna if you turn to PTX-882.

A. Yeah. This is an interesting document that describes a haystack, which at least at a certain point in life of Facebook was

or is used to store actually the pictures of users.

So if you go to the next page, you can see that -- that, you know, it says the photos application is one of Facebook's most popular features.

People have uploaded, you know, billions of photos. And Facebook is a photo-sharing site, which is you know in line with a collaborative nature of sharing information and uploading.

It gives, you know, statistics about how much data is uploaded and later describes different types of infrastructure that rely on distributed systems in order to store the content, the user uploaded information. And it's very technical so -- but it mainly, you know, describes how the storage can be optimized so that, you know, it can be accessed in the fastest way possible.

And it shows that, for example, when they use the photo architecture, there is a -- since each image is stored in its own file, there is an enormous amount of metadata

generated on the storage.

And so this is one example how they use to store this type of information.

Q. If you go down a bit lower in this document, you'll see something titled storage. Do you see that?

A. Yeah.

Q. Is that -- it says the typical hardware configuration of a U2 storage blade provides. Is that indicating that the photos are stored on hardware?

A. These are -- what they're specifying here is actually one -- I mean, it's one -- like one computer. They call it a blade. But it's pretty much a computer exactly as the one sitting down here.

But they use just a number of those computers organized in distributed systems so they can store billions of pictures.

I'd like to move Exhibit 882 into evidence.

MS. KEEFE: No objection.

THE COURT: It's admitted.

BY MR. ANDRE:

Q. Those are three of the public documents that you reviewed; is that correct?

A. Yeah.

Q. Let me show you some of the confidential documents. First, what is a wiki?

A. So a wiki is a web-based means to share information. So imagine that you go to the web page of CNN.com and imagine that instead of just passively looking at the information, you could actually edit some of the articles and add corrections or comments.

That type of interaction is what nowadays we know as a wiki. So a wiki is fundamentally a way to document things in a way that allows many people to comment and contribute to that particular topic.

Okay. So there are wikis on anything. One thing that you might be familiar with is Wikipedia is the idea of creating an encyclopedia by using this shared sort of like information production system.

So, sorry. Now, to answer your question, wikis are often used for documentation of software, because all the developers can

contribute and make sure that the document is up to date and reflects what's going on.

Q. And did you happen to review documents from Facebook's confidential internal wiki?

A. Yes, I did.

Q. And if you will go to Exhibit 252, PTX 252. Could you briefly describe what's in this document?

A. Yeah. So this is a document that describes for internal developers how to -- how uploading a picture happens.

So how the actual capturing of user data and context information is performed by different subcomponents of the system. And it describes how developers can test different flavors of the system if they are to develop enhancement or things like that.

Q. And under the basic upload flow, there's numbers one, two and three. Could you just describe generally what's going on?

A. Well, pretty much it describes, you know, in -- sort of in layman's terms how the different ways in which data can be uploaded

to the system using the components of Facebook.
Q. And number one, where it talks about user navigates to form editfoo.php page on www tier, that let them up load data to Facebook.
A. Mm-hmm.
Q. When they talk about form editfoo.php, what is that referring to?
A. So it's referring to one of different possible components that can be used to upload different kinds of information into the system.
It's trying to be a little more generic in terms of describing how the generic process of uploading different kinds of information is performed.
Q. Then on the number two, it talks about the data is posted. And at the end it says stored in our storage or database.
Do you see that?
A. Yeah. Correct.
Q. Do you understand what that's referring to?
A. Well, I think that refers to the

storage component and storage capability of Facebook. I mean, of course, there is data that is uploaded.
And as we have seen, a moment ago, there is other information in addition to the data that is captured and that has to be put somewhere. And that's the storage component of Facebook.
Q. And then on number three, there's a second sentence. It says other metadata about the write is passed in the get args.
A. Yeah.
Q. What is that referring to?
A. Well, that's -- it's similar to the get -- args are additional arguments. Args is a -- sort of like a computer nickname for arguments so this is additional data that is passed as parameters, and if you remember in the movie when I showed -- so can I show this.
For example, in this case, these are arguments past as part of a post in a different, but similar type of arguments can be passed as a get. There are two operations that can be performed on the web, more than that, but

two are very important, one is get, one is post. And both get parameters, both can provide this type of information that you see right here, so that's the information that they're referring to in the text that you were showing.
MR. ANDRE: Your Honor, I would like to move Exhibit 252 into evidence.
MS. KEEFE: No objection.
THE COURT: It's admitted.
BY MR. ANDRE:
Q. Dr. Vigna, would you please turn to PTX -- do you have a copy of the --
A. I have it somewhere.
MR. ANDRE: I just realized Your Honor, I don't know if we gave him a binder or not.
THE WITNESS: I got it.
Q. If you go to PTX 190.
A. Okay. So this document describes Mulligan which is some kind of code name to describe an improvement to Facebook photos product. So the goal is pretty much to simplify the way in which photos are updated. And if you look at the uploader paragraph, they describe,

you know, that at a certain point their photo uploader wasn't good enough, and it wasn't up to par with respect to their request that users had in terms of uploading pictures. And so they decided to try a new component that would capture users pictures, photos in addition to metadata that is stored in the storage component.
Q. And is this another document from Facebook's confidential internal wiki?
A. Yes, it is.
Q. Have you heard of the term context switching?
A. Yes.
Q. And what does that term refer to?
A. Well, context switching in different context can mean different things. It usually means that you can have certain operation or certain data accessed in different context or environments, and when you move from one to another, you have a context switch.
MR. ANDRE: Your Honor, may I have one moment, please?
THE COURT: You may.